It is well-known that the carbon dioxide gas generated by burning fossil fuels in automobiles contributes to climate change. This is however not their only exhaust and they also generate other troublesome gases; primarily carbon monoxide, nitrogen oxides, and small amounts of partially burnt fuel. These gases in turn contribute to smog, heart and lung conditions, cancer, and other diseases. These toxic emissions are a byproduct of the combustion process and unavoidable even when non-fossil fuels are used. The Three-Way Catalyst (TWC) is one component that can be used to reduce the level of harmful emissions by converting them to non-toxic carbon dioxide, water, and nitrogen gas. The TWC has been very effectively utilized, and can under ideal conditions eliminate nearly all the emissions from gasoline engines. However, the TWC is only effective at removing these emissions when sufficiently hot. This leads to a large initial release of emissions every time the engine is started with a cold TWC.

This thesis focuses on the intersection of optimal control methods and dynamic modeling with the goal of reducing the generated emissions and consumed fuel, particularly in hybrid vehicles. This thesis introduces new variants of general optimal control methods as well as models of the dynamics found in the TWC and hybrid vehicle engines. Using these methods with the developed models allows both for constructing controllers that reduce the level of emissions generated during cold-starts as well as reducing the fuel consumption during changes in the engine's speed. The presented controllers can in the future be implemented in production vehicles, as they do not require any complex calculations to be performed.

JONATHAN LOCK • Towards Optimal Real-Time Automotive Emission Control • 2021

Towards Optimal Real-Time Automotive Emission Control

JONATHAN LOCK

DEPARTMENT OF ELECTRICAL ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2021
www.chalmers.se

THESIS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

Towards Optimal Real-Time Automotive Emission Control

Jonathan Lock

Department of Electrical Engineering
Chalmers University of Technology
Gothenburg, Sweden, 2021

Towards Optimal Real-Time Automotive Emission Control

Jonathan Lock
ISBN 978-91-7905-482-3

Doktorsavhandlingar vid Chalmers Tekniska Högskola
Ny serie nr 4949
ISSN 0346-718X

Department of Electrical Engineering
Chalmers University of Technology
SE-412 96 Gothenburg, Sweden
Phone: +46 (0)31 772 1000
www.chalmers.se

Typeset by the author using LaTeX and the E2 PhD Council template.

Printed by Chalmers Reproservice
Gothenburg, Sweden, May 2021

Towards Optimal Real-Time Automotive Emission Control

Jonathan Lock
Department of Electrical Engineering
Chalmers University of Technology

Abstract

The legal bounds on both toxic and carbon dioxide emissions from automotive vehicles are continuously being lowered, forcing manufacturers to rely on increasingly advanced methods to reduce emissions and improve fuel efficiency. Though great strides have been made to date, there is still a large potential for continued improvement. Today, many subsystems in vehicles are optimized for static operation, where subsystems in the vehicle perform well at constant operating points. Extending optimal operation to the dynamic case through the use of optimal control is one method for further improvements.

This thesis focuses on two subtopics that are crucial for implementing optimal control; dynamic modeling of vehicle subsystems, and methods for generating and evaluating computationally efficient optimal controllers. Though today's vehicles are outfitted with increasingly powerful computers, their computational performance is low compared to a conventional PC. Any controller must therefore be very computationally efficient in order to feasibly be implemented. Furthermore, a sufficiently accurate dynamic model of the subsystem is needed in order to determine the optimal control value. Though many dynamic models of the vehicle's subsystems exist, most do not fulfill the specific requirements set by optimal controllers.

This thesis comprises five papers that, together, probe some methods of implementing dynamic optimal control in real-time. Two papers develop optimal control methods, one introduces and studies a cold-start model of the three-way catalyst, one paper extends the three-way catalyst model and studies optimal cold-start control, and one considers fuel-optimally controlling the speed of the engine in a series-hybrid. By combining the method and model papers we open for the potential to reduce toxic emissions by better managing cold-starts in hybrid vehicles, as well as reducing carbon dioxide emissions by operating the engine in a more efficient manner during transients.

Keywords: Automotive emissions control, optimal control methods, hybrid vehicles, dynamic programming, three way catalyst modelling.

List of Publications

This thesis is based on the following publications:

[A] **Jonathan Lock**, Tomas McKelvey, "A Computationally Fast Iterative Dynamic Programming Method for Optimal Control of Loosely Coupled Dynamical Systems with Different Time Scales". Presented at the IFAC World Congress, Toulouse, France, July 2017.

[B] **Jonathan Lock**, Tomas McKelvey, "Undiscounted Control Policy Generation for Continuous-Valued Optimal Control by Approximate Dynamic Programming". Submitted to the International Journal of Control on July 1, 2020. Preprint available at `https://arxiv.org/abs/2104.11093`.

[C] **Jonathan Lock**, Rickard Arvidsson, Tomas McKelvey, "Optimal Transient Real-Time Engine-Generator Control in the Series-Hybrid Vehicle". Presented at the ASME Dynamic Systems and Control Conference, Park City, Utah, USA, October 2019.

[D] **Jonathan Lock**, Kristoffer Clasén, Jonas Sjöblom, Tomas McKelvey, "A Control-Oriented Spatially Resolved Thermal Model of the Three-Way-Catalyst". Presented at the SAE WCX Digital Summit, Pennsylvania, USA, April 2021.

[E] **Jonathan Lock**, Kristoffer Clasén, Jonas Sjöblom, Tomas McKelvey, "Cold-Start Modeling and On-Line Optimal Control of the Three-Way Catalyst". To be submitted. Preprint available at `https://arxiv.org/abs/2104.12390`.

Acknowledgments

First and foremost, I would like to thank my supervisor Tomas McKelvey. Thank you for your continuous support and around-the-clock availability, encouraging me to follow research ideas that led in unexpected directions (the UCPADP paper in particular), and last but certainly not least for introducing me to road bicycling. I would also like to thank Jonas Sjöblom who has effectively acted as my co-supervisor. Without your crash-course in chemical kinetics and help with countless questions the three-way-catalyst modeling papers would never have reached their full potential.

Special thanks to Kristoffer Clasén for taking the time to set up the TWC experimental rig and help run it, Rickard Arvidsson for the ECU-hacking and help running the hybrid rig, Andreas Henriksson for acting as my own personal graphic designer and for creating probably every good-looking 3D graphic in this thesis, Johan Winges for Matlab tips, tricks, and hacking, Linnéa Ahlman for all the proofreading help, Carl Toft for sanity-checking countless mathematical assumptions, Ann-Brith Strömberg for pointing me towards the topic of Lagrangian Relaxation, and my colleagues at the Electrical Engineering department, Jacob Klintberg, Ebrahim Balouji, Jakob Lindqvist, Jennifer Alvén, Nina Sundström, Johan Nohlert, Erik Stenborg, Måns Larsson, Anders Karlsson, Juliano Pinto, and others. Thanks for sharing both moments of laughter and frustration. Finally, thanks to Louise, Nemo, and my friends and family for kindly keeping me aware of the world outside of the sometimes all-encompassing PhD bubble.

This PhD position has been made possible and funded by the Combustion Engine Research Center at Chalmers (CERC) with financial support from the Swedish Energy Agency.

Jonathan Lock
Gothenburg, May 2021

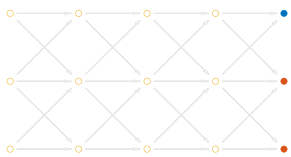
Acronyms

ADP:	Approximate Dynamic Programming
CV:	Continuous-Valued (problem)
DP:	Dynamic Programming
DV:	Discrete-Valued (problem)
EATS:	Exhaust Aftertreatment System
EMS:	Energy Management System
HC:	Homogeneous Charge
HEV:	Hybrid Electric Vehicle
SI:	Spark Ignited
TWC:	Three-Way Catalyst
UCPADP:	Undiscounted Control Policy generation by Approximate Dynamic Programming

Contents

Part I

Introductory chapters



CHAPTER 1

Introduction

The combustion engine has been used for over 100 years in the history of automotive applications. Though optimizing a given engine to run efficiently for stationary operation in "ordinary" combustion modes is a fairly solved problem, there are many sources of dynamics (e.g. temperatures, rotational speeds, and gas flows) that can complicate the task of transitioning the engine from one operating point to another. This holds even more so if we wish to ensure that the engine is operated in a way that is in some sense optimal during the transient. With ever-tightening efficiency and emissions requirements the transient-control problem becomes all the more relevant to study. This leads to the overarching research questions that we discuss in this thesis

Question 1 How should we operate the engine during transients in a close-to optimal manner?

Question 2 Assuming we know how to operate the engine, how can we practically do this in real-time using the limited computational perower available in today's vehicles?

Addressing these three questions touches on many different fields of research, including system modeling, optimal control methods, and computationally efficient control schemes.

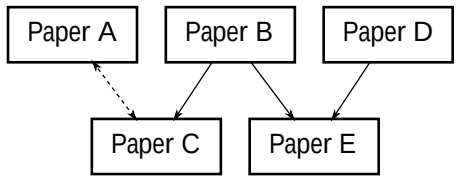


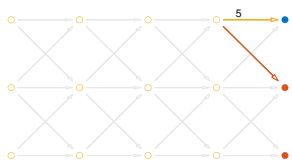
Figure 1.1: Hierarchical relationship of the appended papers. Paper C builds on Paper B and studies a problem similar to Paper A. Paper E builds on Paper B and Paper D.

1.1 Thesis outline

This thesis is divided into two parts. Part I constitutes the introductory chapters, where Chapter 2 starts off by briefly introducing the concepts and methods necessary for understanding the remainder of the thesis. This is followed by Chapter 3 which summarizes the main contributions of the papers appended to the thesis. Afterwards, Chapter 4 provides a concluding discussion and indicates relevant future research directions. Finally, Part II comprises the five appended papers, which constitute the primary contribution of this thesis.

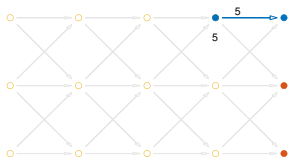
1.2 Thesis aim and scope

The appended papers consist of two papers on numerical methods for solving optimal control problems (Papers A and B), one paper on modeling the three-way-catalyst (TWC) during a cold-start (Paper D), one paper that extends the previous TWC model and generates and benchmarks a real-time implementable optimal cold-start control strategy (Paper E), and one paper on applied optimal control of the combustion engine and generator in the series-hybrid (Paper C). These papers vary significantly in their scope. The method papers (Papers A and B) are general and have the potential for use in a wide range of optimal control applications, including ones beyond the automotive applications discussed here. The TWC papers (Papers D and E) are slightly more specific, with applications primarily in the context of computationally efficiently modeling and controlling the temperature evolution of the TWC during a cold-start. Finally, the series-hybrid optimal control paper (Paper C) has the most limited scope, and exemplifies the improvement given by implementing an on-line real-time controller. It is by this chain of



papers that build on each other (as illustrated in Figure 1.1) that this thesis expands on and contributes to the above research questions. Finally, this thesis does not propose to completely answer the research questions, but is rather a step on the way towards a higher degree of optimality in commercial automobiles.

The scope of optimal control applications studied in this thesis is primarily limited to hybrid (i.e. electric and combustion-engine powered) passenger automobiles with a conventional gasoline (i.e. spark-ignited (SI) homogeneous-charge (HC)) engine and TWC aftertreatment system. Though there are many factors that influence the total performance of an automobile, this thesis focuses on the fuel efficiency and emissions generated during transient operation of the combustion engine and TWC cold-starts.


5
5

CHAPTER 2

Preliminaries

This chapter briefly introduces the theory, concepts, and methods used by the included papers. Section 2.1 introduces the basic concepts of hybrid electric automotive vehicles, and Section 2.2 introduces automotive exhaust aftertreatment of conventional gasoline combustion engines. This is followed by Section 2.3, which introduces the particular type of optimal control method used in the included papers. This is done by first discussing the concept of dynamical systems, followed by the dynamic programming graph optimization method, and then finally showing how dynamic programming can be used to solve optimal control problems.

2.1 Hybrid electric vehicles

The hybrid electric vehicle (HEV) extends on the conventional, purely combustion-engine powered vehicle, by the augmenting it with an additional electric drive-train. The addition of an additional electric drivetrain allows for increasing the vehicle's fuel efficiency, performance, or a combination of both. There are a wide range of different types of hybrid vehicles, both with respect to the drivetrain structure as well as the degree of hybridization. This section

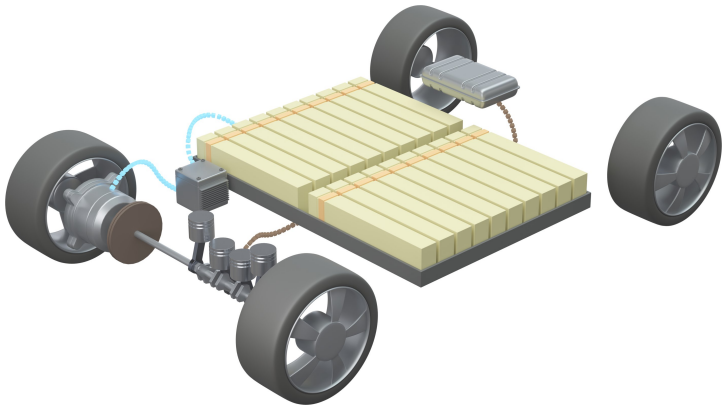
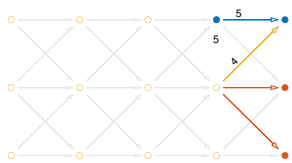


Figure 2.1: In the parallel hybrid, the ICE and EM can separately apply traction power. Opening the clutch allows for the EM to drive the vehicle while the combustion engine is turned off. Electric power flow from the EM to power electronics and battery is illustrated by dotted blue lines, and fuel to the combustion engine by the dotted brown line.

will briefly outline the main categories, benefits, and drawbacks of hybrid drivetrains by highlighting the most commonly found structures and degrees of hybridization. More extensive details covering the contents of this section can be found in nearly any introductory textbook on HEV's, see for instance [1], [2], or [3].

2.1.1 Parallel hybrid

The parallel hybrid structure has an ICE and an EM that can both independently propel the vehicle. In some cases this is implemented simply "through the road", where the front wheels are typically powered from the ICE and the rear wheels from the EM, giving the additional benefit of allowing for four-wheel-drive. In other common structures the EM is instead coupled to the driveshaft, sometimes before and sometimes after the clutch [1, p. 72]. This requires a minimum of additional hardware and in the former case allows the EM to replace the conventional starter motor and alternator (reducing system cost) and in the latter case allowing for the EM to power the vehicle while the ICE is switched off (allowing for pure-electric operation). An illustration of the latter case is shown in Figure 2.1.



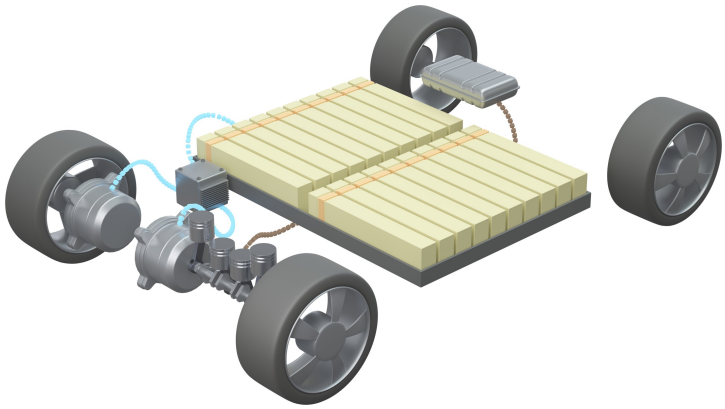


Figure 2.2: In the series-hybrid, the ICE powers a generator, which in turn delivers power to the battery and/or traction motor.

One drawback of the parallel hybrid structure is that the rotational speeds of the ICE and EM are not independent of the vehicle's speed. More specifically, just like a conventional vehicle the ICE's (and EM's) angular velocity is proportional to the vehicles speed and selected transmission gear ratio. Most significantly, this implies that the ICE's angular velocity cannot be arbitrarily controlled, and can be forced to operate at speeds with lower a lower fuel efficiency.

2.1.2 Series-hybrid

In the series-hybrid, the ICE is not mechanically coupled to the wheels, but instead drives an electric generator which in turn supplies power to a battery and/or electric motor which in turn drives the wheels as shown in Figure 2.2. A similar structure, apart from not including a battery, is also commonly used in diesel-electric trains and large ships. By using an "electric" transmission together with a battery, the ICE is free to operate at any velocity and power independently of the vehicle's velocity and traction power (with excess power stored or removed from the battery as needed). This decoupling between the ICE and vehicle velocity and power allows for improved fuel efficiency, particularly in stop-and-go traffic, as the engine can be run at the angular velocity and power level that maximizes its efficiency.

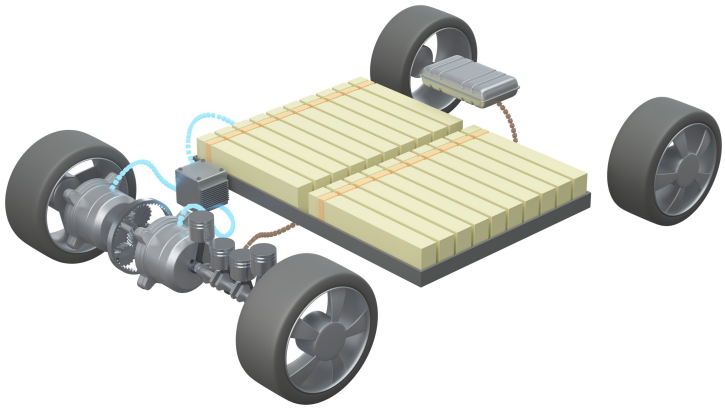


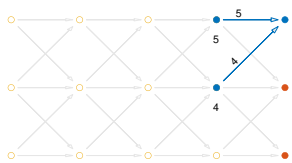
Figure 2.3: In the series-parallel-hybrid, the ICE and EM are connected to the wheels with a power-split device (here shown with a planetary gearset), which allows operation both as a series-hybrid and as a parallel-hybrid.

One drawback with the series-hybrid structure is that the power electronics and traction EM need to be sized for the power demands of the entire vehicle, rather than supplementing the ICE power as in the parallel-hybrid case. Furthermore, the efficiency tends to be lower than the parallel-hybrid when the ICE is heavily utilized, as the losses in each power conversion stages (generator, power electronics, battery losses, traction EM) accumulate.

Some vehicles primarily marketed as (nearly) purely electric vehicles are configured with a large traction EM and battery, as well as a small ICE referred to as a range extender[1, p 104] that is typically only operated when the battery is depleted. These vehicles are typically realized with a series-hybrid structure, thus gaining their benefits and drawbacks.

2.1.3 Series-parallel hybrid

The series-parallel hybrid combines some of the properties of the series- and parallel-hybrid drivetrains, as shown in Figure 2.3. As in the parallel-hybrid case, both the ICE and EM are mechanically connected to the wheels, giving a relatively high total efficiency. However, the ICE (though typically not the EM) retains the ability to control its speed independently of the vehicle's speed



as in the series-hybrid case. This is attained through the use of a power-split device, which acts as a form of continuously-variable transmission, transferring the mechanical power from the ICE and EM to the wheels. There are several different realizations of a power-split device, one of the simpler solutions being a planetary gearset where the ICE, EM, and wheels are connected to the sun, planet, and carrier gears. By correctly choosing the speed of the EM, the ICE can be (close-to) free to rotate at any speed while delivering power to the wheels. This structure maintains the freedom of choosing the ICE speed independently of the vehicle velocity, while also avoiding the need for as powerful an EM as in the series-hybrid case. One drawback of the series-parallel hybrid is that the power-split device is typically realized using two (or more) EM's, which increases the system's cost.

Another method of categorizing hybrid vehicles is by instead considering their degree of hybridization. Typically, a hybrid vehicle will fall into one of three categories.

Mild Hybrid Essentially a conventional vehicle with a moderate-sized electric motor that replaces the starter and alternator of a conventional vehicle, i.e. a parallel-hybrid structure. This is typically combined with a battery with high specific power (capable of delivering a large amount of power) and small capacity (low total stored energy). The primary benefit of this structure is regenerative braking, which allows for capturing the kinetic energy during deceleration and storing it in a battery, and electric boost, wherein the electric motor assists the combustion engine during acceleration. A start/stop scheme is also commonly used, where the combustion engine is turned off when the vehicle is braking, coasting, or stationary. During these intervals accessories run on the energy stored in the battery. This level of hybridization is fairly inexpensive, as there are few additional components, but with limited potential for fuel consumption improvement or performance increases.

Full Hybrid A full hybrid is made using an ICE and electric drivetrain that are each powerful enough to drive the entire vehicle for a meaningful distance, and can be found with all the structures listed above. A battery with high capacity is typically needed in order for the vehicle to be powered purely electrically for reasonable distances. This structure improves on the fuel efficiency and performance of the mild hybrid, but typically at increased complexity and cost. Furthermore, though purely electric

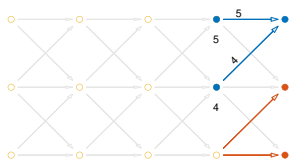
operation is possible, all the energy stored in the battery ultimately originates from the ICE, limiting the total system efficiency.

Plug-in Hybrid A plug-in hybrid extends on a full hybrid by also including the ability to charge the battery by a connection to an external electrical source, as well as typically using a battery with even larger energy capacity. This allows for operation as a purely electric vehicle within the energy and power limitations of the electric drivetrain, and improves on the system efficiency of the full-hybrid.

Regardless of the structure or degree of hybridization, all HEV's require some form of energy management system (EMS) to control when and how much power is delivered from the ICE and electric subsystem respectively.[1] A well-designed EMS should therefore select the power-split ratio (i.e. ratio of ICE power to electric power) in an efficient manner. Typically, an EMS will tend to use the electric subsystem predominantly at lower speeds and in stop/go traffic, which makes good use of the regenerative capabilities of the electric subsystem, avoids exhaust emissions in cities, and consumes the battery's energy at a moderate rate. Similarly, an EMS will typically operate the ICE more during periods of highway operation, which utilizes the ICE in a region where it is efficient and the battery would otherwise drain more quickly. The specific design of the EMS is therefore crucial in order to achieve good vehicle performance and fuel efficiency.

The contents of this thesis can be broadly categorized as studying two aspects of EMS design. One topic concerns controlling the ICE and electric generator in the series-hybrid. Though it is fairly easy to determine the stationary ICE and generator speed that results in the best fuel efficiency for a constant load, it is more difficult to determine how to dynamically transition the engine's speed from one to another when the electrical load changes (i.e. which torques to apply from the ICE and generator with the goal of reaching the new optimal speed). This problem is studied in more detail in Paper C, and builds on Papers A and B. The second topic concerns how to effectively control the ICE in situations where the exhaust aftertreatment system is cold, which is a subject we will introduce in the following section.

[1] This is not the case in a conventional (non-hybrid) vehicle, where the ICE torque is directly controlled by the driver.



2.2 Automotive exhaust aftertreatment

The combustion engine in an automotive vehicle burns a fuel which consists of a mixture of different hydrocarbon molecules. As the combustion engine burns the fuel in air, the majority of its exhaust consists of non-toxic nitrogen gas (N_2), carbon dioxide (CO_2), and water (H_2O). However, some byproducts are unintentionally generated which are harmful to humans and the environment in the concentrations typically present (on the order of 100 ppm to 2%). These harmful emissions are typically categorized as

CO carbon monoxide; produced by the incomplete oxidation of a carbon atom in the fuel,

NO_x nitrogen oxides; primarily formed by the spontaneous combination of the nitrogen and oxygen present in air,

THC gaseous total unburnt hydrocarbons, produced by the incomplete combustion of the fuel, and

PM particulate matter, which is primarily unburnt fuel in the solid phase, i.e. soot.

In this thesis we will limit the scope to the gaseous emissions (i.e. CO, NOx, and THC) from spark ignited (SI) homogeneous charge (HC) combustion engines, i.e. conventional gasoline ICE's. Though PM emissions are increasingly becoming more important, these emissions are beyond the scope of this thesis.

There are legally mandated emission restrictions that serve to limit the emissions generated by automotive vehicles. These restrictions have been successively tightened over time[2], and the automotive industry has had to undertake actions that keep the emissions below these ever-stricter limits. Several methods have been used to reduce the level of emissions, and can broadly be categorized as

- reducing the level of emissions generated by the engine,
- converting the emissions to harmless gases by using an exhaust aftertreatment system (EATS), and

[2] In the EU, the relevant emission standards are labeled EURO 1 through EURO 7 (`https://dieselnet.com/standards/eu/ld.php`). Other regions have similar emissions standards, e.g. Tier 1 through Tier 3 in the USA and China 1 through China 6b in China.

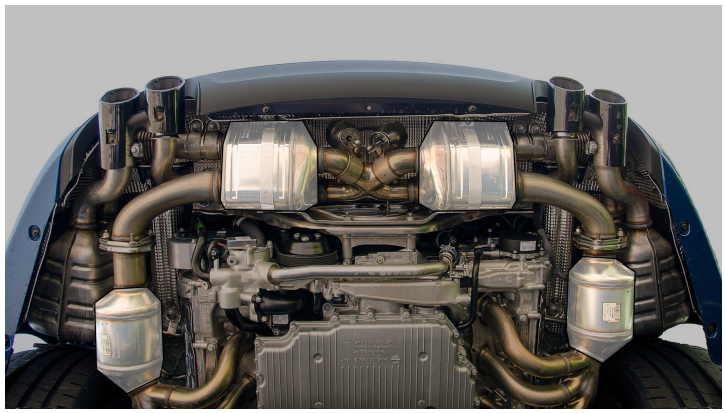
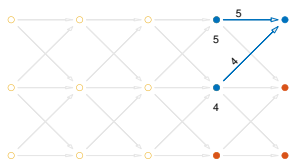


Figure 2.4: The underside of a performance passenger car with exposed EATS highlighted. This particular vehicle has two sequential TWCs per cylinder bank, for a total of four TWCs. Most passenger cars have one or two TWCs.

- using electrification to run the ICE in operating regions where fewer emissions are generated and/or limiting the relative amount of time that the combustion engine is used.

This section will give a brief outline of the EATS typically used with gasoline engines to limit emissions, namely the three way catalyst (TWC). Several excellent textbooks go into considerably more detail on the formation and treatment of emissions, e.g. the classic [4, p. 567], as well as for instance [5, p. 277] or [6, p. 164].

The TWC became widespread in the 1980's, after which new gasoline vehicles were in most countries required to use a TWC to reduce the level of harmful emissions. The TWC converts these emissions to harmless gases (water, nitrogen, and carbon dioxide) by passing the exhaust gas over catalytically active materials (typically a mixture of platinum, palladium, and rhodium). Early TWC's converted $\approx 95\%$ of the emissions generated by the engine, while modern TWC's convert $\approx 99.9\%$ the emissions, ultimately giving a tailpipe exhaust output with toxic emissions several orders of magnitude lower than the engine-out exhaust. The TWC is thus a crucial component in meeting the emissions legislation, and will likely continue to be so in the future.



A photo of a typical TWC in a small passenger vehicle is shown in Figure 2.4. The outside of the TWC is thermally insulated for reasons we will discuss shortly, so we can not directly see the internal workings of the TWC. Figure 2.5 shows an opened TWC instrumented with many temperature sensors. We can identify catalytically active sections (two in this specific TWC), which are shown in more detail in Figure 2.6. We can in these photos start to see the characteristic structure of the conventional TWC, which consists of a large number of long and narrow channels that the exhaust gases are forced to travel through, see Figure 2.7. The channel walls are constructed using either a metallic or ceramic substrate (as previously shown in Figure 2.6), which is then coated with a *washcoat*. The washcoat serves several purposes, but primarily carries the catalytically active materials and is microscopically rough and porous, i.e. has a large the exposed surface area. It is this catalytically active material that ultimately catalyses the reactions where the emissions are converted into safe compounds (Figure 2.8).

There are several conditions that must be met for the TWC to maintain a high conversion efficiency. One of the most noticeable is the TWC's temperature dependence. A conventional TWC has virtually no effect on the gas composition when at room temperature, converts approximately 50% of the emissions at $\approx 350^\circ$ C, and reaches full conversion efficiency when above $\approx 500^\circ$ C. This strong temperature dependence in turn implies that any *cold start*, i.e. starting the engine when the catalyst is at the ambient temperature, is associated with a high level of emissions during a transient period of approximately $10 - 100$ seconds. The total amount of generated emissions during this critical time is typically equivalent to $20 - 30$ minutes of normal operation, and a great deal of work has been performed to reduce the cold start emissions.

There are several other effects that also influence the conversion efficiency. One example is the air-fuel ratio in the engine, where only a very small range of air-fuel ratios near stoichiometry[3] give full conversion of all species[4]. Further-

[3] The HC SI engines studied in this thesis are designed to run at near-*stoichiometric* conditions, i.e. where the ratio of air to fuel is balanced such that both the fuel and oxygen is completely consumed in the combustion process. In the ideal case this would result in exhaust containing neither any unburnt fuel nor any remaining oxygen. It is typically difficult to maintain stoichiometry during high load conditions as well as during dynamic engine operation, i.e. when the engine power is rapidly changed.

[4] The highest conversion efficiency is achieved when the air-fuel ratio is repeatedly switched between slightly rich (excess fuel) and lean (excess air) conditions. See [4] for a brief description on the switching scheme and [7] for more details on why this improves the conversion efficiency

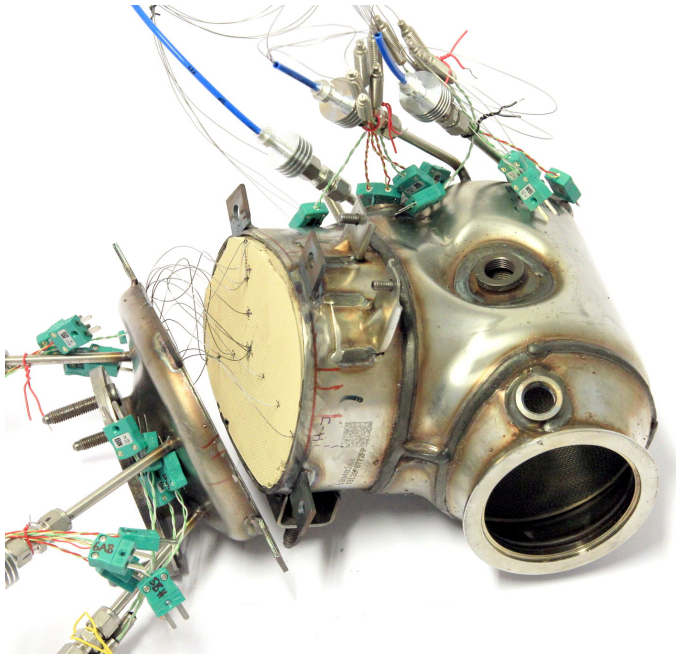


Figure 2.5: A commercial TWC instrumented with many temperature sensors. This TWC has two separate catalytically active sections, the first one seen inside the opening to the lower right, and the second exposed to the left. The entire TWC is approximately 20 cm wide.

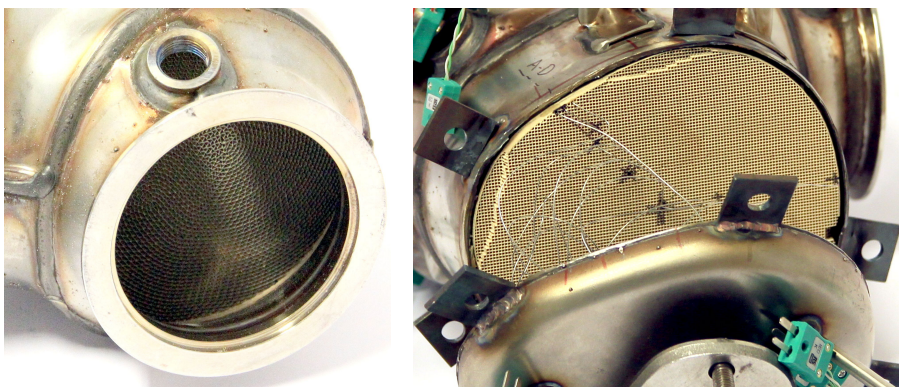
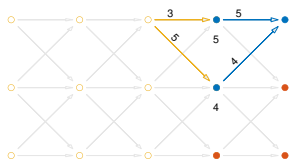


Figure 2.6: Close up of first (metallic) and second (ceramic) sections.



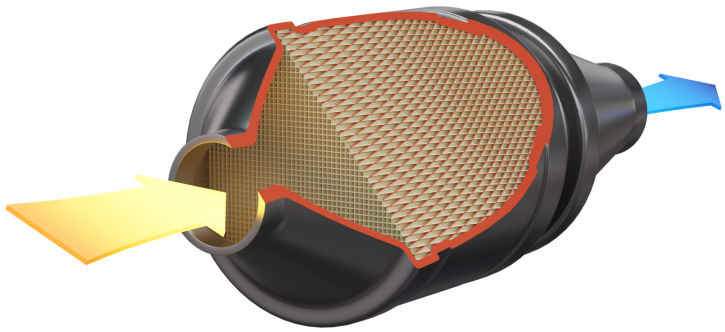


Figure 2.7: A typical TWC, with a cutout region showing the characteristic small channels.

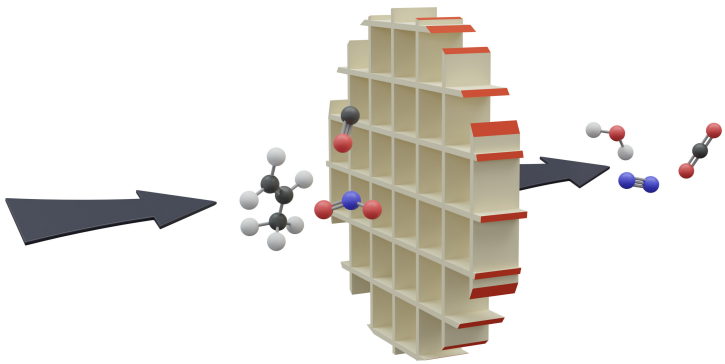


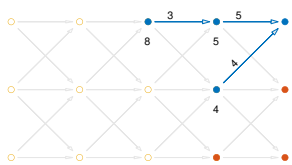
Figure 2.8: A representative slice of the TWC, with catalytically coated walls that convert incoming CO, NO_x, and THC to CO_2, N_2, and H_2O.

more, TWC aging (where older TWC's typically show reduced performance), TWC poisoning (where specific contaminants in the fuel can eventually reach the TWC and permanently reduce its conversion efficiency), and the residence time in the TWC (where slower-moving exhaust spends a longer time in the TWC, leading to improved conversion efficiency) also influence the conversion efficiency [4].

Though the HEV has several attractive properties compared to conventional vehicles, including the potential for fuel-efficiency improvements as described in Section 2.1, it is not clear how the level of generated emissions compares. In conventional vehicles, the ICE is continuously operated, keeping the TWC warm and thus generating one cold start per trip. However, HEV's have the ability to turn the ICE off during extended intervals (which can for instance occur during low-power city driving), causing the TWC to cool down. This in turn leads to repeated cold- (or semi-warm) starts when the combustion engine is subsequently turned on. Whether or not the increase in emissions from additional cold starts in the HEV outweighs the small but continuously generated emissions from conventional vehicles is difficult to determine. However, we can exploit the HEV structure to improve on the cold- and warm-start emissions. By virtue of the electric machine we gain freedom in the choice of power delivered by the ICE. A cold-start-aware EMS can thereby have the ability to, within limits, select e.g. the combustion engine speed, torque, and ignition timing[5] in a manner that conventional (non-hybrid) vehicles cannot.

The second aspect of EMS design studied in this thesis focuses on laying the groundwork for solving the the cold-start problem in HEV's as an optimal control problem that can be run in real-time. We study this problem in Paper D, where we develop a model of the combustion engine and the TWC that captures the engine's exhaust gas composition and the temperature dynamics in the TWC. Furthermore, by developing the method of Paper B we create a foundation for generating an optimal controller that can in turn be implemented in the limited hardware available in a vehicle. Finally, with Paper E we extend the TWC model of Paper D and simulate the performance improvements made by using an EATS-aware EMS. These contributions thus lay the groundwork

[5] Adjusting the specific time when combustion is started in the cylinder allows for trading off fuel efficiency and exhaust gas temperature, allowing for heating the TWC more quickly at cost of increased fuel consumption



for generating a routine for heating the TWC in an efficient manner, both from a completely cold state as well as the likely scenario of a halfway-warm state.

2.3 Optimal control

Optimal control is a branch of control theory that focuses on finding the control (i.e. actuator) signal for a dynamic system that optimizes an objective function. A wide range of applications are found in science and engineering. For example, the dynamical system might be a car with the control being the traction power, and the objective being to minimize the fuel consumption while maintaining a distance range to another vehicle (i.e. optimal adaptive cruise control). However, there exist many possible applications beyond the automotive field studied in this thesis. For instance, the dynamical system might be a fish farm with the control being when and how many fish to harvest, with the objective being to maximize the fish production. Alternatively, the dynamical system could be an aircraft with the control of engines, rudder, aileron, and elevator, the objective being to travel from point A to point B using as little fuel as possible. Other applications can be more abstract, for instance choosing a time-varying economic policy with the goal of minimizing unemployment. In this thesis focus on automotive applications, and primarily consider controlling the combustion engine and TWC.

We will represent optimal control problems using the general form

$$(x^*, u^*) = \arg\min_{x,u} J \tag{2.1a}$$

$$J^* = \min_{x,u} J \tag{2.1b}$$

with the cost function

$$J = \sum_{n=0}^{N} c(x_n, u_n, n) \tag{2.1c}$$

subject to the constraints

$$x_{n+1} = d(x_n, u_n, n) \tag{2.1d}$$

$$g(x_n, u_n, n) \leq 0. \tag{2.1e}$$

Here, x_n and u_n denote the *system state* and *control signal* at time sample $n = [0, 1, 2, \ldots, N]$ respectively. Similarly, x^* and u^* denote the *optimal* states

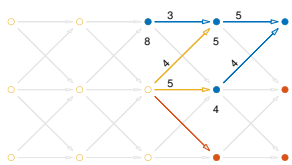
and controls for a given problem formulation. Intuitively, we can view u_n as the control signal we chose at any given time, and x_n captures the information needed to determine the future behavior of the system based on the present. For instance, in the fish farm example we could let x_n indicate the number of fish in the farm and u_n indicate the number of fish harvested at any given day n.

The functions c, d, and g represent the cost, dynamics, and constraints of the problem. The systems dynamics d describes the evolution of the system state, i.e. what the next state will be given the current state, control signal, and time sample. The constraint g and cost c are functions that we as design engineers chose to mathematically represent our optimization goal. g allows us to place hard constraints on the system state and control that must not be violated, while c assigns a cost to any given state and control combination at any given time sample.

At its core, the optimization problem consists of determining a control signal u (a parameter we can directly control) and associated state trajectory x (which is given once u is chosen) that minimizes the total cost J. Much of the complexity of the optimization problem arises from the system dynamics d, in the sense that the choice of u at any given time will typically affect x *at all future times*. Ultimately, solving eq. (2.1), assuming a unique solution exists, gives the *optimal* cost J^*, control u^*, and state trajectory x^* that minimizes the accumulated cost J while satisfying $g \leq 0$.

More formally, we will let $x_n \in \mathbb{R}^l$ and $u_n \in \mathbb{R}^m$, which in turn implies that $c : \mathbb{R}^l \times \mathbb{R}^m \times \mathbb{R} \to \mathbb{R}$, $d : \mathbb{R}^l \times \mathbb{R}^m \times \mathbb{R} \to \mathbb{R}^l$, and $g : \mathbb{R}^l \times \mathbb{R}^m \times \mathbb{R} \to \mathbb{R}$. We will for ease of discussion assume that c, d, and g are continuous and bounded, but may otherwise be arbitrarily nonlinear. Similarly, we will for now assume that c and d are well-defined for all arguments, i.e. their domain is $\mathbb{R}^l$ and $\mathbb{R}^m$ respectively. We will later see that Dynamic Programming (DP), the class of method we introduce to solve the optimal control problem, can easily handle the case where this does not hold and thus avoids undefined regions elegantly. Similarly, we will for convenience assume that J and u^* are unique and exist, but will also see that DP easily detects and can handle the existence of multiple solutions or the lack of a solution.

Wrapping up some loose ends, we will define x_0 as the *initial condition* and view it as a given value. Furthermore, note that eq. (2.1) is a discrete-time control problem, while the physical system we wish to control typically resides



in the continuous-time world. We resolve this by letting the continuous-time control signal $u(t)$, from time t_n to time t_{n+1}, be parameterized by u_n. Most commonly this is done by letting $u(t)$ be a piecewise constant function, constant during each interval of time t_n to t_{n+1} (i.e. zero-order hold reconstruction). This particular choice is motivated as many practical digital control systems operate at a fixed sample rate, i.e. they are configured to output a control signal u_n at a given (constant) rate.

2.3.1 Dynamic systems

In this section, we will briefly cover the concept of dynamic systems and introduce the phase portrait as a visualization tool, which we will later use to connect Dynamic Programming (DP) and optimal control.

An illustration of an idealized pendulum is shown in Figure 2.9 with a support point, a rod, and a bob (the mass at the end of the rod). Here, we assume that the rod is massless and of constant length, all motion occurs in two dimensions, there is no friction or air resistance, the gravitational field is constant and uniform, and the support does not move. We define the torque τ to be some given function of time that we can freely choose, i.e. we can view it as the control input. Furthermore, we will also keep track of the number of turns that the pendulum has rotated clockwise or counterclockwise, i.e. the case where $\theta = 0$ is distinct from the case where $\theta = 2\pi$ even though the weight is directly below the support in both cases.

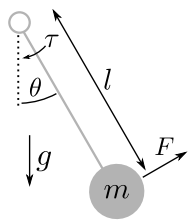


Figure 2.9: Illustration of a simple pendulum.

The idealized pendulum is one example of a dynamic system, i.e. a system whose state (in this case the angle θ and angular velocity $\dot{\theta}$) varies over time. By generating a mathematical function that describes the evolution of these states over time we can then analyze the behavior of the pendulum, which will be necessary for subsequently applying optimal control.

As the pendulum is a simple system, we can analytically determine the function that describes the system dynamics. Applying Newton's second law to the system gives

$$F = -mg\sin\theta - \frac{\tau}{l} \tag{2.2}$$

which in turn implies that the acceleration of the bob along the arc of motion is

$$a = -g\sin\theta - \frac{\tau l}{m}. \tag{2.3}$$

Define s to be the position of the bob along the arc of motion (where $s = 0$ implies that the bob and support are vertically aligned). This gives

$$s = l\theta. \tag{2.4}$$

If we take the derivative of eq. (2.4) twice with respect to time we can see that

$$a = \ddot{s} = l\ddot{\theta}. \tag{2.5}$$

Combining eq. (2.3) and eq. (2.5) gives a second order ODE

$$\ddot{\theta} = -\frac{g}{l}\sin\theta - \frac{\tau}{m}. \tag{2.6}$$

Notably, eq. (2.6) fully defines the change in the pendulum's angle θ over time.

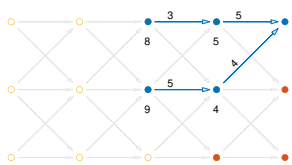
Introducing the notation $x = [\theta\ \dot{\theta}]^T$, $u = \tau$ allows us to represent the continuous-time dynamics eq. (2.6) as a system of coupled first-order ODEs

$$\dot{x} = f(x, u) \tag{2.7a}$$

$$= \begin{bmatrix} \dot{\theta} \\ -\frac{g}{l}\sin\theta - \frac{u}{m} \end{bmatrix}. \tag{2.7b}$$

Solving eq. (2.7) is difficult analytically, but by assuming a constant sample time t_s and letting $u(t)$ be constant between sample times, i.e.

$$u(t) = \begin{cases} u_0 & \text{for } 0 \le t < t_s \\ u_1 & \text{for } t_s \le t < 2t_s \\ u_2 & \text{for } 2t_s \le t < 3t_s \\ \vdots \end{cases} \tag{2.8}$$



we can numerically solve the pendulum's set of first-order differential equations. We can represent this solution as a function of form

$$x_{n+1} = d_{\text{pend}}(x_n, u_n), \tag{2.9}$$

which matches the structure of eq. (2.1d).

For the remainder of this thesis, all illustrations of system behavior will be shown for $g = 1$, $l = 1$, $m = 1$, and $t_s = 0.1$.

The phase portrait

The phase portrait is a graphical representation of a dynamic systems evolution in phase space, i.e. the space spanned by the system's state variables. A phase portrait of the undriven (i.e. $u = 0$) pendulum dynamic system is shown in Figure 2.10. Blue arrows indicate how the state transitions from one sample to the next, i.e. the base of an arrow is located at x_n, while the point is located at the associated next location x_{n+1} (i.e. the state $t_s = 0.1$ seconds later). Solid lines show the state trajectories for a few initial conditions, where the large dot indicates the initial condition x_0 and smaller dots are placed at $x(t) = [1, 2, 3, \dots]$.

Time-optimal pendulum control

We will now show one example of optimal control, specifically bringing the pendulum to a vertical standstill as quickly as possible while limiting the magnitude of the applied torque at any given time instance (u_n) and state at any given time instance (x_n). We can express this problem as minimizing the number of samples where the pendulum is not at the desired state,

$$J_{\text{pend}} = \lim_{N\to\infty} \sum_{n=0}^{N} \begin{cases} 0 \text{ if } x = 0 \\ 1 \text{ otherwise} \end{cases} \tag{2.10a}$$

subject to

$$x_{n+1} = d_{\text{pend}}(x_n, u_n) \tag{2.10b}$$
$$|u_n| \leq 0.5 \tag{2.10c}$$
$$|\theta| \leq 1.5\pi \tag{2.10d}$$
$$|\dot{\theta}| \leq \pi. \tag{2.10e}$$

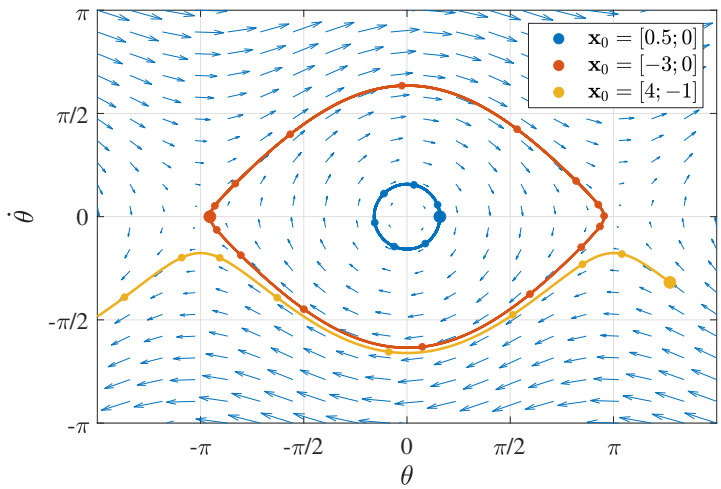
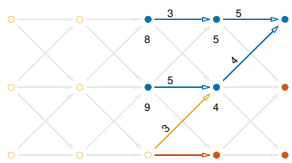


Figure 2.10: Phase portrait of the pendulum for $u = 0$. The blue solid line shows "ordinary" low-angle pendulum behavior with near-sinusoidal angle and velocity, while the red line shows the behavior when the pendulum nearly reaches the upside-down vertical state, and the yellow line shows the case when the pendulum velocity is large enough to rotate in the same direction indefinitely, i.e. $lim_{n\to\infty}\theta_n = -\infty$.



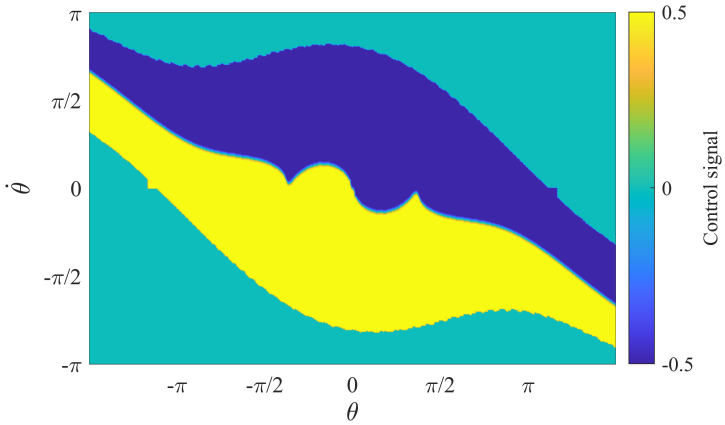


Figure 2.11: Control law for solution to eq. (2.10). The two large regions coded by $u^* = 0$ can never be steered to $x = 0$ without violating the constraints eqs. (2.10c) to (2.10e).

Using the DP-based method we will shortly describe, we can generate an approximate solution to the above problem. An illustration of the generated control law is shown in Figure 2.11. As it turns out, the optimal control u^* for this problem is a function of solely the current system state and can be represented as the *control policy* $u^*(x)$, i.e. explicit, nonlinear, time-invariant state feedback. We will later see that there are many optimal control problems whose solution can be formulated as time-invariant state-feedback, and Paper B details a general numerical method for generating control laws. This is of great practical relevance, as it allows for subsequently implementing a real-time controller if the current system state is measurable by simply storing a look-up-table of the optimal controls to apply. A controller can thus generate an approximation of the optimal control signal by simply consulting the control policy (i.e. table of system states) and interpolating the stored optimal control from among the nearby tabulated states. For example, in Figure 2.11 the optimal control law is shown for a discrete number of points (pixels). If the measured system state were to not lie exactly on one of these points, we can consult the stored signal at neighboring points to approximate the optimal control signal. We will return to this topic in more detail in Section 2.3.3.

Figure 2.12 illustrates the phase portrait of the dynamic system when the optimal controller is applied, showing how several different initial conditions

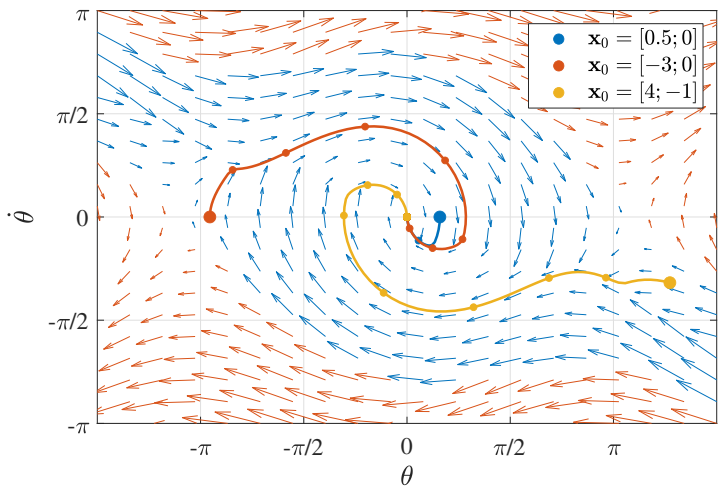


Figure 2.12: Phase portrait of DP solution. Points that can never be steered to $x = 0$ are indicated with red arrows.

all reach the desired state $x = 0$ (compare with Figure 2.10). Red arrows indicate states where there does not exist a solution that satisfies the problem constraints Equations (2.10c) to (2.10e). Their evolution is shown for $u = 0$, and we can see that trajectories in this region eventually violate the problem constraints (this violation would occur regardless of the choice of u). We will later see that this information is also directly generated by the DP method.

An illustration of the set of initial conditions that have a solution (i.e. corresponding to the region with blue arrows in Figure 2.12 and nonzero u in Figure 2.11), and their time evolution is shown in Figure 2.13. The set boundary is shaped like a funnel, showing that all initial conditions that have a solution eventually reach the point $x = 0$.

2.3.2 Dynamic programming

Dynamic programming (DP) is a family of optimization methods developed by Richard Bellman in the 1950's that have found applications in a wide range of fields, including optimal control. Regardless of the specific field, a DP method solves an optimization problem by recursively breaking it down into simpler sub-problems. This section will briefly introduce the concept of dynamic

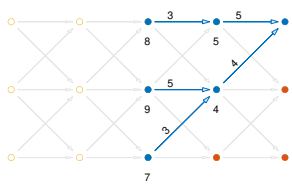


Figure 2.13: The hull of feasible states evolves over time and eventually reaches [0 0]. The phase portrait and a representative solution is shown for reference.

programming. The classic [8] and [9, p. 131] cover dynamic programming and optimization respectively in more detail. A concise derivation is also given by [10, p. 44], with a focus on many of the same topics brought up in this thesis.

One fundamental requirement for applying a DP method is that the problem satisfies *Bellman's principle of optimality*, i.e. that:

> An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.[8, Chap III.3]

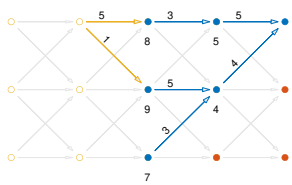
An example of a problem that satisfies the principle of optimality is finding the fastest route between two cities by car. If for instance the shortest path between Stockholm and Gothenburg passes through Jönköping and then Borås, then the shortest path between Jönköping and Gothenburg must also pass through Borås. Essentially, for problems that satisfy the principle of optimality the optimal choice at any given time does not depend on the history of how we arrived to the current situation, but only depends on the present situation (i.e. state). The type of problem we will study in this section — graph optimization — does satisfy the principle of optimality, as we will see shortly.

An example of a problem that does *not* satisfy the principle of optimality is that of finding the least expensive airline tickets between two locations. Assume that the cheapest flight from Stockholm to Gothenburg were to include layovers in Jönköping and Borås. Unlike the car example, it is not necessarily true that the cheapest flight from Jönköping to Gothenburg would also pass through Borås, as airline ticket prices are not usually the sum of each individual flight.

Ultimately, DP is a structured method of finding and storing the best path between connected "locations" (in a general sense) in an efficient way. Despite this simplicity, or perhaps because of it, DP is a very powerful tool that is used in a wide range of applications. In this section we will show how to use DP as a method for solving optimal control problems.

DP in graph optimization

The perhaps most straightforward application for DP is solving a given graph optimization problem. A graph is structure with a set of *vertices*, connected by *edges* with an associated cost, denoted as the *stage cost* (i.e. the cost of



moving from one vertex to a neighbor). A subway map is one example of a graph commonly seen in day-to-day life, with stations (vertices) that are connected to each other by different subway routes (edges), and the stage cost for instance defined as the time it takes to travel from one station to its neighbor. In the context of dynamic programming for optimal control we will be specifically interested in *directed graphs*, where edges only allow traveling from one vertex to another in one direction.

An example of the particular type of directed graph that we will use for DP is shown in Figure 2.14. Vertices are indicated with circles, each painted with one of three different colors that the DP method will later make use of; yellow – unvisited, blue – valid, red – invalid. Edges are shown with gray arrows, with arrowheads indicating permissible directions of movement. For ease of introducing the DP method, we will graphically place the vertices on a grid, with an equal number of vertices in each column and row. Importantly, the edges are constructed to allow traversing the graph from left to right, and any given path will successively pass through each column, while the path through rows is arbitrary. Assume our goal is to determine how to optimally move from one vertex in the leftmost column of vertices to the blue vertex in the rightmost, where each edge has an associated stage cost. For this goal, DP is one method of generating the optimal route (also referred to as the optimal decision policy) that minimizes the total cost of traveling through the graph.

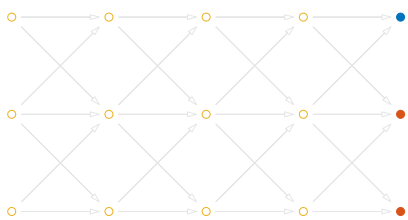


Figure 2.14: Initial directed graph. The stage cost is not shown here to reduce clutter. The back-calculation phase of the DP method we will introduce will make use of the different vertex colors.

The DP process can be divided into two stages. The first is commonly referred to as the *back-calculation phase*, where the optimal edges are determined for all vertices in the graph. In the second stage, referred to as the *forward-calculation*

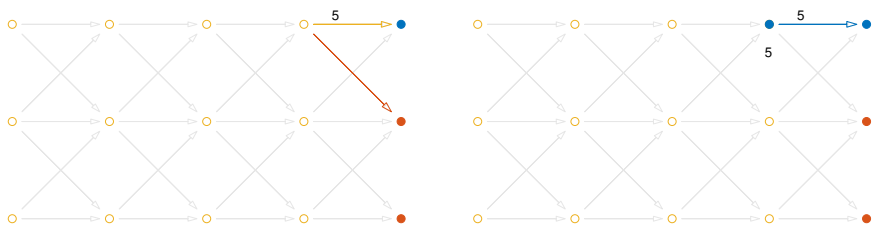
phase, the previous results are used to give the optimal path from any given initial vertex[6].

Back-calculation phase

In the back-calculation phase the optimal path for any vertex in the graph is determined iteratively, starting with the last column of unvisited vertices and successively approaching the first column of vertices.

In the first iteration, we compute the cost from the last column of unvisited vertices (the fourth column in Figure 2.14) to all subsequently connected valid vertices (the rightmost column in Figure 2.14).

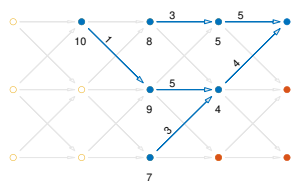
For each vertex we check if any connected vertices are valid, and if so store the edge with the lowest total cost to reach the last column of vertices and its associated cost value, which we will denote as the *cost-to-go*. For the first iteration, the cost-to-go is simply the stage cost (as we immediately reach the last column of vertices). In Figure 2.15 and Figure 2.16, only one connected node is valid, so we store the edge that leads to the valid node and the stage cost (shown above the highlighted edge) of the chosen edge is stored (shown under the marked vertex).

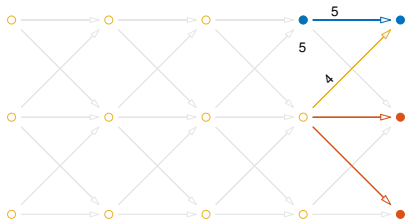


(a) Only one edge is valid... **(b)** ... so we chose it, storing the cost-to-go.

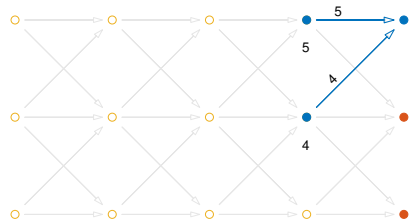
Figure 2.15

[6]Though most DP methods start with a back-calculation and end with a forward-calculation phase, this is not strictly necessary and one could construct a DP method that operates in the reverse order. We will start with the back-calculation phase as this allows us to generate the optimal path for any initial vertex (analogous to the initial condition), in contrast to a reversed method where we could instead generate the optimal path for any terminal vertex.





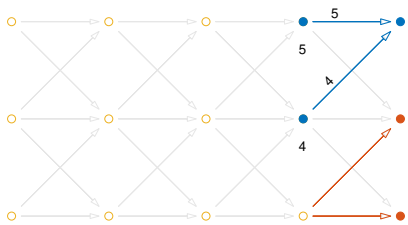
(a) Only one edge is valid...



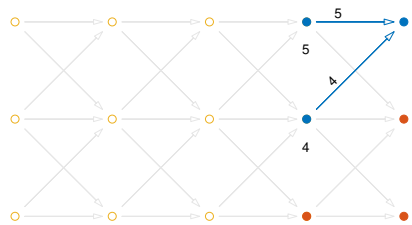
(b) ... so we chose it, storing the cost-to-go.

Figure 2.16

If no connected vertices are valid (2.17a) we mark the vertex as invalid (2.17b).



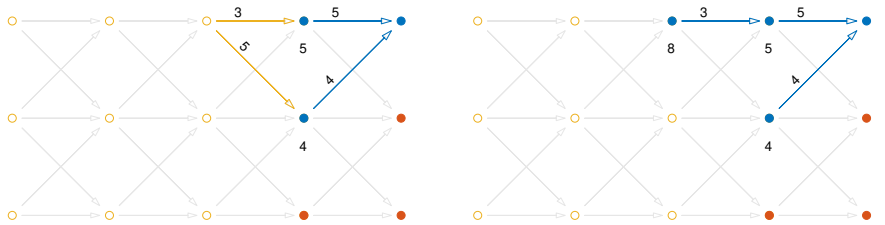
(a) No edges are valid...



(b) ... so we mark the vertex as invalid.

Figure 2.17

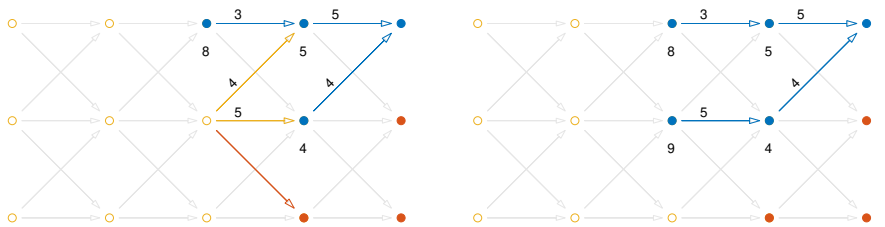
In the second, and all successive iterations, the total cost becomes the sum of the stage and previously stored cost-to-go. In Figure 2.18 we are presented with two options: choosing the upper edge would result in a total cost of $3 + 5 = 8$, while the lower edge would result in a total cost of 9, indicating that the optimal choice is to choose the upper edge.



(a) The upper edge gives a lower total cost... **(b)** ... so we chose it, storing the cost-to-go.

Figure 2.18

It is possible that the total cost is identical for several edges from a single vertex. There are several ways of managing this scenario, but as both choices are equally good one simple method is to arbitrarily select one edge, as shown in Figure 2.19.



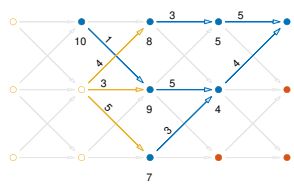
(a) The edges have identical total cost... **(b)** ... so we here just choose one.

Figure 2.19

Notably, the result from the back-calculation phase has several very strong properties;

- The fully solved graph (Figure 2.20) contains the optimal edges to follow for *any* point in the graph.

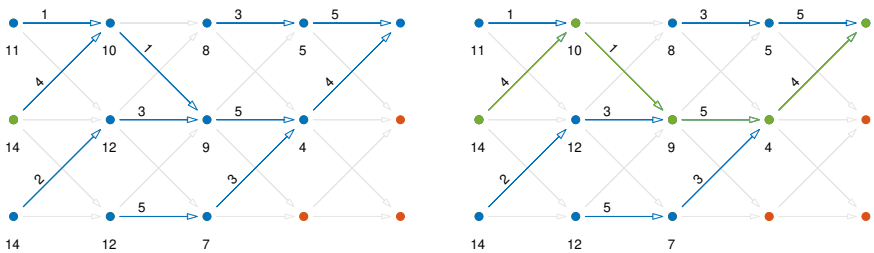
- We know if any given vertex can or cannot lead to the final vertex — a vertex marked blue is guaranteed to reach the final vertex, while we are ensured that a red-marked vertex can not reach the final vertex.



A flipbook animation of the entire back-calculation phase is shown along the bottom edge of the left pages this thesis. View it by holding the thesis in your left hand and flipping through the pages starting from the table of contents.

Forward-calculation phase

Once the backward-calculation phase has been completed and all vertices have been marked as either valid or invalid, the (trivial) forward calculation phase can be performed. Here, we can start from any position in the graph and simply follow the edges that were stored in the back-calculation phase to find an optimal path. For example, in Figure 2.20a we start at the second vertex in the first column (marked green). We can then follow the stored path through the graph and are ensured that we will end up at the rightmost valid vertex with a cost-optimal path (Figure 2.20b).



(a) Start from any vertex... **(b)** ...and follow the stored edges.

Figure 2.20: First (2.20a) and last (2.20b) forward-calculation iteration, shown with green vertices and edges.

2.3.3 DP in optimal control

Recall our original problem eq. (2.1) of solving an optimization problem whose dynamics are of form

$$x_{n+1} = d(x_n, u_n). \tag{2.11}$$

As $x \in \mathbb{R}^l$ and $u \in \mathbb{R}^m$ they are thus continuous-valued. We can refer to eq. (2.11) as a *continuous-value* (CV) problem.

The previously-defined DP method defined allows us to solve an optimization problem for a directed graph, which is a discrete problem (i.e. there are a

finite number of vertices and edges). We must thus first approximate the CV problem as a *discrete-value* (DV) problem, i.e. a problem where x_n and u_n are discretized, in order to use DP to solve the approximated problem.

In this section we will perform this conversion by limiting x_n and u_n to be taken from finite discrete sets of evenly-spaced points, i.e. we demand

$$x_n \in \mathcal{X} \tag{2.12a}$$
$$u_n \in \mathcal{U} \tag{2.12b}$$

where

$$\mathcal{X} = \{(x^1, x^2, \ldots, x^l) | x^1 \in \mathcal{X}^1, x^2 \in \mathcal{X}^2, \ldots, x^l \in \mathcal{X}^l\} \tag{2.12c}$$
$$\mathcal{U} = \{(u^1, u^2, \ldots, u^m) | u^1 \in \mathcal{U}^1, u^2 \in \mathcal{U}^2, \ldots, u^l \in \mathcal{U}^m\} \tag{2.12d}$$

and

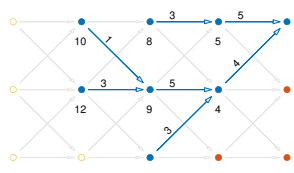
$$\mathcal{X}^i = \{\bar{x}^i, \bar{x}^i + \Delta^{xi}, \bar{x}^i + 2\Delta^{xi}, \ldots, \bar{x}^i + N_{xi}\Delta^{xi}\} \tag{2.12e}$$
$$\mathcal{U}^j = \{\bar{u}^j, \bar{u}^j + \Delta^{uj}, \bar{u}^j + 2\Delta^{uj}, \ldots, \bar{u}^j + N^{uj}\Delta^{uj}\}. \tag{2.12f}$$

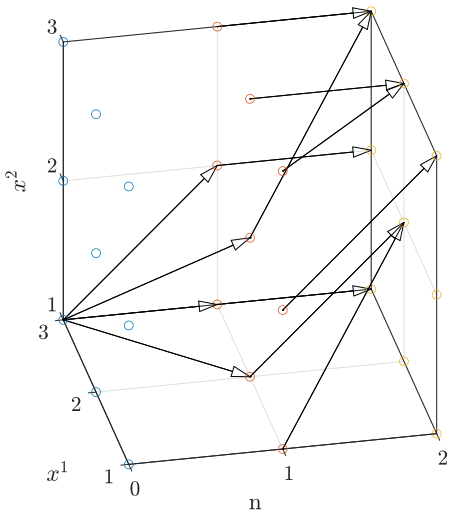
Using this definition, where $\square^i$ indicates the i'th dimension of $\square$, $\mathcal{X}$ and $\mathcal{U}$ are the Cartesian product of l and m sets of discrete evenly-spaced values respectively. We can thus view $\mathcal{X}$ and $\mathcal{U}$ as l- and m-dimensional grids of points, evenly spaced, with different spacing in each dimension. For now, assume that $x_{n+1} = d(x_n, u_n)$ also lies in $\mathcal{X}$. We will return to the typical case where this does not hold in the following section.

With x_n and u_n discretized, we can thus construct the associated graph problem by assigning each element of $\mathcal{X}$ to a vertex and each element of $\mathcal{U}$ to an edge. From a given state configuration (i.e. vertex) we can thus use a control (i.e. edge) to bring us to a new state (another vertex) given by the system dynamics d and with an associated cost given by c.

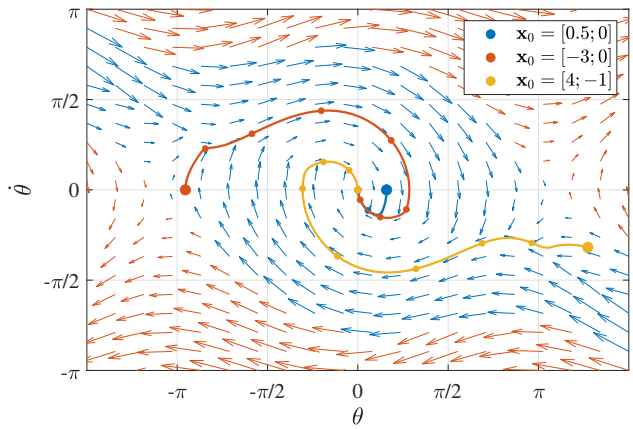
An illustration of this is shown in Figure 2.21a for $l = 2$ and $\mathcal{X}^1 = \mathcal{X}^2 = \{1, 2, 3\}$ (i.e. $\bar{x}^1 = \bar{x}^2 = 1$, $N_{x1} = N_{x2} = 3$, and $\Delta^{x1} = \Delta^{x2} = 1$). Blue, red, and yellow-colored vertices here correspond to samples $n = 1$, $n = 2$, and $n = 3$ respectively. Edges from $n = 0$ to $n = 1$, shown for $x^1 = 3, x^2 = 1$, exemplify four possible transitions for four different control signals u_1. Edges from $n = 1$ to $n = 2$ show representative optimal edges. The stage cost for each edge is not shown to reduce visual clutter.

We can now make an important intuitive connection between the graph used by DP and the evolution of the dynamic system. If we project the edges





(a) Graph structure for problem with two state variables.



(b) The time-optimal pendulum phase portrait, repeated from Figure 2.11.

Figure 2.21: The graph structure used by the DP solver and the phase portrait are similar representations of what can be seen as the same process.

associated with any pair of consecutive samples n in Figure 2.21a onto the plane, we will generate a phase portrait analogous to Figure 2.21b. In the case of Figure 2.21a, where the optimal edges vary with the sample n, this corresponds to a sample-varying phase portrait. Finally, note that though the positions of the vertices in Figure 2.21a help for our understanding of the problem, the position of a node in a graph is ultimately cosmetic — a graph is fundamentally only influenced by the connections between vertices.

Returning to the control problem, we can encode the constraint eq. (2.1e) by augmenting the cost function as

$$c'(x,u,n) = \begin{cases} c(x,u,n) \text{ for } g(x,u,n) \leq 0 \\ \infty \text{ otherwise.} \end{cases} \tag{2.13}$$

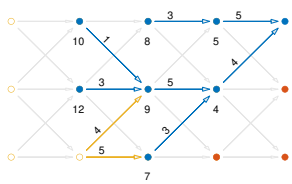
Clearly, by assigning an infinite cost to impermissible state and/or control configurations we are ensured that the resulting control will never violate $g(x,u,n) \leq 0$ if a valid solution exists. This also allows us to manage the case where c, d, and/or g are undefined for some arguments, and can simply set $c' = \infty$ to avoid a solution entering a problematic region.

With this construction we can thus directly apply the DP method described in Section 2.3.2. Furthermore, we can draw conclusions as to the nature of the solution while performing the back-calculation phase. If the total cost J^* was infinite, there no solution exists that satisfies $g(x,u,n) \leq 0$. Furthermore, if at any stage during the back-calculation phase we had a situation where two paths were associated with the same cost (as in Figure 2.19) then we know that the solution is not unique (and vice-versa otherwise know that the solution is unique).

To summarize, we can use DP to generate a solution to the DV-optimal control problem of form eq. (2.1), as follows;

1. Choose a state and control grid $\mathcal{X}$ and $\mathcal{U}$. An appropriate choice of $\mathcal{X}$ and $\mathcal{U}$ (i.e. the discretization of x_n and u_n) is problem-dependent, a reasonable initial might be to let N_{xi} and N_{uj} be on the order of 10–1000, and let the spaces $\mathcal{X}^i$ and $\mathcal{U}^i$ span be given by the space where $g(x,u) \leq 0$.

2. Perform the back-calculation phase of DP. This will typically be computationally demanding, but can be done off-line, and the results stored as a graph or table.



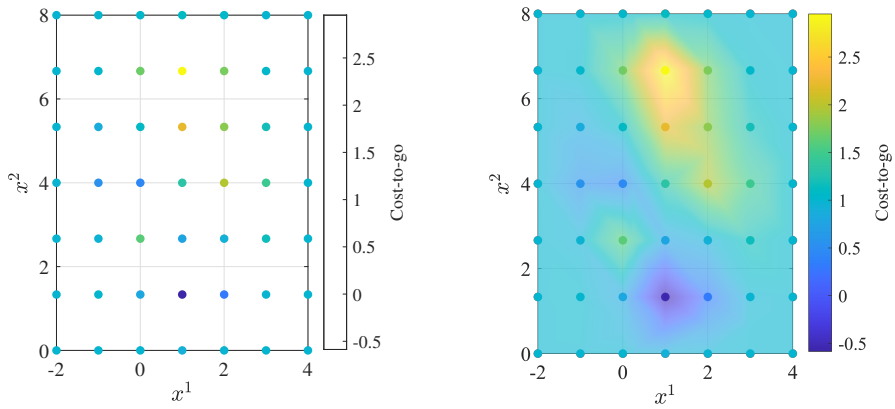
3. When the controller is used, we perform the forward-calculation step in real-time, looking up the optimal control to apply using the current state and sample. This is computationally inexpensive (just a table look-up), requiring very little in the way of computational power, and little memory when l and m are small (we will return to how the memory demand scales with l and m later).

Approximate Dynamic Programming

In this section we will introduce one method in the Approximate Dynamic Programming (ADP) family. The family of ADP methods are a subset of DP methods, and are typically used to resolve cases where some form of information is missing or uncertain. In this thesis, we will use an ADP method that uses interpolation to generate an approximation of the cost-to-go and optimal control functions. See for instance [10, p. 137, p. 144] for more details on interpolation in ADP. For a more general introduction to ADP and it's range of applications in general, the classics [11], [12] are excellent, as is [13].

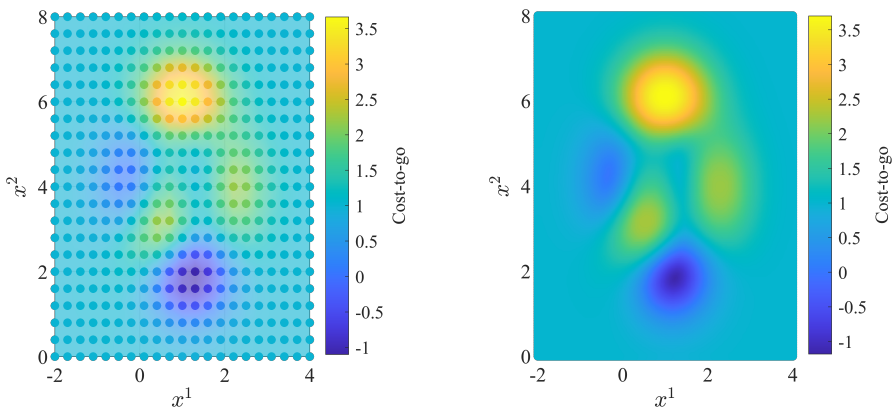
One unresolved issue with the previous scheme of using DP for solving an optimal control problem is that even if $x_n \in \mathcal{X}$ and $u_n \in \mathcal{U}$, we will typically have $d(x_n, u_n) \notin \mathcal{X}$. This is problematic, as if the system evolves to points that are not in $\mathcal{X}$ then we do not know the cost-to-go value in the back-calculation phase and the optimal control in the forward-calculation phase. One method of resolving this is to use interpolation to generate approximations of the cost-to-go and optimal control functions with continuous domains. This is exemplified in Figure 2.22a, where a representative cost-to-go (as computed in the back-calculation phase) is shown for the discrete points in a representative $\mathcal{X}$. Using interpolation allows us to approximate a cost-to-go function with continuous domain as shown in Figure 2.22b. There are many different interpolation schemes that can be used (linear, polynomial, spline, etc), but this thesis uses multi-linear interpolation to reduce the computational demand of the interpolation operation.

Naturally, the accuracy of the interpolated cost-to-go depends on the nature of the "true" (continuous-domain, and hard to determine) cost-to-go function, but tends to improve as the discretization Δ^{xi} decreases. This is illustrated in Figure 2.23a, where the number of discretized points is increased by a factor of three for each state variable. For this cost-to-go function, we can see that the



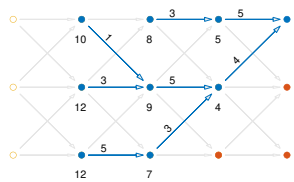
(a) Discrete **(b)** Interpolated

Figure 2.22: Cost-to-go for the discrete and continuous-domain (i.e. interpolated) cases at a given back-calculation iteration.



(a) Tripled resolution **(b)** "True" cost-to-go

Figure 2.23: Increasing the state resolution gives an interpolated cost-to-go that is closed to the true cost-to-go.



case with improved resolution in Figure 2.23a is closer to the "true" cost-to-go in Figure 2.23b than the base case in Figure 2.22b.

Much as with the cost-to-go, we will also face a similar issue in the forward-calculation phase when we need to determine the optimal control signal to apply, as the stored optimal controls are only known at the discrete points in $\mathcal{X}$. This can be resolved by using interpolation in the same manner as above, where an (approximately) optimal control signal is generated by interpolating among the computed controls.

More formally, for any (continuously-valued) state and control we can approximate the cost-to-go and control as

$$c_{\text{ctg,interp}}(x) = \sum_{x' \in \mathcal{X}} h_c(x, x') c_{\text{ctg,stored}}(x') \tag{2.14}$$

$$u^*_{\text{interp}}(x) = \sum_{x' \in \mathcal{X}} h_u(x, x') u_{\text{opt}}(x') \tag{2.15}$$

where h_c and h_u are the weighting functions for the chosen interpolation method and $c_{\text{ctg,stored}}$ and u_{opt} are the stored cost-to-go and optimal control from the back- and forward-calculation phases respectively.

Example problem

Consider the optimal control problem

$$(x^*, u^*) = \arg\min_{x,u} J \tag{2.16a}$$

$$J^* = \min_{x,u} \tag{2.16b}$$

where

$$J = \sum_{n=0}^{6} |x_n| \tag{2.16c}$$

subject to

$$x_{n+1} = x_n + u_n \tag{2.16d}$$
$$x_7 \geq 1 \tag{2.16e}$$
$$x_n \geq -1 \tag{2.16f}$$
$$x_n \leq 2 \tag{2.16g}$$
$$|u_n| \leq 1 \tag{2.16h}$$

where x and u are scalars. By augmenting the cost function we can express this problem as

$$(x^*, u^*) = \arg\min_{x,u} J \tag{2.17a}$$
$$J^* = \min_{x,u} J \tag{2.17b}$$

where

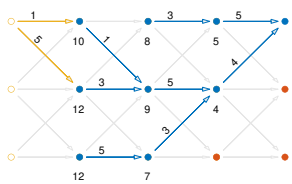
$$J = \sum_{n=0}^{6} c(x, u, n) \tag{2.17c}$$
$$c(x, u, n) = \begin{cases} \infty \text{ if } n = 7 \text{ and } x < 1 \\ \infty \text{ if } x < -1 \\ \infty \text{ if } x > 2 \\ \infty \text{ if } |u| > 1 \\ |x| \text{otherwise.} \end{cases} \tag{2.17d}$$

subject to

$$d(x, u) = x + u. \tag{2.17e}$$

Now, we must select the sets $\mathcal{X}$ and $\mathcal{U}$. It is prudent to chose these sets so that their elements satisfy the constraints in eq. (2.16). We can do this by for instance defining $\mathcal{X} = \{-1, 0, 1, 2\}$ and $\mathcal{U} = \{-1, 0, 1\}$. (The low number of elements in $\mathcal{X}$ and $\mathcal{U}$ is intentional in this specific example.)

With this particular graph construction we arrive at a graph problem that looks similar to the introductory example eq. (2.1). We can select elements in $\mathcal{X}$ and $\mathcal{U}$ and (except for the case $x = -1, u = -1$ and $x = 2, u = 1$) applying them to d will result in a new state that also lies in $\mathcal{X}$.



The solution to eq. (2.17) using DP and the sets $\mathcal{X}$, $\mathcal{U}$ as defined above is shown in Figure 2.24. We can identify that $x^*_{[0,1,\dots,6]} = 0$ (giving the smallest possible cost during this interval), while $x^*_7 = 1$ (satisfying the terminal inequality) and is achieved using $u_{[0,1,\dots,5]} = 0$, $u_6 = 1$. This matches our intuition of minimizing $|x|$, and this solution is in fact the true optimal solution. However, the optimality of our solution is dependent on our selection of $\mathcal{X}$ and $\mathcal{U}$.

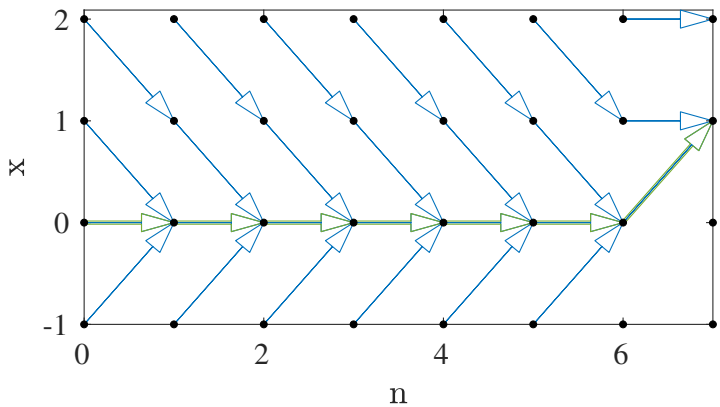


Figure 2.24: Solution to optimal control problem eq. (2.17) for $\mathcal{X} = \{-1, 0, 1, 2\}$ and $\mathcal{U} = \{-1, 0, 1\}$. Blue arrows show the back-calculation results, while the forward-calculation results are shown in green.

In the previous example we chose $\mathcal{X}$ and $\mathcal{U}$ to (except for boundary cases) give $x_{n+1} = d(x_n, u_n) \in \mathcal{X}$. If we had instead used $\mathcal{U} = \{-1, -0.5, 0, 0.5, 1\}$ then $u = \pm 0.5$ would not give a new state in $\mathcal{X}$. An illustration of this is shown in Figure 2.25, which illustrates the back-calculation phase at $x_4 = 0$, $\mathcal{X} = \{-1, 0, 1, 2\}$, and $\mathcal{U} = \{-1, -0.5, 0, 0.5, 1\}$. This illustration is not formally a directed graph, as we have two edges that do not lead to associated vertices. However, with the specific structure of this graph we can interpret the vertical position as continuously indicating the state value, where vertices indicate points where the cost-to-go is known.

Applying $u_4 \in \{-1, 0, 1\}$ (shown in yellow) to 2.17e gives a result that is in $\mathcal{X}$, so we can directly use their stored cost-to-go to compute the total cost. However, for $u_4 = \{-0.5, 0.5\}$, (shown in red) 2.17e does not give a result that is in $\mathcal{X}$, meaning that we do not have a stored cost-to-go to use. In our ADP

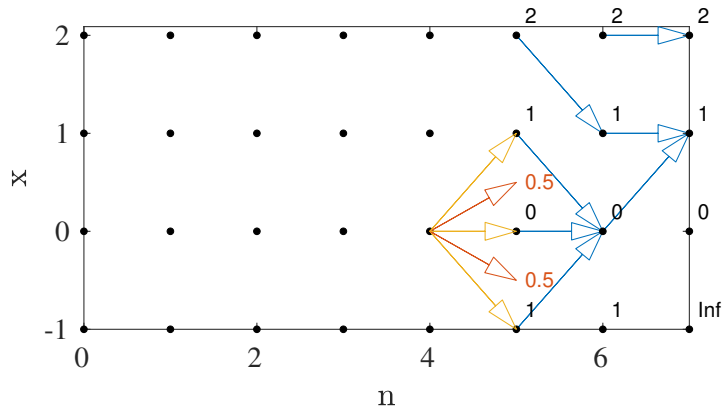
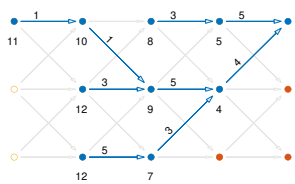


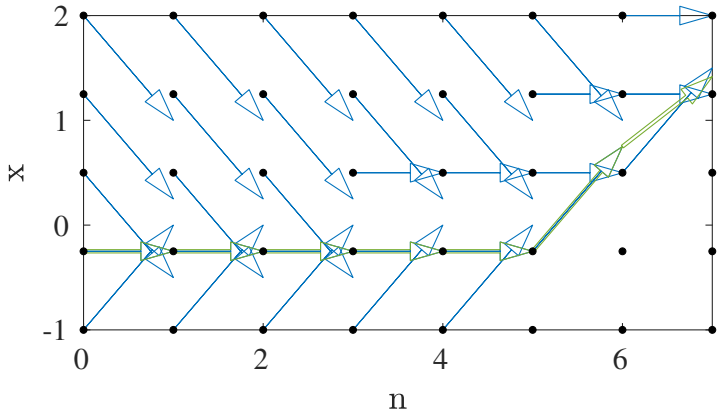
Figure 2.25: During back-calculation phase for problem in eq. (2.17), five different u_4 are tested for $x_4 = 0$. Yellow edges align with existing vertices, allowing for directly consulting the stored cost-to-go (black text), while red edges do not. Here linear interpolation is used to generate an approximate cost-to-go for red edges (red text).

method, we use linear interpolation to give an approximate cost-to-go of 0.5 for both cases. Similarly, during the forward-calculation phase we may end up applying a stored optimal control that brings us to a point that lies "between vertices" in the psuedo-graph. In this case we can use the scheme to select an approximately optimal control based on the optimal controls for the nearby vertices.

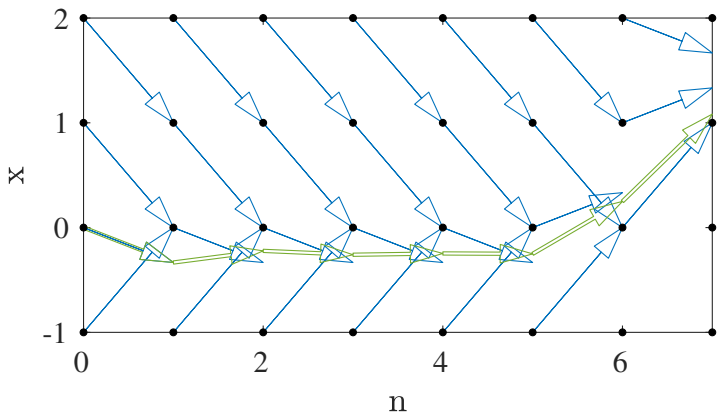
Another potential problem we have not yet discussed in depth is that by selecting $\mathcal{X}$ and $\mathcal{U}$ we have limited the search range for the problem solution. For instance, we can never find a solution where $x_n = 0.5$ if $\mathcal{X} = \{-1, 0, 1, 2\}$ and $\mathcal{U} = \{-1, 0, 1\}$. This issue is due to the quantization of $\mathcal{X}$ and $\mathcal{U}$, and is inherent when using DP to solve the DV approximation of our initial CV problem. An example of this is shown in Figure 2.26, where an imperfectly chosen $\mathcal{X}$ or $\mathcal{U}$ gives solutions that are suboptimal with respect to the continuous-valued problem eq. (2.17).

In general, for meaningful non-linear problems, converting the original CV-problem to a DV-problem will both introduce interpolation and quantization suboptimality, i.e. our resulting solution will no longer be optimal. However, if c and d are continuous we can limit the suboptimality by increasing the





(a)



(b)

Figure 2.26: Suboptimal solutions caused by (Figure 2.26a) imperfectly selected $\mathcal{X} = \{-1, -0.25, 0.5, 1.25, 2\}$ and (Figure 2.26b) imperfectly selected $\mathcal{U} = \{-1, -0.25, 0.25, 1\}$. Compare with optimal solution in Figure 2.24.

resolution of $\mathcal{X}$ and $\mathcal{U}$ (i.e. decreasing Δ_{xi} and Δ_{uj}). Note that we can generate solutions to problems where c and/or d is non-continuous (which we in fact did for c in eq. (2.17d)), at cost of poor interpolation performance near the discontinuity. An example of how increasing the resolution of $\mathcal{X}$ and $\mathcal{U}$ improves the resulting accuracy is shown in Figure 2.27, where $\mathcal{X}$ and $\mathcal{U}$ contain 100 linearly distributed elements in the range $[-1, 2]$ and $[-1, 1]$ respectively. Notably, $\mathcal{U}$ does not contain the "true" optimal element $u = 0$. Similarly, $\mathcal{X}$ does not contain the "true" optimal elements $x = 0$ and $x = 1$. Despite this, the dense sampling of $\mathcal{X}$ and $\mathcal{U}$ gives a solution that is close to the true optimal solution.

In summary, we can approximately solve general non-linear optimal control problems by approximating them as DV problems, converting them to an equivalent graph, and then using interpolation (from ADP) to solve the graph optimization problem. Applying this scheme gives a result that can be seen as a simple interpolated table look-up. The approximate solution can be a fairly accurate despite the interpolation and quantization suboptimality so long as d and c are well-behaved, and interpolating the cost-to-go and optimal control between points in $\mathcal{X}$ and $\mathcal{U}$ gives a reasonable approximation of these functions.

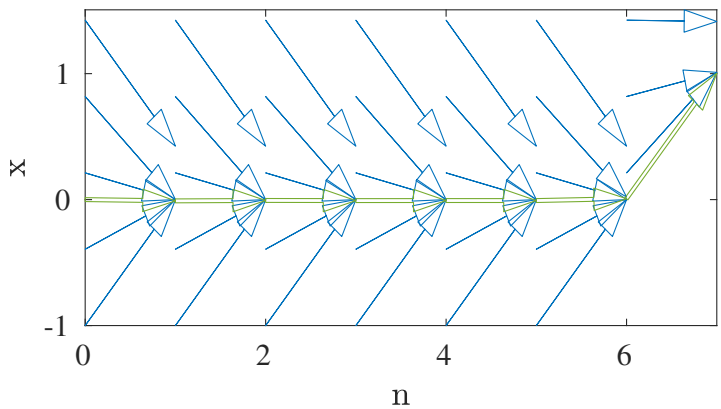
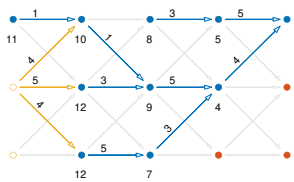


Figure 2.27: As the number of elements in $\mathcal{X}$ and $\mathcal{U}$ grows the solution suboptimality decreases, even if $\mathcal{X}$ and $\mathcal{U}$ are imperfectly selected. Only a subset of the blue back-calculation edges are illustrated to reduce clutter.



Revisiting the pendulum

Now that we have established a DP method for solving nonlinear optimal control problems, we have all the tools needed to solve the pendulum control problem studied in Section 2.3.1. This section will summarize the numerical set-up used to generate the previously presented results.

Recall the nonlinear minimum-time problem in eq. (2.10). One issue we find when converting it to the structure of eq. (2.1) is that the problem is formulated with an infinite time horizon, in contrast to the explicit time horizon in eq. (2.1). One method of resolving the time horizon for this specific problem is to let the horizon N be a large, but finite, value. As the optimal control for this problem is to drive the pendulum to a vertical stationary state as quickly as possible, and then keep the pendulum there, the optimal control will also become stationary for a sufficiently large horizon N. See Paper B for more details on how (and which) infinite-horizon problems can be converted to and solved as finite-horizon problems, as well as a more efficient method of performing this conversion.

We can thus reformulate eq. (2.10) as

$$(x^*, u^*) = \arg\min_{x,u} J \tag{2.18a}$$

$$J^* = \min_{u} J \tag{2.18b}$$

$$J = \sum_{n=0}^{N} c'(x, u) \tag{2.18c}$$

where

$$c'(x,u) = \begin{cases} \infty \text{ if } |u| > 0.5 \text{ or } |\theta| > 1.5\pi \text{ or } |\dot{\theta}| > \pi \\ 0 \text{ if } x = 0 \\ 1 \text{ otherwise} \end{cases} \tag{2.18d}$$

subject to

$$x_{n+1} = d_{\text{pend}}(x_n, u_n) \tag{2.18e}$$

for a sufficiently large N.

The results in Section 2.3.1 and Figure 2.11 in particular shows the solution to eq. (2.18) using ADP. Here, $\mathcal{U}$ contains 101 linearly distributed points in the range ± 0.5, $\mathcal{X}$ contains the outer product of 601 linearly distributed points for

x_1 in the range $\pm 1.5\pi$ and 401 linearly distributed points for x_2 in the range $\pm\pi$, and $N = 200$.

Strengths and weaknesses of ADP

Now that we have introduced ADP in the context of optimal control, it is worth highlighting some of the strengths and weaknesses of the method.

The strengths of ADP include:

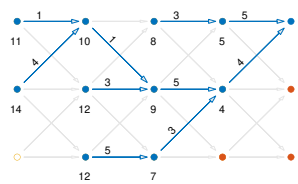
- the ability to solve problems with nonlinear time-varying system dynamics, cost, and constraints (so long as the inherent quantization and interpolation give reasonably correct solutions),
- no need for computing any gradients, Hessians, etc.,
- a solution that approximates the globally-optimal solution (limited by the error introduced by the discretization and interpolation approximation),
- the capacity to solve hybrid control problems[7],
- few tuning parameters (only the selection of $\mathcal{X}$, $\mathcal{U}$, and an interpolation method),
- and, under some conditions, trivially implemented on-line control by table look-up[8].

However, there are weaknesses with DP methods. As introduced in Section 2.3.3, converting the CV problem to a DV problem with ADP typically approximates the solution due to the quantization and interpolation.

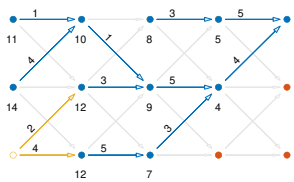
Furthermore, the typically most significant weakness is DP's exponential scaling of computational complexity with the number of state and control variables. This is colloquially referred to as the *curse of dimensionality*. Recall that at each sample n in the back-calculation phase all elements of $\mathcal{U}$ are tested at every element of $\mathcal{X}$. This implies that the computational demand scales exponentially with the sum of the dimensionality of x and u, as adding

[7] A problem where the state and/or control variables consist of both continuous-valued and discrete-valued variables.

[8] For example, in the pendulum example the optimal control signal is determined solely using the current system state (as shown in Figure 2.11), and the approximately-optimal control signal can be determined simply by interpolating from the stored (gridded) optimal control signals



a new dimension implies that all combinations of the existing dimensions must be tested for all points in the new dimension. Though there do exist some methods that reduce the computational complexity of DP (see for example Paper A), with today's computing power there seems to be a practical limit of 3–6 state+control variables (depending on the resolution of $\mathcal{X}$ and $\mathcal{U}$) for most CV optimal control problems.


1
3
5
11
10
8
5
7
4
4
3
5
14
12
9
4
2
4
5
3
12
7

CHAPTER 3

Summary of included papers

This chapter provides a summary of the included papers.

3.1 Paper A

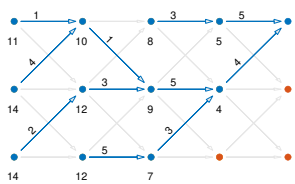
Jonathan Lock, Tomas McKelvey
A Computationally Fast Iterative Dynamic Programming Method for Optimal Control of Loosely Coupled Dynamical Systems with Different Time Scales
Published in IFAC-PapersOnLine,
vol. 50, issue 1, pp. 5953–5960, July 2017.
DOI: 10.1016/j.ifacol.2017.08.1498 .

Dynamic programming methods are used to solve a wide range of optimal control problems. However, one of the more significant drawbacks with dynamic programming methods is their exponential increase in computational demand with the number of state variables. This paper introduces a general method for reducing the computational demand when the state variables are loosely coupled and display different time scales, i.e. where one state variable is varies

significantly more quickly than the other and minor changes to the quickly-varying state variable has only a moderate effect on the slowly-varying state variable.

The proposed method consists of two stages. In the first stage a simplified problem is solved where the quickly-varying state variable is viewed as a control input, i.e. without dynamics. This reduces the dimensionality of the problem, and makes for a problem that is significantly easier to solve. Importantly, solving this simpler problem results in a state trajectory for the slowly-varying state variable. In the second stage we solve the original problem, but can dramatically reduce the computational demand by limiting the evaluated states for the slowly-varying state variable to a region near the solution found in the first phase. Typically this results in a computational speed improvement on the order of the quotient of the state variable's time scales. Furthermore, the method introduces some regularization terms that help reduce the risk of generating an infeasible solution in the forward-calculation phase of the second stage, which can occur when interpolating a control signal near the edge of the set of evaluated states.

The proposed method is in the context of this thesis primarily intended to be used as a tool to generate off-line control solutions, which can for instance be used to benchmark a subsequent on-line control scheme. In this paper we exemplify this by finding the optimal ratio of electrical to combustion engine power when the battery state of charge and engine-generator crankshaft inertia dynamics of the series-hybrid are modeled. This problem displays significantly different time scales, as the battery state of charge typically varies over an hour, while the crankshaft angular velocity varies over seconds. The proposed method solves the optimal control problem approximately 2700 times faster than a direct approach when using typical battery and crankshaft dynamics, while generating a solution of comparable accuracy.



3.2 Paper B

Jonathan Lock, Tomas McKelvey
Undiscounted Control Policy Generation for Continuous-Valued Optimal Control by Approximate Dynamic Programming
Submitted to the International Journal of Control on July 1, 2020
Revised January 7, 2021 and 17 March, 2021.
Preprint available at `https://arxiv.org/abs/2104.11093`.

One crucial element of this thesis is a method for generating near-optimal controllers for real-time (i.e. on-line) systems. This paper introduces a general algorithm for generating a control policy that in turn can be used to implement a near-optimal real-time controller. The method is based on approximate dynamic programming and titled Undiscounted Control Policy generation by Approximate Dynamic Programming (UCPADP). UCPADP is applicable to undiscounted, constant-setpoint, infinite-horizon, nonlinear optimal control problems with continuous state variables. The method runs in an off-line phase and generates a control policy, i.e. a look-up table. Once the control policy has been generated an on-line controller can subsequently be implemented, where the optimal control signal can can be determined by a simple table look-up and interpolation operation, i.e. explicit nonlinear state-feedback. UCPADP is suited for general optimal control problems where the objective is to optimally transition a system to a given constant set-point with costs over an infinite horizon.

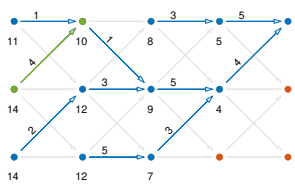
UCPADP's primary contribution is a termination criterion that is amenable to general undiscounted problems, i.e. problems where the cost function is not increasingly discounted as time progresses. UCPADP's termination criterion requires two conditions to be simultaneously satisfied: first, it requires the control law to be near-constant over a sufficiently large period (similarly to the well-studied discounted case); secondly, it also requires the system state to converge to a sufficiently small region near the target set-point for a large number of initial conditions spanning the space of feasible initial conditions.

In this thesis, UCPADP is used as a backend for generating optimal real-time controllers, as exemplified in Papers C and E.

3.3 Paper C

Jonathan Lock, Rickard Arvidsson, Tomas McKelvey
Optimal Transient Real-Time Engine-Generator Control in the Series-Hybrid Vehicle
Published in Dynamic Systems and Control Conference,
vol. 2, October 2019.
DOI: 10.1115/DSCC2019-8964 .

This paper studies an application of the UCPADP method introduced in Paper B. (Note that the UCPADP method was not fully complete at the time of writing this paper, and the method instead referenced to using the provisional name MHDP.) Here, we consider the problem of controlling the combustion engine and electric generator torques in the series hybrid with the goal of delivering a given electrical power and maximizing fuel-efficiency while taking the crankshaft angular velocity dynamics into account. Note that this problem is similar to the example studied in Paper A. The novelty of this controller lies in the optimality of the torques during the transient interval when the crankshaft angular velocity is non-stationary. Simulations show a reduction in fuel consumption of approximately 5–7% during the transient interval. The resulting controller was implemented natively in the vehicle's engine control unit (ECU), requiring only 10–20 CPU instructions per control period, a few bytes of RAM, and 5–20 KiB of nonvolatile memory. Experimental bench-tests showed a fuel consumption improvement of 3.7%, which was limited by the low maximum power of the electric generator and unmodeled turbocharger dynamics. In the context of this thesis, this paper highlights the potential for optimal control to improve the performance of just one of many subsystems, as well as showing the relative ease in which optimal control can be implemented.



3.4 Paper D

Jonathan Lock, Kristoffer Clasén, Jonas Sjöblom, Tomas McKelvey
A Control-Oriented Spatially Resolved Thermal Model of the Three-Way-Catalyst
Published in SAE WCX Digital Summit,
April 2021.
ISSN: 0148-7191 .

In this paper we develop a thermal three-way catalyst (TWC) model and calibrate it with experimental data. Due to the few number of state variables the model is well suited for fast offline simulation as well as subsequent on-line control. Using the model could allow an on-line controller to more optimally adjust the engine ignition timing, the power in an electric catalyst pre-heater, and/or the power split ratio in a hybrid vehicle when the catalyst is not completely hot. The model uses a physics-based approach and resolves both axial and radial temperature gradients, allowing for the thermal transients seen during heat-up to be represented far more accurately than conventional scalar (i.e. lumped-temperature) real-time models. Furthermore, it also use a physics-based chemical kinetics reaction model for computing the exothermic heat of reaction and emission conversion rate which is temperature and residence-time-dependent.

We have performed an experimental campaign with a standard spark-ignited engine and a commercial TWC, where we measured steady-state operation and cold-start transient behavior. This experimental data allowed us to tune the model, where we found excellent matching between the measured and modeled tailpipe emissions. Resolving the radial temperature gradient improved the relative accuracy of the conversion efficiency by 15% compared to a model that does not resolve a radial temperature profile, and simulations indicate the potential for an absolute improvement by 15 percentage points for some cases. Furthermore, the modeled TWC temperature evolution for a cold-start was typically within ± 10°C of the measured temperature (with a maximal deviation of 20°C). The proposed model thus bridges a gap between heuristic models suited for on-line control and accurate models for slower off-line simulation.

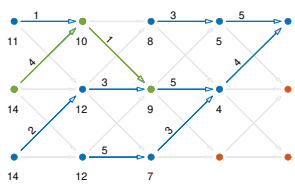
3.5 Paper E

Jonathan Lock, Kristoffer Clasén, Jonas Sjöblom, Tomas McKelvey
Cold-Start Modeling and On-Line Optimal Control of the Three-Way Catalyst
To be submitted
Preprint available at `https://arxiv.org/abs/2104.12390`.

In this paper, the TWC model presented in Paper D is extended in several ways, more extensively tested and validated, and used in combination with the UCPADP method presented in Paper B to construct a Pareto-optimal controller suitable for on-line operation. The model in Paper D is extended to support varying axial discretization lengths, uses tuning parameters expressed in SI units, models the heat generation from the oxidation of hydrogen gas generated by the combustion engine, and models a TWC constructed with two separate monoliths. In this paper we also expand on Paper D by tuning and validating the TWC using separate training and validation datasets.

Our experimental results show that the cumulative tailpipe emissions for a cold-start are typically predicted to be approximately -20% to +80% of the measured emissions. These figures are comparable with those of a significantly more complex model found in the literature whose cumulative accuracy varies between ±20% to ±50%. Furthermore, we hypothesize that the measurement equipment significantly contributes to the modeling error, indicating that the model's accuracy may be underestimated.

Using the model with the UCPADP method in Paper B allows for constructing a near-optimal cold-start controller that is Pareto-optimal with respect to each emission species and fuel-efficiency. The controller is well-suited for on-line control, as the required non-volatile memory can be kept to under 13.9 KiB and the computational operations consist of only a single multi-linear interpolation and linear filtering operation. We compared the simulated performance of an optimal controller with a "reasonable" suboptimal controller that operates the engine at a constant operating point during the heat-up phase. With the specific Pareto-optimal emission weighting studied, we found that the optimal controller generates NO_x emissions that are 35% lower than the suboptimal controller, while the CO and THC emissions and fuel efficiency are otherwise identical. Though the full potential for the model's accuracy and controller's efficacy is not known, these results indicate that there is the potential for practically implementing a more effective cold-start controller.



CHAPTER 4

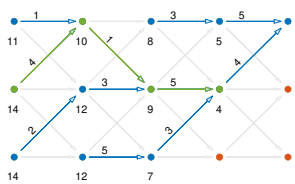
Concluding remarks and future directions

With an increasing level of available computational power there is a clear trend towards more complex control schemes. In this thesis, we particularly look into how near-future EMS systems might improve on the fuel efficiency and generated emissions of HEV's. This thesis contains contributions from a range of topics, from general optimal control methods that can be used for virtually any purpose, to a model of the TWC found in exhaust aftertreatment systems, and evaluations of the performance of optimal real-time controllers for improved fuel efficiency and reduced emissions. Though only scratching the surface of possible optimal control applications, this thesis shows some regions where additional work shows promise.

The optimal control methods presented in papers A and B show promise as general-purpose algorithms. As they are easily used and suited for a wide range of applications they could, in principle, be included in an optimization toolbox in MATLAB, Python, Octave, or other numerical computation software. The catalyst models in papers D and E focus on the TWC, as this is what is typically found in today's production HEVs. However, as the models are physics-based they can in principle be fairly easily adapted to other EATS,

such as an electrically heated TWC or the Selective Catalytic Reduction (SCR) catalyst found in diesel vehicles.

The automotive industry is currently undergoing a large shift, with a very rapid transition towards increased electrification. In the near future it seems likely that battery electric vehicles will become a larger proportion of the vehicle fleet. Though they do not have a combustion engine, there are still a wide range of possible applications for optimal control, for instance ranging from energy-optimal adaptive cruise control to improved drivability in semi-autonomous vehicles. Real-time optimal control is consequently something we can expect to see more of as time progresses.



References

[1] L. Guzzella and A. Sciarretta, *Vehicle Propulsion Systems, Introduction to Modeling and Optimization*, 3rd ed. Springer-Verlag, 2013.

[2] I. Husain, *Electric and Hybrid Vehicles, Design Fundamentals*. CRC press, 2021, ISBN: 9780367693930.

[3] B. de Jager, T. van Keulen, and J. Kessels, *Optimal Control of Hybrid Vehicles*. Springer-Verlag, 2013.

[4] J. B. Heywood, *Internal Combustion Engine Fundamentals*. McGraw-Hill, 1988, ISBN: 007028637X.

[5] W. W. Pulkrabek, *Engineering Fundamentals of the Internal Combustion Engine*. Pearson, 2003, ISBN: 9780131405707.

[6] R. Stone, *Introduction to Internal Combustion Engines*, 3rd ed. Macmillan, 199, ISBN: 978-0-333-74013-2.

[7] P. Kiwitz, C. Onder, and L. Guzzella, "Control-oriented modeling of a three-way catalytic converter with observation of the relative oxygen level profile", en, *Journal of Process Control*, vol. 22, no. 6, pp. 984–994, 2012, ISSN: 09591524.

[8] R. E. Bellman, *Dynamic Programming*. Princeton University Press, 1957.

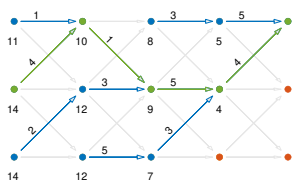
[9] A. Bryson Jr and Y. Ho, *Applied Optimal Control: Optimization, Sstimation and Control*. Blaisdell Publishing Company, 1969.

[10] R. E. Larson, *Principles of Dynamic Programming, Part I*. Marcel Dekker, 1978.

[11] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, 3rd ed. Athena Scientific, 2017, vol. 1.

[12] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, 4th ed. Athena Scientific, 2012, vol. 2.

[13] W. B. Powell, *Approximate Dynamic Programming: Solving the curses of dimensionality*. John Wiley & Sons, 2007, vol. 703.



Part II

Appended papers

PAPER A

A Computationally Fast Iterative Dynamic Programming Method for Optimal Control of Loosely Coupled Dynamical Systems with Different Time Scales

Jonathan Lock, Tomas McKelvey

Published in IFAC-PapersOnLine,
vol. 50, issue 1, pp. 5953–5960, July 2017.
DOI: 10.1016/j.ifacol.2017.08.1498

A Computationally Fast Iterative Dynamic Programming Method for Optimal Control of Loosely Coupled Dynamical Systems with Different Time Scales ⋆

Jonathan Lock * **Tomas McKelvey** **

* *Dept. of Signals and Systems, Chalmers University of Technology, SE-412 96 Gothenburg, Sweden, lock@chalmers.se*
** *Dept. of Signals and Systems, Chalmers University of Technology, SE-412 96 Gothenburg, Sweden, mckelvey@chalmers.se*

Abstract: Iterative dynamic programming is a powerful method that is often used to solve finite-dimensional nonlinear constrained global optimal control problems. However, multi-dimensional problems are often computationally complex, and in some cases an infeasible result is generated despite the existence of a feasible solution. A new iterative multi-pass method is presented that reduces the execution time of multi-dimensional, loosely-coupled, dynamic programming problems, where some state variables exhibit dynamic behavior with time scales significantly smaller than the others. One potential application is the optimal control of a hybrid electrical vehicle, where the computational burden can be reduced by a factor on the order of 100 – 10000. Furthermore, new regularization terms are introduced that typically improve the likelihood of generating a feasible optimal trajectory. Though the regularization terms may generate suboptimal solutions in the interim, with successive iterations the generated solution typically asymptotically approaches the true optimal solution.
Note: Full source code is freely available online with an implementation of the solver, some usage examples, and the test cases used to generate the results shown in this paper.

Keywords: Dynamic programming, Optimal control, Global optimization, Nonlinear control, Bang-bang control, Efficiency enhancement

1. INTRODUCTION

Non-causal global nonlinear constrained optimal control is a notoriously difficult problem which, in general, does not have a known analytical solution. Hence, it is often necessary to use numerical methods. One method that is often used for finite-dimensional problems is dynamic programming (DP). For example, DP is often used for designing hybrid vehicle controllers, where DP is typically used to benchmark the quality of simpler, suboptimal, causal controllers (Liu and Peng (2008); Pérez et al. (2006); Sciarretta and Guzzella (2007)). DP is guaranteed to generate the global optimum for problems that can be represented in a graph. However, DP is computationally complex for multidimensional problems, where the required number of computations scales exponentially with the number of dimensions.

This paper presents a new DP method (and an implementation of it in Matlab) for multidimensional problems that can be described as a loosely coupled set of ordinary differential/difference equations with different time scales. For problems of this type the presented method significantly reduces the time required to generate a solution, and furthermore increases the likelihood of generating a feasible solution. One example of an application that this method works well for is that of hybrid vehicle control, where performance gains on the order of the quotient of the system's time scales are realizable. Typically, this gives a performance improvement on the order of $10^2 - 10^4$.

In this paper DP is used to solve a discrete-valued, discrete-time approximation of a continuous-value problem in discrete- or continuous-time. The DP method used in this paper starts with a backward-calculation phase where, for a sample k, each element from a set of system inputs $\mathcal{U}_k$ is exhaustively applied to each element of a set of system states $\mathcal{X}_k$. The *best* control $u_{opt}[k]$ and corresponding cost $c_{opt}[k]$ are stored for every system state, where the *best* control and cost minimizes the total cost from the current sample to the final sample. This process is repeated for all samples starting from the next-to-last sample and working backwards to the first sample. The optimal control and state trajectories are generated in a forward-calculation phase, where for a given initial state the best stored control signal $u_{opt}[k]$ is successively applied to the system state $x[k]$ for all samples. Interpolation is used when the system state $x[k]$ does not exactly match one of the states evaluated during the back-calculation phase. This directly gives the optimal control and state trajectories $u_{opt}[k]$ and $x_{opt}[k]$. A formal definition of DP for optimal control is beyond the

⋆ This work has been performed within the Combustion Engine Research Center at Chalmers (CERC) with financial support from the Swedish Energy Agency.

scope of this paper, curious readers are referred to any of Bellman (1956); Bertsekas (2005); Sundström and Guzzella (2009).

1.1 Problem definition

For many engineering applications, typical optimal control problems are continuous-time, continuous-variable ($CTCV$) problems. Here, we consider the case where the control input function $u(t)$ from time t_k to time t_{k+1} is linearly parameterized with an l-dimensional control variable $u[k]$. If the function is the step function this representation is known as zero-order-hold sampling (Åström and Wittenmark, 1997, p. 32). This optimal control problem can then be represented as a discrete-time, continuous-variable ($DTCV$) problem, defined as

$$\begin{aligned} J^*_{0,N_s} &= \min_{\mathcal{U}} L_{0,N_s}(\mathcal{U}) \\ &\text{s.t.} \\ L_{0,N_s} &= \sum_{k=0}^{N_s} c(x[k], u[k], k) \\ x[k+1] &= f(x[k], u[k], k),\ k = [0, N_s - 1] \\ b_{in}(x[k], u[k], k) &\leq 0,\ k = [0, N_s] \\ x[k] &\in \mathbb{R}^m \\ u[k] &\in \mathbb{R}^l, \end{aligned} \tag{1}$$

where $x[k]$ is an m-dimensional vector of real-valued state variables and $u[k]$ is an l-dimensional vector of control inputs. The total cost function L_{0,N_s} is minimized with respect to $u[k]$, given the system dynamics $f(\dots)$ and a set of inequality constraints $b_{in}(\dots)$. Here, k is an index that orders the state and control variable trajectories, and for the cases considered here is directly proportional to time.

DP cannot directly be applied to solve (1). Instead, the problem is further approximated by quantizing the state and control variables which yields a discrete-time, discrete-variable ($DTDV$) form, i.e. $x[k]$ and $u[k]$ must each be members of a set with a finite number of elements, denoted $\mathcal{X}_k$ and $\mathcal{U}_k$ respectively. Each element of $\mathcal{X}_k$ can be viewed as a vertex in a directed graph (shown in Figure 1) corresponding to sample k, where the existence of an edge between an element $x[k] \in \mathcal{X}_k$ and an element $x[k+1] \in \mathcal{X}_{k+1}$ implies that there exists a feasible control $u[k] \in \mathcal{U}_k$ so that the constraints in (1) are fulfilled for $x[k]$, $x[k+1]$, and $u[k]$.

In (1), the system dynamics model is given in implicit form — $x[k+1]$ is generated with the function $f(\dots)$ given a state $x[k]$ and control $u[k]$ for sample k. This particular representation is chosen as it is typically difficult to generate a model in explicit form (i.e. of type $u[k] = g(x[k], x[k+1], k)$) in many applications. As a result of this representation, there is no guarantee in the back-calculation phase that applying a member of $\mathcal{U}_k$ to a member of $\mathcal{X}_k$ will generate a value $x[k+1] \in \mathcal{X}_{k+1}$. Similarly, during the forward-calculation phase, if $x[k] \notin \mathcal{X}_k$ then there does not exist an associated stored optimal control signal $u_{opt}[k]$ to apply. A method that resolves this issue is to define the existence of on-demand pseudo-vertices $\bar{\mathcal{X}}_k$, where any $x[k] \notin \mathcal{X}_k$ is defined to be an element of $\bar{\mathcal{X}}_k$, and whose numerical values are derived based on the nearby elements in $\mathcal{X}_k$ using some suitable interpolation method. Similarly, a pseudo-optimal control $\bar{u}_{opt}[k]$ can be generated based on the nearby stored optimal controls $\mathcal{U}_k$. If the elements in $\mathcal{X}_k$ are carefully chosen this becomes a computationally inexpensive gridded interpolation, e.g. nD linear interpolation (Bellman and Dreyfus (2015); Elbert et al. (2013)).

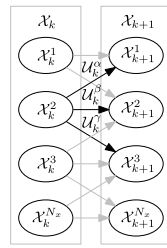


Fig. 1. Directed graph representation of a DTDV problem. For the state configuration $\mathcal{X}_k^2$, only the α'th, β'th and γ'th elements from $\mathcal{U}_k$ are feasible and bring the state to $\mathcal{X}_{k+1}^1$, $\mathcal{X}_{k+1}^2$, and $\mathcal{X}_{k+1}^3$ respectively at the next sample.

1.2 Iterative Dynamic Programming

Iterative dynamic programming (IDP), as defined by Luus (1990), can be used to solve real-valued optimization problems, i.e. problems where the state and control variables take values from the set of real numbers. IDP reduces the state and control quantization to an arbitrarily small amount by first searching over a relatively coarse but large set of system inputs and states using DP, and then successively generating a denser and narrower search range centered about the previous result. This successive reduction in search range is then repeated, eventually allowing for an arbitrarily small variable quantization. This method has, for example, been used in the field of hybrid vehicles, primarily as a solver for limited-horizon nonlinear MPC control, see Wahl and Gauterin (2013).

IDP can also handle problems where the optimal control trajectory lies along a boundary of the feasible set — typically with successive iterations the generated trajectory asymptotically approaches the optimal one. This is an important advantage of IDP as compared to DP defined by e.g. Sundström et al. (2010), which will generate trajectories that avoid the edges of the feasible set, potentially resulting in a suboptimal solution.

Recently, Elbert et al. (2013) implemented a *non-iterative* DP method that correctly handles problems that lie along a boundary of infeasibility. However, this method does not have the additional benefit of reducing variable quantization.

1.3 Current issues

IDP is a powerful method for solving many types of global optimization problems. However, previously it has been unsuitable for certain sub-classes of problems due to issues with poor feasibility guarantees and large search spaces. This paper presents a few extensions that can help resolve these issues.

Poor feasibility guarantees In general, there is no guarantee that a feasible control trajectory will be generated during the forward-calculation phase. It is typically assumed that interpolation of the state and control signal between elements in $\mathcal{X}_k$ and $\mathcal{U}_k$ gives a feasible and close-to optimal control sequence. However, it is possible that selecting this interpolated control signal will lead to a state leaving the feasible set. For problems where the optimal state trajectory lies along the boundary of the feasible set this issue is particularly problematic.

Large search space For problems where the state variable dynamics have very different time scales, the search space quickly becomes unreasonably large for the standard DP algorithm. For example, consider a system where an electric vehicle's velocity and battery state-of-charge (SOC) are modeled as state variables and the vehicle's acceleration is available as an input. Intuitively we expect the velocity to exhibit dynamics on the order of seconds, while the SOC shows dynamics on the order of minutes or hours. Solving this problem directly with IDP requires;

- A time step, 1 second, which is sufficiently small to resolve the fast dynamics of the vehicle velocity.
- That the set of control inputs is dense enough. For a 1% discretization error 100 control signals must be tested for each state configuration.
- A grid density for the vehicle velocity that is dense enough to ensure *reachability*; defined as the ability to *reach* at least one feasible neighboring state — the "nearest" state — at the next sample, given the entire range of state configurations and control signals applied to the system at each sample. In this problem, assuming a velocity grid covering the range 0 – $30\,\mathrm{m/s}$ and a maximum acceleration of $3\,\mathrm{m/s^2}$, 11 equidistant points are required in the velocity grid with the chosen sample rate. As it is crucial that *reachability* is ensured it is generally good practice to inflate this value slightly to take numerical precision into account. Assume a grid density of 15 points is sufficient.
- A grid density for the vehicle SOC that is dense enough to ensure *reachability*. Assuming a battery capacity of $30\,\mathrm{kWh} = 108\,\mathrm{MJ}$ and a maximum power demand of $90\,\mathrm{kW}$, this implies that each grid point must be separated by at most $90\,\mathrm{kW}/1\,\mathrm{s} = 90\,\mathrm{kJ}$. This in turns implies a minimum of $108\,\mathrm{MJ}/90\,\mathrm{kJ} = 1200$ points for the SOC grid. For some added headroom, assume a grid density of 1400 points.

Solving this problem with IDP results in that for every sample k there will be $100 \cdot 15 \cdot 1400 = 2.1 \cdot 10^6$ evaluations of f, b_{in}, and c — the equations that define the system dynamics, constraints, and cost defined in (1). For a time horizon of 1 hour, which captures the dynamics of the SOC, this would imply that the total optimization problem involves $2.1 \cdot 10^6 \cdot 60 \cdot 60 \approx 7.6 \cdot 10^9$ evaluations of the system model for each IDP iteration.

Note that this problem is "only" a 2+1–dimensional problem (two state variables, one control variable), and higher-dimensional problems grow exponentially in complexity. For systems with radically different time scales this becomes computationally exhausting as small time-steps are needed along with a prohibitively large grid for the slowly varying state variable(s).

2. MULTI-PASS ITERATIVE DYNAMIC PROGRAMMING WITH REGULARIZATION (IDP-MP)

The following modifications to the standard IDP scheme, collectively referred to as IDP-MP, mitigate the feasibility and complexity issues described in Section 1.3.

2.1 Improving feasibility with trajectory regularization

Let $\mathcal{F}_k$ denote the feasible set of $x[k]$, i.e. the set of $x[k]$ where there exists a control trajectory $u_{opt}(x[k])$ that satisfies the problem's constraints defined in (1). Let $\mathcal{G}_k$ denote the infeasible set of $x[k]$, i.e. the set of $x[k]$ where there does not exist a state trajectory satisfying (1). Define $J^*_{k,N_s}(x[k])$ as the minimum total cumulative cost from sample k to sample N_s when following the optimal control trajectory from $x[k] \in \mathcal{F}_k$. For convenience, define $J^*_{k,N_s}(x[k]) = \infty \forall x[k] \in \mathcal{G}_k$. Finally, define $\mathcal{O}_k$ as the set of values that $x[k]$ is allowed to cover in (1), i.e. the values of $x[k]$ that satisfies $b_{in}(\dots, k)$. Define a recursive regularized cost function $\hat{J}^*_{k,N_s}$ and constraint function $\hat{b}_{in}(\dots)$ that replaces the terms J^*_{k,N_s} and $b_{in}(\dots)$ in (1) during the back-calculation phase in the DP algorithm as

$$
\begin{aligned}
\hat{J}^*_{k,N_s}(x[k]) &= \min_{u[k]\in\mathcal{U}_k} [c(x[k], u[k], k) \\
&\quad + \hat{J}^*_{k+1,N_s}(x[k+1])] + \mu \\
\hat{J}^*_{N_s,N_s} &= \min_{u[N_s]\in\mathcal{U}_{N_s}} c(x[N_s], u[N_s], N_s) + \mu \\
&\text{s.t.} \\
\hat{b}_{in}(u[k], k) &\le 0,\ k = [0, N_s] \\
x[k+1] &= f(x[k], u[k], k) \\
\mu &= \begin{cases} 0 & d_{min} > d_{thrs} \wedge x[k] \in \mathcal{O}_k \\ \beta & \text{otherwise} \end{cases} \\
d_{min} &= \min_{I_G\in\mathcal{I}_G} \|I_x - I_G\|,
\end{aligned}
\tag{2}
$$

where β is a sufficiently large additive penalization factor, I_x is the grid coordinate of a given element $x[k]$, $\mathcal{I}_G$ contains the grid coordinates for all elements in $\mathcal{G}_k$, and $\hat{b}_{in}(\dots)$ does not depend on $x[k]$ but otherwise has the same constrains as $b_{in}(\dots)$ in (1). In essence, (2) recursively generates the optimal trajectory by removing any hard constraints on x and instead adding a penalty β for every sample $k \in [0, N_s]$ where the optimal trajectory $J^*_{k,N_s}(x[k])$ either exceeds $\mathcal{O}_k$ (i.e. violates the original constraints $b_{in}(\dots)$) or the distance between $x[k]$ and the nearest infeasible state is less than d_{min}. The $d_{min} > d_{thrs}$ term is akin to the use of a barrier function (Bertsekas, 1999, p. 370), though in this application there is no need for it to be continuous or differentiable.

Figure 2 illustrates example values of d_{min} for a two-state problem with a search space of six orthogonal and uniformly distributed values for each state variable, using the 2-norm for defining the distance d_{min}, where the minimum distance between any two grid coordinates is defined as 1. In this example, $\mathcal{I}_G = \{[1,1], [2,2], \dots\}$ (indicated by the red diamonds) and the feasible state grid coordinates are $\{[2,1], [3,1], \dots\}$ (indicated by differently colored circles). For $d_{thrs} = 1.5$ the dashed blue line indicates the boundary between the penalized region (to the left) and the unpenalized region (to the right).

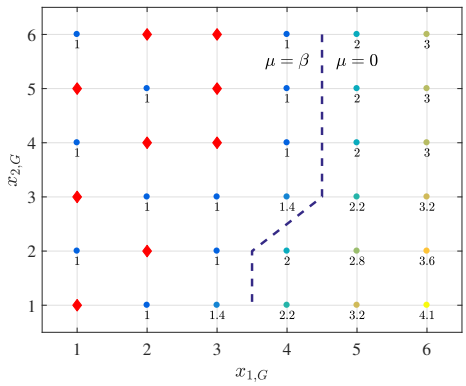


Fig. 2. Feasible (round) and infeasible (diamond) elements for an exemplified $\mathcal{X}_k$ with two variables consisting of six points each, with d_{min}'s value in (2) for each feasible element using the 2-norm. The regions corresponding to $\mu = \beta$ and $\mu = 0$ are indicated for $d_{thrs} = 1.5$.

As the hard constraints on $x\,[k]$ in (1) are replaced by soft constraints in (2), the problem feasibility is significantly improved for problems where the only feasible state trajectory lies near the state variable bounds (e.g. some bang-bang-control problems, (E. Bryson and Ho, 1975, p. 112) among others). Note that there is no need to add soft constraints to $u\,[k]$ as these values can directly be chosen, unlike $x\,[k]$ which evolves over k. (Note that this constraint replacement method is conceptually similar to that of (Bertsekas, 1999, p. 281).)

Determining values for the regularization parameters d_{thrs} and β is not obvious; the examples in this paper use $\beta = \mathrm{c}_{max} - \mathrm{c}_{min}$ (where $\mathrm{c}_{max} = \max c\,(\dots)$ and $\mathrm{c}_{min} = \min c\,(\dots)$, i.e. the maximum and minimum possible sample costs respectively), and 2-norm grid index distances $\sqrt{2} \leq d_{thrs} \leq 5$.

Typically, active regularization terms will generate suboptimal trajectories. However, with successive iterations, the suboptimal path typically asymptotically approaches the true optimal path as

- the suboptimal state and control trajectories introduced by the $d_{min} > d_{thrs}$ condition generates a path that is typically at most d_{thrs} grid indices away from the true optimal path, and with a decreasing grid extent (i.e. a finer grid) the absolute deviation decreases in proportion, and
- an optimal trajectory will only leave $\mathcal{O}_{k+1}$ if there only exists feasible, but penalized, trajectories from $x\,[k]$ (i.e. trajectories that violate the constraints in (1)). With successive IDP iterations the elements in $\mathcal{U}_k$ and $\mathcal{X}_{k+1}$ will be more closely spaced and a control $u\,[k]$ that brings $x\,[k] \to x\,[k+1]$ such that $x\,[k+1] \in \mathcal{O}_{k+1}$ will typically be generated if it exists.

Essentially, with a sufficiently fine grid, and for problems where the region of feasible trajectories is not pathologically shaped, the penalization term β will not be applied, and the initial problem's constraints, (1), will not be violated.

2.2 Improving feasibility with heuristic increase of grid search space

Despite the regularization terms used to improve the feasibility of the problem, as described in Section 2.1, it is possible that a feasible state/control trajectory is not found after reducing the grid size (for example, an unfortunately chosen grid may generate an $\mathcal{X}_k$ and/or $\mathcal{U}_k$ whose members are poorly amenable to interpolation). For the $p+1$'th IDP iteration, let $\mu_{dec} < 1$ and $\mu_{inc} > 1$ be the possible factors to scale $\mathcal{X}_k$ and $\mathcal{U}_k$ with for each sample k. For a given scaling factor, center $\mathcal{X}_k/\mathcal{U}_k$ about the result for the p'th IDP iteration, while leaving the number of grid elements unchanged. Here μ_{dec} is analogous to $1 - \varepsilon$ in Luus (1990).

In this event, a simple heuristic that attempts to eventually generate feasible trajectories is to scale the grid size by μ_{dec} if the previous iteration generated a feasible result and by μ_{inc} if it did not. A good practice for this value is to choose a value that only very slightly increases the grid, i.e. $1 < \mu_{inc} \ll 1/\mu_{dec}$, and selecting μ_{inc} such that $\mu_{inc}^n \neq 1/\mu_{dec}\, \forall n \in \mathbb{Z}$, i.e. select μ_{inc} and μ_{dec} to inhibit repeating cyclic sequences of grid sizes given continuously infeasible iterations after an initially successful iteration. For example, if $\mu_{dec} = 0.8$, one appropriate value is $\mu_{inc} = 1.05$ as $\mu_{inc}^n = [1.05, 1.1025, 1.1576, 1.2155, 1.2763, \dots]$, which does not contain $1/\mu_{dec} = 1.25$.

2.3 Solving the m-1-dimensional problem to reduce the search space

For problems that display loosely coupled states (in the sense that each state's dynamics are relatively independent of the other) and whose dynamics have very different time scales, one can intuitively expect that the state trajectory for the slowly varying state variable(s) is similar to the trajectory generated by solving a simplified problem, where the dynamics of the quickly-varying state variable(s) is neglected. (Note that problems that are stiff and/or chaotic — where the trajectory of the quickly varying state variable(s) has a large effect on the slowly varying state variable(s) — do not lend themselves to this method.)

For these types of problems, the execution time for the optimal control problem can be significantly reduced by,

a solving an approximate lower-dimensional problem using IDP with a possibly longer sample period, where the dynamics of the quickly-varying states are neglected (which is computationally much faster than the full-dimensional problem with a short sample period), followed by

b solving the full-dimensional problem with a sufficiently short sample period and grid spacing density (the latter due to *reachability* requirements) using IDP, where the search space for the slowly varying variable is limited to some region in the vicinity of the solution from the low-dimensional problem *a*.

This procedure allows for significantly reducing the number of elements in $\mathcal{X}$ that correspond to the slowly-varying variable(s) in stage *b*, while maintaining optimality so long as the true optimal control solution for the slowly varying state variable(s) lies within the range of values searched in stage *b*. This method can be easily extended

to m-dimensional problems with n different sets of state variables with different time scales.

The following exemplifies how stages a and b can be applied to the vehicle control problem defined in Section 1.3.

Apply stage a; Solve a 1-state/1-input problem using an equivalent model where the SOC is modeled as a state variable and the model input is the vehicle power (corresponding to the power required to maintain a constant speed). This approximation is equivalent to neglecting the dynamics related to changes in vehicle velocity, i.e. neglecting the vehicle's kinetic energy and allowing the vehicle velocity to be discontinuous. For a longer sample period of 2 seconds (which shifts computational burden from stage a to stage b) *reachability* implies that a grid density of $108\,\mathrm{MJ}/\left(90\,\mathrm{kW}\cdot 2\,\mathrm{s}\right) = 600$ points is required. Assuming 700 points are used this gives a total of $700 \cdot 60 \cdot 60/2 = 1.2 \cdot 10^6$ model evaluations per IDP iteration. Doubling the sample period for this stage results in reducing the number of computations in this stage by a factor of four.

Apply stage b; Solve the full 2-state/1-input problem (i.e. model SOC and velocity as states with acceleration as an input) with a reduced SOC search range. For a 1 second sample rate *reachability* yields a SOC grid separation of 90 kJ, though now a smaller SOC range can be searched compared to the original problem in Section 1.3. A conservative lower bound for the SOC variable extent is to require the permissible variation in battery energy ΔE to be able to accelerate/decelerate the vehicle from zero to maximum speed or vice versa. For equal sample rates in stages a and b and a vehicle mass of 1000 kg, this results in a battery search range of $\pm\Delta E \geq \frac{1}{2}\cdot 10^3\left(30\right)^2 = \pm 450\,\mathrm{kJ}$ centered about the SOC trajectory determined in stage a. In this example, the sample period in stage a was twice the sample period of this stage, suggesting that a search range on the order of $\pm 450 \cdot 2 = \pm 900\,\mathrm{kJ}$ is sufficient, giving a minimum of $2 \cdot 900\,\mathrm{kJ}/90\,\mathrm{kJ} = 20$ grid points (i.e. doubling the sample period in stage a doubles the computational burden in this stage). For some added headroom, assume 24 grid points are used. This gives a total of $100{\cdot}15{\cdot}24 = 36{\cdot}10^3$ state/control combinations to test at each sample, which causes the final two-dimensional problem to consist of $36 \cdot 10^3 \cdot 60 \cdot 60 = 130 \cdot 10^6$ calls to the system model.

In this example, IDP-MP requires $1.2 \cdot 10^6$ and $130 \cdot 10^6$ model evaluations in stage a and b respectively, in contrast to the $7.56 \cdot 10^9$ model evaluations per iteration for the standard IDP method. The total number of model evaluations is reduced by a factor of 58, with a similar reduction in execution time.

In fact, for this particular problem, shifting computational burden from stage a to stage b worsens the net performance, as $\pm\Delta E$ is "only" two orders of magnitude smaller than the total battery capacity. For a 1-second sample rate in both stages, stage a and b would require $4.8 \cdot 10^6$ and $65 \cdot 10^6$ model evaluations respectively, reducing the total number of model evaluations by a factor of 108 compared to the standard IDP method. Naturally, the reduction in computational time is a result of the chosen numerical values in this example, and it will be shown in Section 3.2 that some problems may exhibit a much greater performance improvement.

Note that, in general, determining the minimum search range for stage b and optimally balancing the computational burden between stages a and b is beyond the scope of this paper.

3. RESULTS

This section highlights the benefits of the method presented in Section 2 by solving two optimal-control problems using a Matlab implementation of the presented IDP-MP algorithm.

3.1 Double-integrator

Assume the goal is to optimally control a sampled constrained double integrator with different time scales given by

$$\begin{aligned}
x_2\left[k+1\right] &= x_2\left[k\right] + \alpha T_s u\left[k\right] \\
x_1\left[k+1\right] &= x_1\left[k\right] + T_s x_2\left[k\right] + \alpha u\left[k\right]\frac{T_s^2}{2} \\
c\left[k\right] &= \left(\left|x_1\left[k\right]\right| + \left|\frac{1}{10}x_2\left[k\right]\right|\right) T_s \\
&\text{s.t.} \\
T_s &= 0.125 \\
N_s &= 3/T_s \\
\alpha &= 4 \\
u\left[k\right] &\in \left[-1,1\right] \\
x_2\left[k\right] &\in \left[-1,1\right] \\
\begin{bmatrix} x_1\left[0\right] \\ -x_2\left[0\right] \\ x_1\left[N_s\right] \\ x_2\left[N_s\right] \end{bmatrix} &\geq \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix},
\end{aligned} \tag{3}$$

where α is the difference in time scales between x_2 and x_1, T_s is a constant sampling period, and the magnitudes of x_1 and x_2 are penalized. The penalization terms are selected so that the resulting optimal trajectory is identical to that of a problem using the cost function $c\left[k\right] = \left|x_1\left[k\right]\right| T_s$, while ensuring numerical stability in regions where this simpler cost function displays singular control. Several attributes make this problem a suitable example that illustrates the benefits of the presented method: the optimal control solution is known analytically, the state trajectory of the optimal solution lies on the boundary of the feasible set (as x_2 will be ± 1 for some time), and with $\alpha = 4$ this problem is relatively loosely coupled with respect to x_1 and x_2.

Using the IDP-MP method defined in Section 2.3, some reduction in the search space for x_1 is possible if an approximation of (3) can be generated where the quickly varying state variable is replaced by a control signal. One example of a system that does this is

$$\begin{aligned}
x\left[k+1\right] &= x\left[k\right] + T_s u\left[k\right] \\
c\left(x\right) &= \left|x\left[k\right]\right| T_s
\end{aligned} \quad \text{s.t.} \quad \begin{aligned} u\left[k\right] &\in \left[-1,1\right] \\ \begin{bmatrix} x\left[0\right] \\ x\left[N_s\right] \end{bmatrix} &\geq \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \end{aligned} \tag{4}$$

where x_2 has been directly replaced with an input u. This system is a good approximation of the dynamics of the slowly changing state variable in (3), i.e. a constrained integrator. Furthermore, (4) is easy to solve using an ordinary IDP algorithm; define $\hat{x}\left[k\right], k = \left[0, N_s\right]$ as the solution given by IDP.

Now, the two-dimensional problem in (3) can be solved and the search space can be reduced for x_1 by letting $\mathcal{X}_k^1 = \left\{\hat{x}[k] + \Delta x_1 \frac{n}{N_{g1}}\right\}_{n=-N_{g1}}^{N_{g1}}$. Here, $\mathcal{X}_k^1$ is a set of $2N_{g1}+1$ linearly distributed elements corresponding to x_1 that will be tested at sample k, Δx_1 is the extent of the values to test, and $\hat{x}[k]$ is the generated solution of the 1-state optimal control problem in (4). The particular choice of linearly distributed elements in $\mathcal{X}_k^1$ is convenient, but ultimately arbitrary; the range-reducing method works well for other choices. $\mathcal{X}_k$ can be generated in a number of ways, one practical method is to set $\mathcal{X}_k = \{(a,b) \,|\, a \in \mathcal{X}_k^1 \text{ and } b \in \mathcal{X}_k^2\}$, where $\mathcal{X}_k^2 = \left\{\frac{n}{N_{g2}}\right\}_{n=-N_{g2}}^{N_{g2}}$. Setting $\Delta x_1 = (2 \cdot \text{range}(x_1))/\alpha = 1.5$ results in reducing the number of model evaluations by a factor of two, and the reduction is primarily limited by the quotient of the time scales, α.

The reduced search space decreases the computational burden of the problem, however, this also reduces the feasible set. If reduced too aggressively, the optimal solution will not be completely contained in the search space and solution optimality may be lost. The feasibility of the problem can be improved by introducing the regularization terms defined in Section 2.1. For $\beta = c_{\max} - c_{\min} = \max c(\dots) - \min c(\dots) = 1.1 - 0 = 1.1$, $d_{min} = \sqrt{2}$, using the 2-norm for distance, and an increased search space of $x_2 \in [-1.25, 1.25]$ the feasibility issues are mitigated in this example. It is beyond the scope of this paper to determine the minimum range that can be searched that is guaranteed to contain the optimal solution.

Solving this problem using IDP-MP gives the state trajectory shown in Figure 3 after 50 iterations, taking approximately 10 minutes on a typical desktop computer (primarily CPU-bound, single-core execution on an AMD FX-6300). The feasible set and optimal trajectories for sample $k = 10$ for the first two-dimensional iteration is shown in Figure 4. As can be seen, the results in both figures match the well-known optimal bang-bang control and state space trajectory (E. Bryson and Ho, 1975, p. 112).

Some noteworthy attributes are:

- the optimal control trajectory for states near the infeasible region tend to exhibit a larger magnitude (i.e. the system is more quickly brought to a state at least d_{min} grid indices away from the infeasible bound if such a control exists),
- for states where $x_1 = -0.7$ the optimal control exceeds the soft constraints $x_2 \in [-1, 1]$, implying that these states would belong to the infeasible set had the allowable range not been extended to $x_2 \in [-1.25, 1.25]$,
- the state $[x_1, x_2] = [-0.6, 1.1]$ has an optimal transition to $[x_1, x_2] \approx [-0.5, 1]$, i.e. the optimal control brings the state out of soft constraint violation (i.e. keeps $x_2 \in [-1, 1]$) if there exists a trajectory that does this.

For more details on the numerical values used and the implementation of the IDP-MP solver, see `https://github.com/lerneaenhydra/dpm`.

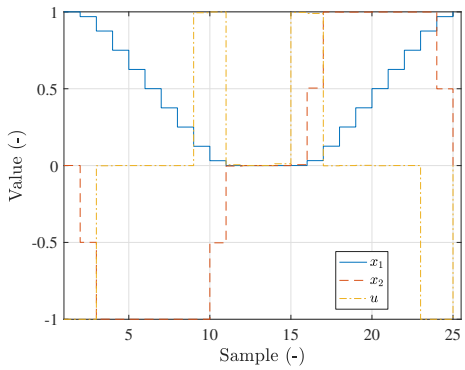


Fig. 3. The state and control trajectory solution for the double integrator (3). This result asymptotically approaches the analytic solution.

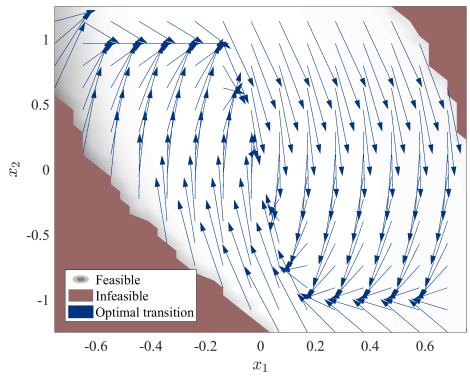


Fig. 4. Optimal trajectory map for the first iteration of the system defined by (3) for sample $k = 10$. Each arrow indicates the optimal state transition from the current sample to the next sample. The feasible set is shown in the greyscale region, with darker areas indicating a larger cost J^*_{13,N_s}.

3.2 Hybrid vehicle example

Determining the optimal control policy for hybrid vehicles is one application that often utilizes DP (Liu and Peng (2008); Pérez et al. (2006)). Typically, this problem is solved for one state variable, namely the SOC, as the search space grows prohibitively quickly with additional state or control variables. However, for a multi-dimensional problem with loosely coupled states with different time scales, the IDP-MP method defined in Section 2 can be effectively used.

Consider a passenger car series-hybrid where the combustion engine crankshaft velocity, ω, and SOC are modeled as state variables. A block diagram illustrating the model is shown in Figure 5, where the internal combustion engine torque, generator torque, and total power demand are model inputs; SOC and ω are state variables; and the SOC and instantaneous fuel consumption $\dot{m}$ are model outputs.

As the crankshaft velocity is loosely coupled and exhibits dynamics several orders of magnitude faster than the SOC,

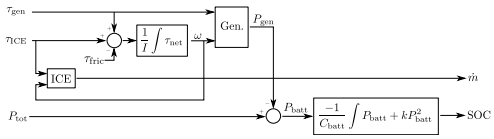


Fig. 5. Simple model of a series-hybrid vehicle.

this problem is amenable to IDP-MP. Solving for model parameters that are representative of a typical passenger car, power requirements given by the US-06 drive cycle (US Environmental Protection Agency (2008)), a 50% SOC at the start and end of the cycle, and solely penalizing fuel consumption gives results shown in Figure 6. (See `https://github.com/lerneaenhydra/dpm` for a full definition of the model set-up.)

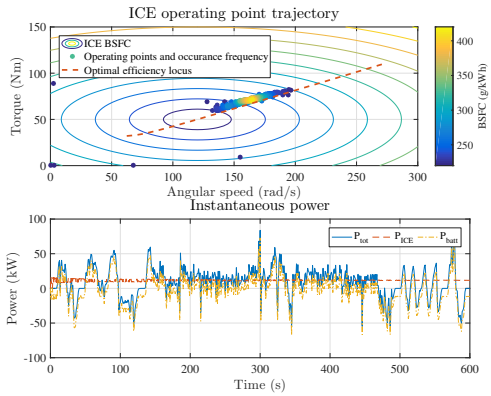


Fig. 6. Internal combustion engine (ICE) operating point and instantaneous subsystem power for the vehicle topology defined in Figure 5 subjected to the US-06 drive cycle. The upper plot shows isolines of the engine's brake specific fuel consumption (BSFC) and operating point at each sample. More frequent points are colored yellow while less frequent points are colored blue. The engine is operated mostly along the optimal-efficiency locus, where the most frequent operating point is 71 Nm and 164 rad/s. The lower plot displays the total, battery, and generator power over time.

This example was solved using both the IDP-MP method as well as a standard IDP method. Here, IDP-MP used 32 times fewer model evaluations than IDP per iteration, and the calculation time was reduced by a similar factor for each iteration. The computational reduction is limited primarily by the quotient of time scales between the two state variables. In this example the battery capacity has been kept as small as possible to let the standard method generate a solution in a reasonable time frame, and a battery time scale on the order of 42 seconds was sufficient (i.e. the battery was sized so that, at maximum power draw, completely depleting the battery from a full SOC takes 42 seconds). The crankshaft displays a time scale on the order of one second, so the state variables time-scales differ by a factor of 42, which matches well with the difference in the number of model evaluations between IDP-MP and IDP. Had the battery time scale instead been on the order of an hour, which is more physically realistic, each IDP-MP iteration would approximately use $1/(32/42 \cdot 60 \cdot 60) \approx 1/2700$ as many system model evaluations as the standard IDP method (as *reachability* implies the battery energy grid density must be constant while the grid range is increased by a factor of approximately $60 \cdot 60/42 \approx 86$).

The IDP-MP method has been verified for this particular example by comparing the results to those given by a standard IDP method. Figure 7 displays the total cost J^*_{0,N_s} for successive IDP iterations. The mean relative difference between the state trajectories of the 7'th iteration of IDP-MP and 5'th iteration of IDP (which have been selected due to their similar total cost) is 0.764% and is believed to primarily be due to the inherent variable quantization of the states and controls. Even though IDP-MP generates inferior results for a given number of iterations the significantly reduced computation time ensures that a result equally good as that given by IDP is generated after a shorter execution time.

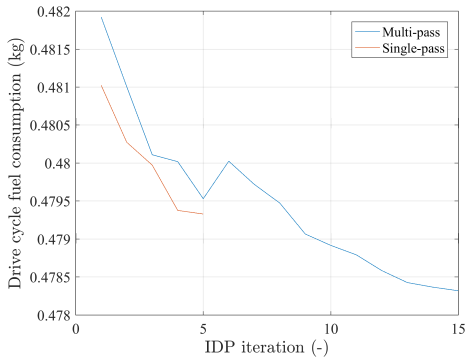


Fig. 7. Net cost (fuel consumption) for the system topology shown in Figure 5 for successive iterations. IDP-MP gives a slightly worse trajectory than IDP for a given iteration, but eventually surpasses IDP while using 1/32 as many model evaluations per iteration. Only five IDP iterations are shown due to the prohibitive execution time, taking approximately ten times as long to calculate as all the IDP-MP iterations.

4. CONCLUSIONS

It has been shown that the IDP-MP method can greatly decrease the computational time required to solve optimal control problems for systems where the system dynamics are both loosely coupled and display significantly different time scales. Typically, the execution time is decreased by a factor on the order of the quotient of the system's time scales. As problems with very different time scales are computationally difficult to solve using IDP, this method is a significant improvement over IDP as these problems are where the largest performance gains with IDP-MP are found. Broadly speaking, the IDP-MP method described in this paper increases the range of optimal control problems that DP/IDP is suited for and, as a result of the large performance improvement, significantly more complex problems can be considered.

REFERENCES

Åström, K.J. and Wittenmark, B. (1997). *Computer-controlled Systems (3rd Ed.)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Bellman, R. and Dreyfus, S. (2015). *Applied Dynamic Programming*. Princeton Legacy Library. Princeton University Press.

Bellman, R. (1956). Dynamic programming and lagrange multipliers. *Proceedings of the National Academy of Sciences*, 42(10), 767–769.

Bertsekas, D.P. (1999). *Nonlinear Programming*. Athena Scientific.

Bertsekas, D.P. (2005). *Dynamic programming and optimal control. Volume I*. Athena Scientific optimization and computation series. Belmont, Mass. Athena Scientific.

E. Bryson, Jr, A. and Ho, Y.C. (1975). *Applied Optimal Control*. Taylor & Francis Group, 270 Madison Avenue, New York, NY.

Elbert, P., Ebbesen, S., and Guzzella, L. (2013). Implementation of Dynamic Programming for n-Dimensional Optimal Control Problems With Final State Constraints. *IEEE Transactions on Control Systems Technology*, 21(3), 924–931.

Liu, J. and Peng, H. (2008). Modeling and control of a power-split hybrid vehicle. *IEEE Transactions on Control Systems Technology*, 16(6), 1242–1251.

Luus, R. (1990). Optimal control by dynamic programming using systematic reduction in grid size. *International Journal of Control*, 51(5), 995–1013.

Pérez, L.V., Bossio, G.R., Moitre, D., and Garcia, G.O. (2006). Optimization of power management in an hybrid electric vehicle using dynamic programming. *Mathematics and Computers in Simulation*, 73(1-4), 244–254.

Sciarretta, A. and Guzzella, L. (2007). Control of hybrid electric vehicles. *IEEE Control Systems*, 27(2), 60–70.

Sundström, O., Ambühl, D., and Guzzella, L. (2010). On Implementation of Dynamic Programming for Optimal Control Problems with Final State Constraints. *Oil & Gas Science and Technology — Revue de l'Institut Franccais du Pétrole*, 65(1), 91–102.

Sundström, O. and Guzzella, L. (2009). A generic dynamic programming matlab function. In *2009 IEEE Control Applications,(CCA) & Intelligent Control,(ISIC)*, 1625–1630. IEEE.

US Environmental Protection Agency (2008). US06 Dynamometer Drive Schedule. `https://www.epa.gov/vehicle-and-fuel-emissions-testing/dynamometer-drive-schedules`. [Online; accessed 2016-10-31].

Wahl, H.G. and Gauterin, F. (2013). An iterative dynamic programming approach for the global optimal control of hybrid electric vehicles under real-time constraints. In *Intelligent Vehicles Symposium (IV), 2013 IEEE*, 592–597. IEEE.

PAPER B

Undiscounted Control Policy Generation for Continuous-Valued Optimal Control by Approximate Dynamic Programming

Jonathan Lock, Tomas McKelvey

Submitted to the International Journal of Control on July 1, 2020
Revised January 7, 2021 and 17 March, 2021.
Preprint available at `https://arxiv.org/abs/2104.11093`

Undiscounted Control Policy Generation for Continuous-Valued Optimal Control by Approximate Dynamic Programming

Jonathan Lock, Tomas McKelvey

***Abstract*—We present a numerical method for generating the state-feedback control policy associated with general undiscounted, constant-setpoint, infinite-horizon, nonlinear optimal control problems with continuous state variables. The method is based on approximate dynamic programming, and is closely related to approximate policy iteration. Existing methods typically terminate based on the convergence of the control policy and either require a discounted problem formulation or demand the cost function to lie in a specific subclass of functions. The presented method extends on existing termination criterea by requiring both the control policy and the resulting system state to converge, allowing for use with undiscounted cost functions that are bounded and continuous. This paper defines the numerical method, derives the relevant underlying mathematical properties, and validates the numerical method with representative examples. A MATLAB implementation with the shown examples is freely available.**

***Index terms*—** Approximate dynamic programming, Control policy, Undiscounted infinite-horizon, Optimal control

I. Introduction

Practical methods for generating the optimal control policy (i.e. the state feedback function) for general non-linear optimal control problems are useful tools for control engineers. If the optimal control policy is known, a real-time optimal controller can be implemented on very computationally limited hardware as the optimal control signal can be generated simply by interpolating the pre-computed optimal control based on the current system state. However, one practical difficultly lies in pre-computing the optimal control policy, which can be very computationally expensive. Although several methods for solving this class of problem are well-studied, dynamic programming (DP) variants being one example, they all have associated limitations or drawbacks. Policy iteration is one extensively studied variant of DP (e.g. Bertsekas 2017, p. 246; Puterman 1994, p. 295; Puterman and Brumelle 1979) that has been used for over 40 years for finding the optimal control policy for discrete-valued, non-linear, infinite-horizon problems, i.e. where the state and control variables are taken from discrete sets.

*Manuscript submitted to the International Journal of Control July 1, 2020; revised January 7, 2021 and 17 March, 2021.

Approximate dynamic programming (ADP) is another well-known extension of DP (see for instance Powell (2009) for a general introduction) that approximates the cost function using a prescribed set of basis functions. One group of ADP methods approximate the cost function by interpolating costs and optimal controls between discrete gridded points (e.g. Munos and Moore (2002); Santos and Vigo-Aguiar (1998)). This approach allows for extending DP to applications with continuous state variables.

Assuming the problem of finding the approximately-optimal control policy for continuous-valued, non-linear, infinite-horizon problems, one might attempt to use traditional policy iteration in concert with ADP. However, this is problematic as traditional policy iteration requires the set of states and controls to be discrete (i.e. finite) to terminate, while the interpolation performed with ADP leads to a continuous (i.e. infinite) number of possible states and controls. This has led to the development of several methods that can be broadly classified as approximate policy iteration (API) methods, where the termination criterion of conventional policy iteration is altered in order to terminate in finite time and generate an approximately optimal solution.

There are several excellent papers that consider different variants of API. However, the vast majority of these are limited to the case where the cost function is discounted, i.e. where future costs are successively weighted less and less (Bertsekas, 2011; Santos & Rust, 2004; Scherrer, 2014; Stachurski, 2008). Though a discounted cost function may be relevant for some problems and allows for more easily determining a termination criterion, a sizeable portion of optimal control problems are better formulated as undiscounted problems (e.g. minimum fuel/energy/time problems, or yield maximisation for chemical plants and cultivation). Guo, Si, Liu, and Mei (2017) introduce one API method for the undiscounted case from a reinforcement learning perspective, but this method is limited both in that the cost function must be a sum of a positive definite function of the state and a quadratically weighted function of the controls, and that the state and control cannot be arbitrarily constrained.

In this paper we will introduce a method similar to API schemes that approximates the solution to the infinite-horizon problem by instead solving a finite-horizon problem. More specifically, the method uses conventional interpolating ADP to approximate the undiscounted, infinite-horizon,

non-linear, optimal control problem where the state is constrained to converge to a unique equilibrium. The primary contribution of this paper is a termination criterion that terminates at a suitable horizon without requiring the presence of a discount factor, while also allowing for (nearly) arbitrary cost, constraints, and problem dynamics — a combination that is novel to the best of the authors knowledge. The method's sole tuning parameter allows for controlling the trade off between memory consumption, computational time, and accuracy. This allows for the method to be used without in-depth knowledge of the method. Furthermore, as the method's output is the optimal control policy (i.e. the optimal control tabulated by the system state) subsequent on-line control can be implemented using a computationally fast interpolation operation.

The structure of this paper follows; in Section II we will define the problem studied in this paper and the structure of the interpolating ADP method we subsequently base our presented method on. We will assume a working knowledge of ADP methods for optimal control. Sundstrom and Guzzella (2009) gives a straightforward introduction while Bertsekas (2017); Puterman (1994) go into more detail. This is followed by Section III, where we derive relevant properties of the studied problem. Though these properties are mostly already known, by deriving them we can both highlight some important details, as well as use a language and notation more commonly seen by control engineers as compared to existing API literature. In Section IV we present our method of generating an approximation of the optimal control policy, as well as highlight how existing API methods compare with our method. Finally, in Section V we use two representative examples to show the results generated by our method. For ease of reference, a list of the symbols and notation used in this paper is shown in Table I.

II. Problem formulation

Assume a dynamic system $f_d : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ whose associated state evolution is recursively given by

$$x_{k+1} = f_d\left(x_k, u_k\right) \tag{1}$$

for the system state $x_k \in \mathbb{R}^n$ and control input $u_k \in \mathbb{R}^m$ at samples $k \in [0, 1, 2, \dots]$. Define the infinite sequences

$$\bar{x} \triangleq [x_0, x_1, x_2, \dots] \tag{2a}$$
$$\bar{u} \triangleq [u_0, u_1, u_2, \dots] \tag{2b}$$

as the *state trajectory* and *control trajectory* respectively. Similarly, define the finite sequences $\bar{x}^N \triangleq [x_0, x_1, \dots, x_{N-1}]$ and $\bar{u}^N \triangleq [u_0, u_1, \dots, u_{N-1}]$. In particular, for both $\bar{x}$ and $\bar{x}^N$ we respectively define x_0 as the *initial condition*.

A. The infinite-horizon problem

Given x_0, introduce

$$J\left(\bar{x}, \bar{u}\right) = \lim_{N\to\infty} \frac{1}{N} \sum_{k=0}^{N-1} f_c\left(x_k, u_k\right) \tag{3a}$$
$$J^* = \min_{\bar{x},\bar{u}} J\left(\bar{x}, \bar{u}\right) \tag{3b}$$
$$\left(\bar{x}^*, \bar{u}^*\right) = \operatorname*{argmin}_{\bar{x},\bar{u}} J\left(\bar{x}, \bar{u}\right) \tag{3c}$$

subject to

$$\left(\bar{x}, \bar{u}\right) \in \mathcal{S} \cap \mathcal{V}_\alpha \tag{3d}$$

for

$$\mathcal{S} = \left\{ \left(\bar{x}, \bar{u}\right) : \lim_{N\to\infty} \begin{array}{c} g\left(x_k, u_k\right) \leq 0 \\ x_{k+1} = f_d\left(x_k, u_k\right) \end{array}, \forall k \in [0, N-1] \right\} \tag{3e}$$

$$\mathcal{V}_\alpha = \left\{ \left(\bar{x}, \bar{u}\right) : \lim_{N\to\infty} \frac{1}{N} \sum_{k=0}^{N-1} f_a\left(x_k, u_k\right) = \alpha \right\} \tag{3f}$$

as the problem we study in this paper. Here, we denote $f_c : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ the *cost function*, $g : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^l$ the *inequality constraint(s)*, a scalar parameter $\alpha \in \mathbb{R}$ the *average constraint*, and $f_a : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ the *average constraint function*. We define a *feasible trajectory* as any trajectory $(\bar{x}, \bar{u})$ that satisfies (3d). The set $\mathcal{S}$ gives a convenient notation for demanding that the "textbook" problem dynamics and inequality constraints hold, while the set $\mathcal{V}_\alpha$ denotes an additional average constraint.

Crucially, as none of the functions in (3) are explicitly dependent on k, its solution satisfies the *principle of optimality* (Bertsekas 2017, p. 20; Bellman 1954). Bertsekas (2017, p. 15) shows that this in turn implies that the optimal control trajectory $\bar{u}^*$ can equivalently be formulated as the control policy (i.e. state-feedback)

$$\bar{u}^* = \left[\mu_0^*\left(x_0\right), \mu_1^*\left(x_1\right), \dots\right], \tag{4}$$

where $\mu_k^* : \mathbb{R}^n \to \mathbb{R}^m$ are functions that are *independent of the initial condition* x_0. Note that while $\bar{x}$ and $\bar{u}$ (with various sub- and super-scripts) are sequences of vectors of scalars, $\bar{\mu}$ (with various sub- and super-scripts) are instead *sequences of functions*. We will refer to $\bar{\mu}^*$ as the *optimal control policy*.

Definition 1. Define $\mathcal{F} \subseteq \mathbb{R}^n$ as the set of initial conditions with feasible solutions, i.e.

$$\mathcal{F} \triangleq \left\{x_0 : \exists \left(\bar{x}, \bar{u}\right) \in \mathcal{S} \cap \mathcal{V}_\alpha\right\}. \tag{5}$$

Assumption 1. *For the remainder this paper we assume:*

A.1 *f_c, f_d, g, and f_a are continuous and bounded.*

A.2 *The optimal solution $(\bar{x}^*, \bar{u}^*)$ associated with x_0 is unique.*

A.3 *The optimal control policy associated with (3) exists, and can be expressed as*

$$\bar{u}^* = \left[\mu^*\left(x_0\right), \mu^*\left(x_1\right), \dots\right], \tag{6}$$

i.e. it is not only independent of the initial condition x_0, but also independent of the sample index k. We will refer to this as a stationary *control policy (Bertsekas & Shreve, 1979).*

Table I: List of used notation, symbols, and first definition.

DP		Dynamic programming
ADP		Approximate dynamic programming
API		Approximate policy iteration
d_x	(8a)	Distance between neighbouring points in $\mathcal{X}$
d_u	(8b)	Distance between neighbouring points in $\mathcal{U}$
f_c	(3a)	Cost function
$f_{c,R}$	(17)	Relaxed cost function
f_d	(1)	System dynamics function
f_α	(3f)	Average constraint function
$\mathcal{F}$	(5)	Set of initial conditions with feasible initial condition
$\mathcal{F}'_k$	(29)	Set of feasible gridded initial conditions after k samples
g	(3e)	Inequality constraint function
J	(3a)	Cost
J^*	(3b)	Optimal cost
J^*_{eq}	(11a)	Optimal equilibrium cost
J_R	(18a)	Relaxed cost
J^*_R	(18b)	Optimal relaxed cost
J^{*N}_R	(25a)	Optimal relaxed N-horizon cost
N_M	(32a)	Finite minimum horizon
N'_M	(33b)	Finite UCPADP horizon
$\mathcal{S}$	(3e)	Set of trajectories with feasible dynamics and inequality constraints
u_k	(1)	Control signal at sample k
$\bar{u}$	(2b)	Control trajectory
$\bar{u}^*$	(3c)	Optimal control trajectory
$u_{\text{eq}}, u^*_{\text{eq}}$	(7)	Optimal equilibrium control (identical by Theorem 4)
$\bar{u}_R$	(18a)	Relaxed control trajectory
$\bar{u}^*_R$	(18c)	Optimal relaxed control trajectory
$\bar{u}^{*N}_R$	(25b)	Optimal relaxed N-horizon control trajectory
$\mathcal{U}$	(9b)	Cartesian grid of sampled controls for ADP routine
$\mathcal{V}_\alpha$	(3f)	Set of trajectories satisfying average equality constraint
x_k	(1)	System state at sample k
$\bar{x}$	(2a)	State trajectory
$\bar{x}^*$	(3c)	Optimal state trajectory
x_{eq},x^*_{eq}	(7)	Optimal equilibrium state (identical by Theorem 4)
$\bar{x}_R$	(18a)	Relaxed state trajectory
$\bar{x}^*_R$	(18c)	Optimal relaxed state trajectory
$\bar{x}^{*N}_R$	(25b)	Optimal relaxed N-horizon state trajectory
$x_{k,CL}$	(27)	Closed-loop state after applying a control policy k times
$\mathcal{X}$	(9a)	Cartesian grid of sampled states for ADP routine
α	(3f)	Average constraint
Δ^k_μ	(30)	Control policy deviation at sample k
Δ^k_x	(31)	State deviation at sample k
ε_x	(32c)	State tolerance
ε_μ	(32b)	Control policy tolerance
λ	(17)	Relaxation parameter
μ^*	(6)	Optimal stationary control law
$\bar{\mu}^{*N}_R$	(26)	Optimal relaxed N-horizon control policies

A.4 *$\mathcal{F}$ is nonempty, $\lim_{k\to\infty}(x_k^*)$ exists and is independent of x_0 for all $x_0 \in \mathcal{F}$, and x_k^* is asymptotically stable in the sense of Lyapunov for x_0 near $\lim_{k\to\infty}(x_k^*)$.*

Note that **A.1** implies that $J(\bar{x},\bar{u})$ is finite for any feasible trajectory, and by **A.4** we can furthermore view J^* as the *average (mean) cost.*

Definition 2. Assuming **A.4** holds, define

$$(x_{\text{eq}}, u_{\text{eq}}) \triangleq \lim_{k\to\infty}(x_k^*, u_k^*) \tag{7}$$

as the problem's equilibrium point.

Note that **A.2**, **A.3**, and **A.4** may be difficult to determine a priori for a given problem. We will briefly discuss the possible effects of them not holding in Section IV.

B. Interpolating ADP

The method we introduce in this paper uses a conventional interpolating ADP scheme, and we will here use the standard method of gridding x and u into finite Cartesian sets. We define

$$d_x \in \mathbb{R}^n \tag{8a}$$
$$d_u \in \mathbb{R}^m \tag{8b}$$

as the distance between neighbouring grid points for each dimension of the states and controls respectively. We also define

$$\mathcal{X} \subset \mathbb{R}^n \tag{9a}$$
$$\mathcal{U} \subset \mathbb{R}^m \tag{9b}$$

as the discrete set of state and control grid points resolved by ADP respectively, separated by d_x and d_u respectively and bounded by the region(s) where $g(x,u) \leq 0$. We then use conventional multilinear interpolation to approximate the cost J and optimal control policy μ for the real-valued states that do not lie in the discrete set $\mathcal{X}$. For example, assuming $x \in \mathbb{R}^2$, $u \in \mathbb{R}^1$, and $g(x,u) = |x|_1 \leq 1 \wedge |u| \leq 1$, choosing the very coarse (but illustrative) $d_x = [2,2]^T$ and $d_u = 0.5$ gives the sets

$$\mathcal{X} = \left\{ \begin{bmatrix} -1 \\ -1 \end{bmatrix}, \begin{bmatrix} -1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right\} \tag{10a}$$
$$\mathcal{U} = \{-1, -0.5, 0, 0.5, 1\}. \tag{10b}$$

III. Infinite-horizon, average-constrained problem properties

In this section we introduce properties of the undiscounted, infinite-horizon, average-constrained problem that will later be utilised by the method we introduce in Section IV.

A. Solution convergence

Definition 3. For $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, using the same functions as in (3), define

$$J_{\text{eq}}^* = \min_{x,u} f_c(x,u) \tag{11a}$$
$$\left(x_{\text{eq}}^*, u_{\text{eq}}^*\right) = \operatorname*{argmin}_{x,u} f_c(x,u) \tag{11b}$$

subject to

$$x = f_d(x,u) \tag{11c}$$
$$g(x,u) \leq 0 \tag{11d}$$
$$f_a(x,u) = \alpha \tag{11e}$$
$$\forall x_0 \in \mathcal{F}, \exists (\bar{x},\bar{u}) \text{ s.t. } \lim_{k\to\infty}(x_k,u_k) = (x,u) \tag{11f}$$

as the *optimal reachable equilibrium operating point* $\left(x_{\text{eq}}^*, u_{\text{eq}}^*\right)$. (Note that we have identical states on both the left- and right-hand side of (11c), i.e. an equilibrium state.) We can view this as the unique stationary point of the system with lowest cost that we can reach for any initial condition in the feasible set $\mathcal{F}$.

Theorem 4. *Given **A.1** and **A.4**,*

$$J^* = J_{eq}^* \tag{12}$$
$$(x_{eq}, u_{eq}) = \left(x_{eq}^*, u_{eq}^*\right), \tag{13}$$

i.e. the equilibrium we reach will be optimal in the sense of (11).

Proof: For $0 \leq i < j$, define

$$J_{i\to j}(\bar{x},\bar{u}) \triangleq \sum_{k=i}^{j} f_c(x_k,u_k). \tag{14}$$

We can then formulate (3b) as

$$J^* = \min_{\bar{x},\bar{u}} \lim_{N\to\infty} \frac{1}{N} J_{0\to i-1}(\bar{x},\bar{u}) + \frac{1}{N} J_{i\to N-1}(\bar{x},\bar{u}). \tag{15}$$

As $N \to \infty$, we are guaranteed that $\frac{1}{N} J_{0\to i-1} = 0$ for any fixed $i > 0$ per our assumption that f_c is bounded. This implies that J^* is only dependent on $J_{i\to N-1}$. By **A.4**, we can make (x_i^*, u_i^*) arbitrarily close to $(x_{\text{eq}}, u_{\text{eq}})$ for sufficiently large i.

Suppose that

$$(x_{\text{eq}}, u_{\text{eq}}) \neq \left(x_{\text{eq}}^*, u_{\text{eq}}^*\right). \tag{16}$$

By **A.4** $(x_{\text{eq}}, u_{\text{eq}})$ is unique, implying that $J^* > J_{\text{eq}}^*$. However, by (11f) there exists trajectories $\bar{x}'$ and $\bar{u}'$ such that $\lim_{k\to\infty}(x_k', u_k') = \left(x_{\text{eq}}^*, u_{\text{eq}}^*\right)$, with corresponding cost $J' < J^*$, contradicting (16).

For an alternate view of the same proof, see Bertsekas (2012, p. 298). ∎

By Theorem 4 we can intuitively view the infinite-horizon problem's solution as ignoring any (finite) costs during the transient phase and driving the state to the reachable stationary point with lowest cost. This is a special case of the *turnpike property* (Trélat & Zuazua, 2015; Zaslavski, 2014), which states that the solution to problems with a sufficiently long (finite) horizon tends to display transient dynamic initial and terminal phases, with a middle stationary phase that is independent of the initial and terminal conditions. Of course, the infinite-horizon problem does not have a terminal phase, and we can thus view the solution to our problem (3) as consisting of an initial transient followed by stationary operation at the optimal reachable equilibrium point.

From a notation perspective, by Theorem 4 we do not need to make the distinction between x_{eq} and x^*_{eq}. For consistency, we will use x^*_{eq} from here on out.

B. Average-constraint relaxation

Definition 5. For a fixed, bounded, scalar relaxation parameter $\lambda \in \mathbb{R}$, define the *relaxed cost* as

$$f_{c,R}(x,u) \triangleq f_c(x,u) + \lambda f_a(x,u). \tag{17}$$

Now we can introduce the *relaxed problem* as

$$J_R(\bar{x}_R, \bar{u}_R) = \lim_{N\to\infty} \frac{1}{N} \sum_{k=0}^{N-1} f_{c,R}(x_k, u_k) \tag{18a}$$

$$J_R^* = \min_{\bar{x}_R, \bar{u}_R} J_R(\bar{x}_R, \bar{u}_R) \tag{18b}$$

$$(\bar{x}_R^*, \bar{u}_R^*) = \operatorname*{argmin}_{\bar{x}_R, \bar{u}_R} J_R(\bar{x}_R, \bar{u}_R) \tag{18c}$$

subject to

$$(\bar{x}_R, \bar{u}_R) \in \mathcal{S}, \tag{18d}$$

where we view $J_R(\bar{x}_R, \bar{u}_R)$ as the *relaxed representation* of $J(\bar{x}, \bar{u})$, and J_R^* and $(\bar{x}_R^*, \bar{u}_R^*)$ as the *optimal relaxed cost* and *optimal relaxed trajectories* respectively. Note that $(\bar{x}_R, \bar{u}_R)$, and therefore also $(\bar{x}_R^*, \bar{u}_R^*)$, are not formally constrained to lie in $\mathcal{V}_\alpha$.

For clarity, we will use the notation $\bar{x}_R$ and $\bar{u}_R$ when referring to trajectories associated with the relaxed problem. We will for ease of notation assume that $(\bar{x}_R^*, \bar{u}_R^*)$ is unique (much as (3c)), though we can in principle use DP (and in turn the method to be presented) to solve problems with non-unique solutions.

Lemma 6. *For a given α, assume for some λ we have $(\bar{x}_R^*, \bar{u}_R^*) \in \mathcal{V}_\alpha$. Then $(\bar{x}_R^*, \bar{u}_R^*) = (\bar{x}^*, \bar{u}^*)$.*

Proof: For convenience, introduce $\zeta^* \triangleq (\bar{x}^*, \bar{u}^*)$, $\zeta_R^* \triangleq (\bar{x}_R^*, \bar{u}_R^*)$, $\zeta \triangleq (\bar{x}, \bar{u})$, $\zeta_R \triangleq (\bar{x}_R, \bar{u}_R)$, and

$$h(\zeta) \triangleq \lim_{N\to\infty} \frac{1}{N} \sum_{k=0}^{N-1} f_a(x_k, u_k) - \alpha. \tag{19}$$

Note that $h(\zeta) = 0 \Leftrightarrow \zeta \in \mathcal{V}_\alpha$.

The weak duality theorem (Andréasson et al., 2016) ensures that

$$J(\zeta^*) \geq J(\zeta_R^*) + \lambda h(\zeta_R^*) = J_R(\zeta_R^*) - \lambda\alpha. \tag{20}$$

In (20), by our assumption $\zeta_R^* \in \mathcal{V}_\alpha$ we are ensured that $h(\zeta_R^*) = 0$, giving

$$J(\zeta^*) \geq J(\zeta_R^*) = J_R(\zeta_R^*) - \lambda\alpha. \tag{21}$$

As $\zeta_R^* \in \mathcal{S} \cap \mathcal{V}_\alpha$, ζ_R^* also minimises (3), allowing us to replace the inequality in (21) with strict equality. By **A.2** ζ^* and ζ_R^* are unique, ensuring that that $\zeta^* = \zeta_R^*$. Finally, as ζ_R^* is independent of constant terms we have that

$$\zeta^* = \zeta_R^* = \operatorname*{argmin}_{\zeta} J_R(\zeta). \tag{22}$$

■

Theorem 7. *Given (3c) and its relaxed counterpart (18c),*

R.1 *If (18c) is infeasible (i.e. a solution does not exist), then (3c) is also infeasible (i.e. **A.4** is violated).*

R.2 *For a given λ and feasible (18c), there exists an α where*

$$(\bar{x}_R^*, \bar{u}_R^*) = (\bar{x}^*, \bar{u}^*). \tag{23}$$

Proof: **R.1**: Trivial, as $\mathcal{S} \supseteq \mathcal{S} \cap \mathcal{V}_\alpha$. ■

Proof: **R.2**: As λ is given and (18c) is feasible, we can thus find $(\bar{x}_R^*, \bar{u}_R^*)$. Let us now define

$$\alpha' \triangleq \lim_{N\to\infty} \frac{1}{N} \sum_{k=0}^{N-1} f_a\left(\bar{x}_{R,k}^* \bar{u}_{R,k}^*\right). \tag{24}$$

For $\alpha = \alpha'$ we (by construction) have $(\bar{x}_R^*, \bar{u}_R^*) \in \mathcal{S} \cap \mathcal{V}_\alpha$, trivially satisfying the requirements of Lemma 6. ■

In essence, for a given λ **R.2** ensures us that $(\bar{x}_R^*, \bar{u}_R^*) = (\bar{x}^*, \bar{u}^*)$ for some value of α. We can intuitively view λ as a tuning parameter, where different values of λ are associated with different solutions, each of which (trivially) have an associated average that we can compute by means of (24).

Using the relaxed problem formulation allows us to avoid the explicit average constraint (3f), which is primarily of use in the sense that the problem becomes more numerically tractable. At its core, the method we will introduce in this paper approximates the solution to (3) by instead solving a finite-horizon problem of sufficient length. One naive method of satisfying the average constraint would then be to introduce an additional state variable that stores the accumulated average, i.e. $z_N = \sum_{k=0}^{N-1} f_\alpha(x_k, u_k)$. We could then add an equality constraint demanding $z_n/N = \alpha$. However, this is computationally demanding (as we need to introduce an additional state variable, which DP schemes scale poorly with) and introduces a bias in the achieved average (as the average z_n/N is taken over both the initial transient and the stationary phase, we therefore only achieve the desired average as $N \to \infty$). Using the relaxed formulation thus avoids these issues entirely.

C. Convergence of finite-horizon problem

We will in this section introduce notation for the finite-horizon problem, which will then be used for constructing the method presented in this paper.

Definition 8. For a given finite horizon N, bounded λ, and initial condition x_0, define

$$J_R^{*N} = \min_{\bar{x}_R^N, \bar{u}_R^N} \frac{1}{N} \sum_{k=0}^{N-1} f_{c,R}(x_k, u_k) \tag{25a}$$

$$\left(\bar{x}_R^{*N}, \bar{u}_R^{*N}\right) = \operatorname*{argmin}_{\bar{x}_R^N, \bar{u}_R^N} \frac{1}{N} \sum_{k=0}^{N-1} f_{c,R}(x_k, u_k) \tag{25b}$$

subject to

$$\left(\bar{x}_R^N, \bar{u}_R^N\right) \in \mathcal{S} \tag{25c}$$

as the *N-horizon relaxed problem* with average cost J_R^{*N} and associated (finite-length) state and control trajectories $\left(\bar{x}_R^{*N}, \bar{u}_R^{*N}\right)$. Furthermore, define

$$\bar{\mu}_R^{*N} = \left[\mu_{R,0}^{*N}, \mu_{R,1}^{*N}, \ldots, \mu_{R,N-1}^{*N}\right], \tag{26}$$

where $\mu_{R,k}^{*N}:\mathbb{R}^n\to\mathbb{R}^m$ is the k'th state-feedback control policy, as the N-horizon sequence of control policies associated with (25).

Definition 9. Define

$$x_{k,CL}\left(\mu,x_0\right) \tag{27}$$

as the (not necessarily optimal) k'th closed-loop state given by repeatedly applying a (sample-independent) control policy μ k times from an initial state x_0, e.g.

$$\begin{aligned}
&x_{0,CL}\left(\mu,x_0\right)\triangleq x_0\\
&x_{1,CL}\left(\mu,x_0\right)\triangleq f_d\left(x_0,\mu\left(x_0\right)\right)\\
&x_{2,CL}\left(\mu,x_0\right)\triangleq f_d\left(x_{1,CL}\left(\mu,x_0\right),\mu\left(x_{1,CL}\left(\mu,x_0\right)\right)\right).
\end{aligned}$$

Note that the method of generating $x_{k,CL}$ is very similar to the forward-calculation stage of ADP, and differs only in that the control policy is kept constant.

Definition 10. For a given control policy μ, define

$$\mathcal{F}_k'(\mu)\triangleq\left\{x_0\in\mathcal{X}:g(x_{k',CL},\mu(x_{k',CL}))\leq 0\,\forall k'\in[0,k]\right\}. \tag{29}$$

We can thus view $\mathcal{F}_k'(\mu)$ as the set of initial conditions in $\mathcal{X}$ that satisfies the problem constraints and dynamics (the latter trivially, as we use μ to apply a control and give the next state) after applying the control policy μ k times.

Definition 11. For $k>0$, introduce the maximum control policy deviation $\Delta_\mu^k\in\mathbb{R}^m$ as

$$\left[\Delta_\mu^k\right]_i\triangleq\max_{\substack{x\in\mathcal{F}'_{\lceil k/2\rceil}(\mu_{R,0}^{*k})\\ k'\in[0,\lceil k/2\rceil]}}\left|\left[\mu_{R,0}^{*k}\left(x\right)-\mu_{R,k'}^{*k}\left(x\right)\right]_i\right|, \tag{30}$$

where the notation $[a]_i$ refers to the i'th element of a vector a and $\lceil\dots\rceil$ refers to the ceiling function. We can view Δ_μ^k as indicating the convergence of $\mu_{R,0}^{*k}$ to μ^*, evaluated at the gridded state points $\mathcal{X}$ whose associated state evolution remains feasible after $k/2$ iterations.

Definition 12. Introduce the maximum state deviation $\Delta_x^k\in\mathbb{R}^n$ as

$$\left[\Delta_x^k\right]_i\triangleq\max_{x\in\mathcal{F}'_{\lceil k/2\rceil}(\mu_{R,0}^{*k})}\left|\left[x_{\lceil k/2\rceil,CL}\left(\mu_{R,0}^{*k},x\right)-\sum_{x'\in\mathcal{F}'_{\lceil k/2\rceil}(\mu_{R,0}^{*k})}x_{\lceil k/2\rceil,CL}\left(\mu_{R,0}^{*k},x'\right)\frac{1}{\left|\mathcal{F}'_{\lceil k/2\rceil}\right|}\right]_i\right|. \tag{31}$$

Note that the notationally heavy second line of (31) is equivalent to the mean feasible state after $\lceil k/2\rceil$ iterations. Similarly to Definition 11, we can thus view Δ_x^k as indicating the convergence of $\left[x_{0,CL},x_{1,CL},\dots,x_{\lceil k/2\rceil,CL}\right]$ to $\bar{x}^*$, evaluated at the points where $x_{\lceil k/2\rceil,CL}$ remains feasible.

Trivially, using Definition 11 and Definition 12 gives:

Proposition 13. *By **A.3** $\lim_{k\to\infty}\Delta_\mu^k=0$, and by **A.4** $\lim_{k\to\infty}\Delta_x^k=0$.*

Definition 14. Given a *control policy tolerance* $\varepsilon_\mu\in\mathbb{R}^m$ and *state convergence tolerance* $\varepsilon_x\in\mathbb{R}^n$, define

$$N_M\triangleq\min_k k \tag{32a}$$

such that

$$\left[\Delta_\mu^k\right]_i<\left[\varepsilon_\mu\right]_i\forall i\in[1,m] \tag{32b}$$

$$\left[\Delta_x^k\right]_i<\left[\varepsilon_x\right]_i\forall i\in[1,n], \tag{32c}$$

as the *minimum horizon*. Proposition 13 ensures us that that for any ε_μ and ε_x there exists an associated finite horizon N_M, which we view as the shortest finite-horizon approximation of the infinite-horizon problem.

IV. The UCPADP method

In this section we introduce the primary contribution of this paper: *Undiscounted Control Policy generation by Approximate Dynamic Programming* (UCPADP), a method that generates an approximation of μ^*. At its core, in UCPADP we generate an approximation of the optimal control policy by iteratively testing successively larger horizons until the termination criteria (32) are satisfied. For computational efficiency reasons we will return to, UCPADP will approximate the control policy as

$$\mu^*\approx\mu_{R,0}^{*N_M'} \tag{33a}$$

$$\text{where } N_M\leq N_M'\leq 2N_M, \tag{33b}$$

i.e. the generated horizon will lie in a range between N_M and $2N_M$.

We can at this stage highlight one of the more significant differences between UCPADP and conventional API: the choice of termination conditions. Conventional API generates improved control policies analogous to $\mu_{R,1}^{*1},\mu_{R,2}^{*2},\mu_{R,3}^{*3},\dots$ with an associated cost $J_R^{*1},J_R^{*2},J_R^{*3},\dots$, and eventually terminates when the difference between either successive policies or cost is below a given threshold, for instance as in Santos and Rust (2004); Stachurski (2008, CPI, PSDP). This is similar to the test performed in (32b), which requires the control policy to be near-stationary. However, in conventional API the termination tolerance (analogous to ε_μ) is sized based on the discount factor, and depending on the specific method chosen the tolerance is either undefined or tends towards zero when the discount factor tends towards one (i.e. becomes the undiscounted case we study here). Bertsekas (2011); Scherrer (2014) review other methods that do not terminate based on the change in the control policy, but instead use some other termination criterion. However, these methods also assume a discounted problem formulation. Guo et al. (2017) is one example of a method that considers the undiscounted case, however their method imposes fairly significant limits on the class of cost and constraint functions (as discussed previously).

The state convergence condition (32c) is to the best of our knowledge novel, and serves a crucial purpose in that it demands the horizon be long enough for *all gridded feasible initial conditions to converge to a region near the equilibrium.*

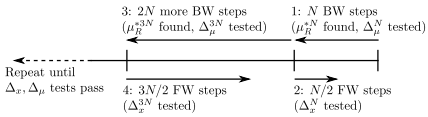


Figure 1: UCPADP steps, successively switching between generating more accurate control policies (backward calculation, steps 1,3,...), and evaluating whether the control policy is constant over the time needed for the state evolution to converge (forward calculation, steps 2,4,...).

Recall that by **A.4** x_k^* (the true optimal state trajectory) is stable in the sense of Lyapunov for initial conditions near the equilibrium, and in concert with Theorem 4 we are thus ensured that an initial condition near the equilibrium will also remain in its vicinity. As we apply test (32c) to all feasible elements in $\mathcal{X}$, at least one initial condition $x_0 \in \mathcal{X}$ will therefore start and then *remain in the nearby vicinity of the equilibrium*. Ultimately, by combining (32b) and (32c) we are ensured that $\mu_{R,0}^{*N'_M}$ is nearly constant during the interval needed for all feasible gridded points in $\mathcal{X}$ to reach the vicinity of the equilibrium.

In UCPADP, we determine $\mu_{R,0}^{*N'_M}$ numerically efficiently in a manner similar to API implemented with ADP. We do this using a nested scheme that repeatedly switches between backward-calculation phases (successively generating control policies with longer associated horizons) and forward-calculation phases (applying tests (32b) and (32c), and eventually terminating when both tests pass). A description of the phases in UCPADP follows, see Fig. 1 for an illustration. For now, assume ε_μ and ε_x are given (fixed) vectors.

First, we arbitrarily choose a small initial horizon N and perform N backward-calculation iterations, giving us (among other data) μ_R^{*N}. We can then perform test (32b) and, by performing $N/2$ forward-calculation steps, test (32c). If both tests pass we terminate and return $\mu_{R,0}^{*N}$ as our approximation of μ^*. Conversely, if either of these tests fail by **A.4** we are ensured that increasing the horizon sufficiently will give a control policy that satisfies the tests. In UCPADP we chose to proceed by increasing the horizon to $3N$. Fortunately, in our DP scheme we can compute μ_R^{*3N} using only $2N$ additional backward-calculation iterations by resuming the backward-calculation from μ_R^{*N}. This is possible as each successive backward-calculation step is independent of the total horizon. After generating μ_R^{*3N} we can now again test (32b) and (32c). Should both tests pass we can return $\mu_{R,0}^{*3N}$ as our approximation of μ^*, and otherwise recursively repeat this procedure of doubling the number of back-calculation steps until the tests pass (i.e. generating and testing horizons $N, 3N, 9N, 27N, \ldots$). A pseudocode implementation of the UCPADP method is listed in Algorithm 1.

Up to this point we have assumed that the problem solution is unique (**A.2**), converges to a stationary control policy (**A.3**), and all states converge to a unique equilibrium (**A.4**). Let us now briefly consider the case where we do not know if these assumptions hold beforehand. Beginning with **A.2**, recall that we can determine whether or not this assumption holds during the backward-calculation phase by checking if the minimum cost is unique, and in the case of a non-unique cost we can resolve this by simply returning one arbitrarily selected optimal solution. Let us now focus on the case where **A.3** and **A.4** are unverified. Applying the UCPADP method gives one of two possible outcomes: UCPADP either never terminates (i.e. (32b) and (32c) never pass), or it terminates after a finite number of back-calculation iterations. If UCPADP never terminates, then one possible cause is that **A.3** and/or **A.4** do not hold (i.e. the termination criteria (32b) and (32c) correctly detected a non-stationary control policy and/or detected that the system states do not converge to a single equilibrium). Alternatively, it is possible that the problem's discretisation and/or tolerances were poorly chosen. Regardless, should UCPADP never terminate it is clear that no valid solution could be generated. If UCPADP *does* terminate, we are assured that either: *(i)* **A.3** and **A.4** do hold and a near-optimal control policy is generated, or *(ii)* the problem is maliciously nonlinear and **A.3** and/or **A.4** do not hold (which went undetected by (32b) and (32c)), ultimately giving a control policy without any clear optimality guarantees. As the class of problems we can attempt to solve with UCPADP covers general non-linear systems it is not surprising that there exist pathological problems that lead UCPADP (and ADP in general) to generate erroneous solutions. Ultimately it is up to the user of UCPADP to determine whether or not the studied problem is of a class that satisfies the (arguably mild) assumptions **A.3** and **A.4**.

In Algorithm 1, we extend the notion of termination used thus far by adding a parameter $N_{\max}$ that allows for configuring a maximum horizon that terminates UCPADP if $N > N_{\max}$. This acts as a safety and guarantees that UCPADP terminates after a finite number of iterations. In the event that this limit triggers UCPADP to terminate we can conclude that either the minimum horizon is larger than $N_{\max}$, that **A.3** and/or **A.4** do not hold, or the discretisation and/or tolerances were poorly chosen. Of course, should this happen then we can not say anything about the stability (let alone the optimality) of the returned control policy.

Tests (32b) and (32c) are straightforward to compute exhaustively, as the initial conditions x_0 come from the discrete set $\mathcal{X}$. Furthermore, in UCPADP we have chosen to double the number of additional back-calculation steps to perform between each test evaluation. This attempts to balance the time spent on backward-calculation iterations and the horizon length sufficiency tests, though we may ultimately solve for problem horizons up to $2N_M$, as indicated by (33b). Ultimately this choice is arbitrary, and it is possible for some problems to use another scheme for selecting a new length.

From a practical perspective, we have found that setting $\varepsilon_\mu \approx 2d_u$ and $\varepsilon_x \approx 2d_x$ (the distance between points in $\mathcal{U}$ and $\mathcal{X}$ respectively) is a good design choice for well-

Algorithm 1. Pseudocode UCPADP algorithm. Here, $\mathrm{DP}_{\text{1-back}}$ and $\mathrm{DP}_{\text{1-fw}}$ are the one-step backward and forward ADP operations. $\mathcal{X}$ is the set of initial conditions tested in the ADP method. N_{init} is the initial problem horizon. C_N is the cost-to-go after N iterations. Note here that a reverse notation is used for the calculated control policy; μ_1 corresponds to the state-feedback control policy from the first back-calculation step (i.e. μ_{N-1}^{*N}) while μ_N corresponds to the last (i.e. μ_0^{*N}). We can view the index k as counting the number of back-calculation steps performed. Note the abuse of notation on line 14 that indicates the Δ_x^N and Δ_μ^N tests respectively.

1: **function** UCPADP($\mathcal{X}$, $N_{\max}$, N_{init})
2: $N_b \leftarrow N_{\text{init}}$ ▷ Batch back-calculation steps
3: $N \leftarrow 0$ ▷ Cumulative back-calculation steps
4: $C_0 \leftarrow 0$ ▷ Set initial cumulative cost to zero
5: **repeat**
6: **for** $N \leftarrow N, N + N_b$ **do**
7: $\mu_{N+1}, C_{N+1} \leftarrow \mathrm{DP}_{\text{1-back}}(C_N)$
8: **end for**
9: $X_{CL} \leftarrow \mathcal{X}$
10: **for** $i \leftarrow 1, \lceil N/2 \rceil$ **do**
11: $X_{CL} \leftarrow \mathrm{DP}_{\text{1-fw}}(X_{CL}, \mu_N)$
12: **end for**
13: $N_{\text{b}} \leftarrow 2 \cdot N_{\text{b}}$ ▷ Raise N_b by doubling
14: **until** $N > N_{\max}$ or $(|X_{CL} - \mathrm{mean}(X_{CL})| < \varepsilon_x$ and $|\mu_N - \mu_k| < \varepsilon_u \ \forall k \in [\lceil N/2 \rceil, N])$
15: **return** μ_N, X_{CL}, N
16: **end function**

behaved problems. Smaller values raise the risk of never terminating, e.g. due to residual state trajectory jitter caused by approximation inherent to interpolation, while larger values give an unnecessarily large approximation of the true control policy $\mu_{R,0}^*$. Ultimately, this implies that UCPADP has to some degree only one tuning parameter: the ADP discretisation, which trades off accuracy with computational time and memory demands.

As UCPADP is based on interpolating ADP (and in turn DP) it is subject to the inherent limitations of DP methods, in particular its poor scaling with problem dimensionality (colloquially referred to as the "curse of dimensionality") (Bellman, 1954; Bertsekas, 2017). This limits UCPADP to low- to moderate-dimensional problems. The examples shown in the following section (with two state variables and one control variable, giving a total of three independent variables) are easily solved using an ordinary desktop computer on the order of one minute to one hour (depending on the demanded solution accuracy). In practice, we expect UCPADP to be viable for up to 4–6 continuous-variable problems, depending on the discretisation of the state and control variables, the nature of the problem, and the available computational power.

A general implementation of the UCPADP method in the MATLAB language, including the numerical examples in the following section, is available at `https://gitlab.com/lerneaen_hydra/ucpadp`.

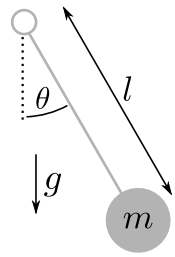


Figure 2: A simple pendulum.

V. Representative examples

We illustrate the UCPADP method, introduced in Section IV, by solving two simple problems. Though "toy" problems in some sense, recall that Assumption 1 allows for significantly more difficult (and practically relevant) problems. First we consider the classical minimum-time inverted pendulum problem, where we highlight the stopping criterion of UCPADP. Afterwards, we consider the problem of maintaining an average pendulum angle with minimum control power, illustrating the average-constraint properties shown in Subsection III-B.

We will consider the dynamical system given by a simple pendulum (Fig. 2) for both problems. For a pendulum with length l, point mass m, gravitational force g, damping coefficient d, angle θ, and applied torque u, the dynamic equation for the system can be derived as

$$\ddot{\theta} + \frac{d}{m}\dot{\theta} + \frac{g}{l}\sin(\theta) = \frac{1}{ml^2}u. \tag{34}$$

In both the following examples we will assume a discrete-time control system with sample rate t_s, i.e. the control input u is piecewise constant over intervals of uniform time t_s. If the problem is reformulated as a set of coupled first-order ordinary differential equations with a state variable vector

$$x \triangleq \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} \tag{35}$$

then we can express the state at the next sample as

$$x_{k+1} = f_p(x_k, u_k), \tag{36}$$

where f_p is given by solving (34) over a time t_s with initial condition x_k and constant control input u_k.

A. The inverted pendulum

To illustrate the mechanics of UCPADP's termination criterion, consider the traditional minimum-time inverted pendulum problem (formulated here as an infinite-horizon problem)

$$J^* = \min_{\bar{x},\bar{u}} \lim_{N\to\infty} \frac{1}{N} \sum_{k=0}^{N-1} f_c(x_k) \tag{37a}$$

$$f_c(x) = \begin{cases} 0 & \text{if } |\theta_k - \pi| < 2d_x, |\dot{\theta}_k| < 2d_x \\ 1 & \text{else} \end{cases} \tag{37b}$$

subject to

$$x_{k+1} = f_p(x_k, u_k) \tag{37c}$$
$$|u_k| \leq 1 \tag{37d}$$
$$-2 \leq \theta_k \leq 3.5 \tag{37e}$$
$$-1.5 \leq \dot{\theta}_k \leq 2. \tag{37f}$$

All the following results are shown for a sample time of $t_s = 0.2$, pendulum parameters set to give the system dynamics equation $\ddot{\theta} + \sin(\theta) = u$, state variables discretised by a Cartesian grid with separation $d_x = [0.05, 0.05]^T$ in the range allowed by (37e) and (37f), and the control variable discretised with even spacing $d_u = 0.01$ in the range allowed by (37d). Setting ε_x and ε_μ to the suggested value of twice the discretisation gives $\varepsilon_x = [0.1, 0.1]^T$ and $\varepsilon_\mu = 0.02$.

Note that the cost function (37b) equally penalises all pendulum configurations other than the single vertical zero-velocity state combination, and with an infinite horizon (and small enough d_x) gives a solution arbitrarily close to the traditional minimum-time formulation. The state bounds (37e) and (37f) have been chosen to give a reasonable range for the specific initial value we will study shortly.

For the above problem, UCPADP terminates after testing a horizon of $N'_M = 135$, indicating that $45 < N_M \leq 135$. An illustration of termination criterion (32b) is shown in Fig. 3, where we can verify the condition is satisfied as all values are above $\lceil N/2 \rceil = 68$. Furthermore, Δ_μ^k will by construction take values from $\mathcal{U} = \{0, \pm 0.01, \pm 0.02, \ldots, \pm 1\}$. For $\varepsilon_\mu = 0.02$ (32b) will thus only be satisfied for values $-0.01, 0, 0.01$. We can see this in Fig. 3, where $\Delta_\mu^{135} = 0$. Similarly, criterion (32c) is illustrated in Fig. 4, where we can verify that representative trajectories all converge to a region bounded by ε_x (shown by the yellow box). An illustration of the control policy ultimately generated by UCPADP is shown in Fig. 5. Solving this specific problem took approximately 10 minutes using a standard desktop PC.

Fig. 6 shows a comparison of the solution generated by UCPADP and a reference solution, generated by formulating a problem with an explicit horizon of $N = 10N'_M = 1350$ (i.e. one order of magnitude longer the UCPADP horizon), for $x_0 = [0, 0]^T$. Here the reference solution is generated using a traditional ADP scheme, configured with the same sample time and state/control grid discretisation. Note that we intentionally compare the UCPADP solution to a traditional ADP solution (in contrast to, for instance, an analytical solution) as we wish to highlight the accuracy of the automatically sized horizon, rather than the accuracy of an interpolating ADP scheme.

The average cost over the time interval shown in Fig. 6 is 0.09664 for the UCPADP solution, while the cost associated with the reference solution is 0.09689, i.e. a deviation* of 0.25%. We can conclude (for this specific problem) that the cost associated with the UCPADP solution is virtually

*The fact that the UCPADP solution has a lower associated cost is likely due to the inherent approximation of interpolating ADP.

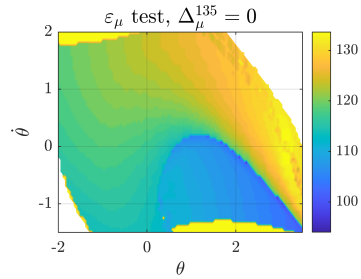


Figure 3: Visualisation of $\varepsilon_\mu = 0.02$ test for $N'_M = 135$. The coloured regions indicate the number of samples that the control policy varies less than ε_μ, while white regions indicate a feasible solution could not be found, i.e. white regions lie outside of $\mathcal{F}'_{135}$. Note that as $\mathcal{U}$ is discrete then Δ_μ^{135} is also, i.e. we here have Δ_μ^{135} identically equal to zero.

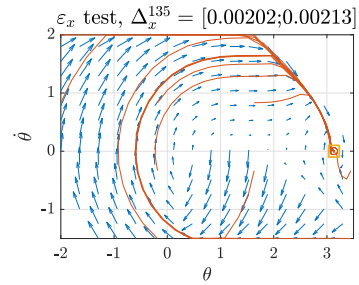


Figure 4: Visualisation of $\varepsilon_x = [0.1, 0.1]^T$ test for $N'_M = 135$. Blue arrows indicate the motion of the system through its phase space. Representative closed-loop trajectories are shown in red. The closed-loop state at $N = 135$ is shown by (overlapping) small red circles near $\theta = \pi$, $\dot{\theta} = 0$. The yellow box indicates the Δ_x^{135} termination criterion, which is satisfied as all states at $N = 135$ lie inside the box.

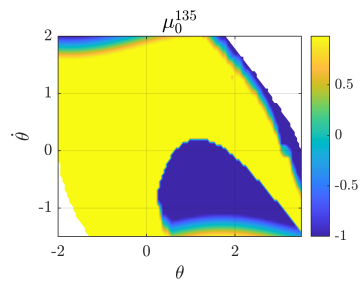


Figure 5: Control policy associated with (37) for horizon $N = 135$. The coloured region shows the optimal control to apply for any given state, while white regions indicate infeasible states, i.e. outside of $\mathcal{F}'_{135}$.

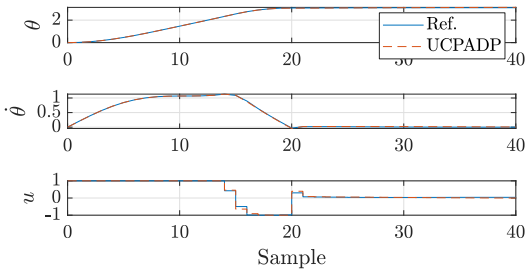


Figure 6: Comparison of the solutions given by UCPADP and an open-loop ADP reference method for $x_0 = [0, 0]^T$.

identical to a conventional ADP solution, indicating that the identified horizon $N'_M = 135$ was sufficient.

B. The constant-angle pendulum

Let us now consider a problem that illustrates the properties of the average constraint introduced in Subsection III-B. Assume we wish to solve

$$J^* = \min_{\bar{x},\bar{u}} \lim_{N\to\infty} \frac{1}{N} \sum_{k=0}^{N-1} u_k^2 \tag{38a}$$

subject to

$$x_{k+1} = f_p\left(x_k, u_k\right) \tag{38b}$$

$$\lim_{N\to\infty} \frac{1}{N} \sum_{k=0}^{N-1} \theta_k = \theta_{\text{ref}} \tag{38c}$$

$$|u_k| \leq 1, |\theta_k| \leq 1, |\dot{\theta}_k| \leq 1, \tag{38d}$$

i.e. the problem of keeping the average pendulum angle at a setpoint θ_{ref} while minimising the quadratic control input u_k^2.

By Theorem 7 we can avoid including the average constraint (38c) by augmenting the cost functional (38a) as

$$J_R^* = \min_{\bar{x}_R,\bar{u}_R} \lim_{N\to\infty} \frac{1}{N} \sum_{k=0}^{N-1} u_k^2 + \lambda\theta_k \tag{39}$$

for a constant scalar λ. Assuming the problem reaches an equilibrium with control u_{eq} and states $\theta = \theta_{\text{ref}}, \dot{\theta} = 0$, by (34) we have $u_{\text{eq}} = mgl\sin(\theta_{\text{ref}})$. We can thus express the equilibrium cost as

$$c_{\text{eq}} = (mgl\sin(\theta_{\text{ref}}))^2 + \lambda\theta_{\text{ref}} \tag{40}$$

which is a function of one variable. Equation (40) has one unique stationary point (a minimum) in the permissible range $|\theta| < 1$, and we can thus find the specific value λ that gives the lowest equilibrium cost at the desired setpoint by setting $\frac{dc_{\text{eq}}}{d\theta_{\text{ref}}} = 0$ and solving for λ, giving

$$\lambda_0 = -2m^2g^2l^2\sin\left(\theta_{\text{ref}}\right)\cos\left(\theta_{\text{ref}}\right). \tag{41}$$

We can now reformulate (38) as the equivalent problem

$$J_R^* = \min_{\bar{x}_R,\bar{u}_R} \lim_{N\to\infty} \frac{1}{N} \sum_{k=0}^{N-1} u_k^2 - \theta_k\lambda_0 \tag{42a}$$

subject to

$$x_{k+1} = f_p\left(x_k, u_k\right), |u_k| \leq 1, |\theta_k| \leq 1, |\dot{\theta}_k| \leq 1. \tag{42b}$$

As in the previous example, we discretise the state and control variables evenly in the permissible space, here with separation $d_x = [0.02, 0.02]^T$ and $d_u = 0.01$ respectively. Solving (42) for pendulum parameters resulting in a system dynamics equation $\ddot{\theta} + \dot{\theta} + \sin(\theta) = u$ and $\theta_{\text{ref}} = 0.5$ gives the results shown in Fig. 7 (again compared with a reference solution given by explicitly choosing a large horizon, one order of magnitude larger than the horizon given by UCPADP).

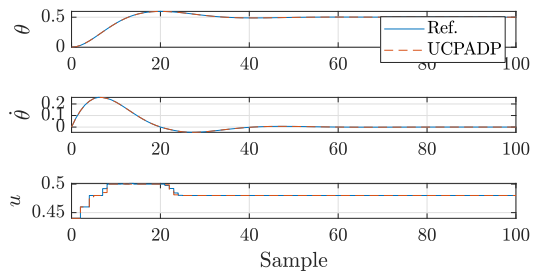


Figure 7: Comparison of the solutions given by UCPADP and an open-loop ADP reference method for $x_0 = [0, 0]^T$. The state trajectories are nearly identical and the control trajectory displays only very small differences.

For this problem, we find that the UCPADP solution generates a solution with control cost (i.e. $\sum u_k^2$) of 0.2321 over the horizon shown in Fig. 7, while the control cost associated with the reference solution is 0.2323 (i.e. a deviation of 0.09%), again showing that the accuracy of

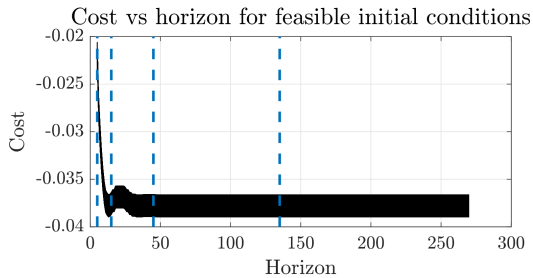


Figure 8: Cost of the applying control policy associated with varying problem horizon lengths. The cost is shown for all feasible initial conditions, resulting in a range of costs for any given horizon (e.g. $[-0.037, -0.039]$ for horizons ≥ 40). Problem horizons tested by UCPADP are shown with dashed lines.

the UCPADP solution is virtually identical to that of a reference ADP solution.

For comparison, in Fig. 8 we also show the solution quality parameterised by different finite horizons. More specifically, we solve the finite-horizon counterpart of (42), i.e. using the notation introduced in (25), for varying finite horizons N (denoted the *problem horizon*), resulting in the associated control policies $\mu_{R,0}^{*N}$. We then apply the control policy to the set of initial conditions feasible with a long horizon $N = 1350$ (denoted the *trajectory horizon*). The plot shows the augmented cost of the trajectory horizons, i.e. J_R^*, parameterised by different problem horizons. We can identify that the average cost is higher for short problem horizons than for long problem horizons, and that the cost associated with problem horizons $\gtrsim 40$ is constant, indicating that for this problem a problem horizon $\gtrsim 40$ is sufficient. It may therefore seem like UCPADP is inefficient in its choice of problem horizon (135 samples). However, computing the average cost of any given problem horizon shown in Fig. 8 is time consuming, with each individual problem horizon taking approximately the same time to compute as the *entire* UCPADP solution, as well as requiring problem-specific knowledge of the initial conditions and trajectory horizon to average over. The trade-off between spending time computing additional back-calculation steps and checking whether a given horizon is sufficiently large thus motivates a scheme like our proposed horizon-doubling method.

VI. Conclusions

In this paper we have introduced UCPADP, a numerical method inspired by API. UCPADP can be used to generate a near-optimal control policy for general undiscounted continuous-valued infinite-horizon nonlinear optimal control problems. The problem can also optionally be constrained to converge to a given equilibrium. The primary contribution of UCPADP is the introduction of a termination criterion that is amenable to the undiscounted case, while still allowing for general costs and constraints. We have evaluated the method by solving two simple, but representative, problems. For both examples we showed that the generated control policy was on par with the accuracy of a reference ADP solution (whose accuracy is determined by the chosen discretisation of the problem).

UCPADP has several properties that render it useful as as one part of the process of constructing an on-line controller. Firstly, it shares a property with other API methods in that it does not require any a-priori information about a suitable horizon, instead performing an indefinite number of iterations and terminating when a suitable problem horizon is found. Secondly, the tuning parameters are simple to grasp, as they trade off solution accuracy with computational time and memory demands. Finally, the output from UCPADP, as with other API methods, is a control policy (i.e. a state feedback table). After this control policy is computed in an off-line phase it can in turn be used to construct a subsequent on-line controller with very low computational demand, only requiring a simple interpolation operation to determine the control signal.

Full source code of the implementation as well as the specific problems studied is available at `https://gitlab.com/lerneaen_hydra/ucpadp`.

Funding

This work was performed within the Combustion Engine Research Center at Chalmers (CERC) with financial support from the Swedish Energy Agency.

References

Andréasson, N., Evgrafov, A., Patriksson, M., Gustavsson, E., Nedelkova, Z., Sou, K. C., & Önnheim, M. (2016). *An introduction to continuous optimization* (Third ed.). Studentlitteratur.

Bellman, R. (1954). The theory of dynamic programming. *Bulletin of the American Mathematical Society*, *60*(6), 503–515.

Bertsekas, D. P. (2011, August). Approximate policy iteration: a survey and some new methods. *Journal of Control Theory and Applications*, *9*(3), 310–335. doi: 10.1007/s11768-011-1005-3

Bertsekas, D. P. (2012). *Dynamic programming and optimal control* (4th ed., Vol. 2). Athena Scientific.

Bertsekas, D. P. (2017). *Dynamic programming and optimal control* (4th ed., Vol. 1). Athena Scientific.

Bertsekas, D. P., & Shreve, S. E. (1979). Existence of optimal stationary policies in deterministic optimal control. *Journal of Mathematical Analysis and Applications*, *69*(2), 607 - 620. doi: 10.1016/0022-247X(79)90171-9

Guo, W., Si, J., Liu, F., & Mei, S. (2017). Policy Approximation in Policy Iteration Approximate Dynamic Programming for Discrete-Time Nonlinear Systems. *IEEE Transactions on Neural Networks and Learning Systems*, 1–14. doi: 10.1109/TNNLS.2017.2702566

Munos, R., & Moore, A. (2002). Variable Resolution Discretization in Optimal Control. *Machine Learning*, *49*(2/3), 291–323. doi: 10.1023/A:1017992615625

Powell, W. B. (2009, April). What you should know about approximate dynamic programming: Approximate Dynamic Programming. *Naval Research Logistics (NRL)*, *56*(3), 239–249. doi: 10.1002/nav.20347

Puterman, M. L. (1994). *Markov decision processes: Discrete stochastic dynamic programming.* John Wiley & Sons.

Puterman, M. L., & Brumelle, S. L. (1979, February). On the Convergence of Policy Iteration in Stationary Dynamic Programming. *Mathematics of Operations Research*, *4*(1), 60–69. doi: 10.1287/moor.4.1.60

Santos, M. S., & Rust, J. (2004, January). Convergence Properties of Policy Iteration. *SIAM Journal on Control and Optimization*, *42*(6), 2094–2115. doi: 10.1137/S0363012902399824

Santos, M. S., & Vigo-Aguiar, J. (1998, March). Analysis of a Numerical Dynamic Programming Algorithm Applied to Economic Models. *Econometrica*, *66*(2), 409. doi: 10.2307/2998564

Scherrer, B. (2014, May). Approximate Policy Iteration Schemes: A Comparison. *arXiv:1405.2878 [cs, stat]*. Retrieved from `http://arxiv.org/abs/1405.2878` (arXiv: 1405.2878)

Stachurski, J. (2008, March). Continuous State Dynamic Programming via Nonexpansive Approximation. *Computational Economics*, *31*(2), 141–160. doi: 10.1007/s10614-007-9111-5

Sundstrom, O., & Guzzella, L. (2009, July). A generic dynamic programming matlab function. In *2009 ieee control applications, (cca) intelligent control, (isic)* (p. 1625-1630). doi: 10.1109/CCA.2009.5281131

Trélat, E., & Zuazua, E. (2015). The turnpike property in finite-dimensional nonlinear optimal control. *Journal of Differential Equations*, *258*(1), 81 - 114. doi: 10.1016/j.jde.2014.09.005

Zaslavski, A. J. (2014). *Turnpike phenomenon and infinite horizon optimal control*. Springer International Publishing. doi: 10.1007/978-3-319-08828-0

PAPER C

Optimal Transient Real-Time Engine-Generator Control in the Series-Hybrid Vehicle

Jonathan Lock, Rickard Arvidsson, Tomas McKelvey

Published in Dynamic Systems and Control Conference,
vol. 2, October 2019.
DOI: 10.1115/DSCC2019-8964

Proceedings of the ASME 2019
Dynamic Systems and Control Conference
DSCC2019
October 9-11, 2019, Park City, UT, USA

DSCC2019-8964

OPTIMAL TRANSIENT REAL-TIME ENGINE-GENERATOR CONTROL IN THE SERIES-HYBRID VEHICLE

Jonathan Lock
Department of Electrical Engineering
Chalmers University of Technology
Gothenburg 41258
Sweden
Email: lock@chalmers.se

Rickard Arvidsson
Department of Electrical Engineering
Chalmers University of Technology
Gothenburg 41258
Sweden
Email: arvidssr@chalmers.se

Tomas McKelvey
Department of Electrical Engineering
Chalmers University of Technology
Gothenburg 41258
Sweden
Email: tomas.mckelvey@chalmers.se

ABSTRACT

We study the dynamic engine-generator optimal control problem with a goal of minimizing fuel consumption while delivering a requested average electrical power. By using an infinite-horizon formulation and explicitly minimizing fuel consumption, we avoid issues inherent with penalty-based and finite-horizon problems. The solution to the optimal control problem, found using dynamic programming and the successive approximation method, can be expressed as instantaneous non-linear state-feedback. This allows for trivial real-time control, typically requiring 10–20 CPU instructions per control period, a few bytes of RAM, and 5–20 KiB of nonvolatile memory. Simulation results for a passenger vehicle indicate a fuel consumption improvement in the region of 5–7% during the transient phase when compared with the class of controllers found in the industry. Bench-tests, where the optimal controller is executed in native hardware, show an improvement of 3.7%, primarily limited by unmodeled dynamics. Our specific choice of problem formulation, a guaranteed globally optimal solution, and trivial real-time control resolve many of the limitations with the current state of optimal engine-generator controllers.

1 Introduction

Engine-generator (gen-set) systems are used in a wide range of applications, from the drivetrain in a series-hybrid passenger vehicle to the larger diesel-electric locomotive and natural gas power plants. Regardless of application a gen-set fundamentally consists of a combustion engine mechanically connected to an electric machine, whose purpose is to convert a combustible fuel into electrical power. One particularly relevant attribute of a well-designed gen-set is it's fuel efficiency, i.e. how efficiently consumed fuel is converted to electrical power. In this paper we will consider the gen-set of a series-hybrid passenger vehicle and specifically a method of improving its fuel efficiency (though this method can easily be extended to other applications). In particular, we will focus on methods that include transient operation, i.e. where the gen-set engine speed is allowed to vary over time.

One method of generating a fuel-efficient control scheme is to formulate the objective as an *optimal control problem*, and then use a method from the field of optimal control to solve the problem. The specific formulation of the optimal control problem and the method used to solve it is crucial. In this paper we will require the solution to

1. accurately capture our intent of minimizing fuel consumption,
2. be (close to) the globally optimal solution, and
3. allow for real-time control with low computational demand.

Previous work in transient gen-set control covers a range

of different optimal control formulations, solution methods, and with varying degrees of real-time applicability.

If we first consider the formulation of the optimization problem, some have elected to use quadratic penalty-based formulations that balance the quadratic gen-set losses and quadratic deviation from requested power [4, 6]. Though a conventional formulation, this results in the gen-set generally never delivering exactly the requested power. Furthermore, this formulation does not generally minimize fuel consumption (as the minimand is quadratic losses and power deviation, not consumed fuel). One alternative formulation is to apply strict equality constraints to the delivered power at all times [3]. This is suitable for applications where there is a true requirement to deliver a given power at every point in time, however, as described in their paper this is at cost of potential infeasibility as well as reducing the range of permissible control actions, virtually guaranteeing a higher fuel consumption compared to the case where delivered power is allowed to vary during the transient phase. A third option is to use an explicit minimum-fuel problem formulation [5]. This choice unequivocally matches our intent, but has the potential of leading to a problem formulation that is difficult to numerically solve.

The specific method used to solve the optimization problem is another aspect relevant to consider. The gradient descent method [4] is a traditional method, but requires a differentiable problem formulation and is not guaranteed to return a globally optimal solution. The multiple shooting and sequential quadratic programming method [5] is more sophisticated, but is still subject to the same fundamental limitations. An alternate approach is the use of stochastic optimization, e.g. a genetic algorithm [6]. This avoids the need for a differentiable problem formulation and is less likely to be trapped in a local optimal solution, but is typically computationally demanding and has no global optimality guarantee. One attractive option is to use a method based on Pontryagin's maximum principle or dynamic programming, as these typically are ensured to give the globally optimal solution. An example of this is seen in [3], though the specific method used requires hours of computational time to solve a problem that covers 10 seconds of operation.

Finally, it is worth considering how well (if at all) solution methods can translate into real-time control. One straightforward method is to limit equilibrium operation to a small number of discrete points [6], allowing for storing pre-computed trajectories and applying them during runtime. Though effective, this scales poorly with the number of equilibrium points, as we will typically need to store the optimal control trajectories over time for every combination of equilibrium points. Alternatively, domain-specific knowledge can be used to implement real-time methods

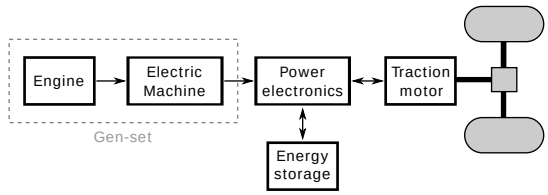


FIGURE 1: Series-hybrid vehicle drivetrain, the gen-set consists of the combustion engine and electrical machine.

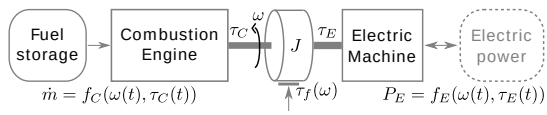


FIGURE 2: Gen-set dynamic model overview.

that approximate the behavior of the computed optimal solutions [3, 4]. Though this can lead to very efficient real-time control methods, we are no longer guaranteed optimality while also requiring extensive knowledge of the specific problem to be solved.

In this paper we will introduce a method that addresses many of the current limitations. In particular, we will formulate the optimization problem as an explicit fuel consumption minimization problem, constrained to deliver a given average electrical power over an infinite time horizon which we solve using dynamic programming, ensuring a (close-to) globally optimal solution. The infinite time horizon is useful both as it ensures us that fuel consumption is balanced between the transient and stationary phase, as well as giving a solution that can be formulated as state-feedback. This ultimately allows for implementing real-time optimal control using a bare minimum of computational power (on the order of 5–20 KiB of nonvolatile memory, a few bytes of RAM, and 10–20 CPU instructions per control period).

2 Problem formulation

In this paper we will study the gen-set of a series-hybrid vehicle, as illustrated in Fig. 1. In particular, we will model the gen-set as a combustion engine directly connected to an electrical machine, with dynamics that arise from the moment of inertia of moving parts (illustrated in Fig. 2).

In the gen-set, we model the combustion engine (CE) as consuming fuel at a rate

$$\dot{m}(t) = f_C(\omega(t), \tau_C(t)), \tag{1}$$

i.e. a function of the instantaneous torque $\tau_C(t)$ and crankshaft angular speed $\omega(t)$. Similarly, the electric machine (EM) delivers or consumes electric power as

$$P_E(t) = f_E(\omega(t), \tau_E(t)) \tag{2}$$
$$= -\tau_E(t)\omega(t) - f_{E,\text{loss}}(\omega(t), \tau_E(t)), \tag{3}$$

for a given instantaneous torque $\tau_E(t)$ and crankshaft speed $\omega(t)$. We will view τ_E and τ_C as control inputs that should ultimately be selected in a way that minimizes the fuel consumption of the gen-set.

Note that we assume f_C and f_E are instantaneous functions of the current crankshaft velocity and their respective torques, without any additional dynamics. This instantaneous formulation allows for determining them by using standard experimentally obtained equilibrium maps (i.e. by experimentally measuring the fuel massflow and delivered electrical power respectively at a wide range of operating points). We will return to the validity of this assumption in Subsection 4.1. Furthermore, note that (3) implies that we can view the EM as an ideal mechanical to electrical converter (with power given by $-\omega(t)\tau_E(t)$), with all non-frictional losses lumped into $f_{E,\text{loss}}$. For brevity, we will no longer explicitly state the time-dependence of the previous terms.

We lump all frictional components in the CE and EM together to form $\tau_f(\omega)$, which is typically non-linear. This implies that we can view τ_C and τ_E as indicated torques rather than crankshaft torques. Similarly, we lump together all inertial terms forming a net moment of inertia J. This gives a net continuous-time dynamic gen-set model

$$\dot{\omega} = \frac{1}{J}\left(\tau_C + \tau_E + \tau_f(\omega)\right), \tag{4}$$

i.e. a non-linear first-order dynamic system. Note that the nonlinearity of τ_f implies that (4) is generally a non-linear differential equation.

This specific choice of modeling and sign notation implies that a negative EM torque will apply a retarding torque to the crankshaft and deliver electrical power. Similarly, for positive EM torques the EM will tend to accelerate the crankshaft and thus consume electrical power. In steady-state operation, where the gen-set delivers electrical power, we will thus have $\tau_C > 0$, $\tau_E < 0$, and $P_E > 0$.

Assume that we will ultimately implement real-time control using a digital controller with a fixed sample rate t_s, where the control signals τ_C and τ_E are held constant between sample times (zero-order hold). For convenience, we will use the notation

$$\omega_k \equiv \omega(kt_s) \tag{5a}$$
$$\tau_{C,k} \equiv \tau_C(kt_s) \tag{5b}$$
$$\tau_{E,k} \equiv \tau_E(kt_s) \tag{5c}$$

to indicate the discrete-time crankshaft speed and torques respectively. Using a numerical ODE solver we can solve (4) for a given initial condition and torques over time t_s, allowing us to introduce a discrete-time dynamic equation

$$\omega_{k+1} = f\left(\omega_k, \tau_{C,k}, \tau_{E,k}\right) \tag{6}$$

where f is given by the ODE solver.

We can now introduce the optimization problem we ultimately wish to solve as

$$(\tau_C^*, \tau_E^*) = \underset{\tau_C, \tau_E}{\operatorname{argmin}} \lim_{N\to\infty} \sum_{k=0}^{N} f_C\left(\omega_k, \tau_{C,k}\right) \cdot t_s \tag{7a}$$

subject to

$$\omega_{k+1} = f\left(\omega_k, \tau_{C,k}, \tau_{E,k}\right) \tag{7b}$$
$$\frac{1}{N}\sum_{k=0}^{N} f_E\left(\omega_k, \tau_{E,k}\right) = P_{\text{tgt}} \tag{7c}$$
$$g\left(\omega_k, \tau_{C,k}, \tau_{E,k}\right) \leq 0, \tag{7d}$$

i.e. over an infinite horizon minimize fuel consumption, while delivering a given average power P_{tgt}, and demanding the CE and EM torques and crankshaft velocity lie in a permissible range given by g. Note that though the sum in (7a) grows arbitrarily large with increasing N, the minimand (τ_C^*, τ_E^*) is for this problem well-defined [1], as $f_C \cdot t_s$ is finite (the CE can only consume a finite amount of fuel in finite time).

Note that the choice of an infinite horizon ($N \to \infty$) obviates the need for determining a control and prediction horizon commonly found in finite-horizon problem formulations (e.g. traditional model predictive control (MPC)).

In this paper we have studied a conventional direct-injection turbocharged gasoline combustion engine coupled with a permanent-magnet synchronous electric machine. These are representative examples of gen-set components

that could be found in a typical light-duty series-hybrid. The functions f_C and f_E are approximated by empirically measuring the physical fuel flow and delivered electrical power respectively for a large set of equilibrium operating points. Similarly, τ_f is determined by measuring the torque required to motor the unloaded crankshaft (i.e. τ_C, τ_E are set to 0) while maintaining a constant speed using an additional external EM. The function g is defined to give box constraints for the permissible state and control signals as

$$g(\omega, \tau_C, \tau_E) = \begin{cases} 0 & \text{if } 0 \le \omega \le 750 \\ & \text{and } 0 \le \tau_C \le 390 \\ & \text{and } -150 \le \tau_E \le 150 \\ 1 & \text{else.} \end{cases} \tag{8}$$

Illustrations of the combustion engine's normalized indicated specific fuel consumption, the electric machine's efficiency (defined as $\eta_{\text{EM}} \equiv (|f_E| - f_{E,\text{loss}})/|f_E|$, and the net friction τ_f are shown in Fig. 3.

3 Method

Minimum Horizon Dynamic Programming (MHDP), a method recently developed by the authors [2], is a method based on dynamic programming and the successive approximation method for solving general average-constrained infinite-horizon nonlinear optimal control problems of form

$$(x^*, u^*) = \underset{x,u}{\operatorname{argmin}} \lim_{N\to\infty} \sum_{k=0}^{N} c(x_k, u_k) \tag{9a}$$

subject to

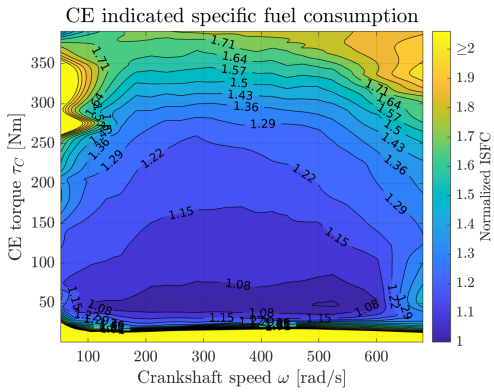
$$x_{k+1} = f_d(x_k, u_k) \tag{9b}$$

$$\frac{1}{N}\sum_{k=0}^{N} f_a(x_k, u_k) = \alpha \tag{9c}$$

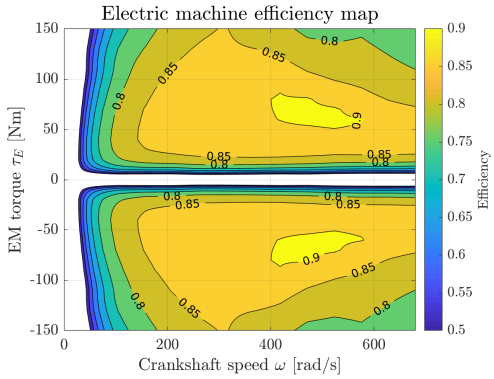
$$g(x_k, u_k) \le 0 \tag{9d}$$

where $x_k \in \mathbb{R}^n$ and $u_k \in \mathbb{R}^m$ are the k'th state and control variables respectively. Ultimately, MHDP determines the optimal controls u^* in the sense that (9a) is minimized while respecting the system dynamics and constraints.

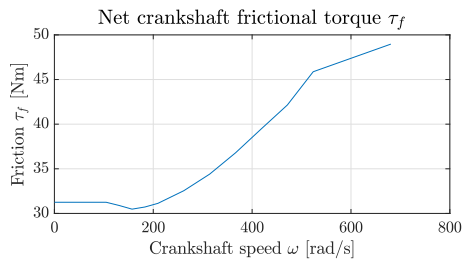
It can be shown that under some mild assumptions, for problems of form (9a), that the optimal control u^* to apply at any given time solely depends on the current system



(a) CE fuel consumption.



(b) EM electrical power.



(c) Net crankshaft friction.

FIGURE 3: Plots of normalized CE fuel consumption, EM efficiency, and τ_f. As the CE and EM are modeled as frictionless devices the crankshaft torques can be views as indicated rather than brake-specific torques.

state [2]. Essentially, there exists a function $f_{\text{sf}} : \mathbb{R}^n \to \mathbb{R}^m$ such that $u_k^* = f_{\text{sf}}(x_k)$. Typically f_{sf} is referred to as an *optimal control policy* or *optimal non-linear state feedback*. MHDP generates an approximation of f_{sf} and returns a look-up table with the optimal controls to apply for a given set of discrete state values. Once f_{sf} is known this allows for near-trivial on-line control, as the controller can simply look up the tabulated state value (typically using some form of interpolation if the current system state lies between tabulated values) and apply the resulting control signal.

Though a detailed description of the MHDP method is beyond the scope of this paper, an overview of the parameters required by the method can aid in understanding and interpreting the generated results. Given a non-linear (but well-behaved) cost c, system dynamics f_d, averaging function f_a, a demanded average α, and general constraints g MHDP can return the optimal control policy associated with (9). Furthermore, MHDP requires selecting a given state and control quantization; more densely sampled points will give a more accurate approximation of f_{sf}, but generating the state-feedback law will take longer as well as requiring more memory in the real-time controller (as the optimal control policy table will contain more entries). Finally, an additional consequence of the MHDP method is that the generated optimal control policies are not explicitly parameterized by the average constraint α (P_{tgt} in our application), but by another scalar parameter λ analogous to a Lagrange multiplier. We resolve this in this paper by searching for values of λ that give the desired average power levels.

Returning to our specific problem, solving (7) with MHDP will generate the optimal control policies

$$\left(\tau_C^*\left(\omega, P_{\text{tgt}}\right), \tau_E^*\left(\omega, P_{\text{tgt}}\right)\right), \tag{10}$$

which we can view as the optimal torques τ_C^* and τ_E^* to apply tabulated by different crankshaft speeds ω and the target-power P_{tgt}.

A block diagram illustrating the final controller construction is shown in Fig. 4. As is shown, the requested power P_{tgt} is fed into the tabulated control laws τ_C^*, τ_E^* along with the current crankshaft speed ω, giving a closed-loop non-linear state-feedback controller.

For our specific problem (7) the real-time system need only perform a two-dimensional interpolation operation every sample. This typically takes on the order of 10–20 CPU instructions and (using the parameters to be presented shortly in Table 1) requires a look-up-table consuming approximately 13 KiB (as $N_{P_{\text{tgt}}} \cdot N_\omega = 6560$ entries are required, using 16-bit words gives ≈ 13 KiB), both of which are trivially performed in virtually any embedded system.

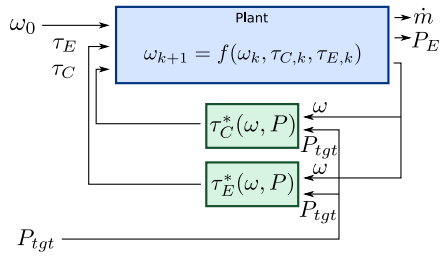


FIGURE 4: Net closed-loop block diagram.

TABLE 1: MHDP parameters used for (7).

Parameter	Value	Unit	Description
t_s	$50 \cdot 10^{-3}$	s	Sample time
$\Delta\tau$	4	Nm	Torque quantization
$N_{P_{\text{tgt}}}$	80	-	Number of evenly spaced equilibrium power levels.
N_ω	82	-	Number of evenly spaced discretized states.

4 Results

We have solved (7) using MHDP (taking approximately 12 hours with a standard desktop PC) with the configuration listed in Table 1, where the state and control signals have been discretized by evenly spacing them in the range allowed by (8). The specific choice of parameters in Table 1 was chosen to give a torque and crankshaft speed quantization that is on par with the measurement accuracy of the system.

Illustrations of the optimal CE and EM torques for representative target powers are shown in Fig. 5a and Fig. 5b. These plots illustrate the low computational demand of the on-line controller — the optimal control is given simply by consulting the stored torque corresponding to the current crankshaft speed and desired power.

Studying Fig. 5a and Fig. 5b allows us to come to some conclusions about the optimal control policy. For instance, for the 37 kW and 62 kW target power the EM will accelerate the gen-set when the crankshaft speed is less than approximately 200 rad/s. We speculate that this is due to the slightly lower efficiency of the CE at low speeds, as is seen in Fig. 3a, and it is thus optimal to spend as little time

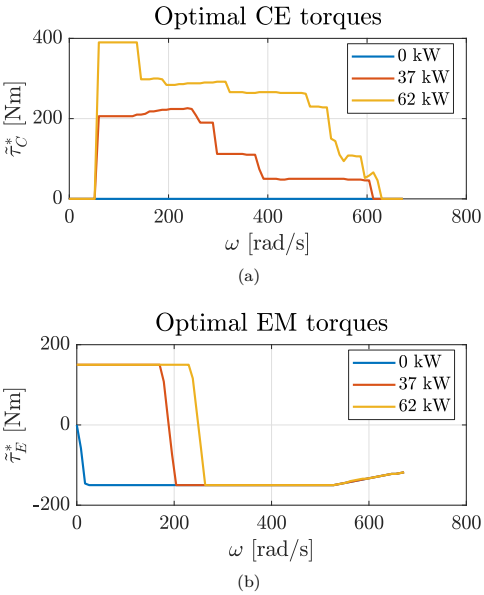


FIGURE 5: Optimal CE and EM torques for representative target powers.

as possible in this region. Furthermore, for the specific case of a 0 kW setpoint, $\tau_C = 0$, i.e. the combustion engine is set to consume no fuel. In the event that $\omega \neq 0$ then $\tau_E < 0$, which we can interpret as the optimal policy being to decelerate the crankshaft to zero speed while converting as much as possible of the crankshaft's stored kinetic energy to electrical energy.

In order to evaluate the performance of the optimal controller generated by MHDP we have chosen to compare it to a traditional gain-scheduled proportional controller. In this paper we will use a controller with control law

$$\begin{bmatrix} \tau_{C,P}\left(\omega, \omega^*_{\mathrm{EQ}}\right) \\ \tau_{E,P}\left(\omega, \omega^*_{\mathrm{EQ}}\right) \end{bmatrix} = \mathrm{clamp}\left(\begin{bmatrix} \tau^*_{C,\mathrm{EQ}}\left(\omega^*_{\mathrm{EQ}}\right) \\ \tau^*_{E,\mathrm{EQ}}\left(\omega^*_{\mathrm{EQ}}\right) \end{bmatrix} + \begin{bmatrix} k_p\left(\omega^*_{\mathrm{EQ}}\right) \\ -k_p\left(\omega^*_{\mathrm{EQ}}\right) \end{bmatrix} \left(\omega - \omega^*_{\mathrm{EQ}}\right) \right) \tag{11a}$$

$$\omega^*_{\mathrm{EQ}} = f^*_\omega\left(P_{\mathrm{tgt}}\right). \tag{11b}$$

Here, $f^*_\omega(P_{\mathrm{tgt}})$ is a function that gives the equilibrium crankshaft speed that consumes the least fuel for a given target power P_{tgt}. $\tau^*_{C,\mathrm{EQ}}$ and $\tau^*_{E,\mathrm{EQ}}$ are functions that return the torques needed to keep the system stationary at the equilibrium speed (which can be viewed as feed-forward terms), clamp is the clamping function that limits $\tau_{C,P}$ and $\tau_{E,P}$ to the permissible values given in (8), and k_p is a parameterized controller gain (which can be viewed as a proportional feed-back gain term). Due to the system's dynamics (4), the structure of (11) ensures that the gen-set will asymptotically approach the optimal operating point but will likely be sub-optimal during the transient approach.

The specific value of the controller gain k_p is critical for the performance of the closed-loop system. With small values a large amount of time is spent operating in regions with low efficiency, while large values will tend to apply very large torques which also typically reduces efficiency. In order to give a fair comparison between the optimal and proportional controllers, we have selected k_p to give a step response time from standstill ($\omega = 0$) to the target speed ($\omega = \omega^*_{\mathrm{EQ}}$) that takes an equally long time as the optimal solution (see Fig. 6). Note that we have now implicitly made our proportional controller to some degree optimal — it is unlikely that a real-world controller would happen to be optimally tuned and would presumably consume even more fuel than the results shown below.

Note that this specific choice of a gain-scheduled proportional controller ensures that the gen-set will deliver the same electrical power and consume fuel at the same rate when at equilibrium operation. In all following comparisons we will thus only consider the transient phase, as it is only in this region where the optimal controller has any potential for improvement.

In all the following results, we have chosen to use the piecewise-constant setpoint function

$$P_{\mathrm{tgt}}(t) = \begin{cases} P & 0 \leq t \leq T \\ 0 & \text{else,} \end{cases} \tag{12}$$

where T is chosen to be large enough to bring the gen-set to near-equilibrium operation. We have chosen this specific setpoint function as this allows us to fairly compare gen-set controllers. A full drive-cycle simulation would require choosing a specific torque-split controller (i.e. when and how much power to draw from the battery and gen-set respectively), the choice of which will greatly influence the total system behavior. Furthermore, we can note that only steps in power will give a truly optimal control signal, as the problem formulation (9) assumes a constant average power constraint. We speculate however that slowly-varying power demands will also be near-optimal (as no significant dynamics are excited). To limit the scope of this paper we will thus only consider the simple power step (12).

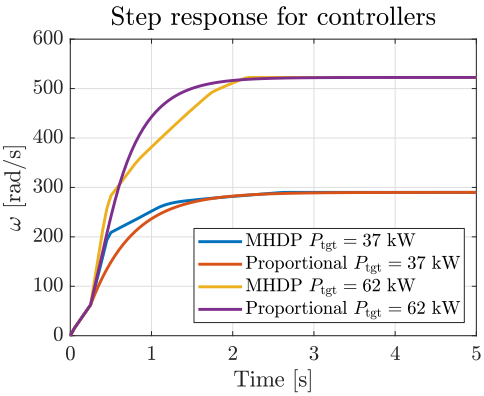


FIGURE 6: Step response from standstill to equilibrium crankshaft speed for optimal and proportional controllers. For fairness, the proportional controller gain is set to make the system reach equilibrium at the same time as the optimal controller.

In Fig. 7 we can study the operating points for the CE and EM for the optimal controller (7) and the traditional controller (11). We can see that though both controllers reach the same equilibrium operation point (indicated by x_{EQ}), their paths to the equilibrium point are different. In particular, in Fig. 7a we can see that the proportional controller generates a trajectory that happens to pass through a region with poor efficiency ($\omega = 75$, $\tau_C = 280$), a region the optimal controller avoids entirely.

As both controllers bring the gen-set to the same operating point in the same time we can numerically compare them by studying their effective efficiency (i.e. delivered electrical energy per consumed fuel mass)

$$\eta = \frac{\int_0^\infty f_E \, \mathrm{d}t}{\int_0^\infty f_C \, \mathrm{d}t}. \tag{13}$$

Note that as $P_{\text{tgt}} = 0$ for $t > T$, (13) can be viewed as a measure of the efficiency of the gen-set during solely transient operation, and indicates the gen-set efficiency both for an increase and decrease in target power.

As the controllers do not necessarily deliver the same amount of energy during the transient phase, the time T in (12) is adjusted separately for each controller until the total delivered power is identical.

In essence, both controllers will start and end at the same crankshaft speed, reach the same equilibrium speed at the same time, and deliver the same amount of energy over the whole cycle. The controller's freedom lies in the choice of transient torques, as well as when electrical power is delivered.

Fig. 8 shows the efficiency improvement during the transient phase of operation, with fuel savings on the order of 2–7%. We believe that the large variation in efficiency gains is primarily due to the traditional controller sometimes performing fairly well (e.g. at 52 kW) and sometimes poorly (e.g. for ≤ 45 kW) compared to the optimal controller. We can see an indication of this in Fig. 7, where in Fig. 7a the proportional controller's trajectory passes through an inefficient region, while in Fig. 7c the proportional controller's trajectory happens to be closer to the optimal trajectory.

4.1 Bench-test results

In this section we verify the previous results by implementing and evaluating the optimal controller in a physical, real-time, control system.

We have performed this test using a light-duty plug-in hybrid electric vehicle with dynamometers mounted on all wheels as shown in Fig. 9. We use the same problem formulation as in the previous section, i.e. solve for (7), (8), with an additional constraint limiting the electrical power to 15 kW due to the vehicle's relatively small inverter. Mathematically we can demand this by adding

$$|f_E| \leq 15 \cdot 10^3$$

as a constraint to (8).

Notably, due to the minimal computational load of implementing the optimal control policy as given by MHDP, the entire control loop is performed natively in the vehicle's engine control unit without any support from external hardware (as shown in Fig. 4). In these tests we have generated state-feedback control laws for 102 discrete crankshaft speeds over the gen-set's operating range (using linear interpolation to determine τ_C and τ_E for intermediate crankshaft speeds) and two power levels ($P_{\text{tgt}} = P$ and $P_{\text{tgt}} = 0$). τ_C^* and τ_E^* were stored as 32-bit floating point values, giving a total nonvolatile memory consumption of 1.6 KiB. Tests have shown that the on-line controller consumes on the order of 10–15 CPU cycles per sample, which is similar to that of the existing (suboptimal) controller. We have compared the optimal controller with a controller of form (11), which is comparable to controllers used in the industry.

We have tested the vehicle's gen-set for a setpoint power of $P = 15$ kW, with a corresponding equilibrium crankshaft speed of 205 rad/s. Due to resource limitations only a single (15 kW) power level was evaluated. In Fig. 10 we can view ensemble plots of the CE and EM torque for several

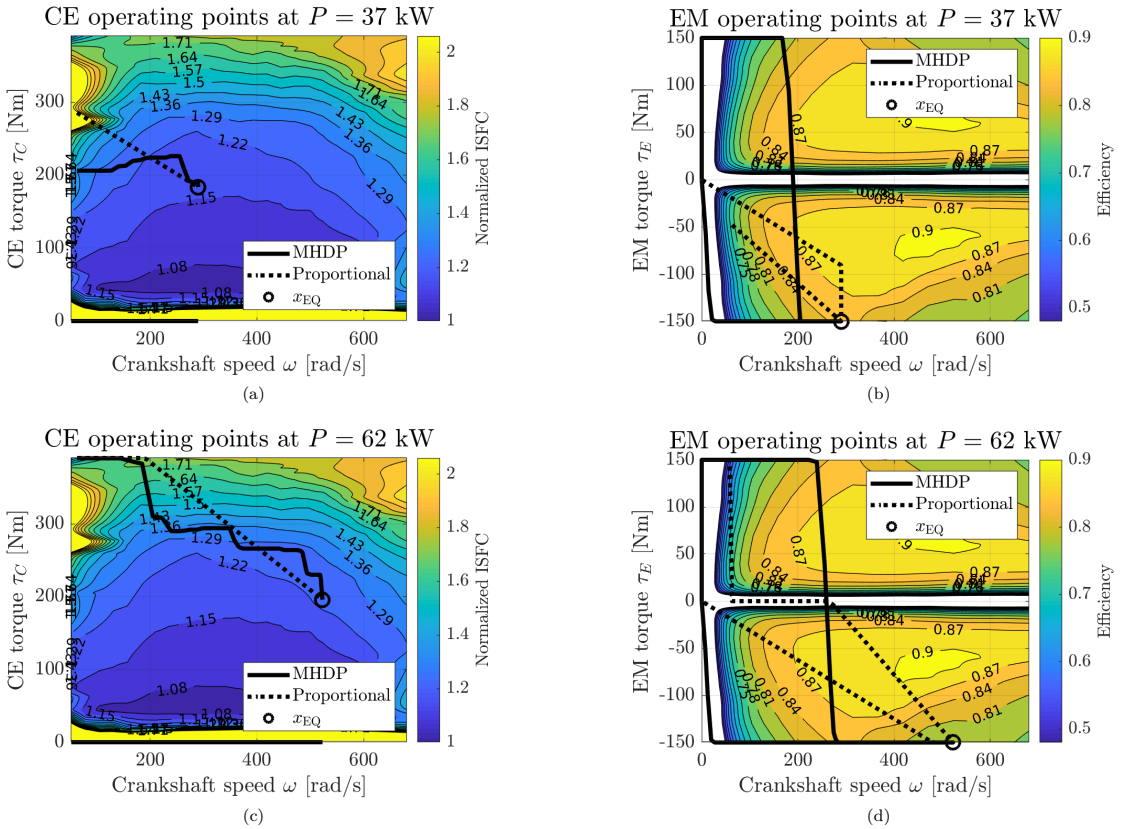


FIGURE 7: Typical CE and EM transient operating point trajectories.

repeated tests, with an associated average efficiency improvement of 3.7%. The difference in equilibrium speed in Fig. 10c is an artifact of the test-bench, but has little effect on the results as the CE efficiency is virtually identical for $\omega = 200$ and $\omega = 210$. Notably, the structure of the proportional controller gives a solution that tends to load the CE more at speeds where it is inefficient ($0.5 \leq t \leq 1$).

There are several potential avenues of further improving the fuel consumption of the optimal controller beyond the results seen in this test setup. One significant limitation is the gen-set's EM inverter power constraint. This limits the permissible operating points to a small portion of the theoretical speed/torque space. For these tests we also have allowed the engine to rest at idle between steps to P (rather than allowing it to reach a standstill) to avoid activating the existing engine start-up routines. Furthermore, there are several unmodeled sources of dynamics in this physical setup, the turbocharger among others. Finally, the functions f_C, f_E, and τ_f only accurately model equilibrium operation, while we have implicitly assumed that they hold during transient operation. We ultimately have a trade-off between model accuracy and the computational load of the controller (whose non-volatile memory usage scales exponentially with the number of state variables).

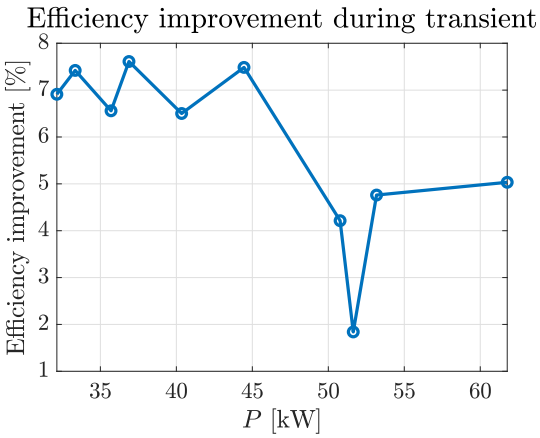


FIGURE 8: Efficiency improvement (delivered energy per consumed fuel mass) for the optimal controller compared to the proportional controller.

FIGURE 9: Test-bench setup. A conventional passenger vehicle is used with the same CE and EM as shown in Fig. 3.

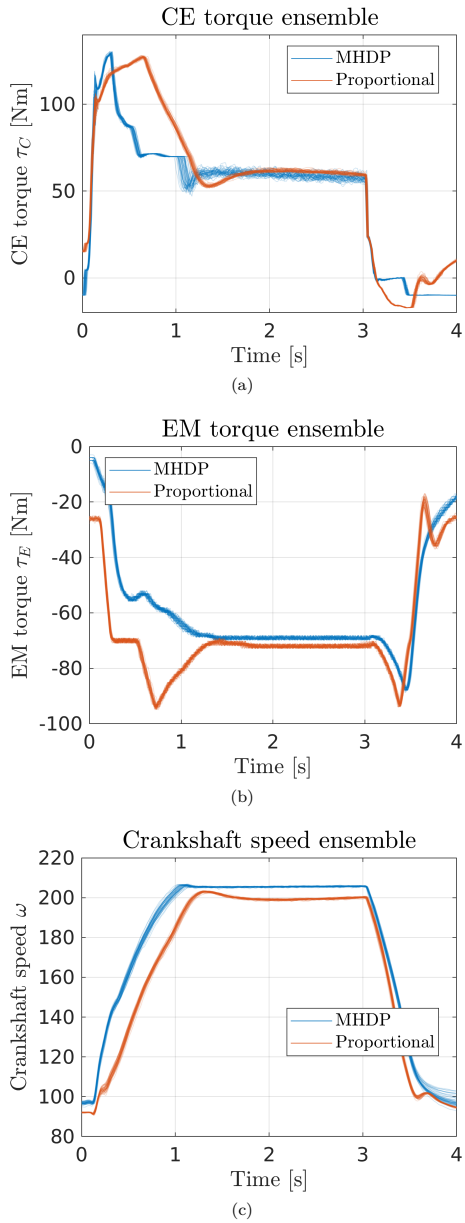


FIGURE 10: Ensemble plots of CE and EM torque and crankshaft speed ω from bench-tests.

5 Conclusions

We have shown that an optimal gen-set controller that takes transient dynamics into account can effectively be implemented and run in real-time. Minimizing the consumed fuel using a dynamic programming method with an infinite horizon ensures that the optimal control policy captures the true intent of the problem studied in this paper; minimize fuel consumption while delivering a requested equilibrium power. This also avoids issues with penalty-based methods, does not require any gradients to be known, and is guaranteed to be close to the globally optimal solution. Using the MHDP method, which is based on dynamic programming and the successive approximation method, allows us to off-line easily construct a state-feedback control law with very low on-line computational demand (on the order of 10–20 CPU instructions and 5–20 KiB of consumed memory).

Simulation results show that a gen-set in a typical light-duty vehicle can expect a fuel consumption improvement on the order of 5–7% during transient operation when compared to current industry-standard controllers. Bench-tests displayed a fuel consumption improvement of 3.7%, indicating that the simulation results are fairly representative. This reduction of fuel consumption may be further increased by using more sophisticated models, in particular a combustion engine fuel consumption model that takes additional transient dynamics, such as a turbocharger, into account.

Relevant future work includes studying the efficacy of the proposed method for the more ultimately relevant case of a whole-vehicle drive cycle. The choice of torque-split controller (i.e. when and how much power is drawn from the gen-set and battery respectively) as well as the selected drive cycle will heavily influence the net effectiveness of the proposed gen-set controller, and should thus be prudently chosen.

Acknowledgements

This research was supported by the combustion engine research center (CERC) at Chalmers. Thanks to Volvo Cars Corporation for setting up experimental test-bench. Special thanks to Robert Buadu for operating the test-bench.

References

[1] Dimitri P. Bertsekas. *Dynamic Programming: Deterministic and Stochastic Models*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1987.

[2] Jonathan Lock and Tomas McKelvey. Constant-setpoint infinite-horizon nonlinear optimal control using dynamic programming. 2019. Available at `https://doi.org/10.5281/zenodo.2613415`.

[3] Tomas Nilsson and Anders Fröberg. Optimal operation of a turbocharged diesel engine during transients. *SAE International Journal of Engines*, 5(2):571–578, April 2012.

[4] Carlos E. Nino-Baron, Abdul R. Tariq, Guoming Zhu, and Elias G. Strangas. Trajectory optimization for the enginegenerator operation of a series hybrid electric vehicle. *IEEE Transactions on Vehicular Technology*, 60(6):2438–2447, July 2011.

[5] Martin Sivertsson and Lars Eriksson. Optimal transient control trajectories in diesel-electric systems – part I: Modeling, problem formulation, and engine properties. *Journal of Engineering for Gas Turbines and Power*, 137(2):021601, September 2014.

[6] Xi Zhang, Zixian Wu, Xiaosong Hu, and Wei Qian. Trajectory optimization-based auxiliary power unit control strategy for an extended range electric vehicle. *IEEE Transactions on Vehicular Technology*, 66(12):10866–10874, December 2017.

PAPER D

A Control-Oriented Spatially Resolved Thermal Model of the Three-Way-Catalyst

Jonathan Lock, Kristoffer Clasén, Jonas Sjöblom, Tomas McKelvey

Published in SAE WCX Digital Summit,
April 2021.
ISSN: 0148-7191

2021-01-0597

A Control-Oriented Spatially Resolved Thermal Model of the Three-Way-Catalyst

Jonathan Lock, Kristoffer Clasén, Jonas Sjöblom, Tomas McKelvey

Abstract

The three-way-catalyst (TWC) is an essential part of the exhaust aftertreatment system in spark-ignited powertrains, converting nearly all toxic emissions to harmless gasses. The TWC's conversion efficiency is significantly temperature-dependent, and cold-starts can be the dominating source of emissions for vehicles with frequent start/stops (e.g. hybrid vehicles). In this paper we develop a thermal TWC model and calibrate it with experimental data. Due to the few number of state variables the model is well suited for fast offline simulation as well as subsequent on-line control, for instance using non-linear state-feedback or explicit MPC. Using the model could allow an on-line controller to more optimally adjust the engine ignition timing, the power in an electric catalyst pre-heater, and/or the power split ratio in a hybrid vehicle when the catalyst is not completely hot. The model uses a physics-based approach and resolves both axial and radial temperature gradients, allowing for the thermal transients seen during heat-up to be represented far more accurately than conventional scalar (i.e. lumped-temperature) real-time models. Furthermore, we also use a physics-based chemical kinetics reaction model for computing the exothermic heat of reaction and emission conversion rate which is temperature and residence-time-dependent. We have performed an experimental campaign with a standard spark-ignited engine and a commercial TWC, where we measured steady-state operation and cold-start transient behavior. This experimental data allowed us to tune the model, where we found excellent matching between the measured and modeled tailpipe emissions. Modeling the radial temperature gradient improved the relative accuracy of the conversion efficiency by 15%, and simulations indicate the potential for an absolute improvement by 15 percentage points for some cases. Furthermore, the modeled TWC temperature evolution for a cold-start was typically within ±10°C of the measured temperature (with a maximal deviation of 20°C). The proposed model thus bridges a gap between heuristic models suited for on-line control and accurate models for slower off-line simulation.

Introduction

The three way catalyst (TWC) is an essential part of the powertrain in virtually all spark-ignited (SI) automotive vehicles, significantly reducing the level of harmful emissions and keeping them below legislated limits. These emissions are generated by combustion processes in the engine, primarily consisting of carbon monoxide (CO), nitrogen oxides ($\mathrm{NO_x}$), and residual unburnt hydrocarbons (THC). In normal operation virtually all of the generated emissions are converted to non-toxic carbon dioxide ($\mathrm{CO_2}$), nitrogen gas ($\mathrm{N_2}$) and water ($\mathrm{H_2O}$) [1, 2]. In order to reach a high conversion ratio the TWC must be sufficiently warmed up by the engine exhaust, typically to at least 250–350°C. However, when a cold vehicle is started there is a short period, on the order of 10–30 seconds, where the TWC is not sufficiently warm to convert the exhaust emissions. This gives rise to a high level of tailpipe emissions, and for many regulatory test procedures these cold-start emissions are responsible for 60–80% of the emissions generated from an entire test (which are for comparison on the order of 30 minutes) [2].

Methods for reducing cold-start emissions have been studied extensively from several different perspectives, including TWC design methods that reduce the cold-start time [3], additional hardware that can pre-heat the TWC before starting the engine [4], and control schemes that control the engine's operation to reduce the generated emissions and/or heat the TWC more quickly [2, 5, 6, 7, 8]. One shared requirement for making a good design choice for all these tasks is a sufficiently accurate cold-start thermal model of the TWC. Ideally, a model should be able to predict both the spatially varying thermal dynamics and the conversion efficiency of the TWC to a sufficiently accurate degree. Naturally, more complex models allow for a higher degree of accuracy (for instance, a full 3D model), while simpler models are computationally faster.

In this paper we introduce and experimentally study a physics-based thermal TWC model that is both computationally fast enough to be used for on-line control methods, while simultaneously using a kinetic reaction model for emission species conversion and resolving both axial and radial temperature gradients. This is a significant improvement over many other numerically fast methods, which use simpler heuristics for emission conversion and/or assume a constant temperature profile in the TWC. Furthermore, the model allows for adjusting the number of axial and radial segments in order to tune the computational complexity to the available processing power. Ultimately, the model allows for more accurately simulating the behavior of the TWC, both for off-line applications that must be numerically fast as well as on-line control methods with limited computational capacity. For instance, an on-line controller could use the presented TWC model in combination with a model-based control method to balance emissions and fuel consumption by controlling the engine's ignition timing, an auxiliary catalyst heater, gear selection, and/or power split in the case of a hybrid vehicle. Similarly, off-line tuning of heuristic cold-start strategies is typically time-consuming when done by experimental test-bench studies or numerically slow simulations. The time needed to tune a heuristic controller could thus be reduced by instead simulating the TWC behavior with the presented model.

In this paper we will first give a brief overview of existing models and highlight their strengths, weaknesses, and possible applications. Following this we introduce and define the model, after which we

describe the experimental setup used to tune it. Finally, we compare the experimental data and the predictions from the model.

Literature survey

Many authors have considered the problem of modeling a TWC in a manner that is amenable to on-line control or very fast simulation. These models can generally be categorized by how they trade off accuracy and computational demand as well as which sources of dynamics the models choose to include.

The computationally fastest models typically assume one (scalar) TWC temperature that is influenced by the exhaust emissions and use a suitable function for determining the temperature-dependent conversion efficiency. One example of this is [9], which assumes a scalar TWC temperature whose state evolution is linearly influenced by the engine exhaust gas temperature, ambient temperature, and converted emission species. They further use the $\arctan$ function to approximate the temperature-dependence of the reactions occurring in the TWC. Finally, they use a dynamic programming (DP) method to solve an optimal control problem, which could be implemented as an on-line controller. See also [5, 6, 10] for similar (scalar TWC temperature) approaches.

Some authors chose to improve the accuracy of the TWC model by also modeling a TWC phenomenon wherein oxygen is absorbed and released from the TWC. The stored oxygen content influences the TWC's ability to remove emissions, where CO and THC are more efficiently oxidized when there is an overabundance of stored oxygen in the TWC, while $\mathrm{NO_x}$ is more efficiently converted when the oxygen content is low. For instance, [11] uses two state variables, one to store the TWC temperature similarly to the previous authors, and one to store the total stored oxygen in the TWC. This addition results in a model that more accurately captures the behavior of the TWC during phases where the air-fuel-ratio, often referred to as *lambda*, deviates from a stoichiometric condition. This occurs most prominently during fuel-cut operation, where the engine motors without any fuel injection, but also occurs during a short interval when the engine speed and load is changed [12, p. 69], as well as to a lesser extent during normal operation where lambda is periodically switched between slightly-rich and slightly-lean in order to improve the net TWC conversion efficiency [13]. A significant excursion from stoichiometric operation (i.e. beyond the ordinary rich/lean switching scheme) is typically handled by running the TWC at slightly rich or lean in order to bring the average lambda back to stoichiometry, and is crucial in order to maintain a high conversion efficiency.

One common approach of further improving model accuracy is by using a heterogeneous TWC model. Most commonly this is done by using a model that can represent varying TWC temperatures and/or stored oxygen. As the TWC is typically constructed with hundreds of identical parallel channels [1, p. 650], one might choose to assume that the gas flow, composition, and temperature is identical for all channels. By using this *single-channel approximation* our modeling focus can be directed to studying the behavior in one channel in isolation, after which we can compute the total TWC behavior via scaling by the number of channels. As each single channel is long and narrow one obvious method is to divide each channel into separate axial slices and then allow each slice to take on its own temperature and/or stored oxygen (e.g. [14, 15, 16], where temperature is axially resolved). Ultimately this allows for capturing transient behavior in the TWC, such as a cold-start, where the front of the TWC is typically several hundreds of degrees warmer than the rear.

The most practically significant drawback with more complex models is their increased computational demand. In particular, on-line control systems intended for native vehicle implementation are typically highly computationally limited, and any controller must be of low numerical complexity. There are several different approaches that can be used, ranging from using optimal control strategies as an inspiration for heuristic controllers [14], to more computationally expensive methods, for instance based on Pontryagin's Maximum Principle [15], while others primarily target simulation using hardware much more powerful than that found in vehicles [16].

Several powerful on-line control methods, e.g. nonlinear state feedback and some explicit MPC variants, can be implemented for nearly any practically relevant model complexity (nonlinear, non-convex, and non-differentiable) while being very computationally inexpensive to run on-line. These methods are computationally inexpensive as the model is never evaluated in the real-time controller. Instead, the control signal is generated by referring to a pre-computed look-up-table [17, 18], i.e. the optimal control can be computed by an interpolation operation (taking on the order of 5–10 CPU instructions) in the case of nonlinear state feedback, and a region identification and function evaluation step in the case of explicit MPC. However, these methods are limited by a computational complexity that scales very poorly (exponentially) with the number of state variables. This applies both to the off-line phase with respect to the computational demand of generating the model-derived look-up-table, as well as during the on-line phase with respect to memory requirements (as well as computational demand in some explicit MPC cases). This motivates a model that has a limited number of state variables, but may otherwise (within reasonable limits) be complex and nonlinear. Note that there exist on-line control methods that do not place as large demands on the number of state variables (e.g. linear programming methods, which scale polynomially with the number of state variables). However, these methods instead limit the complexity of the model dynamics, and can require them to be e.g. linear, convex in cost, and so on. In this paper we take the approach of limiting the number of state variables while allowing the dynamics to be nonlinear, which in turn makes for instance DP-based methods an attractive choice for subsequent simulation or controller generation.

Our goal with this paper is to develop an efficient TWC model suited for cold-starts, whose temperature evolution is driven by the time-evolution of the combustion engine's exhaust gas composition, temperature, and mass-flow. Our model primarily captures axial temperature variations as well as to some extent radial temperature variations, which we can view as using the axially-sliced single-channel approximation while also allowing the temperature of each channel to vary based on its radial position in the TWC. Given this goal and the limit on the number of state variables, it is natural to use a nested controller structure. For instance, an inner control loop to regulate the engine's lambda with a goal to maintain the correct stored oxygen, in concert with an outer control loop to regulate the engine's operating point (e.g. ignition timing, speed, and torque) with a goal to regulate the TWC temperature. Our model is therefore primarily designed for use as a supervisory controller, and assumes a lambda controller is already implemented. Note that since the behavior of the physical TWC depends on the lambda-controller, then a tuned thermal TWC model is also only valid for a given lambda-controller (i.e. it must be re-tuned if the lambda-controller is modified).

The presented model is to our knowledge novel in the sense that it uses a first-principles approach for both thermal conduction and chemical kinetics in a control-oriented context, meaning that we can avoid the use of heuristics to describe the thermal and chemical behavior. Furthermore, the model is configurable in the axial and radial resolution, allowing for balancing the model's accuracy and computational demand to the target application.

TWC model

In this section we will present the TWC model, which resolves both axial and radial temperature dynamics, uses a first-principles model for thermal conduction and exhaust gas species conversion reactions, and is easy to tune to experimental data.

In principle, the model is a nonlinear ordinary differential equation,

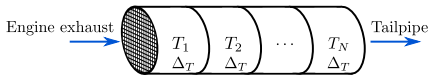


Figure 1: Axially and radially discretized TWC with state variable $\bar{T} = [T_1, T_2, \ldots, T_N, \Delta_T]$.

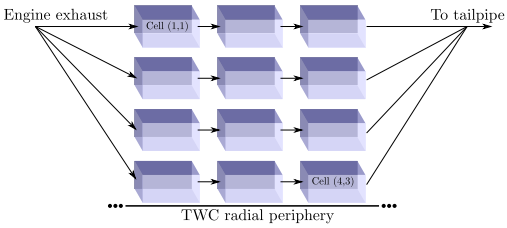


Figure 2: Fully thermally resolved TWC, here shown for $M = 4$, $N = 3$. Arrows show the exhaust gas flow through the different cells in the model.

where the temperature derivative depends on the current temperature and incoming gas properties. For ease of understanding we will introduce the model in a constructive manner, similar to an algorithmic implementation, divided into the following parts:

1. For given state variable values (which inherently only encode the TWC temperature at a few positions) a densely-sampled representation of the TWC's temperature distribution is generated where the radial resolution is significantly increased.
2. Once the dense temperature distribution is known, the properties of the gas entering the TWC are used to determine the outgoing gas properties and thermal flux in the TWC.
3. For a given thermal flux, the densely-sampled representation is converted to the time-derivative of the low-dimensional state variables.

Each of the above phases are defined in the following sections.

Generating the full temperature distribution

Figure 1 illustrates a typical cylindrical TWC monolith, which we have axially divided into N equally long slices. Each slice, n, has an associated state variable T_n that corresponds to the radially central temperature. A final state variable, Δ_T, represents the difference in temperature between the radial center of each slice and its associated periphery, giving the total state vector

$$\bar{T} = [T_1, T_2, \ldots, T_N, \Delta_T]. \tag{1}$$

Note that Δ_T is *not* axially resolved, i.e. the difference in temperature between the radial center and periphery of each slice is assumed constant throughout the entire TWC. This assumption is made in order to reduce the number of state variables.

In the first model step, we construct the full representation of the TWC temperature by increasing the radial resolution. An illustration of the full TWC model is shown in Figure 2, which consists of the same equally long N axial slices represented in the state vector, but is also extended radially with M evenly spaced elements. This gives a total of $N \cdot M$ *cells*, where each cell can be viewed as a small reactor with homogeneous temperature. Cells $(1, n)$ correspond to the radially centermost parts of the TWC, while cells (M, n) correspond to the radially outermost parts of the TWC. We naturally assume that

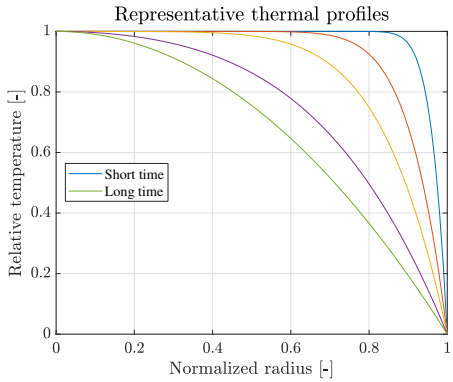


Figure 3: Representative solutions to the transient heat equation for varying times t. The shown solution's temperatures have been normalized to the range $[0, 1]$. With this normalization, the solution at radii $[0, 1)$ starts off at unit relative temperature, and as time progresses gradually approaches the closer-to linear distribution.

gas flows from one given axial position to its successor, without any radial transport between cells (as indicated by the arrows in Figure 2), i.e. we assume there is no gas diffusion between neighboring channels in the TWC. Cells $([1 \ldots M], 1)$ are fed with engine exhaust, cell $(m, n+1)$ is fed with the output of cell (m, n), and the output from cells $([1 \ldots M], N)$ are combined to form the total tailpipe exhaust. Note that as the TWC is circular, cells nearer the periphery have a larger associated open frontal area. We compensate for this by assuming a constant gas flow-rate, and weight the incoming massflow relative to each cell's relative area. Letting $\dot{m}_{\text{eng ex}}$ be the total massflow from the engine, the flow into each cell is then

$$\dot{m}_{m,1} = \frac{m^2 - (m-1)^2}{M^2} \cdot \dot{m}_{\text{eng ex}} \tag{2}$$

$$\dot{m}_{m,n+1} = \dot{m}_{m,n}. \tag{3}$$

In essence, we can view the full TWC model as consisting of M parallel single-channel models, each of which consists of N sections with different temperatures, and whose incoming massflow is proportional to its associated frontal area.

Each cell is assigned a temperature as follows:

- cells $(1, 1)$ through $(1, N)$ are assigned temperatures T_1 through T_N respectively (i.e. the radially centermost temperature),
- cells $(M, 1)$ through (M, N) are assigned temperatures $T_1 + \Delta T$ through $T_N + \Delta T$ respectively, (i.e. the radially outermost temperature), and
- the remaining cells, $(2, 1)$ through $(M-1, N)$, are assigned temperatures following a physics-based interpolation scheme described below.

Radial temperature profile

The radial temperature distribution (i.e. the temperature profile between the radial center and periphery) in the physical TWC varies depending on the properties of the incoming gas, and by extension the operating point of the combustion engine.

We have chosen to model the true radial temperature profile as a solution to the heat equation. More specifically, we solve the transient

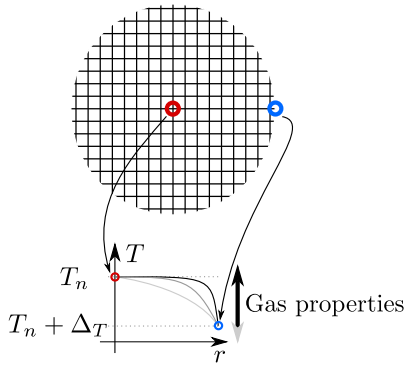


Figure 4: Each axial slice is assigned a radial temperature distribution that depends on the properties of the incoming exhaust gas. The radially centermost temperature (T_n) and radially outermost temperature ($T_n + \Delta_T$) in combination with the operating point is used to construct the radially-varying temperature distribution.

heat equation in a flat circular plate with radius R and homogeneous initial temperature $\hat{T}(r) = 0,\ r = [0, R]$ (note that despite the similar notation, $\hat{T}$ has no direct relation to the model state vector $\bar{T}$ or the axial slices T_n). Furthermore, we apply the boundary condition $\hat{T}(R) = 0$ at the periphery (i.e. a Dirichlet boundary condition), and assume the bulk of the plate develops a constant and homogeneous power (analogous to the energy delivered by the incoming gas to a given axial TWC slice). This is a textbook problem (e.g. [19, p. 148]) with a well-known solution that can be expressed as a Fourier-Bessel series. We can easily numerically solve this problem over time and along the plate's radius (for instance using MATLAB's `pdepe` function), generating the radially-varying time-evolution of the plate's temperature. We show normalized solutions, i.e. where $\hat{T}$ is offset and scaled so that $\hat{T}(0) = 1$ and $\hat{T}(R) = 0$, for some representative time instances in Figure 3 for $R = 1$.

In this paper we select the modeled TWC temperature profile at any given time (i.e. which profile we select from Figure 3) based on the engine's current operating point, as indicated in Figure 4, and we will return to the specific method for choosing a profile in the experimental results section. Ultimately, this gives an easily implemented interpolation method for cells $(2, 1)$ through $(M - 1, N)$ that matches the temperatures at the radial center and periphery, represented by $T_1, \ldots, T_N$ and $T_1 + \Delta_T, \ldots, T_N + \Delta_T$ respectively. The scheme is physically motivated, capturing the essence of the radial thermal conduction and incoming power from gas convection and exothermic heat of reaction.

Chemical kinetic model

Though the total set of reactions occurring in the TWC are highly complex and involve many different compounds, there are fewer components that contribute to the legislated emissions and significant heat generation. We therefore are only interested in the following net reactions (adapted from [11, 20])

$$2\,CO + O_2 \longrightarrow 2\,CO_2 \tag{4}$$
$$2\,NO + 2\,CO \longrightarrow N_2 + 2\,CO_2 \tag{5}$$
$$2\,NO_2 \longrightarrow N_2 + 2\,O_2 \tag{6}$$
$$2\,C_3H_6 + 9\,O_2 \longrightarrow 6\,CO_2 + 6\,H_2O \tag{7}$$
$$C_3H_8 + 5\,O_2 \longrightarrow 3\,CO_2 + 4\,H_2O. \tag{8}$$

Typically, engine-out nitrogen oxides and hydrocarbon emissions are lumped together [1, pp. 572, 597], and we thus categorize the the emission species as carbon monoxide (CO), nitrogen oxides (NO and NO_2), and hydrocarbons (C_3H_6 and C_3H_8). By [1, p. 578] we assume a constant molar ratio of 99:1 for NO to NO_2 (a similar ratio was found in our experimental tests). Similarly, by [21] and [22] we choose to assume a a 3:1 ratio for C_3H_6 to C_3H_8. For convenience, we will refer to these lumped emissions as NO_x and THC respectively.

We use an Arrhenius expression to model the reaction rate $k^s_{n,m}$ of a given emission species s (i.e. CO, NO_x, and THC) in any given cell n, m as

$$k^s_{n,m} = A^s e^{\frac{-E^s_a}{RT_{n,m}}} \tag{9}$$

where R is the ideal gas constant, $T_{n,m}$ is the temperature of cell n, m, E^s_a is the activation energy of emission species s, and A^s is the apparent pre-exponential factor for species s (which is dependent on the cell's volume). Using the notation $y^s_{n,m}$ to indicate the mole fraction of emission species s, we model the evolution of the mole fraction as

$$\frac{\mathrm{d}y^s_{n,m}}{\mathrm{d}t} = -k^s_{n,m} y^s_{n,m}. \tag{10}$$

For a typical TWC, the gas residence time in each slice is short enough for the temperature to be close-to constant, which we will now motivate. In [6, p. 64], the authors find that the gas residence time in the *entire* TWC is typically in the range of 0.05–0.1 s, e.g. with 4 axial slices the residence time in each slice is typically 12.5–25 ms. Furthermore, [6, p. 66] also finds that the typical temperature time-derivative in the TWC during a cold-start is on the order of 10°C/s (this is similar to our findings, which we will see later). Ultimately this implies that the temperature in a given axial slice changes by approximately 0.1–0.25°C during the time the gas is in each slice, which we view as insignificant.

Using the constant-temperature approximation, (10) has the explicit solution

$$y^s_{n,m}(t_r) = y^s_{n,m}(0) \cdot e^{-k^s_{n,m} t_r} \tag{11}$$

for a residence time t_r. We will assume a simple plug flow reactor model (i.e. no axial mixing), which gives

$$t_r = \frac{V_{\text{slice}}}{\nu} \tag{12}$$

where V_{slice} is the volume of each axial slice and ν is the volumetric gas flow rate. We generate the slice volume as

$$V_{\text{slice}} = \frac{V_{\text{TWC}}}{N}, \tag{13}$$

where V_{TWC} is the gas volume of the entire TWC. We further estimate ν as

$$\nu = \frac{\dot{m}_{\text{eng ex}}}{P/(R_{\text{specific}} T_{n,m})} \tag{14}$$

where P is the absolute pressure in the TWC (typically close to ambient pressure) and R_{specific} is the specific gas constant for the post-TWC ratio of N_2, CO_2, O_2, and H_2O we measured in our experimental trials during stoichiometric operation. We have chosen to use this specific gas constant as it is easily determined and the remaining gasses contribute minimally to R_{specific}.

Ultimately, using (9) through (14) gives a simple, but physics-based, model for the reactions occurring in the TWC that takes temperature, gas composition, and residence time into account. This implies that the model is suited for quasi-stationary combustion engine operation, where the engine-out exhaust temperature, emission species mole fraction, and massflow varies slowly with respect to the residence time. Though the model will not accurately capture the true behavior during engine transients, as gas at different positions in the TWC will

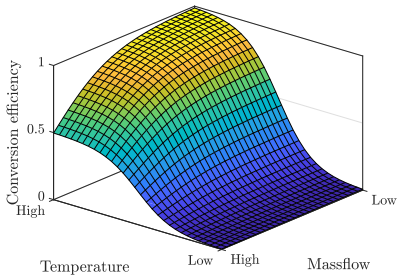


Figure 5: Representative conversion efficiency for a given emission species and cell, with maximal efficiency at low massflow (i.e. longer residence time) and high temperature.

have originated from the engine operated at different regimes, we hypothesize (but were not able to verify with the available measurement equipment) that the predicted TWC performance converges to the measured performance on a time scale that is comparable with the residence time in the TWC.

By (11), the massflow emitted from cell n, m is

$$\dot{m}^{s,\text{out}}_{n,m} = \dot{m}^{s,\text{in}}_{n,m} e^{-k^s_{n,m} t_r} \tag{15}$$

and, by the conservation of mass, the converted massflow is

$$\dot{m}^{s,\text{conv}}_{n,m} = \dot{m}^{s,\text{in}}_{n,m} - \dot{m}^{s,\text{out}}_{n,m}. \tag{16}$$

Furthermore, we define the conversion efficiency of each cell as

$$\eta^s_{n,m} = 1 - \frac{\dot{m}^{s,\text{out}}_{n,m}}{\dot{m}^{s,\text{in}}_{n,m}}. \tag{17}$$

A qualitative illustration of the conversion efficiency as a function of massflow and temperature is shown in Figure 5.

Finally, we model the tailpipe emission of species s as the sum of outputs from each element in the last axial segment, i.e.

$$\dot{m}^s_{\text{tp}} = \sum_{m=1}^{M} \dot{m}^s_{m,N}. \tag{18}$$

In order to determine the exothermic reaction power generated by the above reactions, we can compute the (temperature-dependent) heat of reaction for each mole of reactant species as

$$\mathrm{d}H_{\text{CO}} = H_{0,\text{CO}_2} - H_{0,\text{CO}} - 1/2 H_{0,\text{O}_2} \tag{19}$$

$$\mathrm{d}H_{\text{NO}} = 1/2 H_{0,\text{N}_2} + H_{0,\text{CO}_2} - H_{0,\text{NO}} - \mathrm{d}H_{\text{CO}} \tag{20}$$

$$\mathrm{d}H_{\text{NO}_2} = 1/2 H_{0,\text{N}_2} + H_{0,\text{O}_2} - H_{0,\text{NO}_2} \tag{21}$$

$$\mathrm{d}H_{\text{C}_3\text{H}_6} = 3H_{0,\text{CO}_2} + 3H_{0,\text{H}_2\text{O}} - H_{0,\text{C}_3\text{H}_6} - 9/2 H_{0,\text{O}_2} \tag{22}$$

$$\mathrm{d}H_{\text{C}_3\text{H}_8} = 3H_{0,\text{CO}_2} + 4H_{0,\text{H}_2\text{O}} - H_{0,\text{C}_3\text{H}_8} - 5H_{0,\text{O}_2}. \tag{23}$$

Note that we have neglected to indicate the temperature dependence of these above terms for brevity, but include the temperature dependence in the model by using the Shomate equation and reference constants given by the NIST available at `https://webbook.nist.gov/`. The heat of reaction for the lumped emission terms is then similarly given by the weighted average

$$\mathrm{d}H_{\text{NO}_\text{x}} = (99\mathrm{d}H_{\text{NO}} + \mathrm{d}H_{\text{NO}_2})/100 \tag{24}$$

$$\mathrm{d}H_{\text{THC}} = (3\mathrm{d}H_{\text{C}_3\text{H}_6} + \mathrm{d}H_{\text{C}_3\text{H}_8})/4. \tag{25}$$

Using (19), (24), and (25) the total temperature-dependent heat of reaction generated in each cell is given as

$$P_{n,m} = \dot{m}^{\text{CO,conv}}_{n,m} \cdot \mathrm{d}H_{\text{CO}} + \dot{m}^{\text{NO}_\text{x},\text{conv}}_{n,m} \cdot \mathrm{d}H_{\text{NO}_\text{x}} + \dot{m}^{\text{THC,conv}}_{n,m} \cdot \mathrm{d}H_{\text{THC}}. \tag{26}$$

Generating the state vector derivative

In this stage, we use the previously computed converted massflow and associated exothermic reaction power per cell to generate the state vector derivative. Note that the state derivative is of size $N+1$, i.e. the information encoded in the $N \cdot M$ cells is reduced to $N+1$ dimensions. The state derivative is constructed from three different terms:

$$\frac{\mathrm{d}\bar{T}}{\mathrm{d}t} = \bar{T}_{\text{cond}} + \bar{T}_{\text{exo}} + \bar{T}_{\text{convect}}. \tag{27}$$

Here, $\bar{T}_{\text{cond}}$, $\bar{T}_{\text{exo}}$, and $\bar{T}_{\text{convect}}$ correspond to bulk thermal conduction, exothermic reaction power, and the convective power from the incoming gasses. Theses terms are in turn defined as follows.

Thermal conduction

We model thermal conduction in the TWC, $\bar{T}_{\text{cond}}$, both axially and radially as

$$\bar{T}_{\text{cond}} = \frac{1}{\tau_{\text{ax}}} \begin{bmatrix} T_2 - T_1 \\ \vdots \\ T_{n-1} + T_{n+1} - 2T_n \\ \vdots \\ T_{N-1} - T_N \\ 0 \end{bmatrix} \tag{28}$$

$$+ \frac{1}{\tau_{\text{ra}}} \begin{bmatrix} \Delta_T \\ \vdots \\ \Delta_T \\ -\Delta_T \end{bmatrix} \tag{29}$$

$$+ \frac{1}{\tau_{\text{amb}}} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ T_{\text{amb}} - \frac{1}{N}\sum_{n=1}^{N}(T_n + \Delta_T) \end{bmatrix}, \tag{30}$$

where τ_{ax}, τ_{ra}, and τ_{amb} are scalar tuning parameters that capture the net axial, radial, and ambient thermal resistance respectively and T_{amb} is a given ambient temperature. Note that $\bar{T}_{\text{cond}}$ is given as the sum of three terms, corresponding to axial conduction (28), radial conduction (29), and heat transfer to the ambient environment (30) respectively. Each of these three terms are motivated by the 1-dimensional heat equation [19]. In particular, note that in (30) $\frac{\mathrm{d}\Delta_T}{\mathrm{d}t}$ is given by the difference in temperature between the ambient and *mean* peripheral temperature. This is ultimately due to our choice of assuming a scalar core/periphery temperature difference, i.e. one that is identical for all axial slices.

Exothermic terms

We introduce

$$P_{\text{ctr},n} = \sum_{m=1}^{M} P_{n,m}\left(1 - \frac{m-1}{M-1}\right) \tag{31}$$

as the weighted exothermic reaction power associated with the radial center of the TWC for axial position n. Note that the term $\frac{m-1}{M-1}$ varies from 0 to 1 for m from 1 to M, i.e. a linear weighting. We have chosen this specific weighting as it matches our intuition in that exothermic power near the center of the fully resolved TWC should be assigned to the state variables corresponding to the axial center, and vice-versa for the periphery. However, this is only a first approximation and an optimal weighting might be different.

Similarly, we introduce

$$P_{\text{per},n} = \sum_{m=1}^{M} P_{n,m}\frac{m-1}{M-1} \tag{32}$$

as the weighted exothermic reaction power associated with the periphery of the TWC for axial position n. In essence, $P_{\text{ctr},n}$ and $P_{\text{per},n}$ collect the exothermic reaction power developed in each cell, weighting power near the center to $P_{\text{ctr},n}$ and power near the periphery to $P_{\text{per},n}$.

We can now define the state derivative terms due to the heat of reaction as

$$\bar{T}_{\text{exo}} = k_{\text{conv}} \begin{bmatrix} P_{\text{ctr},1} \\ \vdots \\ P_{\text{ctr},N} \\ \frac{1}{N}\sum_{n=1}^{N}(P_{\text{per},n} - P_{\text{ctr},n}) \end{bmatrix}, \tag{33}$$

where k_{conv} is a tuning parameter for the proportionality between heat of reaction and the temperature derivative, i.e capturing the heat capacity of the TWC. Note that if P_{ctr} and P_{per} are expressed in Watts and $\bar{T}_{\text{exo}}$ as K/s then k_{conv} will have units J/K. Finally, note that the last element of $\bar{T}_{\text{exo}}$ can be seen as the average difference in power between the radially central and peripheral exothermic reaction powers, which is thus proportional to $\frac{\mathrm{d}\Delta_T}{\mathrm{d}t}$.

Convective terms

We model the temperature derivative from convective terms driven by the incoming gas as

$$\bar{T}_{\text{convect}} = \frac{\dot{m}_{\text{eng ex}}}{\tau_{\text{convect}}} \begin{bmatrix} T_{\text{eng ex}} - T_1 \\ \vdots \\ T_{n-1} - T_n \\ \vdots \\ T_{N-1} - T_N \\ 0 \end{bmatrix} \tag{34}$$

where $T_{\text{eng ex}}$ is the temperature of the gas entering the TWC and τ_{convect} is a tuning parameter proportionality constant, capturing the the specific heat of the exhaust gas and the heat capacity of the TWC. Note that (34) implies that the gas leaving each given cell is assumed to have the same temperature as the cell itself, i.e. each cell can be viewed as a reactor that is long enough for the moving gas to attain the same temperature as the cell before leaving (motivated by [6, p. 62] for $N \leq [2, 5]$). Also, recall that the final element of $\bar{T}_{\text{convect}}$ stores the difference in temperature between the radially central and peripheral part, i.e. setting the final element of (34) to 0 implies radially uniform gas flow through the TWC.

Experimental setup

Engine

The engine setup consisted of a production Volvo Cars two liter in-line four-cylinder direct injected spark ignited turbocharged engine rated at 187 kW and 350 Nm, as listed in Table 1. The engine exhaust was connected to an exhaust aftertreatment system consisting of a TWC equipped with wide-band lambda sensors, 14 thermocouples, and five exhaust gas sampling locations, both measuring average and point source gas compositions. The engine was connected to an electrical dynamometer that regulated the engine speed and measured the delivered torque. A prototyping ECU was used to allow sampling and changing engine parameters.

An auxiliary air valve was added to the exhaust manifold, allowing for flushing the entire exhaust aftertreatment system with room-temperature air for the cold-start tests. Cutting all fuel to the engine (while pumping air through the engine) and flushing with auxiliary air allowed for rapidly cooling the exhaust aftertreatment system, taking approximately 5 minutes to cool the entire TWC to under 100°C. The auxiliary airflow was kept small enough to avoid the turbo spooling up, which might cause an undesirable transient when resuming normal operation. During all normal engine operation the auxiliary air valve was kept closed in order to not alter the composition and temperature of the engine exhaust.

Table 1: Engine properties.

Engine type	VEA Gen I, VEP4 MP
Number of cylinders	Four, in-line
Displaced volume	1969 cc
Bore/Stroke	82 mm/93.2 mm
Compression ratio	10.8:1
Valve train	DOHC, 16 valves
Intake camshaft	Variable 0-48° CA advance
Exhaust camshaft	Variable 0-30° CA retard
Ignition system	DCI, standard J-gap spark plugs
Fuel system/Injection pressure	DI/200 bar
Fuel	Gasoline RON95 E10
Start of injection	308-340 CAbTDCf
Boosting system	Turbocharger
Rated power/Rated torque	187 kW/350 Nm
Stoichiometric air/fuel ratio	14.01:1

Figure 6: Experimental setup, TWC housing highlighted. Above the TWC is a 90°elbow. The turbocharger exhaust is just visible to the right of the elbow. Below the TWC is a GPF (not used in this experiment) and a flexible bellows.

A photograph of the aftertreatment system is shown in Figure 6, with a schematic representation is shown in Figure 7 and a detailed view of the TWC in Figure 8. The schematic shows the engine exhaust and auxiliary cool-down air feed entering the turbo, which then proceeds through a 90°elbow, enters the TWC, and then is finally exhausted. Gas composition measurement points are located before and after the TWC which sample the average gas composition. An additional three sample points are located at the left, center, and right of the TWC, which allow sampling the local gas composition leaving a few TWC channels.

Data Acquisition

Data was sampled using two data acquisition systems. Emissions signals from instruments, fuel, and the dynamometer were sampled

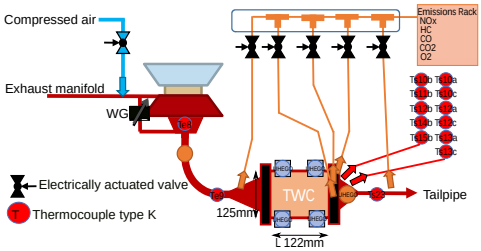


Figure 7: Schematic diagram of experimental setup.

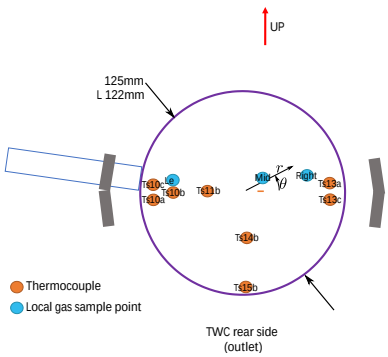


Figure 8: TWC instrumentation as seen from rear (outlet) side. Thermocouple locations indicated with orange, local gas sample points indicated with blue. See Table 2 for detailed thermocouple positioning.

Table 2: Thermocouple positioning. Depth indicates position of thermocouple relative to rear (outlet) of monolith. TWC length, diameter, and radius/azimuth reference as shown in Figure 8.

Thermocouple	Radius r [mm]	Azimuth θ [°]	Depth [mm]
Ts10a	61	180	-100
Ts10b	61	180	-60
Ts10c	61	180	-20
Ts11b	30	180	-60
Ts12a	0	0	-100
Ts12b	0	0	-60
Ts12c	0	0	-20
Ts13a	61	0	-100
Ts13c	61	0	-20
Ts14b	30	270	-60
Ts15b	61	270	-60

Table 3: Steady-state operating points. The load point at 3000 rpm, 8 bar BMEP, 1.05λ was not measured due to time constraints.

Speed [RPM]	BMEP [bar]	Lambda [-]
1000	2	0.95/1.00/1.05
1500	5	0.95/1.00/1.05
2000	15	0.95/1.00/1.05
3000	8	0.95/1.00

with a National Instruments DAQ and an associated LabVIEW program. Engine parameters such as temperatures, pressures, and target lambda were sampled using acquisition units connected via CAN to an ETAS-module. All temperatures were measured using type K thermocouples. Fuel flow mass was measured with a Coriolis meter. All parameters were sampled at a rate of 10 Hz.

Exhaust samples were extracted from five different locations (as indicated in Figure 7). Regardless of sample location, all were extracted via a heated hose (180°C), followed by a heated conditioning unit (190°C) with a heated filter and pump. Emissions concentrations were then analyzed by separate instruments. Total hydrocarbon concentration (THC) was measured using a flame ionization detector, NO_x concentrations using a chemiluminescence analyzer, and CO using a non-dispersive infrared detector. The propagation delay and axial dispersion in hoses and instruments was identified by generating a step in emissions (by disabling the fuel-cut signal as will be described in the cold-start experimental procedure). From this data we could compensate for the propagation delay as well as apply a first-order high-pass filter to mitigate some of the axial dispersion in the measured pre-catalyst emission data. This compensation was then applied to all other emission sampling locations, allowing for studying transient gas composition changes fairly well using an instrument rack primarily intended for steady-state analysis. Ultimately however, our experimental set-up only allowed for measuring emissions at any one given location at a time. We therefore chose to keep the engine in stationary operation in order to maximize the repeatability of the engine-out emissions, which was critical for generating an accurate estimate of the TWC's conversion efficiency.

Measurement procedure

The emission measurement equipment was calibrated before measurements using calibration gasses, and the engine was heated to its working temperature by running at a moderate load until the coolant temperature remained constant. Afterwards, we performed two different tests.

Steady-state analysis

The goal of this test was to identify the steady-state engine-out emissions and associated steady-state radial temperature distribution in the TWC for the engine operating points that can plausibly occur during a low- to medium-load cold-start. This was performed by statically running the engine and storing the steady-state values of all measured signals. The target lambda value was set swept between slightly rich, stoichiometric, and slightly lean in order to characterize the steady-state engine-out emissions[1]. This was then repeated for the operating points listed in Table 3.

[1]The conventional cold-start lambda-switching routine was not altered during this test, i.e. the instantaneous lambda value was automatically switched to slightly above and slightly below the target lambda. We only discuss the target (average) lambda here.

Cold-start thermal evolution

This goal of this test was to characterize the thermal evolution of the catalyst during a cold start. We performed this experiment for each emission sampling point by:

- bringing the engine to a moderate speed and torque (1500 RPM and 5 bar BMEP, giving a suitably long warm-up phase),
- disabling fuel injection (i.e. pumping air only) and opening the auxiliary air valve until the thermocouples in the TWC reported a temperature of under 100°C,
- first closing the auxiliary air valve and then enabling ordinary fuel injection until the TWC reached near-equilibrium temperature.

To capture the ordinary behavior of the engine during a cold-start, we ensured that the lambda control was run in open-loop, i.e. the narrowband lambda sensor after the TWC was not used to fine-tune the engine's air-fuel ratio. Instead, the ECU's conventional open-loop scheme was kept active during all tests, which periodically switches between lean and rich phases with $\lambda \approx \{0.98, 1.02\}$. This switching period is reduced as engine speed and torque increases, leading to poor lambda control at moderate loads and above. Due to this we kept the engine load fairly low during tests, at 1500 RPM and 5 bar BMEP.

Experimental results

Steady-state

The results from the steady-state experiment are listed in Table 4, showing the measured engine-out temperature, massflow, and emission concentration. The relative midway temperature

$$t_{\text{mid}} = \frac{\text{Ts11b} - \text{Ts10b}}{\text{Ts12b} - \text{Ts10b}},$$

indicates the relative temperature of thermocouple Ts11b (the axial and radial midpoint of the catalyst), where 0 corresponds to the temperature of the radial periphery (i.e. Ts10b) and 1 corresponds to the temperature of the radial center (i.e. Ts12b).

Cold-start thermal evolution

An illustration of the temperature evolution in the TWC is shown in Figure 9. It shows the radial and azimuth temperature evolution (Figure 9a) and the radially central and radially peripheral axial thermal evolution (Figure 9b) for the tested load step. Notably, Figure 9a indicates that there are no significant azimuth thermal variations (as Ts11b and Ts14b exhibit virtually identical trajectories), in turn indicating that the gas flow through the TWC is fairly evenly distributed azimuthally and that the radius of the 90°elbow after the turbocharger is sufficiently large for near-uniform flow. Furthermore, some radial variations are visible, as the radially outermost thermocouples (Ts10b, Ts15b) show a lower temperature after ≈ 100 s. Figure 9b shows a very significant axial thermal variation, with a clearly visible thermal front progressing over time from the incoming gas (Te9), to the front of the TWC (Ts10a/Ts12a), followed by the middle (Ts10b/Ts12b) and the rear (Ts10c/Ts12c) of the TWC, and finally the gas exiting the TWC (Ts23). Furthermore, the radial peripheral temperature is bears a close resemblance to the radially central temperature of each given axial slice. This motivates our choice of modeling the radially peripheral temperature as $T_n + \Delta_T$ (i.e. the peripheral temperature varies axially), in contrast to assuming a constant peripheral (canning) temperature. These results thus lend weight to our approach of modeling primarily axial, and to some extent radial, thermal variations.

An illustration of the net conversion efficiency of the TWC, defined as the ratio of the total converted mass to total incoming mass, is illustrated with dashed lines in the upper part of Figure 10. This figure was generated by using time-resolved emission data from two different sampling points, one before the TWC (i.e. measuring the total generated emissions) and one after the TWC (i.e. measuring the total remaining emissions). The figure illustrates that CO and NO_x reach light-off more quickly than THC, which is to be expected [1, p. 652]. Furthermore, we can reach a more fundamentally important conclusion; any attempt to characterize the entire TWC's conversion efficiency as a function of a single temperature during a thermal transient is bound to be limited in its accuracy. As Figure 9b shows, at for instance $t = 75$ s, the front of the catalyst is ≈ 375°C, while the rear is ≈ 125°C. We see a similar, albeit smaller, radial thermal difference in Figure 9a. Ultimately, this again motivates our modeling choice of a thermally resolved axi-radial model.

Model tuning

We utilized the experimental data in the previous section to tune the model constants. The tuning phase was divided into three different stages:

Radial temperature profile

By using the relative mid temperature, as found in the steady-state experiment and listed in Table 4, we uniquely determined the radial temperature profile associated with a given engine operating point. We followed the method described in the TWC model section, where we solved and normalized the heat equation solution over time. Afterwards, we selected the solution associated at a given time where the half-radius relative temperature was equal to the measured relative mid temperature t_{mid}.

Reaction rate parameters

The parameters E_a and A, as used in (9), were estimated using the cold-start experimental results. First, we assigned the time-evolution of the model's states to the measured thermal evolution (as shown in Figure 9). With our experimental data using three axial slices is most convenient, as we can generate $\bar{T} = [T_1, T_2, T_3, \Delta_T]$ directly by Ts12a, Ts12b, Ts12c, and Ts10b − Ts12b respectively. This explicitly gives the sample-evolution of the state vector $\bar{T}(k)$, where $k = 0, 1, 2, \dots$ indicates the time-sample of the state vector, sampled at rate of 10 Hz. The radial temperature profile was estimated using $M = 15$ independent radial channels. We chose to use 15 radial channels as this is sufficiently large to resolve the most significant radial temperature distribution, as will be shown later.

For given E_a^s and A^s we computed the modeled tailpipe output emissions $\dot{m}_{\text{tp}}^s(k)$ using (9) through (18). Finally, we tuned E_a^s and A^s by solving the numerical optimization problem

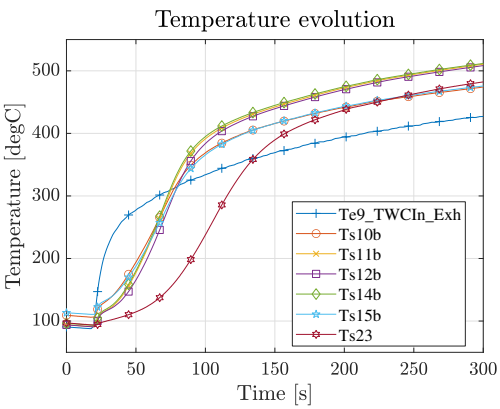
$$\min \sum_k |\dot{m}_{\text{tp}}^s(k) - \dot{m}_{\text{tp,meas}}^s(k)|^2 \tag{35}$$

for all measured samples (as shown in Figure 9), and where $\dot{m}_{\text{tp,meas}}^s(k)$ was the measured tailpipe emissions of each emission species.

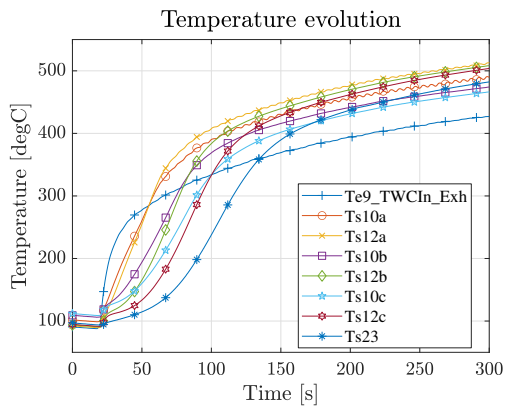
We solved (35) using the `patternsearch` optimization tool in MATLAB (a gradient-free direct-search method bearing some similarity to the Nelder-Mead method), which gave good matching between the measured and simulated output emissions, as shown in Figure 10. Note that the first 5 – 10 seconds of experimental data is of lower accuracy, particularly for THC, as the sharpening filter used could not completely compensate for the axial dispersion in the sampling lines and pump leading to the measurement equipment. This otherwise very good match indicates our choice of using a Arrhenius expression (9) and a first-order rate equation (10) can accurately approximate the true reactions in the TWC. The numerically

Table 4: Steady-state engine-out temperature, massflow, relative gas composition, and TWC midway temperature.

SPEED [RPM]	BMEP [BAR]	LAMBDA [-]	TEMPERATURE [°C]	MASSFLOW [G/S]	CO [PPM]	NOX [PPM]	THC [PPM]	t_{MID} [-]
1000	2	0.95	356	5.81	14500	151	586	0.723
		1.00	368	5.95	4660	183	470	0.787
		1.05	361	6.13	1370	325	405	0.757
2000	15	0.95	755	56.9	16000	1660	319	0.797
		1.00	755	55.6	3920	2720	204	0.815
		1.05	744	57.1	1270	3070	117	0.816
3000	8	0.95	812	50.0	15600	1050	293	0.912
		1.00	860	53.3	2140	1530	11.3	0.858
1500	5	0.95	583	16.3	14900	357	384	0.919
		1.00	623	17.6	4130	529	225	0.940
		1.05	642	19.7	943	782	59.7	0.983



(a) Radial and azimuthal temperature evolution of axially central slice. The radially central portions of the TWC heats earlier (Ts11b-Ts14b) than the radially outer parts (Ts10b, Ts15b). No noticeable azimuth variation.



(b) Axial temperature variation of radially central and radially peripheral sections. The sensors closest to the engine exhaust (Ts10a, Ts12a) heat up first, while sensors successively further back (Ts10b/Ts12b, and Ts10c/Ts12c) heat up after each other. Sensors of the same axial position (Ts12a and Ts10a, Ts12b and Ts10b, Ts12c and Ts10c) show a strong interdependence, motivating our modelling approximation of Δ_T being constant axially along the TWC.

Figure 9: Temperature evolution of TWC. Sensor locations as shown in Figure 7 and Figure 8.

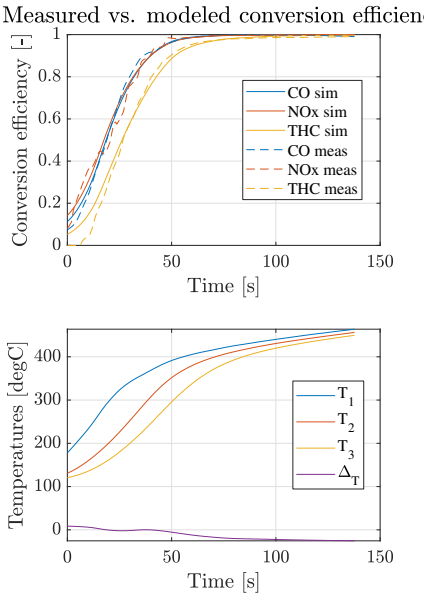


Figure 10: Simulated and measured output emissions. The core temperature of each axial slice T_1, T_2, T_3 as well as Δ_T is shown.

estimated parameters are shown in Table 5, and our estimated values of E_a are on the same order of magnitude (±50–100%) as those found in earlier studies, e.g. [23]. Note that we can not practically expect a more precise match as we do not model mass transport limitations, stored oxygen in the catalyst, and lambda-switching, all of which influence the apparent reaction kinetics.

Thermal dynamics parameters

The thermal dynamics tuning parameters τ_{ax}, τ_{ra}, τ_{amb}, k_{conv}, and τ_{convect} were also estimated using the cold-start thermal evolution. First, we set the initial state vector $\bar{T}(0) = [T_1, T_2, T_3, \Delta T]$ to the temperatures measured by Ts12a, Ts12b, Ts12c, and Ts10b − Ts12b respectively. Then, for given tuning parameter values we computed the modeled thermal evolution $\bar{T}(k)$, using an explicit 4'th order Runge-Kutta method. As the tuning parameter's influence on $\bar{T}(k)$ involves non-linear dynamics, conventional optimization methods tend to behave poorly. Here, we have used MATLAB's genetic algorithm (`ga`) tool, with the fitness function

$$\min \sum_k |\bar{T}(k) - \bar{T}_{\text{meas}}(k)|^2, \tag{36}$$

where $\bar{T}_{\text{meas}}$ is the measured temperature evolution at the sampled times (i.e. Ts12a, Ts12b, Ts12c, Ts10b − Ts12b), and otherwise using the default `ga` solver settings. Though other optimization methods tend to converge more quickly, the genetic algorithm is convenient in that it searches a wide range of initial values and the presented model evaluates quickly enough for the slow rate of convergence to not be problematic. Here, generating a solution required approximately 4 hours on a standard desktop PC (AMD Ryzen 2700x with 16 gb 3200 MHz RAM), where each model call required approximately 30 ms of compute time and generated modeled data corresponding to 425 seconds at sample rate of 50 ms (i.e. ≈ 280 000 model calls per second, or ≈ 15 000 times faster than realtime). The results gave good matching between the measured and simulated thermal trajectories, as shown in Figure 11. We re-ran the algorithm 10 times, and received virtually identical results each time.

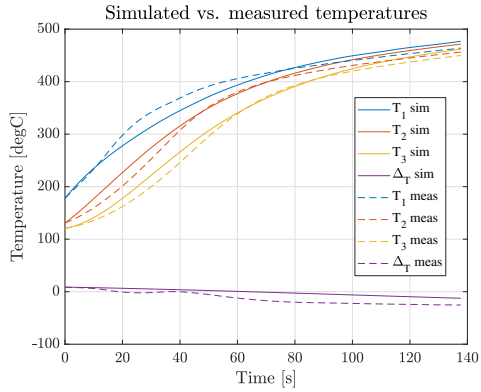


Figure 11: Simulated and measured thermal evolution of the TWC.

Table 5: Estimated reaction rate parameters.

Species	E_A [J/mol]	A
CO	$40.7 \cdot 10^3$	$161 \cdot 10^3$
NO_x	$37.5 \cdot 10^3$	$86.0 \cdot 10^3$
THC	$44.6 \cdot 10^3$	$215 \cdot 10^3$

Table 6: Estimated thermal parameters.

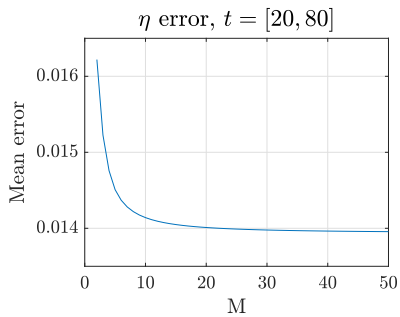
Parameter	Value
τ_{ax}	$88.2 \cdot 10^3$
τ_{ra}	385
τ_{amb}	$2.18 \cdot 10^3$
k_{conv}	$34.5 \cdot 10^{-6}$
τ_{convect}	171

The estimated parameters are listed in Table 6.

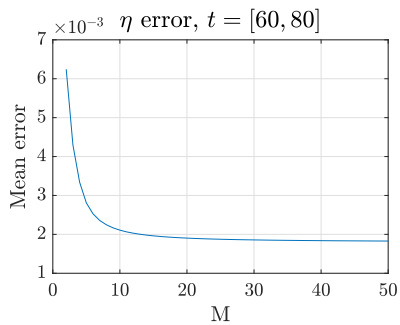
The difference between the measured and simulated temperatures in Figure 11 are typically within ±10°C, with a maximal deviation of 20°C. We hypothesize that the deviation between the simulated and measured T_1 at 20–60 seconds is caused by exhaust species that were adsorbed on the TWC during the first 20 seconds that start reacting as the TWC reaches lightoff (≈ 300°C). We have chosen not to model adsorbed exhaust species as this would require at least one additional state variable.

We can note that the thermal TWC model captures convective gas terms (seen during the first 20 seconds, where the entire TWC is below the light-off temperature) as well as exothermic reaction power generated in the TWC (seen after the first 80 seconds, where the TWC is well above the light-off temperature and is hotter than the incoming gasses). Furthermore, the model captures the characteristic thermal front that progresses from the front to the rear, as well as a radial temperature gradient that increases with increasing core temperatures, motivating our choice of a model that is axially and radially resolved.

Note that unlike many black-box modeling approaches, the tuning parameters in the model correspond to physical parameters. We hypothesize that this implies that we can tune the model for a given axial resolution M, and then easily generate tuning parameters for a different axial resolution M', by scaling the slice-volume-dependent parameters A and τ_{ax} by M/M', and using the remaining parameters as-is. However, we have as of yet not conclusively validated this hypothesis.

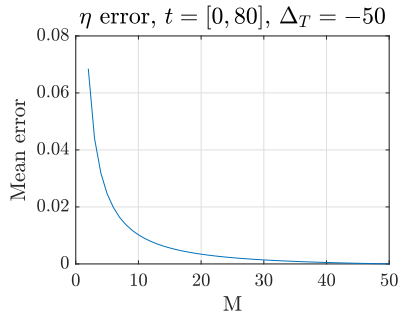


(a)

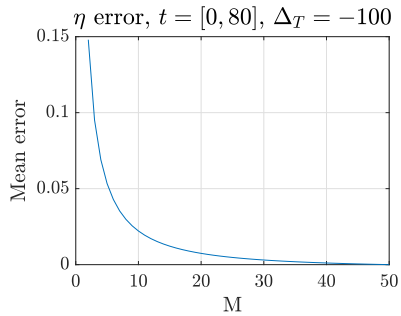


(b)

Figure 12: Mean absolute deviation between predicted and measured conversion efficiency for varying M. In Figure 12a this is computed over $t = [20, 80]$ s, while Figure 12b focuses on the region $t = [60, 80]$ s where Δ_T is more significant, as seen in Figure 10.



(a)



(b)

Figure 13: Mean absolute deviation between the simulated emission for $M = 50$ and the simulated emission for varying M, with Δ_T fixed to -50 and -100°C.

Radial resolution

One of the primary contributions of this paper is the inclusion of a radially resolved temperature profile. However, in order for this addition to be motivated, this should result in a model that more accurately predicts the TWC behavior. In Figure 12 we illustrate the mean deviation between the modeled and measured conversion efficiency (as previously studied in Figure 10) for varying radial resolutions M. Here, we have swept $M = [1, 50]$ while keeping the reaction rate parameters fixed at the values found for $M = 15$ (see Table 5). Both figures show an improvement in the estimated conversion efficiency with increasing M, indicating that resolving the radial temperature profile improves the model performance. Furthermore, the figures also indicate that the model is not over-fitted, as the prediction improves when M is increased from the value we used to tune the model. Finally, we can note that our choice of $M = 15$ is plausible, as the majority of the gains are had for $M \gtrsim 7$.

More specifically, if we consider Figure 12a, we can state that resolving the radial temperature profile marginally improves the average predicted conversion efficiency over $t = [20, 80]$, with a maximal relative improvement of approximately 15%. However, if we consult Figure 10 it is clear that $|\Delta_T|$ is very small at times $t = [20, 60]$. Limiting the averaging window to times where $|\Delta_T|$ is larger, e.g. $t = [60, 80]$, gives Figure 12b, where we see a relative improvement of approximately 350%. Though this is a very large relative improvement in accuracy, the absolute deviation between the modeled and measured conversion efficiency is small for all tested M here, as the conversion efficiency is very close to 1. However, had the engine been run in a manner that gave a large $|\Delta_T|$ near light-off we can expect the inclusion of a radially resolved temperature profile to give a significant absolute accuracy improvement. An illustration of this is shown in Figure 13, where we simulate the effect of a constant $\Delta_T = \{-50, -100\}$, but otherwise use the same thermal evolution and engine-out emissions. Though a constant Δ_T is not practically representative, it is plausible for $|\Delta_T|$ to be large during intervals near light-off. (For example, if a hybrid vehicle starts the combustion engine when reaching highway speeds the TWC periphery will be significantly cooled by the passing air.) In these figures we compare the mean emissions for $M = 50$ with the emissions for varying M over the entire cold-start phase. These results show that resolving the radial temperature distribution can lead to an average absolute accuracy improvement by 7–15 percentage points over the entire cold-start cycle, with gains that increase as the magnitude of Δ_T increases.

Conclusion

In this paper we have introduced a physics-based TWC model that resolves axial and radial thermal dynamics, while simultaneously keeping the number of state variables low, ultimately allowing for use as a model for fast off-line simulation or on-line control methods (e.g. nonlinear state-feedback methods and explicit MPC). The number of axial slices (and thereby the number of state variables) and radial resolution is configurable, allowing for balancing the computational demand and accuracy for a specific application. The model uses a first-principles approach for chemical kinetics, exothermic reaction power, gas convection, axial and radial thermal conduction, and an interpolation method for increasing the radial resolution.

Resolving the axial and radial temperature profile allows for implementing more nuanced control schemes, where if for instance during a cold-start the first axial slice(s) of the TWC are hot, then the exhaust mass-flow can be kept sufficiently low in order to increase the residence time in the hot part(s) of the TWC. This awareness of the TWC's condition is lost with most conventional on-line models that only characterize TWC temperature as a scalar value.

We have performed an experimental campaign where we measured the temperatures and emission massflow in the TWC using a conventional experimental rig. This experimental campaign required only a few hours of effective measurements, after which we were able to identify the model's tuning parameters. These consist of three activation energies, three pre-exponential scaling terms, one radial thermal distribution table, and five thermal parameters.

Subsequent analysis showed that the modeled and experimentally measured conversion efficiencies very closely match each other, even with only three axial slices. Furthermore, the modeled temperature evolution matched the experimentally measured temperature well, typically deviating no more than ±10°C. The most significant discrepancies were seen just after lightoff, plausibly due to exhaust species that adsorbed onto the cold TWC that then exothermically reacted after lightoff. Resolving the radial temperature profile improved the relative accuracy of conversion efficiency by approximately 15%, but would likely be much more significant for operating regimes with a more prominent radial temperature gradient. Simulations show the potential for the *absolute* conversion efficiency accuracy to improve by 7–15 percentage points when the temperature of the TWC periphery differs from the center by 100°C.

The fast computational speed of the model (approximately 15000 times faster than realtime on a desktop PC (AMD Ryzen 2700x with 16 gb 3200 MHz RAM) with unoptimized code) and the low number of state variables open up for several possible use-cases. Examples include closed-loop on-line control methods suited for low-dimensional non-linear models (such as nonlinear state-feedback and explicit MPC) and very numerically fast off-line TWC simulations. This can ultimately allow for on-line optimal cold-start strategies, TWC sizing, or more quickly evaluating the performance of exhaust aftertreatment systems with several different TWC elements. Relevant future work includes constructing a suitable state observer for on-line applications that have a limited number of measured temperatures.

References

1. J. B. Heywood, *Internal Combustion Engine Fundamentals*. McGraw Hill, 1988.

2. R. Hedinger, P. Elbert, and C. Onder, "Optimal Cold-Start Control of a Gasoline Engine," *Energies*, vol. 10, p. 1548, Oct. 2017.

3. K. Ramanathan, D. H. West, and V. Balakotaiah, "Optimal design of catalytic converters for minimizing cold-start emissions," *Catalysis Today*, vol. 98, no. 3, pp. 357–373, 2004.

4. J. Gao, G. Tian, A. Sorniotti, A. E. Karci, and R. Di Palo, "Review of thermal management of catalytic converters to decrease engine emissions during cold start and warm up," *Applied Thermal Engineering*, vol. 147, pp. 177–187, 2019.

5. M. Schori, T. Boehme, T. Jeinsch, and M. Schultalbers, "Optimal catalytic converter heating in hybrid vehicles," *SAE Technical Paper*, 2014.

6. R. S. Pannag, "Coldstart Modeling and Optimal Control Design for Automotive SI engine," 2009.

7. G. Fiengo, L. Glielmo, S. Santini, and G. Serra, "Control of the exhaust gas emissions during the warm-up process of a TWC-equipped SI engine," *IFAC Proceedings Volumes*, vol. 35, no. 1, pp. 301–306, 2002.

8. F. Keynejad and C. Manzie, "Suboptimal Cold Start Strategies for Spark Ignition Engines," *IEEE Transactions on Control Systems Technology*, vol. 21, pp. 1295–1308, July 2013.

9. D. Kum, H. Peng, and N. Bucknor, "Modeling and Control of Hybrid Electric Vehicles for Fuel and Emission Reduction," in

ASME 2008 Dynamic Systems and Control Conference, Parts A and B, (Ann Arbor, Michigan, USA), pp. 553–560, ASMEDC, Jan. 2008.

10. G. Fiengo, L. Glielmo, S. Santini, and G. Serra, "Control oriented models for TWC-equipped spark ignition engines during the warm-up phase," in *Proceedings of the 2002 American Control Conference (IEEE Cat. No.CH37301)*, (Anchorage, AK, USA), pp. 1761–1766 vol.3, IEEE, 2002.

11. E. Brandt, Yanying Wang, and J. Grizzle, "Dynamic modeling of a three-way catalyst for SI engine exhaust emission control," *IEEE Transactions on Control Systems Technology*, vol. 8, pp. 767–776, Sept. 2000.

12. U. Kiencke and L. Nielsen, *Automotive Control Systems*. Springer, 2000.

13. H. Bauer, ed., *Gasoline-engine management*. Robert Bosch GmbH, 1999.

14. P. Michel, "Optimizing fuel consumption and pollutant emissions of gasoline-HEV with catalytic converter," *Control Engineering Practice*, p. 8, 2017.

15. N. L. Azad, P. R. Sanketi, and J. K. Hedrick, "Determining Model Accuracy Requirements for Automotive Engine Coldstart Hydrocarbon Emissions Control," *Journal of Dynamic Systems, Measurement, and Control*, vol. 134, p. 051002, Sept. 2012.

16. Z. Zhu, S. Midlam-Mohler, and M. Canova, "Development of physics-based three-way catalytic converter model for real-time distributed temperature prediction using proper orthogonal decomposition and collocation," *International Journal of Engine Research*, p. 146808741987612, Sept. 2019.

17. D. P. Bertsekas, *Dynamic Programming and Optimal Control*, vol. 1. Athena Scientific, 4 ed., 2017.

18. M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 1994.

19. J. Logan, F. Gehring, and P. Halmos, *Applied Partial Differential Equations*. Springer Undergraduate Mathematics Series, Springer, 1998.

20. K. N. Pattas, A. M. Stamatelos, P. K. Pistikopoulos, G. C. Koltsakis, P. A. Konstandinidis, E. Volpi, and E. Leveroni, "Transient modeling of 3-way catalytic converters," *International Congress & Exposition*, 1994.

21. S. Tischer, Y. Jiang, K. W. Hughes, M. D. Patil, and M. Murtagh, "Three-way-catalyst modeling - a comparison of 1d and 2d simulations," *SAE World Congress & Exhibition*, pp. 2007–01–1071, 2007.

22. I. Svraka and Ö. W. Linus, "Model based catalyst control," Master's thesis, Linköping University, 2019.

23. S. B. Kang, S. J. Han, I.-S. Nam, B. K. Cho, C. H. Kim, and S. H. Oh, "Detailed reaction kinetics for double-layered pd/rh bimetallic TWC monolith catalyst," *Chemical Engineering Journal*, vol. 241, pp. 273–287, 2014.

Nomenclature

TWC	Three way catalyst
DP	Dynamic programming
SI	Spark-ignited
DOHC	Dual Over Head Camshaft
CO	Carbon monoxide
NO_x	Nitrogen oxides
THC	Total hydrocarbons
$\bar{T}$	Model state vector
T_n	Temperature of axial slice n
Δ_T	Temperature difference between radial center and radial periphery
N	Number of axial elements
M	Number of interpolated radial elements
$T_{n,m}$	Temperature of cell n, m
$\dot{m}_{n,m}$	Total massflow into cell n, m
$k^s_{n,m}$	Reaction rate of emission species s in cell n, m
A^s	Apparent pre-exponential scaling factor for Arrhenius expression for emission species s in any given cell
E^s_a	Activation energy for emission species s
$y^s_{n,m}$	Mole fraction of emission species s in cell n, m
t_r	Residence time of exhaust gas in a given cell
V_{slice}	Gas volume of each axial slice
V_{TWC}	Gas volume of the entire TWC
ν	Volumetric flow rate in a given cell
$\dot{m}_{\text{eng ex}}$	Total exhaust massflow
P	Absolute pressure in the TWC
$\dot{m}^{s,\text{out}}_{n,m}$	Massflow of emission species s leaving cell n, m
$\dot{m}^{s,\text{in}}_{n,m}$	Massflow of emission species s entering cell n, m
$\dot{m}^{s,\text{conv}}_{n,m}$	Massflow of emission species s converted in cell n, m
$\eta^s_{n,m}$	Conversion efficiency of emission species s in cell n, m
$\dot{m}^s_{\text{tp}}$	Tailpipe massflow of emission species s
$\mathrm{d}H_x$	Molar heat of reaction of x
$P_{n,m}$	Exothermic reaction power in cell n, m
$\bar{T}_{\text{cond}}$	State derivative due to conductive terms
$\bar{T}_{\text{exo}}$	State derivative due to exothermic reaction terms
$\bar{T}_{\text{convect}}$	State derivative due to convective terms
τ_{ax}	Tuning parameter, axial thermal resistance
τ_{ra}	Tuning parameter, radial thermal resistance
τ_{amb}	Tuning parameter, ambient thermal resistance
T_{amb}	Ambient temperature
$P_{\text{ctr,n}}$	Weighted exothermic reaction power, center of axial slice n
$P_{\text{per,n}}$	Weighted exothermic reaction power, periphery of axial slice n
k_{conv}	Tuning parameter, exothermic reaction power proportionality constant
τ_{convect}	Tuning parameter, gas convection proportionality constant
$T_{\text{eng ex}}$	Engine exhaust temperature
$\bar{T}_{\text{meas}}$	Experimentally measured temperatures corresponding to model state vector
t_{mid}	Relative radially midway temperature
$\dot{m}^s_{\text{tp,meas}}$	Experimentally measured tailpipe massflow of emission species s

Contact Information

Jonathan Lock
+46317725792
lock@chalmers.se

Acknowledgments

This work has been performed within the Combustion Engine Research Center at Chalmers (CERC) with financial support from the Swedish Energy Agency.

PAPER E

Cold-Start Modeling and On-Line Optimal Control of the Three-Way Catalyst

Jonathan Lock, Kristoffer Clasén, Jonas Sjöblom, Tomas McKelvey

To be submitted
Preprint available at `https://arxiv.org/abs/2104.12390`

Cold-Start Modeling and On-Line Optimal Control of the Three-Way Catalyst

Jonathan Lock, Kristoffer Clasén, Jonas Sjöblom, Tomas McKelvey

***Abstract*—We present a three-way catalyst (TWC) cold-start model, calibrate the model based on experimental data from multiple operating points, and use the model to generate a Pareto-optimal cold-start controller suitable for implementation in standard engine control unit hardware. The TWC model is an extension of a previously presented physics-based model that predicts carbon monoxide, hydrocarbon, and nitrogen oxides tailpipe emissions. The model axially and radially resolves the temperatures in the monolith using very few state variables, thus allowing for use with control-policy based optimal control methods. In this paper we extend the model to allow for variable axial discretization lengths, include the heat of reaction from hydrogen gas generated from the combustion engine, and reformulate the model parameters to be expressed in conventional units. We experimentally measured the temperature and emission evolution for cold-starts with ten different engine load points, which was subsequently used to tune the model parameters (e.g. chemical reaction rates, specific heats, and thermal resistances). The simulated cumulative tailpipe emission modeling error was found to be typically -20% to +80% of the measured emissions. We have constructed and simulated the performance of a Pareto-optimal controller using this model that balances fuel efficiency and the cumulative emissions of each individual species. A benchmark of the optimal controller with a conventional cold-start strategy shows the potential for reducing the cold-start emissions.**

I. Introduction

The Three-Way Catalyst (TWC) is used in nearly all conventional vehicles with spark-ignited (SI) engines to reduce the level of harmful emissions generated by the combustion engine that would otherwise exit the tailpipe. The toxic emissions generated by the combustion engine, broadly categorized as nitrogen oxides (NO_x), carbon monoxide (CO), and residual hydrocarbons (THC), are in the TWC converted to primarily form non-toxic nitrogen gas (N_2), carbon dioxide (CO_2), and water (H_2O) [2, 3]. Modern TWCs are very effective at removing emissions, with conversion efficiencies of over 95% being commonplace [3] and in some cases significantly higher, as was found in the experimental results in this paper. However, the TWC must be sufficiently hot to function, which is in ordinary operation maintained by virtue of the hot exhaust gases passing through it from the combustion engine. However, when a vehicle is cold-started (i.e. started after the TWC has had sufficient time to cool to the ambient temperature) the tailpipe emissions are much larger until the TWC is heated to its ordinary operation temperature, an interval typically taking on the order of 40–100 seconds[3]. These cold-start emissions are very significant, and for many regulatory test procedures are responsible for 60–80% of the emissions generated from an entire test (which are for reference on the order of 30 minutes).

Several methods for reducing cold-start emissions have been studied from a multiple perspectives. These range from methods of constructing the TWC that reduces the cold-start time [4], methods for preheating the TWC before starting the combustion engine [5], and control schemes that focus on controlling the combustion engine's operation to limit the emissions generated during the cold-start [3, 6, 7, 8, 9].

In this paper our goal is to develop a model-based optimal TWC cold-start controller that can be feasibly be implemented in a standard engine control unit (ECU). We will consider a conventional SI engine and TWC, where the engine's load point can be freely controlled during the cold-start, making the controller suitable for e.g. hybrid vehicles. More specifically, we view the TWC cold-start problem as determining the optimal engine speed, load, and spark timing to apply over time while balancing the conflicting goals of maximizing fuel efficiency and minimizing the cumulative emissions. Generating a Pareto-optimal controller will therefore both require a dynamic thermal model of the TWC as well as a suitable optimal control design method. This paper can naturally be divided into two parts, one where we develop a TWC cold-start model, and one where we evaluate the performance of an optimal controller generated using said model.

The model presented in this paper extends on a TWC model previously developed by the authors [1]. In the previous work, we derived a thermal model of the TWC with very few state variables suited for fast off-line simulation or on-line control systems. The model resolved both axial and radial temperature variations from a first-principles perspective. In this paper we extend the previous model by allowing the axial discretization to vary along the TWC's length, model the heat generation caused by oxidation of hydrogen in the exhaust gas, and reformulate the tuning parameters (heat capacity, thermal conductivity, etc) to be expressed in well-known units ($\mathrm{J\,K^{-1}\,kg^{-1}}$, $\mathrm{W\,m^{-1}\,K^{-1}}$, etc). Furthermore, we expand on our previous work by here considering a TWC consisting of two separate monoliths and use separate training and validation datasets for tuning and evaluation.

In this paper we will first briefly discuss categories of existing models and their strengths, weaknesses, and relevant applications. Following this we will introduce our extended model and the experimental setup used to calibrate the model. Finally, we will study the experimental results and evaluate the simulated performance of the optimal control scheme using the calibrated model. A listing of all abbreviations is shown in Table I, while all model parameters and their units are presented in Table II.

TABLE I
TABLE OF USED ABBREVIATIONS.

CAbTDC	Crank Angle before Top Dead Center
CO	Carbon Monoxide
ECU	Engine Control Unit
SI	Spark Ignited
NO_x	Nitrogen Oxides
SA	Spark Angle
THC	Total Hydrocarbon
TWC	Three Way Catalyst
TWC1	The first monolith in the TWC
TWC2	The second monolith in the TWC

A. Literature survey

The topic of modeling the TWC for cold-start purposes has been considered from a wide range of perspectives. Some authors derive fully three-dimensional or two-dimensional models that capture many of the fundamentally complex chemical kinetics, transport dynamics, and temperature dynamics [10, 11, 12]. Though accurate, models of this caliber are very computationally demanding and primarily suited for in-depth analysis.

One commonly used method of reducing the computational demand is to exploit the characteristic structure of modern TWC's, which consist of a large number of parallel channels. Assuming each channel's construction and composition is identical it is sufficient to study the behavior of a single channel and afterwards scale the result by the number of channels in the TWC. This approximation is typically referred to as the single channel approximation or single channel model, and results in models with significantly reduced computational demands.

Several authors[13, 14, 15, 16, 17, 18, 19] model the TWC with the single channel approximation, and depending on the number of modeled chemical phenomenon and the level of spatial resolution some models can approach realtime simulation speeds on powerful PC's (i.e. where generating one second's worth of data requires one second of computation time). Beyond modeling the temperature dependence of the TWC, many of these models also include terms to capture a TWC phenomenon where oxygen is absorbed and released in the TWC. The stored oxygen greatly influences the TWC's capacity to convert emissions, where CO and THC are more effectively oxidized when the stored level of oxygen is large, while NO_x is more effectively reduced when the level is low.

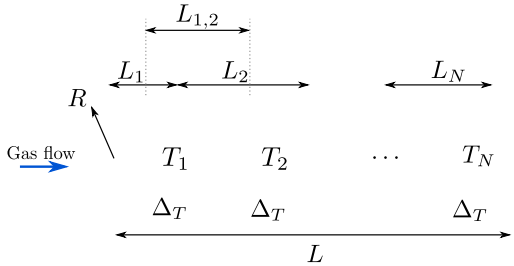
These models allow for simulating either the whole TWC or a representative channel to a fairly high degree of accuracy. However, they are primarily of use for analyzing the performance of the TWC in an off-line manner, for instance in a TWC design process. In this paper we instead focus on on-line TWC control, and in particular consider the cold-start problem. This requires a model that is significantly simpler computationally, both as optimal control methods place specific demands on the complexity and structure of the system models, and as the ECU has a very limited computational capacity. There are several classes of control-oriented models, ranging from models that approximate the spatially varying temperature distribution as a scalar temperature [20, 6, 7, 8], to more complex models that

TABLE II
TWC PARAMETERS.

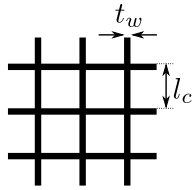
Parameter	Unit	Description
L	m	Total TWC length
R	m	TWC radius
L_n	m	Length of axial slice n
$L_{n-1,n}$	m	Center-to-center distance between axial slice $n-1$ and n
t_w	m	TWC wall thickness
l_c	m	TWC channel width
OFA	-	TWC open frontal area
m_{TWC}	kg	Monolith mass
c_p	$\mathrm{J\,K^{-1}\,kg^{-1}}$	Net monolith specific heat
$\dot{m}_{\mathrm{exh}}$	$\mathrm{kg\,s^{-1}}$	Exhaust massflow
$c_{p,\mathrm{exh}}$	$\mathrm{J\,K^{-1}}$	Exhaust gas specific heat
T_{amb}	°C	Ambient temperature
T_{exh}	°C	Temperature of gas feeding TWC
k_{ax}	$\mathrm{W\,m^{-1}\,K^{-1}}$	Effective axial thermal conductivity
k_{rad}	$\mathrm{W\,m^{-1}\,K^{-1}}$	Effective radial thermal conductivity
k_{amb}	$\mathrm{W\,m^{-1}\,K^{-1}}$	Effective thermal conductivity to ambient
t_{amb}	m	Thickness of insulation to ambient
N	-	Number of axial TWC segments
M	-	Number of resolved radial TWC channels
T	°C	Temperature state vector
T_n	°C	Central temperature in axial slice n
Δ_T	°C	Center/periphery temperature difference
$k^s_{n,m}$	$\mathrm{mol\,s^{-1}}$	Reaction rate, species s in cell n,m
A^s	-	Pre-exponential factor, species s
E^s_a	$\mathrm{J\,mol^{-1}}$	Activation energy of emission species s
$y^s_{n,m}$	-	Mole fraction, species s in cell n,m
$\dot{m}^{\mathrm{s,conv}}_{n,m}$	$\mathrm{kg\,s^{-1}}$	Mass conversion, species s, cell n,m
$\dot{m}^{\mathrm{s,in}}_{n,m}$	$\mathrm{kg\,s^{-1}}$	Incoming mass, species s, cell n,m
$\dot{m}^{\mathrm{s,conv}}_{n,m}$	$\mathrm{kg\,s^{-1}}$	Outgoing mass, species s, cell n,m
$\dot{m}^s_{\mathrm{tp}}$	$\mathrm{kg\,s^{-1}}$	Tailpipe massflow, species s
$t_{r,n}$	s	Gas residence time in slice n
p_{TWC}	Pa	Absolute pressure in TWC
P_{ctr}	W	Heating power, radial center
P_{per}	W	Heating power, radial periphery
P_{ax}	W	Heating power, axial conduction
P_{rad}	W	Heating power, radial conduction
$P_{\mathrm{con,ctr}}$	W	Heating power, convection, center
$P_{\mathrm{con,per}}$	W	Heating power, convection, periphery
$P_{n,m}$	W	Exothermic power generated in cell n,n
$P_{\mathrm{exo,ctr}}$	W	Weighted central exothermic power
$P_{\mathrm{exo,per}}$	W	Weighted peripheral exothermic power
P_{amb}	W	Heating power, loss to ambient
q_{ax}	$\mathrm{W\,m^{-2}}$	Axial heat flux
q_{rad}	$\mathrm{W\,m^{-2}}$	Radial heat flux
q_{amb}	$\mathrm{W\,m^{-2}}$	Heat flux to ambient

axially resolve the TWC temperature and/or include oxygen-storage terms [21, 22, 23]. The simpler models are fairly well suited for direct use with on-line optimal control methods, while most of the more complex models are used primarily as a starting point for creating a suboptimal controller.

The model presented in this paper (an extension of [1]) has been constructed for the specific purpose of subsequently being implemented for optimal control in conventional ECUs. More specifically, it is well-suited to control-policy based optimal control methods, where the optimal control signal (e.g. engine speed, load, spark angle, and so on) is precomputed in an offline phase and stored in a table for a discrete set of TWC temperatures. A subsequent realtime controller can ultimately determine the optimal control signal by consulting the table of stored temperatures and associated optimal control signals [24, 25]. Controllers of this class are very powerful, as they allow for nearly arbitrarily nonlinear model dynamics, costs, and constraints, but are limited in that their memory demand scales exponentially with the number of state variables and require the states variables to either be measured or estimated. In an effort to limit the number of state variables we have chosen to not dynamically model the stored oxygen in the TWC as this is not as significant as the temperature dynamics during a cold-start [20, 1].



(a) TWC body. Engine exhaust travels from left to right. Figure derived from [1].



(b) Detail of TWC channels.

Fig. 1. Illustration of geometry and measurement definitions for a single TWC monolith.

II. TWC Model

We will in this section introduce the TWC model. This model is based on and extends a model previously presented by the authors [1]. The model is extended by allowing the size of the axially discretized slices to vary over the length of the catalyst, reformulated so all parameters are based on easily determined physical parameters, and we consider the case where two separate TWC monoliths are placed in series.

The model can naturally be divided into three distinct subsections; one modeling the chemical kinetics, one modeling the temperature dynamics, and one that interpolates the low-dimensional state variables to a higher-dimensional temperature distribution using a physics-based method. We will initially consider a single TWC monolith, and later return to the case where two are placed in series.

An illustration of the assumed TWC geometry is shown in Figs. 1a and 1b. Specifically, we assume that the TWC is cylindrical with radius R and length L. We make the modeling choice of dividing the TWC into N different axial slices, and extend the previously presented model [1] that assumed equally-sized slices by allowing the associated lengths $L_1, L_2, \ldots, L_N$ (where $\sum L_n = L$) to be different for each slice. We also define the lengths $L_{1,2}, L_{2,3}, \ldots, L_{N-1,N}$ as the axial distances between the midpoints of neighboring slices. Finally, we assume the TWC has a monolithic structure with square, axially traversing channels of wall thickness t_w and channel length l_c, as illustrated in Fig. 1b.

Importantly, we make the approximation of axially discretizing the TWC temperature. More specifically, we model the TWC temperature in the radial center of slice n as T_n. This implies that the temperature in the radially central channel is modeled as N segments of constant temperature. Furthermore, we model the difference in temperature between the radial center and radial periphery of each slice as as Δ_T. Note that Δ_T is not axially resolved, i.e. we assume that Δ_T is identical for all axial slices, i.e. the periphery temperature of slice 1 is $T_1 + \Delta_T$, slice 2 is $T_2 + \Delta_T$, and so on. Finally, as is described in more detail in Section II-A, we use an interpolation scheme to approximate the temperature at M different radial locations ranging from the radial center to the periphery. Ultimately, a single TWC monolith is at any given instance in time characterized by the state variable vector

$$T = \begin{bmatrix} T_1 \\ T_2 \\ \ldots \\ T_N \\ \Delta_T \end{bmatrix}. \tag{1}$$

In the following subsections we will detail the individual parts of the model. First, we define how the state variable is used to generate a dense representation of the TWC temperature that is axially and radially resolved. This is followed by the chemical kinetics model that determines the conversion efficiency of the TWC as well as the heating power generated by the exothermic reactions. Finally, we introduce the thermal model that generates the state variable derivative and define the interface between the two separate TWC monoliths.

A. Radial temperature interpolation

The model presented in this work uses the single-channel approximation for fundamental TWC material properties, while resolving the radial temperature profile by simulating several

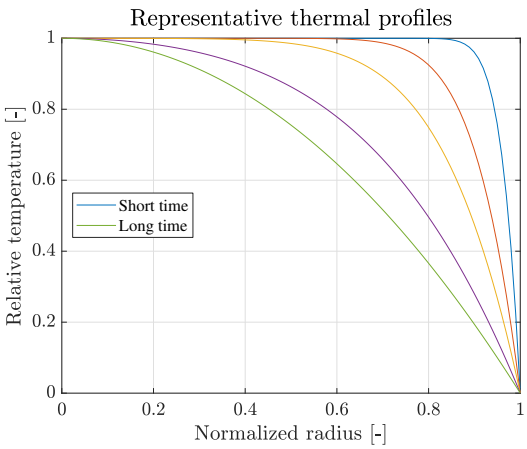


Fig. 2. Representative solutions to the transient heat equation in a flat disc with a homogeneous power term and a Dirichlet boundary condition $\hat{T}(R) = 1$. The displayed temperatures and radii have been normalized to the range $[0, 1]$. Figure reused from [1].

parallel channels corresponding to different radial positions with different associated temperatures. This allows for capturing the experimentally observed behavior where the periphery of the TWC is significantly colder than the radially central sections (as we will be discussed in Section IV-B).

More specifically, we model the radial temperature profile $\hat{T}(t, r)$, at time t and radius r, as a solution to the transient heat equation in a flat circular disc with radius R and an initial temperature of zero, i.e. $\hat{T}(0, r) = 0, r = [0, R]$. Note that $\hat{T}$ has no relation to the state variable T or the axial slice temperatures T_n despite the similar notation. Furthermore, we assume a Dirichlet boundary condition, i.e. $\hat{T}(t, R) = 0$, and assume that the plate develops a constant homogeneous power. This power is intended to be analogous to the power delivered to a slice in the TWC by convection, axial conduction, and exothermic heat generation. Solving the time-evolution of $\hat{T}(t, r)$ is a textbook problem (e.g. [26, p. 148]) with a solution that can be expressed as a Fourier-Bessel series. Solving this numerically over time and radii can also be done easily (for instance with MATLAB's `pdepe` function), generating the radially-varying time evolution of the plate's temperature. Normalized solutions (where $\hat{T}(t, r)$ is scaled to the range $[0, 1]$, and $R = 1$) are shown in Fig. 2.

In this paper we interpolate the radial TWC temperature profile by using precomputed solutions to the above flat-plate problem. More specifically, we assume that the radial temperature profile in slice n is given by $A_1\hat{T}(t', r) + A_2$ for a given time instant t'. Here, A_1 and A_2 are selected so that $\hat{T}(t', 0) = T_n$ and $\hat{T}(t', R) = T_n + \Delta_T$ (i.e. $\hat{T}$ is scaled and offset to match the known radial center and periphery temperatures), and t' is selected to give a radial temperature profile that matches the experimentally measured temperature profile for a given engine operating point (more on this in Section IV-A). In summary, we interpolate the

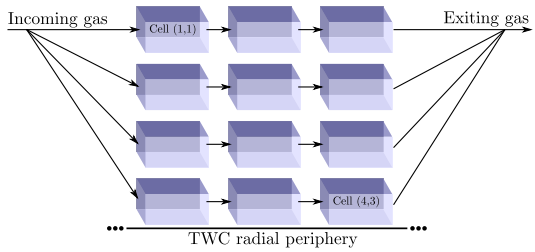


Fig. 3. Fully-resolved TWC, here shown for $N = 3$ and $M = 4$. Figure reused from [1].

radial temperature distribution using the known radially central and peripheral temperatures and the instantaneous engine operating point. Letting M denote the number of independent single-channel models we wish to resolve, this interpolation scheme allows us to convert the $N + 1$ state variables to a representation with M single-channel models of N segments. This is illustrated in Fig. 3, where cells $([1 \dots M], 1)$ are fed with the incoming gas, cell $(m, n + 1)$ is fed with the output of cell (m, n), and the output from cells $([1 \dots M], N)$ are combined to form the total exhaust from the TWC.

As the TWC is assumed to be circular, regions near the periphery have a larger associated area than regions near the axial center. This is taken into account in our model by approximating the massflow in the physical TWC as equal at all locations, and scaling the proportion of gas passing through each channel to match. Letting $\dot{m}_{\text{exh}}$ be the total exhaust massflow from the engine and scaling by the relative area of an annular ring with a major radius of m/M and a minor radius of $(m - 1)/M$ gives the massflow into a given cell as

$$\dot{m}_{m,1} = \frac{\pi(m^2 - (m-1)^2)}{\pi M^2}\dot{m}_{\text{exh}} = \frac{(2m-1)}{M^2}\dot{m}_{\text{exh}} \tag{2a}$$

$$\dot{m}_{m,n+1} = \dot{m}_{m,n}\,. \tag{2b}$$

B. Chemical kinetics

The total range of chemical reactions occurring in the TWC are very complex and involve a wide range of compounds. However, there are fewer that contribute to the legislated emissions or significant heat generation. We will therefore limit our scope to net reactions (i.e. without considering intermediary steps). This is done both for simplicity, and as a detailed approach would require the addition of numerous state variables that track the concentration of the emission species and their intermediaries in the TWC. [19, 27, 1] give the most significant

reactions (apart from Eq. (3b)) as

$$2\,CO + O_2 \longrightarrow 2\,CO_2 \tag{3a}$$
$$2\,H_2 + O2 \longrightarrow 2\,H_2O \tag{3b}$$
$$2\,NO + 2\,CO \longrightarrow N_2 + 2\,CO_2 \tag{3c}$$
$$2\,NO_2 \longrightarrow N_2 + 2\,O_2 \tag{3d}$$
$$2\,C_3H_6 + 9\,O_2 \longrightarrow 6\,CO_2 + 6\,H_2O \tag{3e}$$
$$C_3H_8 + 5\,O_2 \longrightarrow 3\,CO_2 + 4\,H_2O\,. \tag{3f}$$

We will assume all reactions are balanced, i.e. for a TWC with 100% conversion efficiency all CO, H_2, C_3H_6, and C_3H_8 emitted from the engine are fully oxidized and all NO and NO_2 are fully reduced.

In this paper we include the heat generated by the oxidation of hydrogen gas, i.e. (3b), which is generated in the engine by the water-gas shift reaction. As H_2 is not typically experimentally measured we will instead estimate its mole-fraction from the measured mole-fractions of CO and CO_2, as given by [2, eq. (4.68)]

$$y_{H_2} = \frac{y_{H_2O} y_{CO}}{K y_{CO_2}} \tag{4a}$$

where

$$y_{H_2O} = \frac{m'}{2n'} \frac{y_{CO} + y_{CO_2}}{1 + y_{CO}/(K y_{CO_2}) + (m'/2n')(y_{CO} + y_{CO_2})}\,. \tag{4b}$$

Here, y corresponds to the mole-fraction of each corresponding compound, K is a constant value set to 3.8 [2, eq (4.63)], and n' and m' correspond to the number of carbon and hydrogen atoms respectively[1] in each molecule of the fuel. With the RON95 E10 fuel studied in this paper we used the supplier-specified value of $m'/2n' = 0.258$.

Typically, nitrogen oxides (NO and NO_2) and hydrocarbon (C_3H_6 and C_3H_8) emissions are lumped together and denoted as NO_x and THC respectively [2, pp. 572–597]. We will in this paper assume a constant ratio of 99:1 for NO to NO_2 as indicated by [2, p. 578], and by [28, 29] a constant ratio of 3:1 for C_3H_6 to C_3H_8.

We model the reaction rate $k^s_{n,m}$ of an emissions species s in any given cell n, m using an Arrhenius expression of form

$$k^s_{n,m} = A^s e^{\frac{-E^s_a}{RT_{n,m}}}\,, \tag{5}$$

where R is the ideal gas constant, $T_{n,m}$ is the temperature of cell n, m, E^s_a is the activation energy of emission species s, and A^s is the apparent pre-exponential factor for species s. Letting $y^s_{n,m}$ indicate the mole fraction of emission species s in cell n, m, we model the evolution of the mole fraction as

$$\frac{\mathrm{d}y^s_{n,m}}{\mathrm{d}t} = -k^s_{n,m} y^s_{n,m}\,. \tag{6}$$

Note that we do not include an inhibition factor in Eq. (6) in order to limit the complexity of the model and the model tuning process. However, including an inhibition term (e.g. as in [18]) is viable and would not interfere with the optimal control method that will be described in Section V. Furthermore, note that Eq. (6) does not include an oxygen concentration term. As the engine is operated stoichiometrically the O_2 concentration is fairly constant, implying that it can be lumped into $k^s_{n,m}$. This is beneficial as we avoid the need to explicitly measure or model the O_2 concentration.

Though Eq. (4) allows for generating an estimate of the hydrogen gas concentration for a given CO and CO_2 concentration (quantities which are easily measured with conventional emissions-measurement equipment), as y^{H_2} is not typically measured it is difficult to tune the associated reaction rate parameters. By [18, 29] we have chosen to instead model the reaction rate of Eq. (3b) as identical to that of Eq. (3a), i.e. we assume $E^{H_2}_a = E^{CO}_a$ and $A^{H_2} = A^{CO}$. Finally, as H_2 is not typically viewed as a problematic emission species we will in the remainder of this paper only consider Eq. (3b) from the perspective of determining the heat of reaction, in contrast to CO, THC, and NO_x emissions which both contribute with their associated heat of reaction and whose tailpipe emissions are important to track.

As described in more detail in [1], the gas residence time in each axial slice is short enough for the monolith temperature to be close-to constant. Using this constant-temperature approximation we can explicitly solve Eq. (6) as

$$y^s_{n,m}(t_{r,n}) = y^s_{n,m}(0) e^{-k^s_{n,m} t_{r,n}} \tag{7}$$

for a residence time in slice n of $t_{r,n}$. We will approximate the residence time by assuming a plug-flow reactor model (i.e. assuming there is no axial dispersion), giving

$$t_{r,n} = \frac{V_{\mathrm{slice},n}}{\nu} \tag{8}$$

where $V_{\mathrm{slice},n}$ is the gas volume of slice n and ν is the volumetric flow-rate of the exhaust gases. Using the geometry of the TWC as defined in Fig. 1, and using the ideal gas law we can approximate Eq. (8) as

$$t_{r,n} = \frac{\mathrm{OFA} \cdot L_n \pi R^2}{\dot{m}_{\mathrm{exh}} R_{\mathrm{specific}} T_{n,m} p^{-1}_{\mathrm{TWC}}}\,, \tag{9}$$

where OFA is the open frontal area of the TWC, defined by

$$\mathrm{OFA} = (l_c - t_w)^2 l_c^{-2}\,. \tag{10}$$

In this paper we extend [1] to use physically meaningful SI units for all parameters. Here $\dot{m}_{\mathrm{exh}}$ is the exhaust massflow ($\mathrm{kg\,s^{-1}}$), P is the absolute pressure (Pa) in the TWC, which is typically close to the ambient pressure, and R_{specific} is the specific gas constant ($\mathrm{J\,K^{-1}\,kg^{-1}}$) for tailpipe ratio of N_2, CO_2, O_2, and H_2O that was experimentally measured for a hot TWC. This specific gas constant is used as it is easily determined and the remaining gases only marginally contribute to R_{specific}.

Ultimately, Eqs. (5) to (10) give a simple physics-based model of the most significant reactions that occur in the TWC that takes temperature, gas composition, and residence time into account.

Note that we have implicitly assumed that the incoming gas composition is time-invariant, as this significantly reduces

[1] [2] indicates in (4.68) that n' and m' correspond to hydrogen and carbon respectively. Consulting the previous derivations instead indicates that n' and m' correspond to carbon and hydrogen.

the number of required state variables. (Explicitly modeling a time-varying incoming gas concentration can require an additional $3N$ state variables, one for each emission species concentration in each slice.) This implies that the model is suited for quasi-static combustion engine operation, where the engine-out emission species and massflow varies slowly with respect to the residence time in the TWC. Fortunately, as the residence time in the entire TWC is fairly short (on the order of 0.05 – 0.1 s) [7, p. 64] we hypothesize that moderately-varying dynamic operation with transitions on the order of 0.5 – 1 s will show accuracy similar to that of constant engine operation.

By Eq. (7), we can compute the massflow emitted from cell n, m as

$$\dot{m}^{s,\text{out}}_{n,m} = \dot{m}^{s,\text{in}}_{n,m} e^{-k^s_{n,m} t_{r,n}}\,, \tag{11}$$

and, by the conservation of mass, the converted massflow is trivially

$$\dot{m}^{s,\text{conv}}_{n,m} = \dot{m}^{s,\text{in}}_{n,m} - \dot{m}^{s,\text{out}}_{n,m}\,. \tag{12}$$

This lets us model the tailpipe emissions of emission species s as the sum of the outputs from each individual cell in the last axial segment, i.e.

$$\dot{m}^s_{\text{tp}} = \sum_{m=1}^{M} \dot{m}^s_{m,N}\,. \tag{13}$$

For convenience, we also define the conversion efficiency of the entire TWC for a given emission species as

$$\eta^s = 1 - \frac{\dot{m}^s_{\text{tp}}}{\dot{m}^s_{\text{exh}}}\,, \tag{14}$$

which we can view as the proportion of emissions converted in the TWC.

We generate an estimate of the exothermic reaction power generated by the above reactions by computing the (temperature-dependent) heat of reaction for each mole of reactant species as

$$\mathrm{d}H_{\text{CO}} = H_{0,\text{CO}_2} - H_{0,\text{CO}} - 1/2 H_{0,\text{O}_2} \tag{15a}$$
$$\mathrm{d}H_{\text{H}_2} = H_{0,\text{H}_2\text{O}} - H_{0,\text{H}_2} - 1/2 H_{0,\text{O}_2} \tag{15b}$$
$$\mathrm{d}H_{\text{NO}} = 1/2 H_{0,\text{N}_2} + H_{0,\text{CO}_2} - H_{0,\text{NO}} - \mathrm{d}H_{\text{CO}} \tag{15c}$$
$$\mathrm{d}H_{\text{NO}_2} = 1/2 H_{0,\text{N}_2} + H_{0,\text{O}_2} - H_{0,\text{NO}_2} \tag{15d}$$
$$\mathrm{d}H_{\text{C}_3\text{H}_6} = 3H_{0,\text{CO}_2} + 3H_{0,\text{H}_2\text{O}} - H_{0,\text{C}_3\text{H}_6} - 9/2 H_{0,\text{O}_2} \tag{15e}$$
$$\mathrm{d}H_{\text{C}_3\text{H}_8} = 3H_{0,\text{CO}_2} + 4H_{0,\text{H}_2\text{O}} - H_{0,\text{C}_3\text{H}_8} - 5H_{0,\text{O}_2}\,. \tag{15f}$$

For brevity, we have not explicitly stated the temperature dependence of the above terms but include their temperature dependence in the numerical model. We use the Shomate equation and reference constants given by the NIST (available at https://webbook.nist.gov) to compute the numerical values of the above terms. Using the previous concentration ratios for the lumped terms gives the effective reaction power

$$\mathrm{d}H_{\text{NO}_x} = (99\mathrm{d}H_{\text{NO}} + \mathrm{d}H_{\text{NO}_2})/100 \tag{16a}$$
$$\mathrm{d}H_{\text{THC}} = (3\mathrm{d}H_{\text{C}_3\text{H}_6} + \mathrm{d}H_{\text{C}_3\text{H}_8})/4\,. \tag{16b}$$

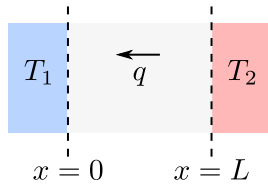


Fig. 4. Heat flux between two materials of known temperature.

By Eqs. (15a), (15b), (16a) and (16b) the total temperature-dependent heat of reaction generated in each cell is thus

$$P_{n,m} = \dot{m}^{\text{CO,conv}}_{n,m} \cdot \mathrm{d}H_{\text{CO}} + \dot{m}^{\text{H}_2\text{,conv}}_{n,m} \cdot \mathrm{d}H_{\text{H}_2} + \dot{m}^{\text{NO}_x\text{,conv}}_{n,m} \cdot \mathrm{d}H_{\text{NO}_x} + \dot{m}^{\text{THC,conv}}_{n,m} \cdot \mathrm{d}H_{\text{THC}}\,. \tag{17}$$

C. Temperature dynamics

We model the temperature dynamics using a heat balance ODE. Introducing the relative length-weighting matrix

$$\mathbf{W}_L = \begin{bmatrix} L_1/L & 0 & \dots & 0 \\ 0 & L_2/L & \dots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & L_N/L \end{bmatrix}, \tag{18}$$

we can define the heat-balance ODEs as

$$m_{\text{TWC}} \mathbf{W}_L c_p \frac{\mathrm{d}T_{\text{ctr}}}{\mathrm{d}t} = P_{\text{ctr}} \tag{19a}$$
$$m_{\text{TWC}} \mathbf{W}_L c_p \frac{\mathrm{d}T_{\text{per}}}{\mathrm{d}t} = P_{\text{per}}\,, \tag{19b}$$

where

$$P_{\text{ctr}} = P_{\text{ax}} - P_{\text{rad}} + P_{\text{con,ctr}} + P_{\text{exo,ctr}} \tag{20a}$$
$$P_{\text{per}} = P_{\text{ax}} + P_{\text{rad}} + P_{\text{con,per}} + P_{\text{exo,per}} - P_{\text{amb}}\,. \tag{20b}$$

Here, m_{TWC} is the mass of the TWC, c_p is its specific heat, P_{ctr} and P_{per} are $N \times 1$ vectors corresponding to the total power developed in the radial center and periphery respectively, and $\frac{\mathrm{d}T_{\text{ctr}}}{\mathrm{d}t}$ and $\frac{\mathrm{d}T_{\text{per}}}{\mathrm{d}t}$ are $N \times 1$ vectors representing the temperature derivative in the radial center and periphery. With these terms, we can construct the total state vector ODE as

$$\frac{\mathrm{d}T}{\mathrm{d}t} = \begin{bmatrix} \frac{\mathrm{d}T_{\text{ctr}}}{\mathrm{d}t} \\ N\text{mean}(\mathbf{W}_L(\frac{\mathrm{d}T_{\text{per}}}{\mathrm{d}t} - \frac{\mathrm{d}T_{\text{ctr}}}{\mathrm{d}t})) \end{bmatrix}, \tag{21}$$

where $\text{mean}([x_1, x_2, \dots, x_n])$ corresponds to the arithmetic mean of the elements in x, i.e. $1/n \sum_{i=1}^{n} x_i$. Note that we can interpret $N\text{mean}(\mathbf{W}_L(\frac{\mathrm{d}T_{\text{per}}}{\mathrm{d}t} - \frac{\mathrm{d}T_{\text{ctr}}}{\mathrm{d}t}))$ as corresponding to the average difference between radially central and peripheral powers, weighted by the relative length of each slice.

The power terms in the right hand side of Eqs. (20a) and (20b) are separated into axial, radial, convection, exothermic, and ambient loss terms respectively, which we will define below. This extends on our previous work [1], which used lumped-element parameters without an explicit power-balance formulation.

1) Axial conduction: The axial heat conduction power P_{ax} is modeled by Fourier's heat law [26]. Using conventional notation, the heat flux between two materials of constant temperature separated by a material of thickness l is in general

$$q = k\frac{T_2 - T_1}{l}, \tag{22}$$

where q is the heat flux ($\mathrm{W/m^2}$), T_1 and T_2 are two known temperatures (K), and k is the thermal conductivity of the material ($\mathrm{W\,m^{-1}\,K^{-1}}$), as illustrated in Fig. 4.

In this paper, we extend our previous model [1] by modeling the axial flux between successive axial slices as

$$q_{\mathrm{ax}} = k_{\mathrm{ax}}\begin{bmatrix} \frac{T_2 - T_1}{L_{1,2}} \\ \frac{T_3 - T_2}{L_{2,3}} \\ \vdots \\ \frac{T_N - T_{N-1}}{L_{N-1,N}} \end{bmatrix} \tag{23}$$

where k_{ax} is the axial thermal conductivity and $L_{n,n+1}$ is the distance between the center of axial slice n and $n+1$, as illustrated in Fig. 1a. Note that for N axial slices we thus have $N-1$ axial fluxes between slices.

We model the power associated with each flux term by scaling by the surface area of the solid mass of the TWC, i.e. $q_{\mathrm{ax}}(1-\mathrm{OFA})R^2\pi$. We can then model the total developed power in each axial slice due to conduction as the difference in incoming and outgoing power fluxes, i.e.

$$P_{\mathrm{ax}} = \begin{bmatrix} 0 \\ q_{\mathrm{ax}}(1-\mathrm{OFA})R^2\pi \end{bmatrix} - \begin{bmatrix} q_{\mathrm{ax}}(1-\mathrm{OFA})R^2\pi \\ 0 \end{bmatrix}. \tag{24}$$

2) Radial conduction: The radial heat conduction is modeled in a manner similar to the axial heat conduction. The radial flux is modeled as

$$q_{\mathrm{rad}} = k_{\mathrm{rad}}\frac{\Delta_T}{R/2}, \tag{25}$$

i.e. a temperature difference of Δ_T and separation of $R/2$. Approximating the surface area conducting heat as that of a cylinder with half the radius of the TWC and length equal to the TWC's length gives a developed radial conduction power of

$$P_{\mathrm{rad}} = q_{\mathrm{rad}}\pi RL\mathbf{1}, \tag{26}$$

where $\mathbf{1}$ is the ones vector of size $N \times 1$.

3) Convection: The convection heat powers $P_{\mathrm{con,ctr}}$ and $P_{\mathrm{con,per}}$ are modeled under the assumption that each cell is sufficiently long and narrow for the gas temperature to reach the cell temperature, i.e. the gas travels slowly enough to reach thermal equilibrium with the TWC walls. This was considered in our previous model [1], where we found that five axial slices was a suitable upper limit. As we now also allow for the slice lengths to vary, it is thus reasonable to require that $L_n \leq L/5\,\forall n$.

This gives the convection powers as

$$P_{\mathrm{con,ctr}} = \dot{m}_{\mathrm{exh}}c_{p,\mathrm{exh}}\begin{bmatrix} T_{\mathrm{exh}} - T_1 \\ T_1 - T_2 \\ \vdots \\ T_{N-1} - T_N \end{bmatrix} \tag{27}$$

$$P_{\mathrm{con,per}} = \dot{m}_{\mathrm{exh}}c_{p,\mathrm{exh}}\begin{bmatrix} T_{\mathrm{exh}} - (T_1 + \Delta_T) \\ (T_1 + \Delta_T) - (T_2 + \Delta_T) \\ \vdots \\ (T_{N-1} + \Delta_T) - (T_N + \Delta_T) \end{bmatrix} \tag{28}$$

where $\dot{m}_{\mathrm{exh}}$ is the exhaust massflow ($\mathrm{kg\,s^{-1}}$), $c_{p,\mathrm{exh}}$ is the constant-pressure specific heat of the exhaust gases ($\mathrm{J\,kg^{-1}\,K^{-1}}$), and T_{exh} is the temperature of the exhaust gas fed into the TWC (K).

4) Exothermic power: The exothermic power terms $P_{\mathrm{exo,ctr}}$ and $P_{\mathrm{exo,per}}$ are modeled by weighting the densely-resolved single-channel exothermic power into an effective central and peripheral powers. Here we use a linear weighting scheme as a first approximation, given as

$$P_{\mathrm{exotherm,ctr}} = \sum_{m=1}^{M} P_{n,m}\left(1 - \frac{m-1}{M-1}\right) \tag{29}$$

$$P_{\mathrm{exotherm,per}} = \sum_{m=1}^{M} P_{n,m}\frac{m-1}{M-1}. \tag{30}$$

Note that the term $\frac{m-1}{M-1}$ varies from 0 to 1 as m varies from 1 to M.

5) Ambient losses: The heat losses to the ambient environment are modeled as conductive, with a total flux of

$$q_{\mathrm{amb}} = k_{\mathrm{amb}}\begin{bmatrix} (T_1 + \Delta_T - T_{\mathrm{amb}})t_{\mathrm{amb}}^{-1} \\ (T_2 + \Delta_T - T_{\mathrm{amb}})t_{\mathrm{amb}}^{-1} \\ \vdots \\ (T_N + \Delta_T - T_{\mathrm{amb}})t_{\mathrm{amb}}^{-1}. \end{bmatrix} \tag{31}$$

Here, k_{amb} is the effective thermal conductivity of the insulating material and t_{amb} is its associated thickness. Modeling the exposed surface area as a cylinder with radius and length equal to the whole TWC (and thus neglecting heat loss through the circular ends of the cylinder) gives the power loss to the ambient environment as

$$P_{\mathrm{amb}} = q_{\mathrm{amb}}2\pi RL. \tag{32}$$

D. Two-monolith structure

In this paper we extend the model previously presented in [1] by studying a TWC with series-coupled monoliths. More specifically, we consider two physically separated TWC's, where the gas leaving the first is assumed to be completely mixed and then fed into the second, as illustrated in Fig. 5. We model this by assuming two completely independent sets of TWC parameters (as listed in Table II), and let the exhaust gas first travel from the engine through the first TWC. As a first approximation, the gas leaving the first TWC is assumed to be

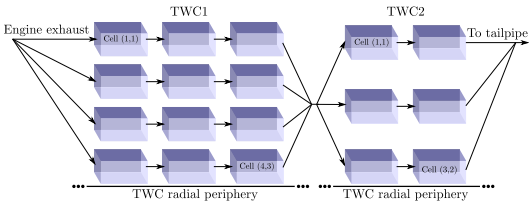


Fig. 5. The series-connected TWC is modeled as two independent TWC's with the gas exiting the first being mixed and fed into the second.

TABLE III
ENGINE PARAMETERS.

Engine type	VEA Gen I, VEP4 MP
Number of cylinders	Four, in-line
Displaced volume	1969 cc
Bore/Stroke	82 mm/93.2 mm
Compression ratio	10.8:1
Valve train	DOHC, 16 valves
Intake camshaft	Variable 0-48°CA advance
Exhaust camshaft	Variable 0-30°CA retard
Ignition system	DCI, standard J-gap spark plugs
Fuel system/Injection pressure	DI/200 bar
Fuel	Gasoline RON95 E10
Start of injection	308-340 CAbTDCf
Boosting system	Turbocharger
Rated power/Rated torque	187 kW/350 Nm
Stoichiometric air/fuel ratio	14.01:1

perfectly mixed (both with respect to temperature and emission species concentrations) and then fed through the second TWC, which then finally exits to the tailpipe.

Ideal mixing implies that the temperature of the gas feeding the second TWC is given as

$$T_{\mathrm{exh,TWC2}} = \sum_{m=1}^{M_{\mathrm{TWC1}}} \frac{m^2-(m-1)^2}{M_{\mathrm{TWC1}}^2} T_{N,m,\mathrm{TWC1}}\,, \qquad (33)$$

where i.e. the gas leaving the first TWC (TWC1) is combined and scaled by the its relative flow rate. Trivially, we also have that the emission species concentration entering TWC2 is

$$\dot{m}_{1,m,\mathrm{TWC2}}^{s,\mathrm{in}} = \frac{m^2-(m-1)^2}{M_{\mathrm{TWC2}}^2} \dot{m}_{\mathrm{tp,TWC1}}^{s}\,, \qquad (34)$$

i.e. we use the same weighting scheme previously defined in Eq. (2a).

III. Experimental setup

As in our previous study in [1], the experimental setup consisted of a production Volvo Cars two liter in-line four-cylinder direct injected spark ignited turbocharged engine rated for 187 kW and 350 Nm, as listed in Table III. The engine was connected to an electrical dynamometer that regulated the engine speed and measured the generated torque. A prototyping ECU was used to sample and change engine parameters. The TWC was close-coupled to the turbocharger outlet.

The TWC was instrumented with 28 thermocouples (14 in each monolith) and three exhaust gas sampling locations. The thermocouples, 0.5 mm type-K with a grounded hot junction[2], were inserted into a TWC channel and held by friction. A close-up of the instrumented TWC is shown in Fig. 6, which also shows the direction of gas flow through the two monoliths. A more detailed drawing of the TWC construction and the thermocouple locations is shown in Fig. 7.

Fig. 6. The instrumented TWC, with exposed ceramic (left) and metallic (right) sections. The exhaust gas flow is shown with the highlighted arrow. The cover to the left is open here for illustrative purposes and tightly connected to the main body during operation.

Several thermocouples failed during the experimental campaign. We believe this to be due to the combination of fairly sharply bending the thermocouples in order to reach the required monolith channels and a high level of vibration in an initial TWC mounting fixture. Fortunately, the most critical sensors (in slices 1, 3, 4, and 6) were fully functional and only sensors in slices 2 and 5 were damaged. We excluded data from the damaged sensors in our analysis.

An auxiliary air feed was added to the exhaust manifold, which allowed for flushing the entire exhaust subsystem with room-temperature air. By running the engine in fuel-cut mode (i.e. disabling fuel injection and motoring the engine with the dynamometer) and injecting auxiliary air into the exhaust manifold the exhaust aftertreatment system could be cooled to under 100°C in approximately 5 minutes. The auxiliary airflow was set to 1000 $\mathrm{L\,min^{-1}}$ STP, which was the maximum flow-rate supported by the mass flow controller. The auxiliary airflow was completely disabled during normal operation (i.e. when fuel injection was enabled). A photograph of the experimental setup is shown in Fig. 8, where the engine is visible and the TWC is highlighted. A schematic representation of the experimental set-up and gas flows is shown in Fig. 9, which also highlights the auxiliary air feed, exhaust gas flow, and gas sampling locations.

The emission sampling points after TWC1 and after TWC2

[2]Manufacturer: RS PRO, model number: 847-1110

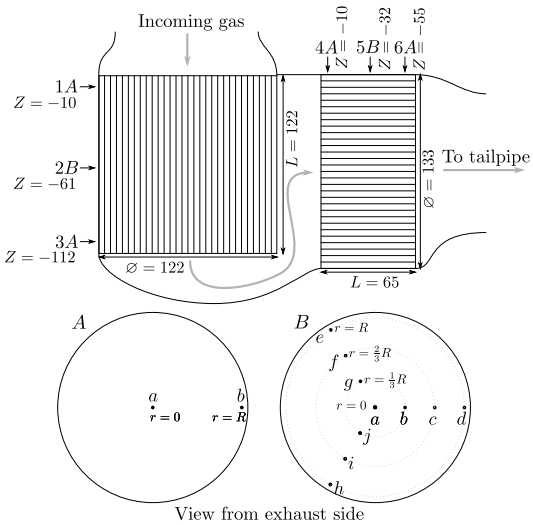


Fig. 7. Detailed cross-section of the TWC structure and thermocouple locations. A total of six axial positions were measured (three per monolith) as indicated. Each axial position was sampled in one of two configurations, A and B, as indicated. Each specific sensor is referenced as TS`XYZ`, where `X` is the slice number (1-6), `Y` is the slice type (A/B), and `Z` is the sensor position (A/B or A-J).

measured the emissions exiting the radially central channel of each respective TWC. As both TWCs at times displayed a large radial temperature differential, this implies that the average emissions leaving each TWC can be significantly different from the emissions measured at the radial center.

A. Data Acquisition

Emissions signals from instruments, fuel consumption, and dynamometer readings were sampled with a National Instruments DAQ and an associated LabVIEW program. Engine temperatures, pressures, and the air-fuel ratio was sampled using acquisition units over a CAN ETAS module. All thermocouples were of type K. Fuel massflow was measured with a Coriolis meter. All parameters were sampled at a 10 Hz rate.

Exhaust gases were sampled from three different locations (as illustrated in Fig. 9). All sampled gases were extracted with a heated hose (180 °C), followed by a heated conditioning unit (190 °C) with a heated filter and pump. Emissions concentrations were measured with separate instruments. THC emissions were measured using a flame ionization detector, NO_x using a chemiluminescence analyzer, and CO using a non-dispersive infrared detector. The propagation delay and axial dispersion in hoses and instruments was identified by recording the measured engine-out emissions during the transition from fuel-cut operation to normal operation (as will be described in Section III-B). With this data we compensated for the propagation delay and applied a first-order high-pass filter to mitigate some of the axial dispersion. This compensation was applied to the remaining two sampling locations, allowing for

Fig. 8. The experimental setup, with the TWC housing and heat shield highlighted. The turbocharger is just visible to the right of the TWC. The exhaust from the TWC is fed down through the visible ducting.

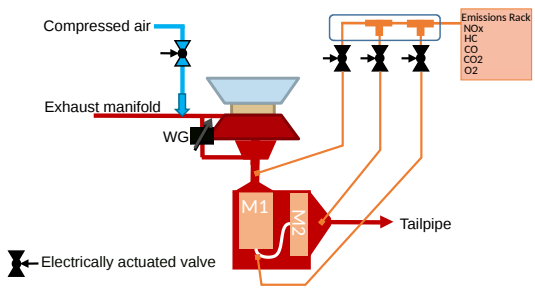


Fig. 9. Schematic representation of the experimental setup, with exhaust gas passing through the turbocharger, through monolith 1 (M1), monolith 2 (M2), and finally exiting to the tailpipe. Exhaust gases were sampled at three locations; directly after the turbocharger, between the monoliths, and after monolith 2.

TABLE IV
TESTED STEADY-STATE ENGINE OPERATING POINTS. THE LISTED SPEEDS, LOADS, AND SPARK ANGLES (SA) ARE SETPOINT VALUES.

Speed [RPM]	BMEP [bar]	SA [CAbTDC]
1000	2	[12, 14, 16, 18, 20, 22, 24]
1000	5	[6, 8, 10, 12, 14, 16, 18]
1000	8	[-2, 2, 4]
1500	2	[12, 14, 16, 18, 20, 21, 22, 24]
1500	5	[6, 8, 10, 12, 14, 16, 18]
1500	8	[-2, 2, 4, 6, 8, 10]
2000	2	[16, 18, 20, 22, 24, 26, 28]
2000	5	[8, 10, 12, 14, 16, 18, 20]
2000	8	[2, 4, 6, 8, 10, 12]
2000	10	[-4, 4]
2000	12	[-2, 2, 4]
2000	14	[-2, 2]
2500	2	[14, 16, 18, 20, 22, 24, 26]
2500	5	[12, 14, 16, 18, 20, 22, 24]
2500	8	[8, 10, 12, 14, 16]
2500	13	[4, 6]
3000	8	[8, 10, 12, 14, 16, 18]

studying transient emission concentration changes moderately well using an instrument rack primarily intended for steady-state analysis.

As our experimental set-up only allowed for measuring the emissions at one location at any given time it was crucial for the engine-out emissions to be consistent between different runs. Due to this we chose to run the combustion engine in stationary operation, with the goal of maximizing the exhaust gas composition repeatability. We hypothesize that using hardware that measures the emission species at every sample point simultaneously would allow for non-stationary engine operating during cold-start tests.

B. Measurement procedure

The emission measurement equipment was calibrated before measurements using calibration gases and the engine was heated to its working temperature by operating it at a moderate load until the coolant reached its working temperature. The engine was kept warm during the entire test procedure, implying that the cold-starts studied in this paper refer to the case where the TWC is initially cold while the engine is at operating temperature. Furthermore, the TWC was instrumented with heated lambda sensors, and the engine operated with the conventional closed-loop lambda control scheme during TWC cold-start tests.

1) Steady-state analysis: The goal of this test was to identify the steady-state engine-out emissions and the associated steady-state radial temperature distribution in the TWC. This was performed by statically running the engine at a given speed and BMEP and sweeping the spark angle from the default value and retarding it to the edge of combustion stability. Table IV lists the tested speeds, BMEPs, and spark angles tested.

2) TWC cold-start characterization: The goal of this test was to characterize the cold-start parameters of the two TWC's. The combustion engine was kept at a warm and constant temperature throughout these tests, i.e. we evaluated the behavior of a cold TWC and warm engine. We performed this experiment by

- disabling fuel injection (i.e. motoring the engine with the dynamometer) and opening the auxiliary air valve until all the TWC thermocouples reported a temperature of under 100 °C,
- first closing the auxiliary air valve, and then immediately enabling ordinary fuel injection until the TWC reached near-equilibrium temperature and emissions.

TABLE V
TWC COLD-START LOAD POINTS. SA SET TO ECU DEFAULT VALUE.

Index	Speed [rpm]	BMEP [bar]	SA [CAbTDC]
1	1000	2	24
2	1000	5	18
3	1500	5	18
4	1500	2	24
5	2000	2	28
6	2000	5	20
7	3000	8	18
8	1000	8	4
9	1500	8	10
10	2000	8	12

This procedure was repeated for each emission sample point for each of the load points listed in Table V. These test points were chosen so that some generated a heating profile that gave a long time to light-off (primarily load points 1–5), while others reached light-off more quickly (load points 6–10). The low-load points gave a longer data stream and reduced the relative error due to axial dispersion in the emission sampling lines. The remaining load points (points 6–10) were more representative of a conventional heating strategy, where light-off is reached more quickly. Furthermore, the load points were characterized by BMEP rather than IMEP due to limitations in the measurement equipment. The engine was kept at a warm and constant temperature to ensure that the exhaust gas composition was not influenced by changes in friction from one load point to the next. It is plausible that regulating for a given IMEP would give an exhaust gas profile that is less sensitive to engine temperature.

IV. EXPERIMENTAL RESULTS

A. Steady-state

The steady-state experimental results were used to generate a table of the mean equilibrium TWC temperatures, engine-out emissions, exhaust massflow, and engine BSFC for each of the load points listed in Table IV. Representative data is shown in Table VI for TWC1. The parameters TT_0, $TT_{R/3}$, $TT_{2R/3}$, and TT_R correspond to the mean thermocouple temperature for thermocouples in slice 2 at radius $r = 0$, $r = R/3$, $r = 2R/3$, $r = R$ respectively (i.e. $TT_{R/3}$ is the mean of TS2BB, TS2BG, TS2BJ). Importantly, we also also use this experimental data to determine the radial interpolation profile outlined in Section II-A, i.e. for each load point we generate an associated interpolated radial temperature profile. More

TABLE VI
TWC1 STEADY-STATE DATA FOR REPRESENTATIVE LOAD POINTS (SPEED, BMEP, AND SPARK ADVANCE). FIGURES SHOWN WITH MEASURED VALUES.

Speed [RPM]	BMEP [bar]	SA [CAbTDC]	$\dot{m}_{\text{exh}}$ [g/s]	BSFC [g/kWh]	CO [ppm]	NO_x [ppm]	THC [ppm]	TT_0 [°C]	$TT_{R/3}$ [°C]	$TT_{2R/3}$ [°C]	TT_R [°C]
1010	6.9	10	13	257	3580	2500	427	568	567	566	537
3000	7.9	18	41	259	7620	2550	284	839	839	832	813
1000	1.8	12	5	410	6090	391	548	620	615	609	569
1000	2.0	24	5	384	6650	1150	684	498	497	491	442
1000	4.7	6	9	299	5950	1200	459	525	526	519	472
1000	5.0	18	9	280	6550	2070	533	552	552	545	503
1570	4.8	6	16	313	7720	533	345	726	725	719	684
1570	5.1	19	15	279	9070	1350	459	668	667	661	627
2000	5.0	8	21	316	6730	279	325	715	716	710	674
2000	5.2	21	19	274	9150	686	442	696	697	693	664
3000	8.1	8	46	277	7060	1610	195	893	893	884	864
3010	8.2	19	43	258	7560	2590	328	803	805	797	773

specifically, for each load point we assume an interpolation function of form

$$f_{\text{interp}} = \hat{T}(t', r) \tag{35a}$$

where

$$t' = \underset{t}{\operatorname{argmin}} \sum_{n=0}^{3} |\hat{T}(t, nR/3) - TT_{nR/3}|\,, \tag{35b}$$

i.e. we let f_{interp} be the optimal solution in the one-norm sense that minimizes the deviation between the measured temperatures and the solution to the heat equation over all time t'. The one-norm is consistently used in this paper in an effort to reduce the effect of outliers. Measured temperatures and the associated interpolation function is shown Fig. 10 for two representative load points.

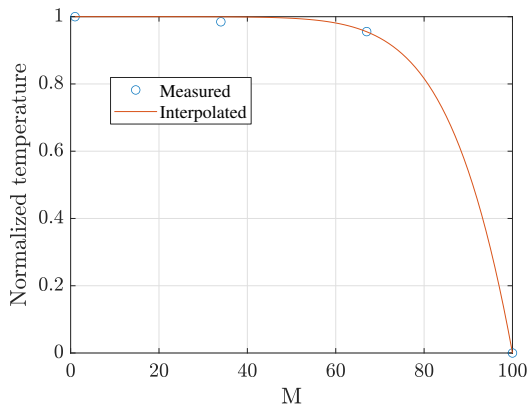
B. Cold-start

Figure 11 shows a representative cold-start temperature evolution, here for 1000 RPM and 2 bar BMEP. We show this specific load point as it gives the longest system dynamics. We can draw several useful conclusions from this test;

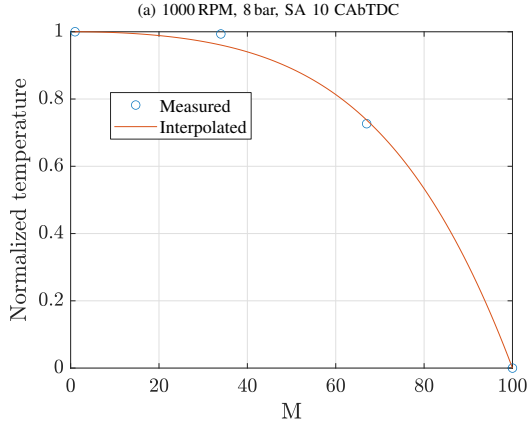
- The radial temperature distribution is significant throughout both TWC's, with a temperature difference between the radial center and periphery of up to 100 °C near light-off.
- The first TWC shows no major azimuth temperature variation.
- The second TWC does show variations along the azimuth, with the hottest regions nearer the bottom section of the cross-section (see Fig. 12). We hypothesize that this is due to increased massflow near the lower sections, as the sharp bend in the TWC housing causes an uneven pressure distribution across the inlet to the second TWC.
- TWC2 shows less pronounced axial temperature variations when compared to TWC1. This could plausibly be due to the length of TWC2, which is only half of TWC1.

These results are consistent for the other load points, which display similar results.

Based on these results we have chosen to model TWC1 as consisting of three axial slices, while TWC2 is modeled with



(a) 1000 RPM, 8 bar, SA 10 CAbTDC



(b) 3000 RPM, 8 bar, SA 18 CAbTDC

Fig. 10. Measured normalized radial temperature distribution and least-squares interpolation for $M = 100$ at two representative load points.

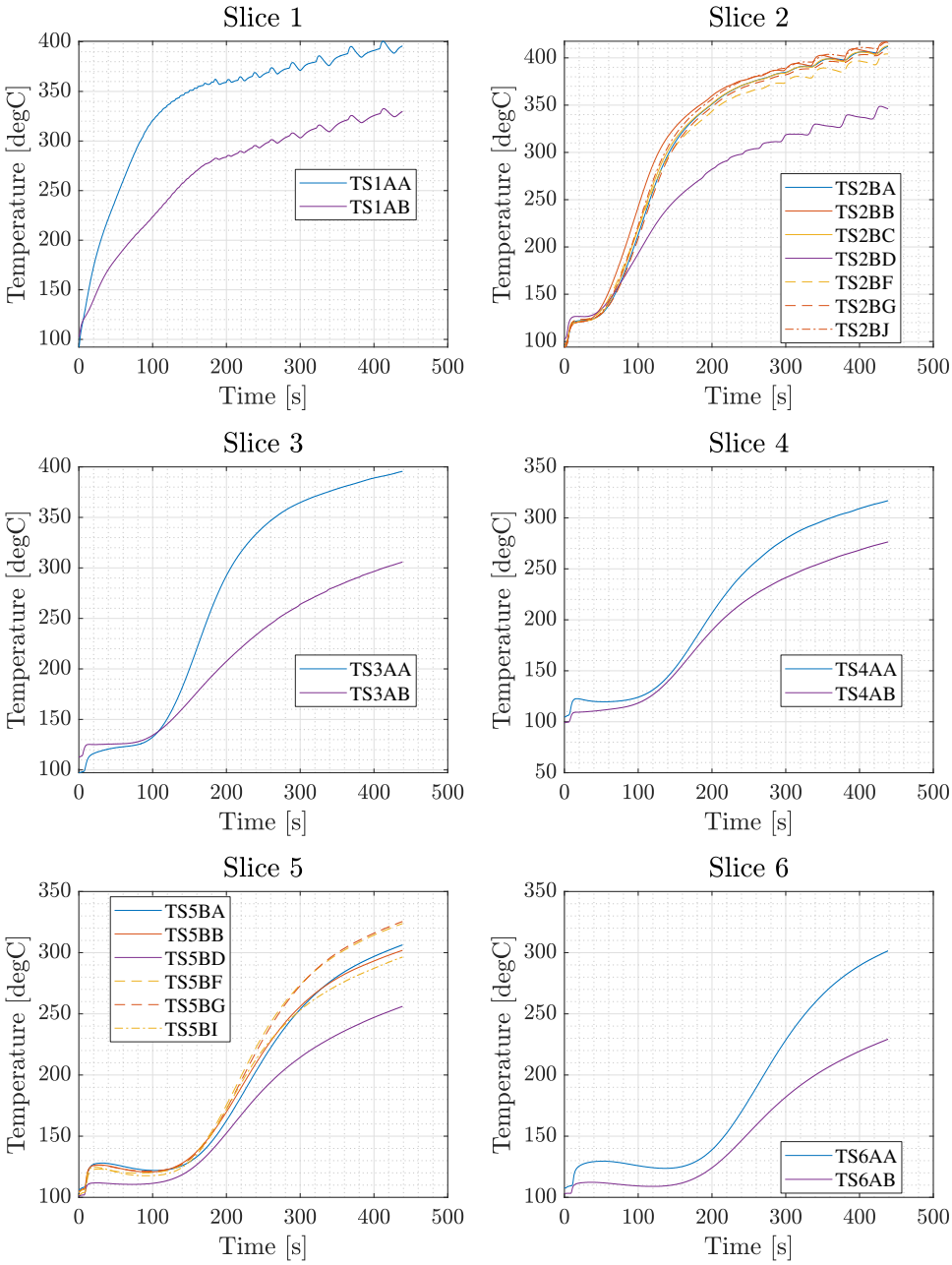


Fig. 11. Representative cold-start temperature evolution. Here shown for load point 1 in Table V (1000 RPM, 2 bar BMEP). Thermocouple locations listed in Fig. 7. Damaged thermocouples (TS2BE, TS2BH, TS2BI, TS5BC, TS5BE, TS5BH, TS5BJ) excluded from plots.

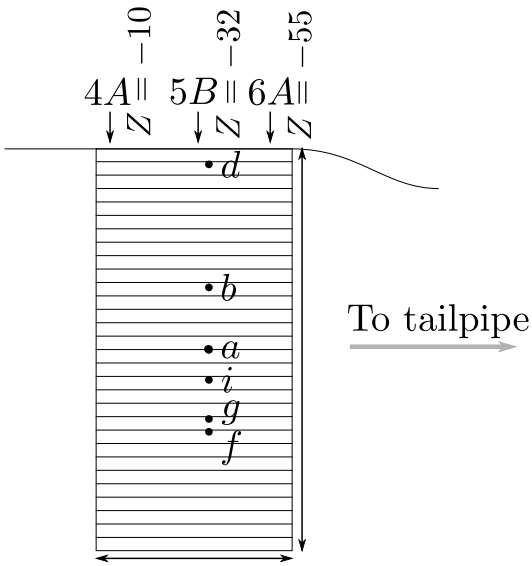


Fig. 12. Projection of TS5Bx thermocouples onto the plane. Approximate gas flow shown by gray arrows. The thermocouple closest to the bottom (TS5BF) reaches the highest temperature in Fig. 3, while the thermocouple closest to the top (TS5BD) reaches the lowest.

a single axial slice. This gives a total of 3+1 state variables for the TWC1 and 1+1 state variables for TWC2, i.e. a total of six state variables. Though additional slices would have the benefit of improving accuracy, the dynamic-programming based optimal control method we use in Section V is suited for no more than 4–6 state variables. We have chosen to allocate more slices to the first TWC as it displays the most significant axial temperature variations.

C. Model tuning

The primary goal of the experimental work is to generate measurement data that is used to tune the TWC model. The tuning process was divided into two distinct sections where we first tuned the reaction rate parameters, and afterwards tuned the temperature dynamics parameters. The problem was divided into two sections to reduce the number of degrees of freedom in each optimization step. Furthermore, as the temperature dynamics depends on the exothermic power, it is prudent to first determine the reaction rate parameters.

Of the 10 cold-start simulations listed in Table V, we designated operating points $i_{\text{train}} = [1, 3, 5, 7, 9]$ as a training set, and $i_{\text{valid}} = [2, 4, 6, 8, 10]$ as a validation set. All model tuning was done solely using i_{train}, allowing us to later study the model's accuracy by studying results of applying the model to the validation set's operating points.

D. Reaction rate parameters

Here, we consider tuning the per-species reaction-rate parameters A^s and E_a^s, giving a total of six parameters to tune per TWC. With the experimental setup as described in Section III, the gas composition entering and leaving TWC1's radially central channel is well-measured. However, the gas composition entering TWC2 is not as well characterized, as the gas composition leaving TWC1 is inhomogeneous (due to the large radial temperature gradient) and partially mixed before entering TWC2. Due to this, we have chosen to first tune the reaction rate parameters for TWC1, and afterwards make use of the identical precious metal composition of TWC1 and TWC2 (which differ only in their washcoat thickness and loading). This allows us to estimate TWC1's reaction rate parameters using experimental data and then compute the equivalent parameters for TWC2.

With respect to TWC1, for a given set of reaction-rate parameters we used the (measured) temperature evolution of each axial slice to simulate the outgoing emission concentration. More specifically, we let the measured state evolution for each operating point be

$$T_{\text{meas,TWC1}} = [T_1, T_2, T_3, \Delta_T] \tag{36a}$$

where

$$T_1 = \text{TS1AA} \tag{36b}$$
$$T_2 = \text{TS2BA} \tag{36c}$$
$$T_3 = \text{TS3AA} \tag{36d}$$
$$\Delta_T = \text{mean}([\text{TS1AB} - \text{TS1AA}, \text{TS2BD} - \text{TS2BA}, \text{TS3AB} - \text{TS3AA}]), \tag{36e}$$

i.e. the measured radially central temperatures and the mean difference between the radial center and periphery respectively. This gives a state vector time-evolution $T(k)_{\text{meas,TWC1}}$, where $k = 0, 1, 2, \ldots$ indicates the time-sample of the state vector, sampled at a rate of 10 Hz. In this and later stages we simulated 100 radial channels. We chose to simulate a relatively large number of channels as the kinetics submodel was implemented in a semi-parallel manner that did not require significantly longer to evaluate than for instance 10 channels[3].

We started the tuning process by first determining an initial guess, where we selected E_a^s as determined by [18] and A^s was selected to give a light-off temperature near the experimentally measured behavior. We then used MATLAB's `patternsearch` utility (a zero'th order gradient-descent optimization method) to minimize the 1-norm penalty

$$J^* = \min_{E^s_{a,\text{TWC1}}, A^s_{\text{TWC1}}} \sum_{i \in i_{\text{train}}} \sum_{k=1}^{K} |\dot{m}(k)^{s,\text{out}}_{3,1,\text{TWC1},i} - \dot{m}(k)^{s}_{\text{meas,TWC1out,i}}| \tag{37}$$

[3]This was in part due to the nature of our model implementation in MATLAB. As the chemical kinetics submodel consists of only basic arithmetic operations we could implement the model in a manner that effectively makes use of MATLAB's numerically efficient matrix operations. This gave a model implementation whose execution time was dominated by the the number of function calls, i.e. the sample rate and simulation time, rather than the number of radial channels.

TABLE VII
TUNED REACTION RATE PARAMETERS.

Parameter	Value	Unit
$E_{a,\mathrm{TWC1,2}}^{\mathrm{CO}}$	$84.0 \cdot 10^3$	$\mathrm{J\,mol^{-1}}$
$E_{a,\mathrm{TWC1,2}}^{\mathrm{NO}_x}$	$82.1 \cdot 10^3$	$\mathrm{J\,mol^{-1}}$
$E_{a,\mathrm{TWC1,2}}^{\mathrm{THC}}$	$51.0 \cdot 10^3$	$\mathrm{J\,mol^{-1}}$
$A_{\mathrm{TWC1}}^{\mathrm{CO}}$	$59.5 \cdot 10^9$	-
$A_{\mathrm{TWC1}}^{\mathrm{NO}_x}$	$27.6 \cdot 10^6$	-
$A_{\mathrm{TWC1}}^{\mathrm{THC}}$	$11.2 \cdot 10^9$	-
$A_{\mathrm{TWC2}}^{\mathrm{CO}}$	$23.8 \cdot 10^9$	-
$A_{\mathrm{TWC2}}^{\mathrm{NO}_x}$	$16.1 \cdot 10^6$	-
$A_{\mathrm{TWC2}}^{\mathrm{THC}}$	$5.63 \cdot 10^9$	-

using the `MADSPositiveBasis2N` polling method and with the `UseCompletePoll` flag set. Here, $\dot{m}(k)_{3,1,\mathrm{TWC1},i}^{s,\mathrm{out}}$ is the simulated concentration of emission species s emitted by the third (i.e. last) axial slice at the radial center at sample k and operating point i, $\dot{m}(k)_{\mathrm{meas,TWC1out,i}}^{s}$ is the measured emissions at the same time instance and operating point, and i iterates over the operating points in the training dataset.

With the kinetics parameters for TWC1 determined, we estimate the parameters for TWC2 by assuming an identical activation energy and with the pre-exponential term scaled by the amount of catalytically active material. This gives the estimates

$$E_{a,\mathrm{TWC2}}^{s} = E_{a,\mathrm{TWC1}}^{s} \tag{38a}$$

$$A_{\mathrm{TWC2}}^{s} = \frac{t_{\mathrm{wash,TWC2}} \cdot w_{\mathrm{TWC2}}}{t_{\mathrm{wash,TWC1}} \cdot w_{\mathrm{TWC1}}} A_{\mathrm{TWC1}}^{s}\,, \tag{38b}$$

where t_{wash} and w are the washcoat thickness and loading respectively for each TWC.

Table VII lists the identified parameters for both TWCs, which are on the same order of magnitude as literature suggests [18, 30]. Note that the `patternsearch` method is similar to gradient-descent methods, and as the problem is non-convex is therefore not guaranteed to return a globally optimal solution.

Time-resolved plots of the measured and simulated emissions profiles and the simulated conversion efficiencies are shown in Fig. 13 for the lowest-load operating point. The simulated outgoing emissions are shown for the simulated radially central channel, which corresponds to the location of the measured emissions. The most significant deviations are seen at $t \in [0, 30]$ for NO_x and at $t \in [0, 50]$ for THC. We hypothesize that the former is due to adsorption and the latter due to the poor transient response of the measurement equipment.

E. Temperature dynamics parameters

With the reaction rate parameters determined we consider the temperature dynamics parameters k_{ax}, k_{ra}, k_{amb}, c_p, and L_n separately for each TWC. Note that as TWC2 is modeled with only a single axial slice, there is no modeled axial conduction and k_{ax} and L_n therefore have no meaning. We tuned each TWC to the measured data by assigning the initial state to the measured temperature at the start of the cold-start test and

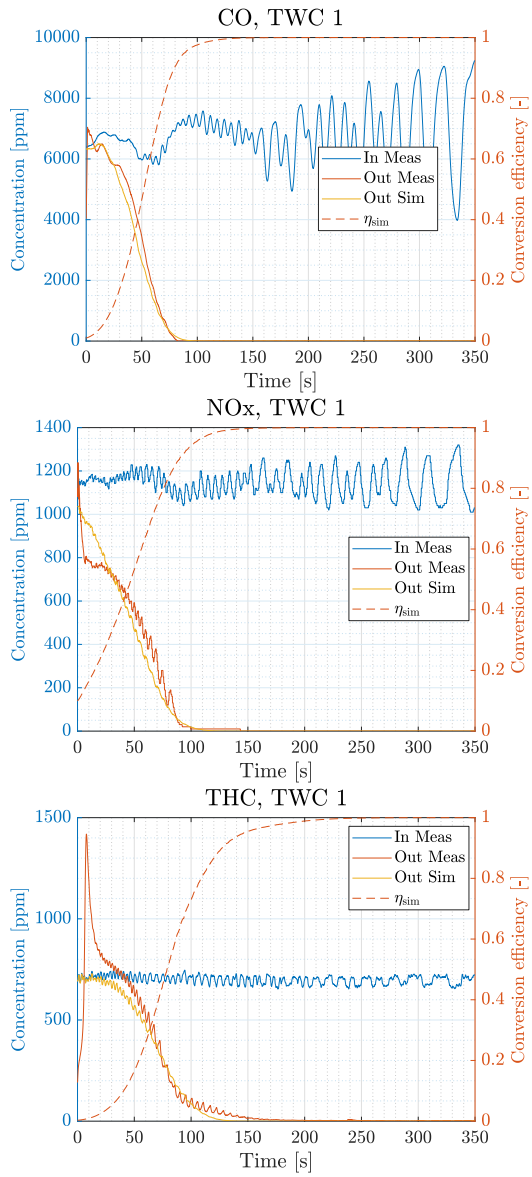


Fig. 13. Emissions profiles for 1000 rpm, 2 bar BMEP load point. See Fig. 11 for the associated measured temperature evolution and Fig. 14a for the associated state evolution.

TABLE VIII
TUNED AND FIXED TEMPERATURE DYNAMICS PARAMETERS.

Tuned Parameter	Value	Unit
$k_{\mathrm{ax,TWC1}}$	319	$\mathrm{W\,m^{-1}\,K^{-1}}$
$k_{\mathrm{ra,TWC1}}$	46.6	$\mathrm{W\,m^{-1}\,K^{-1}}$
$k_{\mathrm{amb,TWC1}}$	0.421	$\mathrm{W\,m^{-1}\,K^{-1}}$
$c_{p,\mathrm{TWC1}}$	2318	$\mathrm{J\,K^{-1}\,kg^{-1}}$
$L_{1,\mathrm{TWC1}}$	$32.1 \cdot 10^{-3}$	m
$L_{2,\mathrm{TWC1}}$	$48.2 \cdot 10^{-3}$	m
$L_{3,\mathrm{TWC1}}$	$41.8 \cdot 10^{-3}$	m
$k_{\mathrm{ra,TWC2}}$	4.53	$\mathrm{W\,m^{-1}\,K^{-1}}$
$k_{\mathrm{amb,TWC2}}$	0.602	$\mathrm{W\,m^{-1}\,K^{-1}}$
$c_{p,\mathrm{TWC2}}$	2360	$\mathrm{J\,K^{-1}\,kg^{-1}}$

Fixed Parameter	Value	Unit
$\mathrm{OFA_{TWC1}}$	0.935	-
$\mathrm{OFA_{TWC2}}$	0.846	-
m_{TWC1}	0.418	kg
m_{TWC2}	0.248	kg
$c_{p,\mathrm{exh}}$	1050	$\mathrm{J\,K^{-1}\,kg^{-1}}$
$t_{\mathrm{amb,TWC1}}$	$10 \cdot 10^{-3}$	m
$t_{\mathrm{amb,TWC2}}$	$10 \cdot 10^{-3}$	m
T_{amb}	25	°C

then applying a 1-norm penalty to the deviation between the simulated and measured states, i.e.

$$J^* = \min_{k_{\mathrm{ax}},k_{\mathrm{ra}},k_{\mathrm{amb}},c_p,L_n} \sum_{i\in i_{\mathrm{train}}} \sum_{k=1}^{K} |T(k)_{\mathrm{meas},i} - T(k)_{\mathrm{sim},i}| \tag{39}$$

where T_{sim} is the simulated state evolution generated by solving Eq. (21) using an explicit fourth-order Runge-Kutta solver with a fixed time-step of 0.1 s, $T(0)_{\mathrm{sim}}$ is initialized as $T(0)_{\mathrm{sim}} = T(0)_{\mathrm{meas}}$, T_{meas} is defined by Eq. (36) for TWC1, and for TWC2 defined as

$$T_{\mathrm{meas}} = [T_1, \Delta_T] \tag{40}$$

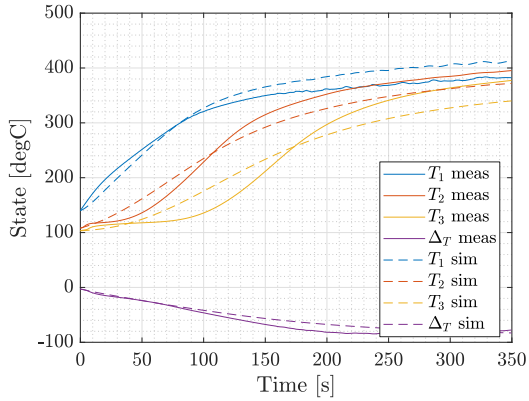
where

$$T_1 = \mathrm{mean}([\mathrm{TS4AA}, \mathrm{TS5BA}, \mathrm{TS6AA}]) \tag{41}$$

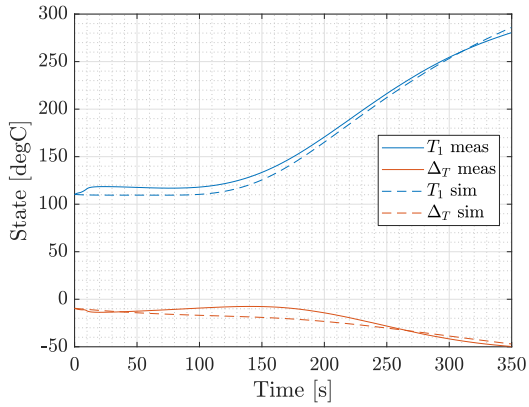
$$\Delta_T = \mathrm{mean}([\mathrm{TS4AB} - \mathrm{TS4AA}, \mathrm{TS5BD} - \mathrm{TS5BA}, \mathrm{TS6AB} - \mathrm{TS6AA}]). \tag{42}$$

As in the reaction rate parameters, we used the `patternsearch` method to determine the optimal parameters. We supplied the initial guess for k_{ax}, k_{ra}, k_{amb}, and c_p by setting them to the values specified by the TWC manufacturer, and L_n to geometrically ideal values, where we assume the thermocouples are placed in the center of each slice. Referencing Fig. 7 gives the initial guess $L_1 = 20 \cdot 10^{-3}$ m, $L_2 = 102 \cdot 10^{-3}$ m, and $L_3 = 20 \cdot 10^{-3}$ m. Table VIII lists the parameter values found after tuning, as well as the known (i.e. assigned) fixed model parameters.

An illustration of a representative temperature evolution is shown in Fig. 14a and Fig. 14b. Though the first slice of TWC1 and TWC2 capture the measured temperature evolution well, the second and third slices of TWC1 do not capture the characteristic delay shown in the measured data. We hypothesize



(a) Measured and simulated temperature evolution for TWC1 with 3 axial slices.



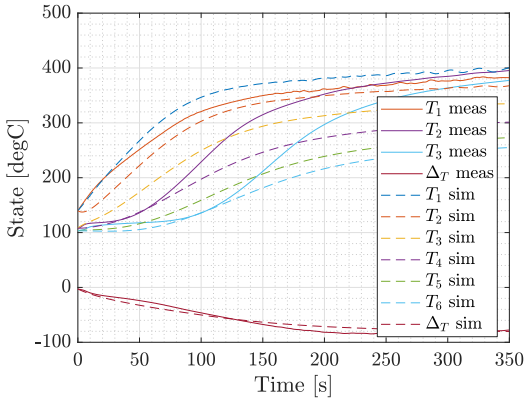
(b) Measured and simulated temperature evolution for TWC2.

Fig. 14. Temperature evolution for 1000 rpm, 2 bar BMEP operating point.

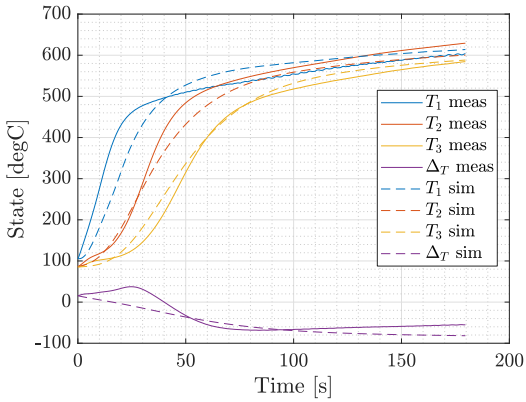
that this is independent of the chosen tuning parameters and an inherent limitation of our modeling assumption of a small number of axial slices. More specifically, as a general discrete-time delay of n samples (trivially) requires storing the values of the n samples, our shown model can only represent a true delay of three samples, i.e. an insignificant 0.3 seconds, before the last axial segment starts displaying a positive temperature derivative. This can be alleviated somewhat either by increasing the number of axial segments or by considering cold-starts with a less prominent delay, as is shown in Fig. 15.

F. Cumulative emissions accuracy

With the model tuned, we will now turn to quantitatively evaluating the TWC model's accuracy. Here, we consider the relative difference between the cumulative measured and simulated emissions (i.e. cold-start "bag emissions") for each emission species and TWC. Using the notation where Δ_i^s



(a) Measured and simulated temperature evolution at the 1000 rpm, 2 bar BMEP operating point, TWC1 model modified to use 6 axial slices. Other model parameters unchanged.



(b) Measured and simulated temperature evolution for TWC1 at the 1500 rpm, 8 bar BMEP load point.

Fig. 15. Increasing the number of resolved slices (Fig. 15a) and/or heating the TWC more quickly (Fig. 15b) reduces the modeling error caused by the limited ability to represent a delay.

corresponds to the i'th TWC for emission species s gives

$$\Delta_1^s = \frac{\sum_{k=0}^{K} \dot{m}(k)^{s,\text{out}}_{3,1,\text{TWC1}}}{\sum_{k=0}^{K} \dot{m}(k)^{s}_{\text{meas,TWC1out}}} - 1 \tag{43a}$$

$$\Delta_2^s = \frac{\sum_{k=0}^{K} \dot{m}(k)^{s,\text{out}}_{1,1,\text{TWC2}}}{\sum_{k=0}^{K} \dot{m}(k)^{s}_{\text{meas,TWC2out}}} - 1\,. \tag{43b}$$

An illustration of the simulated and measured cold-start emissions is shown in Fig. 16 for the 1000 rpm, 2 bar BMEP load point, along with the associated cumulative simulation error. The figure indicates that one significant contribution to the cumulative error is due to inaccuracies in the measurement equipment (which is primarily designed for analyzing stationary operation). This is most clearly seen during the first 20 seconds of operation for THC, where the measured emissions are significantly larger than than the engine-out emissions.

TABLE IX
RELATIVE MODEL ACCURACY FOR TRAINING (UPPER HALF) AND VALIDATION (LOWER HALF) LOAD POINTS.

Speed [RPM]	BMEP [bar]	Δ_2^{CO} [-]	$\Delta_2^{\text{NO}_x}$ [-]	Δ_2^{THC} [-]
997	1.99	-11.6%	-11.6%	-39.6%
1500	4.84	+106.5%	-20.8%	+66.0%
2000	2.07	+44.8%	+34.2%	-7.8%
3000	8.17	+39.5%	+38.9%	-18.2%
1500	8.09	+72.7%	+56.4%	-10.7%
998	4.93	+2.0%	-19.5%	-23.4%
1500	2.07	+38.6%	+41.7%	-10.8%
2010	4.93	+73.3%	+75.8%	+13.1%
999	7.96	+81.2%	+65.8%	-13.8%
2000	8.08	+63.0%	+48.6%	-26.7%

It is possible that the NO_x emissions are also incorrectly measured, as the measured emissions are only half of the engine-out emissions after 3-4 seconds (while light-off occurs after approximately 60 seconds at this load point). However, it is also plausible that the low NO_x emissions are correctly measured and this anomaly is instead due to unmodeled adsorption in the TWC.

A table listing the relative cumulative tailpipe error is shown in Table IX for each load point. The validation dataset (the lower half) displays an accuracy comparable to the training dataset (the upper half), indicating that the model is not overfitted. Furthermore, though there is a significant degree of variability between the measured and predicted cumulative emissions this is only somewhat worse than a significantly more complex model [18] which displays a typical cumulative error on the order of $\pm 20\%$ to $\pm 50\%$. Furthermore, we hypothesize that the most significant outliers (e.g. the 1500 rpm, 2 bar load point) are to some extent due to process variability and/or measurement error. Consulting the time evolution for this load point (Fig. 17, here shown for CO emissions) indicates that this can be a factor, as the majority of the modeling error arises after 10 seconds when the engine-out emissions significantly increase but a similar increase is not seen in the measured emissions.

V. Optimal Control

We will here illustrate optimal cold-start control as one application of the presented model. Specifically, we generate an optimal state-feedback controller suitable for on-line operation that balances the combustion engine's fuel efficiency and tailpipe emissions. We model the combustion engine exhaust using a static mean-value engine model, and allow the controller to freely choose the engine's speed, BMEP, and spark angle [31]. We will then use the controller to generate simulated cold-start temperature and emission trajectories and compare the results for different weightings of fuel efficiency and tailpipe emissions.

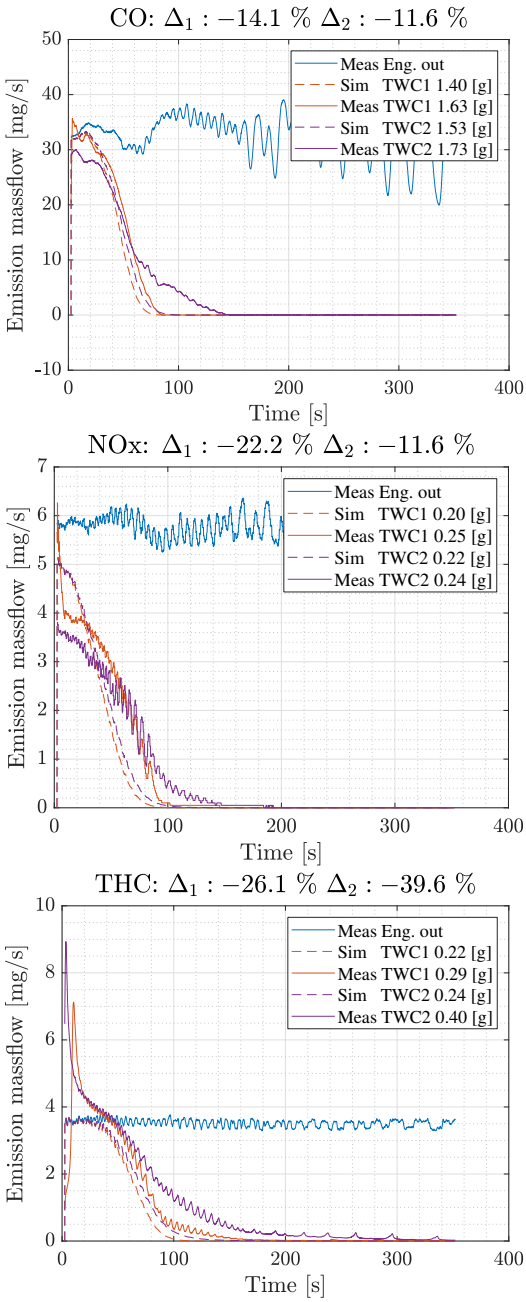


Fig. 16. Measured and simulated emissions for each emission species at a 1000 rpm, 2 bar BMEP load point.

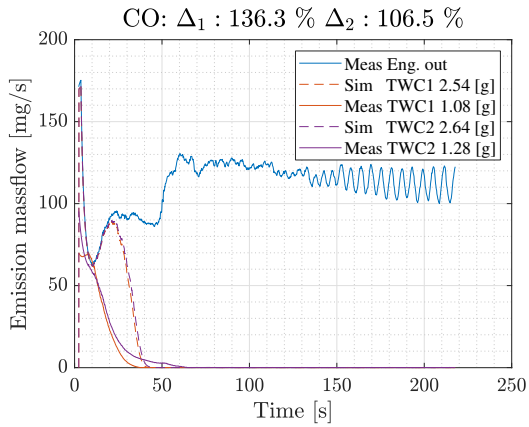


Fig. 17. Measured and simulated emissions for CO emissions at the 1500 rpm, 5 bar BMEP load point. Note the large increase in engine emissions after 10 seconds, giving rise to an increase in simulated emissions without an associated increase in measured emissions.

A. *Problem formulation*

We introduce the optimal control problem as

$$J^* = \min_u \lim_{K\to\infty} \sum_{k=0}^{K} \mathrm{BSFC}(k) + \Lambda^T \cdot \begin{bmatrix} \dot{m}_{\mathrm{tp}}^{\mathrm{CO}}(k) \\ \dot{m}_{\mathrm{tp}}^{\mathrm{NO}_x}(k) \\ \dot{m}_{\mathrm{tp}}^{\mathrm{THC}}(k) \end{bmatrix} \tag{44a}$$

subject to

$$x(k+1) = f_d(x(k), u(k)) \tag{44b}$$

$$g(x(k), u(k)) \leq 0\,. \tag{44c}$$

Here, x is a state vector corresponding to both TWC's (i.e. $[T_{\mathrm{TWC1}}; T_{\mathrm{TWC2}}]$), u is a discrete control variable corresponding to the requested operating point of the combustion engine (i.e. an integer value that indexes the operating points in Table IV), $\mathrm{BSFC}(k)$ is the mean BSFC associated with operating point $u(k)$, Λ is a 3×1 tuning parameter that balances the relative weight given to fuel-efficient operation and minimizing emissions (where smaller Λ prioritizes the BSFC and larger Λ prioritizes the level of emissions), f_d is the system dynamics given by solving Eq. (21) for a given sample time for each TWC, and g is a constraint function that bounds u to the integer values that index the tested operating points and bounds x to safe TWC temperatures.

The cost function Eq. (44a) is specifically formulated to be of form $a + \Lambda^T \cdot b$, as this is equivalent to minimizing a while limiting $b \leq B$, i.e. minimizing the average BSFC while limiting the vector of cumulative emissions to a given level. The same structure is also commonly seen in Equivalent Consumption Minimization Strategy (ECMS) controllers [32, 33] for the equivalent purpose balancing fuel consumption and electric energy consumption. We here notationally use Λ rather than λ to avoid confusion with conventional notation where λ is used to denote the air-fuel ratio.

Note that Eq. (44a) is formulated as an undiscounted infinite-horizon problem, as this penalizes the BSFC and emissions without requiring time to heat the TWC to be explicitly specific or known beforehand. Furthermore, by permitting the engine to operate at any of the points in Table IV we also allow the engine power to freely vary. The hybrid vehicle cold-start problem is one example of an application that is well-suited to this cost formulation, as the electric machine can typically either supply or consume the difference between the combustion engine power and traction power.

We have solved Eq. (44) using a method developed by the authors [34] based on approximate dynamic programming and similar to policy iteration methods. The method, *Undiscounted Control Policy generation by Approximate Dynamic Programming* (UCPADP) extends on existing approximate dynamic programming policy iteration methods by allowing for undiscounted problem formulations, i.e. infinite-horizon problems where the cost function does not decay with increasing k. In principle, we can solve Eq. (44) without using the above method by setting K to a sufficiently large value and using a conventional ADP method [24, 35] to generate a solution. However, it is difficult to manually determine a sufficiently but not excessively large value K. Conveniently, the UCPADP method also returns a sufficient horizon, which for the specific TWC and cost formulation studied here was found to be 145 seconds.

One major benefit with UCPADP and other policy iteration methods is that the optimal control signal can be represented as a control law, i.e. the optimal control signal can be simply tabulated by the state values. This implies that a controller can be implemented by simply looking up the optimal control for the current state. However, this does require knowledge of the current system state, either by direct measurement or by a state observer that estimates the system state. Furthermore, note that the selection of the engine's operating point is not formulated as a dynamic problem, which in principle implies that there is no cost associated with rapidly changing the engine's operating point. Though our solutions did not exhibit very impractical operating point changes, we have in our presented results applied a 5-second rolling average filter to the engine's target speed, BMEP, and spark angle. Though this makes for solutions that are not optimal with respect to Eq. (44), we believe that this results in more suitable engine behavior with somewhat damped transients.

Solving Eq. (44) using an ADP methods first requires the state and control variables to be discretized. As u is inherently discrete (indexing the operating points in Table IV) we only need to discretize the states x. A denser discretization will give a solution closer to the true optimal solution, but at cost of increased memory and computational demand. We have chosen to discretize the states in TWC1 as

$$T_1 = [0, 25, 50, 75, \dots, 900] \tag{45a}$$
$$T_2 = [0, 100, 200, \dots, 900] \tag{45b}$$
$$T_3 = [0, 100, 200, \dots, 900] \tag{45c}$$
$$\Delta_T = [-200, 100] \tag{45d}$$

and for TWC2 as

$$T_1 = [0, 100, 200, \dots, 900] \tag{45e}$$
$$\Delta_T = [-200, 100]\,, \tag{45f}$$

i.e. we resolve the first axial slice in TWC1 with fairly high detail, while the remaining slices and Δ_T is more coarsely resolved.

B. Optimal results

We have solved Eq. (44) for a range of different normalized weights Λ_n, defined element-wise as

$$\Lambda_n^s = \frac{\Lambda^s}{\min \mathrm{BSFC} / \min \dot{m}_{\mathrm{exh}}^s} \tag{46}$$

with results listed in Table X. We use the normalized Λ_n for ease of reference as $\Lambda_n = [1, 1, 1]$ in some sense equally weighs the fuel consumption and engine-out emissions. As tailpipe emissions approach zero when the TWC heats up we can thus view $\Lambda_n = 1$ as a lower bound of relevant values to consider.

We have simulated the performance of the optimal controller and list the cumulative emissions, fuel efficiency, and consumed fuel in Table X for several different Λ_n and two initial conditions. As expected, with increasing Λ_n the sum of penalized emissions decrease, while the mean BSFC increases. This data also indicates that the potential for reducing NO_x emissions is significantly larger than CO and THC emissions, as shown in the last 3 rows where NO_x emissions are reduced by 94% compared to the unpenalized case, while CO and THC emissions are reduced by 35% and 41% respectively. Furthermore, there seems to be some degree of conflict with respect to the individual emissions, as penalizing one species tends to increase the production of others. This indicates that the solutions shown in Table X are Pareto optimal, i.e. a given emission species mass cannot be reduced without either increasing another species' or the BSFC.

An illustration of the controller's time-evolution is shown in Fig. 18 for $\Lambda_n = [10^2, 10^2, 10^2]$. The cold-start trajectory can be divided into three sections;

$t < 5$*: Initial heating phase.* The engine-out species massflow is kept low (reducing tailpipe emissions) and the BSFC is not prioritized. At the end of this phase the first axial slice is hot enough to convert emissions at low mass-flows.

$5 < t < 25$*: Intermediary phase.* With increasing conversion efficiency the engine-out emissions are gradually allowed to increase, allowing for the BSFC to be increasingly prioritized. At the end of this phase the first two axial slices are hot enough to convert emissions at the massflow associated with the minimum-BSFC operating point.

$t > 25$*: Sufficiently-heated phase.* Here the TWC is sufficiently hot for operation at the minimum-BSFC operating point, which the engine is statically operated at while the TWC converts virtually all emissions. Note that the entire TWC is well above light-off after 40 s, i.e. a relatively short heating interval [3].

These sections can largely be seen for other values of Λ_n, with shorter times allocated to the initial- and intermediary

TABLE X

PERFORMANCE OF OPTIMAL CONTROLLER FOR VARYING Λ_n DURING A COLD-START (25 °C, NO RADIAL DISTRIBUTION) AND HALF-WARM START (200 °C SOME RADIAL DISTRIBUTION). GREEN-COLORED CELLS INDICATE CASES WHERE Λ_n PENALIZES ONLY ONE EMISSION SPECIES, WHILE OTHER ARE IGNORED (RED), AND CAN BE COMPARED WITH THE EQUALLY-PENALIZED CASE (BLUE).

(a) $T_{1-3,\mathrm{TWC1}} = T_{1,\mathrm{TWC2}} = 25\,^{\circ}\mathrm{C}$, $\Delta_{T,\mathrm{TWC1}} = \Delta_{T,\mathrm{TWC2}} = 0\,^{\circ}\mathrm{C}$.

Λ_n^{CO} [-]	Λ_n^{THC} [-]	$\Lambda_n^{\mathrm{NO}_x}$ [-]	CO [mg]	THC [mg]	NO_x [mg]	BSFC [g/kWh]	m_{fuel} [g]
0	0	0	521	19.5	262	250	165
1	1	1	521	19.5	262	250	165
10	10	10	423	18.7	153	252	155
0	0	10	423	18.6	152	252	155
0	10	0	521	19.5	262	250	165
10	0	0	388	21.2	239	250	162
10^2	10^2	10^2	526	35.0	41	262	130
0	0	10^2	687	42.6	38	262	126
0	10^2	0	494	19.7	242	250	162
10^2	0	0	340	24.6	229	251	157
10^3	10^3	10^3	529	32.7	29	271	117
0	0	10^3	530	32.9	29	270	116
0	10^3	0	432	11.5	151	254	158
10^3	0	0	338	33.4	245	253	143
10^4	10^4	10^4	526	37.0	26	298	90
0	0	10^4	668	85.1	15	356	60
0	10^4	0	433	11.4	145	255	155
10^4	0	0	340	33.0	232	255	140

(b) $T_{1-3,\mathrm{TWC1}} = T_{1,\mathrm{TWC2}} = 200\,^{\circ}\mathrm{C}$, $\Delta_{T,\mathrm{TWC1}} = \Delta_{T,\mathrm{TWC2}} = -50\,^{\circ}\mathrm{C}$.

Λ_n^{CO} [-]	Λ_n^{THC} [-]	$\Lambda_n^{\mathrm{NO}_x}$ [-]	CO [mg]	THC [mg]	NO_x [mg]	BSFC [g/kWh]	m_{fuel} [g]
0	0	0	265	11.9	153	250	165
1	1	1	265	11.9	153	250	165
10	10	10	188	14.2	100	251	155
0	0	10	187	14.1	99	251	155
0	10	0	266	11.9	153	250	165
10	0	0	207	14.3	129	250	162
10^2	10^2	10^2	219	19.5	20	258	138
0	0	10^2	271	22.8	21	257	137
0	10^2	0	219	12.1	126	250	161
10^2	0	0	167	23.0	120	251	149
10^3	10^3	10^3	212	17.4	16	267	125
0	0	10^3	235	19.4	13	264	126
0	10^3	0	201	9.0	80	253	157
10^3	0	0	131	32.0	122	252	132
10^4	10^4	10^4	203	26.9	12	303	86
0	0	10^4	208	35.3	6	322	75
0	10^4	0	212	9.9	76	255	152
10^4	0	0	168	36.0	81	265	117

heating phases as Λ_n decreases and longer times with larger Λ_n.

Though an open-loop control scheme could be easily implemented, i.e. by "playing back" the speed, BMEP, and spark angle trajectory shown in Fig. 18 without any temperature feedback, this type of controller is potentially sensitive to system variations. This includes both the initial temperature of the TWC (where a hotter initial condition will reach light-off more quickly) as well as variations in the exhaust gas temperature due to the fuel's composition, combustion

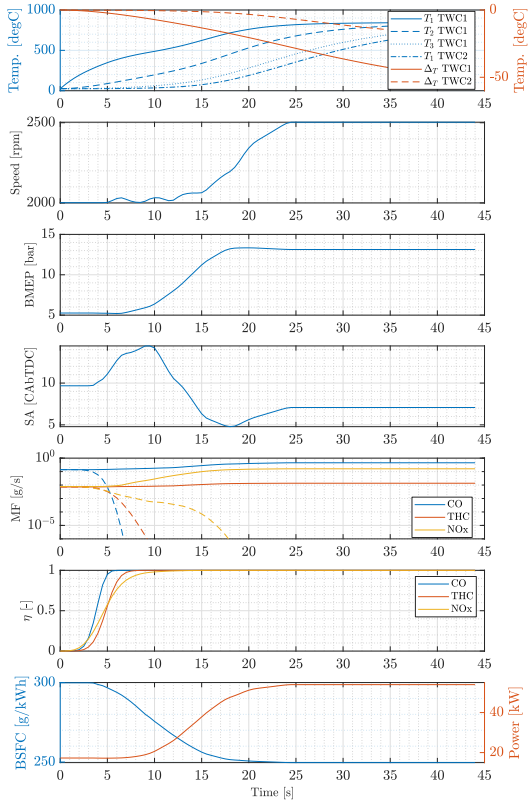


Fig. 18. Simulated trajectories for $\Lambda_n = [10^2, 10^2, 10^2]$. SA indicates the spark angle, MF the engine-out emissions (solid) and tailpipe emissions (dashed), and η the net conversion efficiency.

variability, and so on. We found that the optimal control trajectory for half-warm starts is for some values of Λ_n nearly identical to a time-shifted version of Fig. 18, as exemplified in Fig. 19, while others displayed significant differences. This indicates the potential for implementing a quasi-optimal open-loop heating strategy for some Λ_n.

C. Comparison to suboptimal control

We have compared the optimal controller with a traditional suboptimal heating strategy. The suboptimal controller was defined such that combustion engine is run at a constant operating point for t' seconds, and then switches to the minimum-BSFC operating point. The initial operating point was chosen to be the same as the point chosen by the optimal controller at $t = 0$, and t' selected to give the same average BSFC as the case for $\Lambda_n = [10^2, 10^2, 10^2]$. A comparison of the optimal and suboptimal controllers is listed in Table XI, and the time-evolution is shown in Fig. 20. The CO and THC emissions are virtually identical, but the NO_x emissions are reduced by 35% in the optimal controller. We can see the

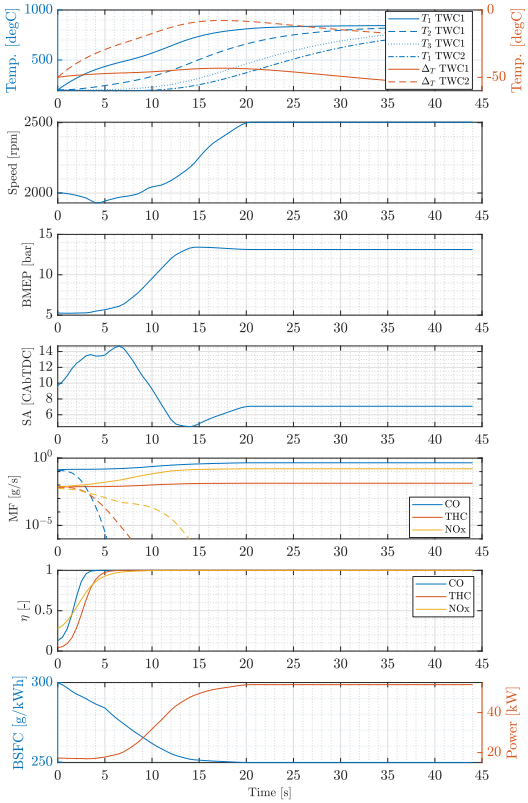


Fig. 19. Simulated trajectories for $\Lambda_n = [10^2, 10^2, 10^2]$ with an initial condition $T_{1-3,\mathrm{TWC1}} = T_{1,\mathrm{TWC2}} = 200\,^\circ\mathrm{C}$, $\Delta_{T,\mathrm{TWC1}} = \Delta_{T,\mathrm{TWC2}} = -50\,^\circ\mathrm{C}$.

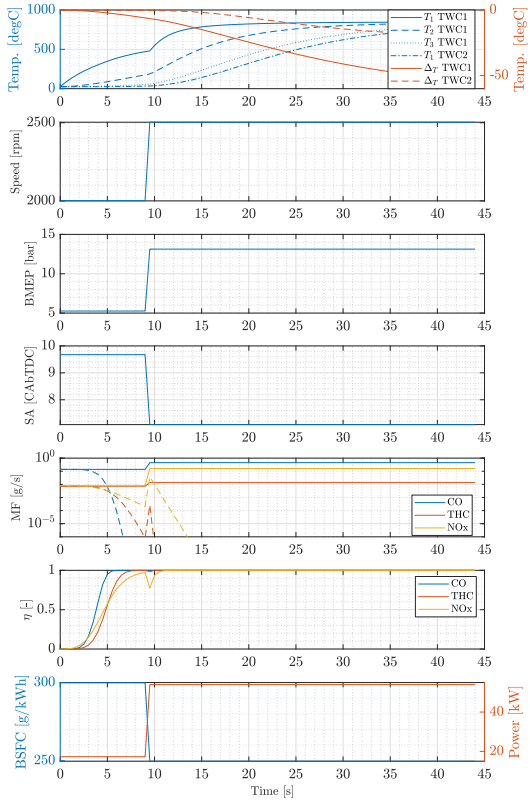


Fig. 20. Simulated trajectory for the suboptimal controller. Note the significant NO_x slip at $t = 10\,\mathrm{s}$, which is not present in Fig. 18.

source of this in Fig. 20, where there is significant NO_x slip at $t \in [9, 13]$ when the engine transitions from the heating phase to the minimum-BSFC operation phase. Though the heating phase could be extended, this would be at cost of reduced average BSFC.

We can also compare the optimal and suboptimal controllers for the half-warm start case. It we consider the same optimal and suboptimal controllers, Table X indicates that the optimal controller attains a mean BSFC of $258\,\mathrm{g/kWh}$, in comparison to the suboptimal controller's $262\,\mathrm{g/kWh}$. This difference corresponds to a 33% reduction relative to the minimum BSFC of $250\,\mathrm{g/kWh}$, indicating the potential for fuel savings by using a closed-loop cold-start strategy.

D. Memory footprint

Though manageable in a PC, the memory demand associated with the discretization in Eq. (45) (with 148,000 permutations) can be problematic in an ECU that has a wide range of other tasks to perform. However, we can apply a simple space reduction scheme to significantly reduce the used memory.

TABLE XI
COMPARISON OF THE OPTIMAL (HERE SHOWN FOR $\Lambda_n = [10^2 10^2 10^2]$) AND SUB-OPTIMAL CONTROLLERS AND THE RELATIVE REDUCTION IN EMISSIONS FOR THE OPTIMAL CONTROLLER.

Cont [-]	CO [mg]	THC [mg]	NO_x [mg]	BSFC [g/kWh]	m_{fuel} [g]
Optimal	526	35.0	41.0	262	130
Suboptimal	525	34.9	62.9	262	143
Difference	-0.2%	-0.3%	34.8%	-	-

Rather than store the full discretization, we can reduce the number of stored elements by letting successive axial slices only be resolved for temperatures equal to or below the preceding slices (rounded up to the nearest discretized value), i.e. storing a table of form similar to that in Table XII.

Furthermore, we can reduce the resolution of T_1 for temperatures significantly below and above light-off by resolving T_1 with $100\,^\circ\mathrm{C}$ increments for temperatures below $100\,^\circ\mathrm{C}$ and above $400\,^\circ\mathrm{C}$, i.e.

$$T_1 = [0, 100, 125, 150, \ldots, 350, 375, 400, 500, \ldots, 900]\,. \tag{47}$$

TABLE XII
REPRESENTATIVE TABLE OF STORED STATES. HERE SORTED FROM FIRST TO LAST COLUMN IN ASCENDING ORDER.

TWC1				TWC2	
T_1	T_2	T_3	Δ_T	T_1	Δ_T
0	0	0	-200	0	-200
0	0	0	-200	0	100
0	0	0	-200	100	-200
⋮	⋮	⋮	⋮	⋮	⋮
250	300	300	100	200	-200
250	300	300	100	200	100
250	300	300	100	300	-200
250	300	300	100	300	100
275	0	0	-200	0	-200
275	0	0	-200	0	100
275	0	0	-200	100	-200
⋮	⋮	⋮	⋮	⋮	⋮
900	900	900	100	900	100

Reducing the range of considered values in this manner reduces the number of stored states to 9500 permutations. Each state permutation is associated with an optimal engine speed, BMEP, and spark angle. Assuming 4 bits of information are allocated for each parameter (allowing resolving the speed, BMEP and spark angle to 16 different and independent values) gives a total storage requirement of 12 bits per state permutation, for a total non-volatile memory requirement of $12/8 \cdot 9500 \approx 13.9$ KiB, which is feasible with existing ECU hardware. More sophisticated compression schemes have the potential to further reduce the required memory, for instance by using decision trees to avoid the need to exhaustively storing every state permutation in regions where the optimal control is constant.

VI. CONCLUSIONS

In this paper we have extended a physics-based TWC model previously presented by the authors [1] suited for on-line optimal control. The previously presented model resolves both axial and radial temperature variations while limiting the number of state variables, allowing for use with optimal control methods that construct an optimal control policy (e.g. nonlinear state-feedback and explicit MPC). In this paper we extended the model to support varying axial discretization lengths, use tuning parameters expressed in well-known SI units, model heat generation by the oxidation of hydrogen, consider a TWC consisting of two separate monoliths of different construction, and use a more rigorous evaluation method with separate tuning and validation datasets. Finally, we have used the model to generate a near-optimal controller [34] that can easily be implemented in existing ECU hardware, requiring no more than 13.9 KiB (14250 bytes) of nonvolatile memory and at virtually no computational cost (as the optimal control is given by a simple linear interpolation operation and a linear rolling average filter). The specific construction of the cost function allows for systematically trading off fuel consumption and each individual emission species, giving the ability to tune the cold-start controller to minimize fuel consumption while individually limiting the specific level CO, THC, and NO_x emissions.

Our experimental study, though limited by the measurement equipment, shows the potential for use both for off-line simulation as well as for generating a near-optimal cold-start controller. Though we experimentally studied the case of a warm engine and a cold TWC for improved experimental repeatability, we hypothesize that the controller can be extended to the cold engine case by a suitable update to the combustion engine exhaust model. Though the measured predictive accuracy is fairly low (with cumulative cold-start emissions typically estimated at -20% to +80% of the measured emissions), it is likely that the experimental setup significantly contributes to this error. The second monolith is situated after a sharp bend, giving a temperature distribution that is not particularly well-captured with an axi-radial model. However, as the majority of the emissions can be converted in the first monolith for low to moderate load-points this inaccuracy might not be of great importance. It may therefore be a prudent design decision to solely model and control the first monolith dynamics in an effort to further reduce the memory requirements of the controller.

We have simulated the performance of the Pareto-optimal cold-start controller for several different relative weightings of fuel efficiency (BSFC) and cumulative emissions for each emissions species, i.e. different points on the Pareto front. For one representative weighting the optimal controller gives NO_x emissions that are 35% lower than a traditional cold-start controller with otherwise identical BSFC and CO and THC emissions. This indicates that an optimal controller that is generated using the presented model has the potential to reduce the cold-start emissions, as well as allowing for systematically adjusting the trade-off between each emission species and fuel consumption. Furthermore, for some regions on the Pareto-front the optimal controllers display similar speed, load, and spark angle trajectories for varying initial TWC temperatures (up to a shift in time). It is therefore plausible that a close-to optimal controller could be implemented with only a single temperature sensor by "playing back" a section of the temporally-resolved optimal control trajectory based on the measured temperature.

Relevant future work includes performing an experimental study using measurement equipment more suited to transient conditions and that is capable of measuring emissions at two locations simultaneously. Furthermore, it would be prudent to experimentally validate the performance of the presented controller, which in turn requires a method for measuring or estimating the temperatures in the TWC. Additionally, we hypothesize that characterizing the engine emissions for different average air-fuel ratios near stoichiometry could allow for the optimal controller to gain an additional degree of freedom in balancing the ratio of CO and THC emissions to the NO_x emissions during a cold-start.

FUNDING

This work was performed within the Combustion Engine Research Center at Chalmers (CERC) with financial support from the Swedish Energy Agency.

References

[1] Jonathan Lock et al. "A Control-Oriented Spatially Resolved Thermal Model of the Three-Way-Catalyst". In: *SAE International* (Apr. 6, 2021). URL: https://www.sae.org/publications/technical-papers/content/2021-01-0597/.

[2] John B. Heywood. *Internal Combustion Engine Fundamentals*. McGraw Hill, 1988.

[3] Raffael Hedinger, Philipp Elbert, and Christopher Onder. "Optimal Cold-Start Control of a Gasoline Engine". en. In: *Energies* 10.10 (Oct. 2017), p. 1548. ISSN: 1996-1073. DOI: 10.3390/en10101548. URL: http://www.mdpi.com/1996-1073/10/10/1548.

[4] Karthik Ramanathan, David H. West, and Vemuri Balakotaiah. "Optimal design of catalytic converters for minimizing cold-start emissions". In: *Catalysis Today* 98.3 (Dec. 2004), pp. 357–373. ISSN: 09205861. DOI: 10.1016/j.cattod.2004.08.003. URL: https://linkinghub.elsevier.com/retrieve/pii/S0920586104005139.

[5] Jianbing Gao et al. "Review of thermal management of catalytic converters to decrease engine emissions during cold start and warm up". In: *Applied Thermal Engineering* 147 (Jan. 2019), pp. 177–187. ISSN: 13594311. DOI: 10.1016/j.applthermaleng.2018.10.037. URL: https://linkinghub.elsevier.com/retrieve/pii/S1359431118336081.

[6] Markus Schori et al. "Optimal catalytic converter heating in hybrid vehicles". In: *SAE Technical Paper* (2014).

[7] R. Sanketi Pannag. *Coldstart Modeling and Optimal Control Design for Automotive SI engine*. 2009.

[8] Giovanni Fiengo et al. "Control of the exhaust gas emissions during the warm-up process of a TWC-equipped SI engine". en. In: *IFAC Proceedings Volumes* 35.1 (2002), pp. 301–306. ISSN: 14746670. DOI: 10.3182/20020721-6-ES-1901.01518. URL: https://linkinghub.elsevier.com/retrieve/pii/S1474667015399390.

[9] Farzad Keynejad and Chris Manzie. "Suboptimal Cold Start Strategies for Spark Ignition Engines". In: *IEEE Transactions on Control Systems Technology* 21.4 (July 2013), pp. 1295–1308. DOI: 10.1109/TCST.2012.2203821. URL: http://ieeexplore.ieee.org/document/6236300/.

[10] Joachim Braun et al. *Three-dimensional simulation of the transient behavior of a three-way catalytic converter*. Tech. rep. SAE Technical Paper, 2002.

[11] David KS Chen et al. "A three-dimensional model for the analysis of transient thermal and conversion characteristics of monolithic catalytic converters". In: *SAE transactions* (1988), pp. 177–189.

[12] Kyriacos Zygourakis. "Transient operation of monolith catalytic converters: a two-dimensional reactor model and the effects of radially nonuniform flow distributions". In: *Chemical Engineering Science* 44.9 (1989), pp. 2075–2086.

[13] GN Pontikakis, GS Konstantas, and AM Stamatelos. "Three-way catalytic converter modeling as a modern engineering design tool". In: *J. Eng. Gas Turbines Power* 126.4 (2004), pp. 906–923.

[14] Naoki Baba, Katsuyuki Ohsawa, and Shigeki Sugiura. "Numerical approach for improving the conversion characteristics of exhaust catalysts under warming-up condition". In: *SAE transactions* (1996), pp. 2064–2079.

[15] Se H Oh and James C Cavendish. "Transients of monolithic catalytic converters. Response to step changes in feedstream temperature as related to controlling automobile emissions". In: *Industrial & Engineering Chemistry Product Research and Development* 21.1 (1982), pp. 29–37.

[16] Marcelo Real et al. "Modelling three-way catalytic converter oriented to engine cold-start conditions". In: *International Journal of Engine Research* 22.2 (Feb. 2021), pp. 640–651. DOI: 10.1177/1468087419853145. URL: http://journals.sagepub.com/doi/10.1177/1468087419853145.

[17] Xieyang Yan et al. "Modeling Three-Way Catalyst Converters During Cold Starts And Potential Improvements". In: *SAE International* (Dec. 2019), pp. 2019–01–2326. DOI: 10.4271/2019-01-2326. URL: https://www.sae.org/content/2019-01-2326/.

[18] Karthik Ramanathan and Chander Shekhar Sharma. "Kinetic Parameters Estimation for Three Way Catalyst Modeling". en. In: *Industrial & Engineering Chemistry Research* 50.17 (Sept. 2011), pp. 9960–9979. DOI: 10.1021/ie200726j. URL: https://pubs.acs.org/doi/10.1021/ie200726j.

[19] E.P. Brandt, Yanying Wang, and J.W. Grizzle. "Dynamic modeling of a three-way catalyst for SI engine exhaust emission control". en. In: *IEEE Transactions on Control Systems Technology* 8.5 (Sept. 2000), pp. 767–776. ISSN: 10636536. DOI: 10.1109/87.865850. URL: http://ieeexplore.ieee.org/document/865850/.

[20] Byron T. Shaw, Gerald D. Fischer, and J. Karl Hedrick. "A SIMPLIFIED COLDSTART CATALYST THERMAL MODEL TO REDUCE HYDROCARBON EMISSIONS". en. In: *IFAC Proceedings Volumes* 35.1 (2002), pp. 307–312. ISSN: 14746670. DOI: 10.3182/20020721-6-ES-1901.01519. URL: https://linkinghub.elsevier.com/retrieve/pii/S1474667015399407.

[21] Pierre Michel. "Optimizing fuel consumption and pollutant emissions of gasoline-HEV with catalytic converter". en. In: *Control Engineering Practice* (2017), p. 8.

[22] Nasser L. Azad, Pannag R. Sanketi, and J. Karl Hedrick. "Determining Model Accuracy Requirements for Automotive Engine Coldstart Hydrocarbon Emissions Control". en. In: *Journal of Dynamic Systems, Measurement, and Control* 134.5 (Sept. 2012), p. 051002. DOI: 10.1115/1.4006217. URL: https://asmedigitalcollection.asme.org/dynamicsystems/article/doi/10.1115/1.4006217/394995/Determining-Model-Accuracy-Requirements-for (visited on 10/22/2019).

[23] Zhaoxuan Zhu, Shawn Midlam-Mohler, and Marcello Canova. "Development of physics-based three-way catalytic converter model for real-time distributed temperature prediction using proper orthogonal decomposition

and collocation". en. In: *International Journal of Engine Research* (Sept. 2019), p. 146808741987612. DOI: 10.1177/1468087419876127. URL: http://journals.sagepub.com/doi/10.1177/1468087419876127 (visited on 02/18/2020).

[24] Dimitri P. Bertsekas. *Dynamic Programming and Optimal Control*. 4th ed. Vol. 1. Athena Scientific, 2017.

[25] Arthur Earl Bryson. *Applied optimal control: optimization, estimation and control*. CRC Press, 1975.

[26] J.D. Logan, F.W. Gehring, and P.R. Halmos. *Applied Partial Differential Equations*. Springer Undergraduate Mathematics Series. Springer, 1998.

[27] K. N. Pattas et al. "Transient Modeling of 3-Way Catalytic Converters". In: *International Congress & Exposition* (Mar. 1, 1994). DOI: 10.4271/940934. URL: https://www.sae.org/content/940934/.

[28] Steffen Tischer et al. "Three-Way-Catalyst Modeling - A Comparison of 1D and 2D Simulations". In: *SAE World Congress & Exhibition* (Apr. 16, 2007), pp. 2007–01–1071. DOI: 10.4271/2007-01-1071. URL: https://www.sae.org/content/2007-01-1071/.

[29] Irman Svraka and Österdahl Wetterhag Linus. "Model Based Catalyst Control". MA thesis. Linköping University, 2019.

[30] Sung Bong Kang et al. "Detailed reaction kinetics for double-layered Pd/Rh bimetallic TWC monolith catalyst". In: *Chemical Engineering Journal* 241 (Apr. 2014), pp. 273–287. ISSN: 13858947. DOI: 10.1016/j.cej.2013.12.039. URL: https://linkinghub.elsevier.com/retrieve/pii/S1385894713016203 (visited on 08/27/2020).

[31] S. H. Chan and J. Zhu. "The Significance of High Value of Ignition Retard Control on the Catalyst Lightoff". In: *SAE International* (Oct. 1996), p. 962077. DOI: 10.4271/962077. URL: https://www.sae.org/content/962077/.

[32] Cristian Musardo et al. "A-ECMS: An Adaptive Algorithm for Hybrid Electric Vehicle Energy Management". In: *European Journal of Control* 11.4 (2005), pp. 509–524. ISSN: 0947-3580. DOI: https://doi.org/10.3166/ejc.11.509-524. URL: https://www.sciencedirect.com/science/article/pii/S0947358005710487.

[33] G. Paganelli et al. "Equivalent consumption minimization strategy for parallel hybrid powertrains". In: *Vehicular Technology Conference. IEEE 55th Vehicular Technology Conference. VTC Spring 2002 (Cat. No.02CH37367)* 4 (2002), 2076–2081 vol.4. DOI: 10.1109/VTC.2002.1002989.

[34] Jonathan Lock and Tomas McKelvey. *Undiscounted Control Policy Generation for Continuous-Valued Optimal Control by Approximate Dynamic Programming*. 2021. arXiv: 2104.11093 [math.OC].

[35] Dimitri P. Bertsekas. *Dynamic Programming and Optimal Control*. 4th ed. Vol. 2. Athena Scientific, 2012.