CHALMERS
UNIVERSITY OF TECHNOLOGY

Communication Scheduling by Deep Reinforcement Learning for Remote Traffic State Estimation with Bayesian Inference

Downloaded from: https://research.chalmers.se, 2026-04-04 13:44 UTC

Citation for the original published paper (version of record):
Peng, B., Xie, Y., Seco-Granados, G. et al (2022). Communication Scheduling by Deep Reinforcement Learning for Remote Traffic State Estimation
with Bayesian Inference. IEEE Transactions on Vehicular Technology, 71(4): 4287-4300.
http://dx.doi.org/10.1109/TVT.2022.3145105

N.B. When citing this work, cite the original published paper.

© 2022 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, or reuse of any copyrighted component of this work in other works.

(article starts on next page)

Communication Scheduling by Deep Reinforcement Learning for Remote Traffic State Estimation with Bayesian Inference

Bile Peng, *Member, IEEE*, Yuhang Xie, Gonzalo Seco-Granados, *Senior Member, IEEE*, Henk Wymeersch, *Senior Member, IEEE*, and Eduard A. Jorswieck, *Fellow, IEEE*

***Abstract*—Traffic awareness is the prerequisite of autonomous driving. Given the limitation of on-board sensors (e.g., precision and price), remote measurement from either infrastructure or other vehicles can improve traffic safety. However, the wireless communication carrying the measurement result undergoes fading, noise and interference and has a certain probability of outage. When the communication fails, the vehicle state can only be predicted by Bayesian filtering with a low precision. Higher communication resource utilization (e.g., transmission power) reduces the outage probability and hence results in an improved estimation precision. The power control subject to an estimate variance constraint is a difficult problem due to the complicated mapping from transmit power to vehicle-state estimate variance. In this paper, we develop an estimator consisting of several Kalman filters (KFs) or extended Kalman filters (EKFs) and an interacting multiple model (IMM) to estimate and predict the vehicle state. We propose to apply deep reinforcement learning (DRL) for the transmit power optimization. In particular, we consider an intersection and a lane-changing scenario and apply proximal policy optimization (PPO) and soft actor-critic (SAC) to train the DRL model. Testing results show satisfactory power control strategies confining estimate variances below given threshold. SAC achieves higher performance compared to PPO.**

***Index Terms*—Autonomous driving, Bayesian filtering, interacting multiple model, resource allocation, power control, deep reinforcement learning, proximal policy optimization, soft actor-critic.**

I. Introduction

AWARENESS of the traffic state (i.e., positions and velocities of surrounding vehicles) is the prerequisite for autonomous driving [1], which is usually enabled by either onboard sensors of vehicles or sensors on a road-side unit (RSU) as infrastructure. However, they are inherently limited by measurement accuracy, cost, and range. For example, radar and camera in bad weather have high measurement noise. Lidar provides a high measurement accuracy but the cost limits its application. Besides, onboard sensors are often blocked by surrounding vehicles and cannot measure vehicles behind the blockage. These problems can be solved to a large extent by remote measurement and vehicular communication, where vehicles or RSU transmit the estimated traffic state to other vehicles with wireless communication [2]. In this way, a high-cost Lidar can be equipped on an RSU, which broadcasts the traffic state to all nearby vehicles. The RSU can be installed on a high place (such as on the traffic light) and a vehicle can transmit the traffic state estimates (either of itself or of adjacent vehicles) obtained by its own sensors to other vehicles. In this way, problems of unfavorable sensing conditions, e.g., blockage, can be relieved. Furthermore, the received estimates from the RSU can also be combined with measurements by the on-board sensors (if available) for better precision or richer details.

However, the wireless communication cannot be assumed always reliable. When the communication fails, the traffic state can only be predicted using previous estimates. Besides, the inherent measurement noise is also a constraint that needs to be addressed. Due to the relatively tractable dynamics of traffic state, the Bayesian filter is widely applied for state estimation and prediction. In particular, Kalman filter (KF) and extended Kalman filter (EKF) are popular because of the optimality of the KF given a linear system dynamics and a good compromise between complexity and performance of the EKF given a nonlinear system dynamics.

Bayesian inference, remote estimation and corresponding communication scheduling have been widely studied for various applications. For example, the Cramér-Rao lower bound (CRLB) was derived in the vehicular context in [3]. The vehicle movement is modeled with Bayesian approaches in [4]–[6]. The problem of estimation over lossy network is addressed in [7]–[9]. Communication scheduling is optimized for Bayesian inference with remote measurement in [10]–[16]. Distributed sensor fusion over lossy channels is discussed in [17], [18]. As a relevant topic, the metrics age of information (AoI) and value of information (VoI) are optimized in [19]–[23]. In these works, the estimation with irregular measurements has been well discussed but the lossy communication network is either assumed given (e.g., [7]–[9], [17], [18]) or the optimization is naive and empirical (e.g., [10]–[13]) or is derived for simple scenarios and the solution is given as a closed-form solution [14], [15]. For nonstationary vehicular

B. Peng, Y. Xie and E. A. Jorswieck are with Institute for Communications Technology, TU Braunschweig, 38106 Braunschweig, Germany (e-mail: {b.peng,yuhang.xie,e.jorswieck}@tu-braunschweig.de).

G. Seco-Granados is with Department of Telecommunications and Systems Engineering, School of Engineering, Universitat Autònoma de Barcelona (UAB), Bellaterra 08193 - Barcelona, Spain (e-mail: gonzalo.seco@uab.cat).

H. Wymeersch is with the Department of Electrical Engineering, Chalmers University of Technology, 41296 Gothenburg, Sweden (e-mail: henkw@chalmers.se).

The of Bile Peng and Eduard Jorswieck was in part supported by the Federal Ministry of Education and Research (BMBF, Germany) as part of the 6G Research and Innovation Cluster 6G-RIC under Grant 16KISK020K. The work of Gonzalo Seco-Granados is partly funded by the Spanish Ministry of Science and Innovation PID2020-118984GB-I00 and by the Catalan ICREA Academia Programme.

environment, an empirical solution might be highly suboptimal and an analytical solution is difficult or impossible to obtain. To the best of our knowledge, the above-mentioned problem in a complicated nonstationary environment remains open. Machine learning is a suitable tool to handle such extremely difficult problems. Machine learning has been widely applied for communication system optimization [24]. Since we consider an optimization problem in a dynamic process, the communication in current time step impacts the estimation precision not only in current time step, but also in time steps in near future, the deep reinforcement learning (DRL) [25] is a good candidate as a solution. It optimizes a sequence of actions in a dynamic environment such that the sum of rewards over time is maximized. In recent years, DRL has been applied to resource allocation [26]–[29], signal processing [30], [31] and has achieved good results when an analytical approach is impossible.

In this paper, we present a traffic state estimator in nonstationary traffic environments with remote source, we formulate the problem of transmission power control optimization, we define the corresponding reinforcement learning (RL) environment, and finally we train a policy with state-of-the-art DRL algorithms. This paper in also an early contribution to realize the vision of *semantic communication* [32], which does not only optimize the communication performance (e.g., data rate, delay and outage probability), but also considers the utility tasks of the receiver (i.e., vehicles near the RSU) brought by the received information (i.e., estimation precision in our application). This is a difficult problem because it requires the interdisciplinary consideration of both communication and estimation, and its formulation is usually too complicated to solve analytically. In recent years, the semantic communication has been considered in a Bayesian game in [33]. Network optimization based on value of information has been considered in [34]. Joint communication, computation, caching and control for edge computation has been studied in [35]. To the authors' best knowledge, joint consideration of estimation and communication in context of transportation safety and automation is still an open problem. This is the objective of this paper. Our specific contributions are:

- We extend the standard KF, EKF and interacting multiple model (IMM) to cases with and without measurements. If measurements are available, the standard filtering (predicting and updating) is applied. If measurements are not available due to failed communication, we do predictions until measurement is available again.
- We develop a problem formulation to minimize radio resource utilization while keeping the expected estimation precision at the other vehicles better than a threshold without feedback from them. Resource allocation without feedback is advantageous because of two reasons: firstly, it entails lower communication load since no feedback is required; and secondly, similar to the transmission from the RSU to the receivers, the feedback is not always reliable either and resource allocation without feedback has to be addressed anyway. Assuming the unfavorable Rayleigh fading channel, the worst case is considered without knowledge of the actual channels to the receivers.
- We use the posterior estimates to simplify the problem formulation, which considers all possibilities (whether the transmissions are successful or not, called *histories* later in the paper) at the receivers, whose probabilities are determined by the transmit powers in the past. By transmitting posterior estimates instead of raw measurements, the number possible histories is greatly reduced from 2^T to T where T is the total number of time steps. The problem becomes then feasible.
- We apply the state-of-the-art DRL algorithms proximal policy optimization (PPO) and soft actor-critic (SAC) to solve the problem, which are able to minimize the sum of transmit powers while keeping the estimation precision above a given level. We show that even with the worst case assumption of Rayleigh fading channels, the sum of transmit powers computed with the DRL algorithms is still significantly lower than constant transmit powers.

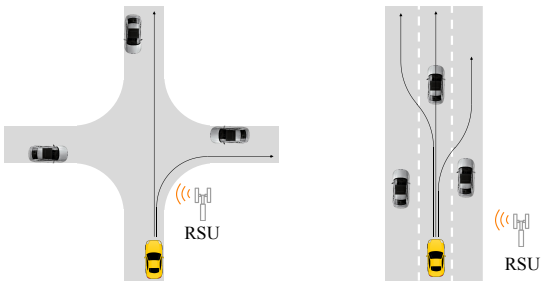
The remaining part of the paper is structured as follows: Section II formulates the problem, Section III describes the communication model, Section IV explains the involved RL techniques and defines the RL environment. Training and evaluation results will be presented in Section V and the conclusion will be drawn in Section VI.

Notation: Throughout this paper we use the following notations: $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes the Gaussian distribution with expectation $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. $(\cdot)^T$ denotes the transpose operator. $\mathcal{N}(\mathbf{z}; \boldsymbol{\mu}, \boldsymbol{\sigma}^2)$ denotes the probability density of $\mathbf{z}$ given Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. $\mathbb{E}(\cdot)$ denotes the expectation operator. $\mathrm{D}_{\mathrm{KL}}(A||B)$ denotes the Kullback–Leibler (KL)-divergence between distributions A and B. Indicator function $\mathrm{I}(c)$ is 1 if condition c is true and 0 otherwise. $\mathcal{N}(\mathbf{z}; \boldsymbol{\mu}, \boldsymbol{\sigma}^2)$ denotes the probability density at $\mathbf{z}$ given the normal distribution with expectation $\boldsymbol{\mu}$ and variance $\boldsymbol{\sigma}^2$, $\mathcal{N}(\mathbf{z}; \boldsymbol{\mu}, \boldsymbol{\sigma}^2) = \exp(-\frac{1}{2}(\mathbf{z}-\boldsymbol{\mu})^T \boldsymbol{\sigma}^{-2}(\mathbf{z}-\boldsymbol{\mu}))/\sqrt{(2\pi)^k |\boldsymbol{\sigma}|^2}$, where k is the dimension of $\mathbf{z}$.

II. Model and Problem Formulation

The vehicle state and its predictability depend on the environment. In this study, we consider two typical traffic environments with varying vehicle state dynamics and therefore challenging predictability: an intersection and a multi-lane highway, as shown in Fig. 1. In the intersection, the vehicle uses the rightmost lane and can either go straight or turn right. On the highway, the vehicle is initially on the middle lane and can either keep the lane or change to left or right lane at any location. Since the vehicles are not controlled by the RSU, which measures the vehicle states, controlling inputs [36, eq. (1.1)] are neglected and the different possible driving maneuvers are modeled by the IMM available at both RSU and receiving vehicles, as will be described in the next section. Vehicles are not assumed to have any sensors.

We consider one vehicle of this paper. Estimating states of multiple vehicles can be done with multiple target tracking [37] and communication of multiple vehicles' state estimates can be done using different resource blocks. Joint scheduling and estimation optimization of multiple vehicles is therefore a straightforward extension of the current paper.



(a) Intersection (b) Lane changing

Fig. 1. Considered scenarios. In the intersection scenario, a vehicle (yellow) on the rightmost lane can either go straight or turn right. In the multi-lane highway scenario, a vehicle (yellow) on the middle lane can keep lane, change to the left or right lane at any place. The RSU measures and estimates the state of the yellow vehicle and transmits it to the adjacent traffic participants (gray) who are interested in the state of the yellow vehicle.

A. Vehicle Dynamics

Below we describe the vehicle dynamics, as modeled at the RSU or the vehicles. We define the vehicle state at time step t as $\mathbf{x}_t = (p_{x,t}, p_{y,t}, v_{x,t}, v_{y,t})$ with $(p_{x,t}, p_{y,t})$ and $(v_{x,t}, v_{y,t})$ being position and velocity of the vehicle at time step t, respectively, where x is the coordinate in direction from west to east and y is the coordinate in direction from south to north. In general, the vehicle dynamics is described by

$$\mathbf{x}_t = f(\mathbf{x}_{t-1}) + \mathbf{w}_t, \tag{1}$$

where f is a general nonlinear function of system dynamics reflecting possible driving maneuvers, e.g., straight driving, accelerating, decelerating or turning, $\mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{N})$ is the inherent randomness of the process and is assumed normally distributed with expectation $\mathbf{0}$ and covariance matrix $\mathbf{N}$. In the following we choose straight driving and turning as two examples of f. It is to note that the set of system dynamics models can be generalized to include other possible driving behaviors.

1) Straight Driving: Given the state $\mathbf{x}_{t-1}$ at time step $t-1$ and assuming the vehicle is going straight with a constant velocity, the state $\mathbf{x}_t$ at time step t is

$$\mathbf{x}_t = \mathbf{F}\mathbf{x}_{t-1} + \mathbf{w}_t, \tag{2}$$

where $\mathbf{F}$ is a linear state-transition matrix as a concrete realization of f in (1), which is defined as

$$\mathbf{F} = \begin{pmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

where Δt is the time difference between two adjacent time steps.

2) Turning: When the vehicle is turning with velocity direction change of $\Delta\theta$ between time step $t-1$ and time step t, the state-transition function is nonlinear and the velocity at time step t is

$$v_{x,t} = v_{x,t-1}\cos(\Delta\theta) - v_{y,t-1}\sin(\Delta\theta) + w_{v_x,t} \tag{3}$$
$$v_{y,t} = v_{x,t-1}\sin(\Delta\theta) + v_{y,t-1}\cos(\Delta\theta) + w_{v_y,t} \tag{4}$$

where $(w_{v_x,t}, w_{v_y,t})^T \sim \mathcal{N}(\mathbf{0}, \mathbf{N}_v)$ is the normally distributed perturbation of velocity with expectation $\mathbf{0}$ and covariance matrix $\mathbf{N}_v$. The position at time step t is

$$\begin{pmatrix} p_{x,t} \\ p_{y,t} \end{pmatrix} = \begin{pmatrix} p_{x,t-1} \\ p_{y,t-1} \end{pmatrix} + \Delta t \cdot \begin{pmatrix} v_{x,t} \\ v_{y,t} \end{pmatrix} + \mathbf{w}_{p,t}, \tag{5}$$

where $\mathbf{w}_{p,t} \sim \mathcal{N}(\mathbf{0}, \mathbf{N}_p)$ is the normally distributed perturbation of position with expectation $\mathbf{0}$ and covariance matrix $\mathbf{N}_p$[1]. If the vehicle turns right, we only need to set $\Delta\theta$ to a negative value. In the intersection scenario, the vehicles either goes straight all the time or goes straight and turns right in the intersection by 90 degree and goes straight again. In the lane changing scenario, the vehicle either goes straight all the time, or first turns left and then turns right (to change to the left lane), or first turns right and then turns left (to change to the right lane) at a random position.

B. Sensing Model

We consider sensing of the vehicle state at the RSU. Assuming a general nonlinear measurement model, the measurement at time t is

$$\mathbf{z}_t = h(\mathbf{x}_t) + \mathbf{v}_t, \tag{6}$$

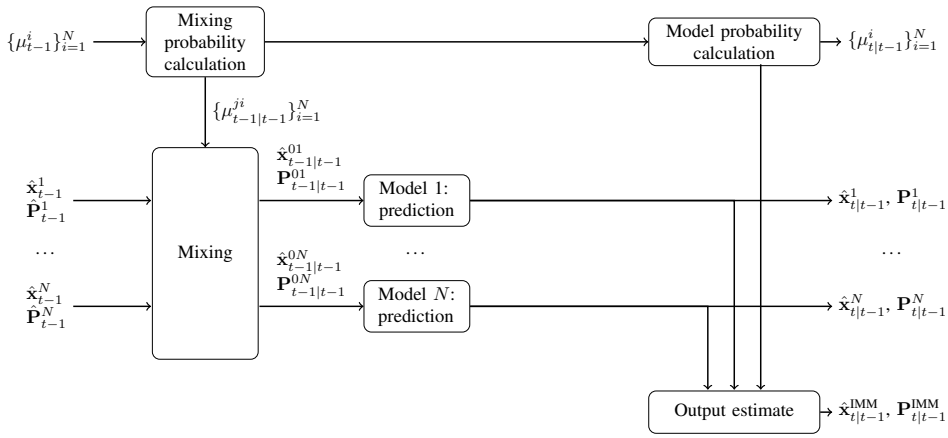
where h is the measurement model and $\mathbf{v}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_t)$ is the normally distributed measurement noise, with $\mathbf{R}_t$ being the covariance matrix of measurement noise at time step t. Based on these measurements, the RSU runs a tracking method. In this work, we apply KF, EKF and IMM for this purpose. Their standard formulation is briefly presented in the appendices. The IMM which provides the current state estimate $\hat{\mathbf{x}}_t^{\text{IMM}}$ with the associated covariance $\mathbf{P}_t^{\text{IMM}}$. The block diagram of the estimator in one time step is shown in Fig. 2. The model probabilities $\{\mu_{t|t}^i\}_{i=1}^N$, state estimates $\{\hat{\mathbf{x}}_{t|t}^i\}_{i=1}^N$ and covariance matrices $\{\mathbf{P}_{t|t}^i\}_{i=1}^N$ (i.e., the input of Fig. 2(a)) are transmitted from RSU to adjacent traffic participants after the estimation, where N is the number of models in IMM. Note that the estimator can operate both with and without measurements. In the latter case, which will be relevant to the vehicle receiver, we perform an *open loop prediction*, which is equivalent to assuming an infinite measurement noise variance.

C. Communication and Information Model

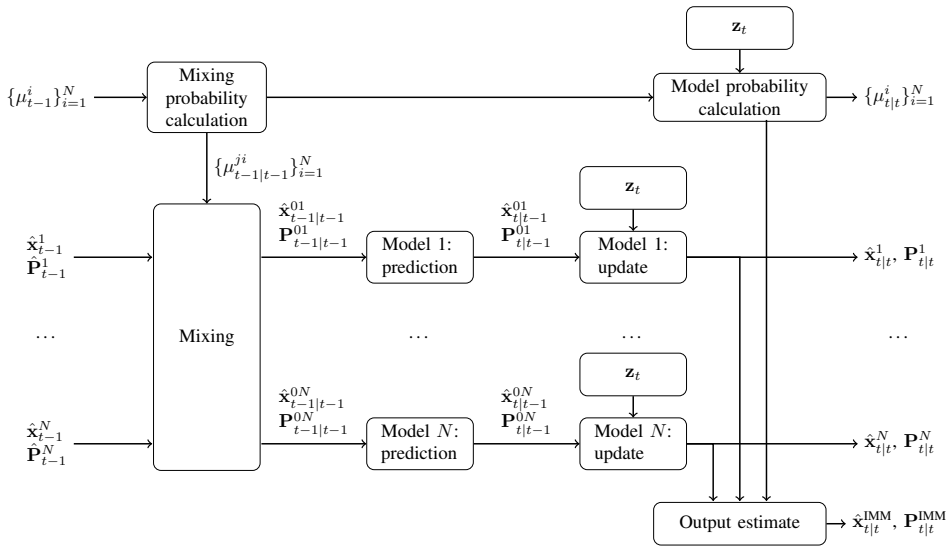
The RSU computes the posterior estimate of the vehicle state and broadcasts it via wireless communication. The receiving vehicle do not measure the target vehicle's state but can do prediction without measurement (i.e., without (29) and (32)) if the transmission of RSU's estimate fails. Hence, both the vehicle and the RSU run tracking methods with different information models

- *Receiving vehicle* is equipped with an estimator without updating with measurement, which is shown in Fig. 2(a). It does not do measurement but knows the history of successful transmissions $\mathbf{h}_t = (c_0, c_1, \ldots, c_t)$, where $c_k \in \{0, 1\}$ indicates whether or not a downlink packet

[1]Note that we apply the discrete EKF [36], which assumes constant velocity in a time step. In reality, this velocity can be the mean velocity in the time step such that the error caused by the approximation is minimized.



(a) Without measurement (for receiving vehicles to predict state without measurement if communication fails and for RSU to estimate the estimation variance of receiving vehicles which experience communication failure)



(b) With measurement (for RSU to estimate the target vehicle's state)

Fig. 2. Block diagram of IMM in one time step. At the beginning of each time step, the mixing probabilities are computed with (35) (block "mixing probability calculation"). The mixed state estimates and estimate variances are computed with (36) and (37), respectively, for each model (block "mixing"). After that, prediction and prior variance are computed with (27) and (28), or (33) and (34), respectively, depending on whether KF or EKF is applied (block "model n: prediction"). If measurement is available, the estimates and the variances are updated with the measurement with (29) and (32), respectively, for each model (block "model n: update"). The model probabilities are updated with (39) or (38), for cases without and with measurement (block "model probability calculation"). Finally, the output estimate and variance of IMM is computed with (40) and (41), respectively (block "output estimate"). This figure is based on Figure 1 in [38], which only considers IMM with measurement.

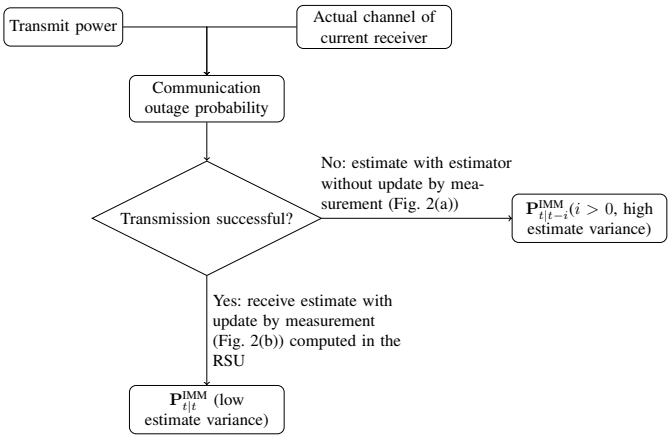
was received. The estimation is done with measurement if the transmission is successful or as open loop prediction otherwise. Therefore, the estimate at the vehicle of the state at time t $\hat{\mathbf{x}}_t^{\text{IMM}}$ and the associated covariance $\mathbf{P}_t^{\text{IMM}}$ depend on $\mathbf{h}_t$ and are therefore denoted as $\hat{\mathbf{x}}_t^{\text{IMM}}(\mathbf{h}_t)$ and $\mathbf{P}_t^{\text{IMM}}(\mathbf{h}_t)$, respectively.

- *RSU* has access to all measurements and runs multiple estimators (Fig. 2(a) and Fig. 2(b)) with all possible histories to estimate all possible $\mathbf{P}_t^{\text{IMM}}(\mathbf{h}_t)$ at the receivers. Besides, the RSU also estimates the probabilities of the histories with its transmit powers based on the worst case fading channel assumption with the highest communication outage probability. Because history $\mathbf{h}_t$ depends on the transmit power $(P_1, \ldots, P_t)$ where P_t is the transmit power at time step t, we denote the history as $\mathbf{h}_t(P_1, \ldots, P_t)$. In this way, it can compute the upper bound of $\mathbb{E}_{\mathbf{h}_t}\left(\mathbf{P}_t^{\text{IMM}}(\mathbf{h}_t(P_1, \ldots, P_t))\right)$ without feedback from the vehicles because $\mathbb{E}_{\mathbf{h}_t}\left(\mathbf{P}_t^{\text{IMM}}(\mathbf{h}_t)\right)$ is an increasing function of the outage probability.

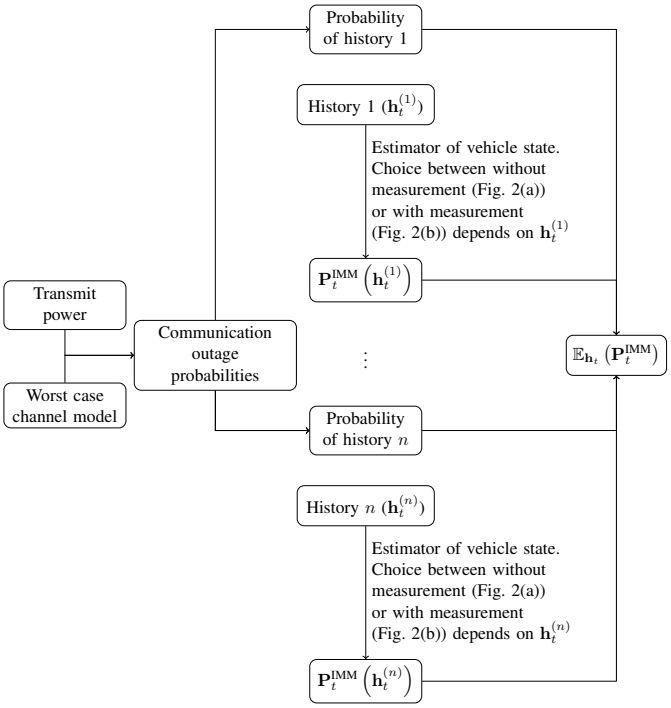
The process described above is illustrated in Fig. 3.

D. Problem Statement

As was described, if the transmission is successful, the vehicle receives an estimate with a high precision, otherwise it can only predict the current state with a lower precision. In some cases (e.g., when the vehicle goes straight outside the intersection), its state is relatively predictable and it is expected that the transmission power can be kept low. The state becomes highly unpredictable when the vehicle is in the intersection without an unambiguous trend whether it is going straight or turning right (in the first scenario) and the vehicle changes lane (in the second scenario). Our objective is to optimize the



(a) At the traffic participants: the transmission power determines whether the transmission is successful. The outage probability is determined by the transmit power and the actual channel for the current receiver (which is different from user to user). If the transmission is successful, traffic participants receive the estimate with low variance, otherwise they predict the state without measurement with high variance.



(b) At the RSU: the transmission power of the RSU determines the communication outage probability. Since we do not know the channel to the receiving vehicles, we assume the worst case channel model, which has the highest possible outage probability among all channels in the area of the RSU's responsibility. The outage probabilities determine the probability of each possible history (denoted by the superscript i). The sum of the estimate variances weighted by the probabilities of histories is the expected estimate variance at the traffic participants.

Fig. 3. Work flows of receiving vehicles and the RSU.

communication, such that the expected estimate variance at the receiving vehicles is below a given threshold to ensure a certain safety level while minimizing the sum of transmit powers. The problem can be formulated as

$$\begin{aligned} \min_{P_1,\ldots,P_T} \quad & \sum_{t=1}^{T} P_t \\ \text{s.t.} \quad & \operatorname{tr}\left(\mathbb{E}_{\mathbf{h}_t}\left(\mathbf{P}_t^{\text{IMM}}\left(h_t(P_1,\ldots,P_t)\right)\right)\right) < p_{\text{th}}, \end{aligned} \tag{7}$$

where P_t is the transmit power at time step t, T is the horizon (i.e., the maximum time step where the vehicle is in the responsible range of the RSU) and p_{th} is the threshold that should not be violated. The complicated relation between P_t and $\mathbf{P}_t^{\text{IMM}}$ will be presented in Section III.

III. Transmission in Unreliable Channel and Consideration of Multiple Histories

A. Transmission in an Unreliable Fading Channel

We assume vehicles with autonomous driving or driving assistance systems are able to use the IMM estimator to predict and estimate states of surrounding vehicles with estimates transmitted from the RSU (i.e., outputs of Fig. 2). The transmission is over a fading channel and the channel capacity is a random variable depending on the channel gain. Therefore, there is a certain outage probability (i.e., the probability that the transmission fails), which is computed as

$$\begin{aligned} p_t^{\text{out}} &= \mathbb{P}\left(R > W\log_2\left(1+\frac{|g|^2 P_t}{W N_0}\right)\right) \\ &= F_g\left(\sqrt{\frac{\left(2^{R/W}-1\right)W N_0}{P_t}}\right), \end{aligned} \tag{8}$$

where g is the randomly distributed channel gain, F_g is the cumulative distribution function (CDF) of random variable g, R is the required data rate and W is the bandwidth (both assumed to be constant for simplicity), P_t is the transmission power, N_0 is the noise power spectrum density per Hertz.

If F_g is a monotonically increasing function (which is usually this case with conventional distributions of fading channels), when we increase P_t, p_t^{out} will decrease and the traffic participants have a higher probability to receive the posterior estimate, which leads to a lower estimate variance. Until now we have completed the description of relationship between P_t and $\mathbf{P}_t^{\text{IMM}}$.

The choice of the distribution and the associated parameters should be determined by on-site measurement and reflect the worst case in the area of the RSU's responsibility, i.e., the actual outage probability at any position in the area of the RSU's responsibility cannot be higher than p_t^{out} computed in (8).

B. Consideration of Multiple Histories

Problem (7) requires the expectation of $\mathbf{P}_t^{\text{IMM}}$. From the above sections we know that $\hat{\mathbf{x}}_t^{\text{IMM}}$ and $\mathbf{P}_t^{\text{IMM}}$ depend on which measurements are available and are denoted as $\hat{\mathbf{x}}_t^{\text{IMM}}\left(\mathbf{h}_t^{(i)}\right)$

and $\mathbf{P}_t^{\text{IMM}}\left(\mathbf{h}_t^{(i)}\right)$, respectively, where i is the index of the possible history. The probability of $\mathbf{h}_t^{(i)}$ is

$$p\left(\mathbf{h}_t^{(i)}\right) = \prod_{k=0}^{t}\left(\left(1-p_k^{\text{out}}\right)c_k^{(i)} + p_k^{\text{out}}\left(1-c_k^{(i)}\right)\right), \tag{9}$$

the expectation of $\hat{\mathbf{x}}_t^{\text{IMM}}$ is

$$\mathbb{E}_{\mathbf{h}_t^{(i)}}\left(\hat{\mathbf{x}}_t^{\text{IMM}}\right) = \sum_i \hat{\mathbf{x}}_t^{\text{IMM}}\left(\mathbf{h}_t^{(i)}\right) p\left(\mathbf{h}_t^{(i)}\right) \tag{10}$$

and the expectation of $\mathbf{P}_t^{\text{IMM}}$ is

$$\mathbb{E}_{\mathbf{h}_t^{(i)}}\left(\mathbf{P}_t^{\text{IMM}}\right) = \sum_i \mathbf{P}_t^{\text{IMM}}\left(\mathbf{h}_t^{(i)}\right) p\left(\mathbf{h}_t^{(i)}\right). \tag{11}$$

If we have T time steps in total, there would be 2^T possible histories, which is an enormous number given an ordinary T, say, 60 - 70 in our scenarios. However, the RSU can do the estimation with all measurements in the past (because the RSU always has all the measurements) and transmits the estimates (i.e., $\{\mu_{t|t}^i, \hat{\mathbf{x}}_{t|t}^1, \mathbf{P}_{t|t}^1\}_{i=1}^N$) to the receiving vehicles. If the transmission is successful (i.e., $c_t^{(i)} = 1$), the previous history $h_{t-1}^{(i)}$ does not impact the estimate because the transmitted estimate is based on all measurements in the past. Only when the transmission fails, the receiving vehicles needs to estimate (predict) the current state with methods described above. Therefore, if the RSU transmits the estimates instead of the raw measurements, what matters is only the last time when the transmission was successful. In this case, we have

$$p\left(\mathbf{h}_t^{(-K)}\right) = \left(1-p_{t-K-1}^{\text{out}}\right)\prod_{k=t-K}^{t} p_k^{\text{out}}, \tag{12}$$

instead of (9), where $\mathbf{h}_t^{(-K)}$ is the history that the previous K transmissions fail and the last successful transmission was at time step $t-K-1$. Hence, the number of possible histories reduces from 2^T to T (which is a considerable complexity reduction). The expectation of $\mathbf{P}_t^{\text{IMM}}$ is computed with (11) as before.

Another advantage of transmitting estimates instead of raw measurements is that the estimates have higher precision than the measurements and the receiving vehicles do not have to carry out the estimation themselves. Instead, the estimation is done only once at the RSU. The overall computational effort is therefore reduced. Besides, RSUs often have fixed power supply and are less energy-sensitive compared to vehicles, which usually run computations on batteries.

IV. Transmit Power Control with Deep Reinforcement Learning

The model-free DRL algorithms can be roughly classified into off-policy algorithms in style of Q-learning and on-policy policy optimization algorithms, with SAC and PPO as the state-of-the-art representatives in each category, respectively. In addition, the on-policy algorithms require an external advantage estimator, where the generalized advantage estimation (GAE) is widely applied. In this section, we first introduce some fundamental definitions in DRL and then describe the above-mentioned algorithms, which are later used to solve the proposed problem.

A. RL Problem Formulation and Fundamental Definitions

RL aims to optimize actions in a dynamic environment such that the expected sum of rewards is maximized. As shown in Fig. 4, in a time step t, an agent observes the environment state s_t, and uses its policy π_θ parameterized by θ to determine an action[2] a_t, i.e., $a_t = \pi_\theta(s_t)$. The action a_t changes the environment state from s_t to s_{t+1} with the system dynamics $s_{t+1} = f(s_t, a_t)$ and determines the reward r_t with the reward function $r_t = r(s_t, a_t)$. The system dynamics function f and reward function r are given in the problem formulation and the policy π_θ is to be optimized. Formally, the problem can be formulated as

$$\begin{aligned} \max_\theta \quad & \mathbb{E}\left(\sum_{t=1}^T r(s_t,a_t)\middle| s_0\right) \\ \text{s.t.} \quad & a_t = \pi_\theta(s_t) \\ & r_t = r(s_t, a_t) \\ & s_{t+1} = f(s_t, a_t). \end{aligned} \tag{13}$$

Note that the expectation operator is necessary because of the inherent randomness of the considered problem: both system dynamics $f(s,a)$ and policy $\pi(s)$ might be stochastic. For example, the driving behavior (going straight or turning) is unknown in advance, which causes randomness in system dynamics. The stochastic policy is widely used by modern RL algorithms, which returns a distribution of action parameterized by state s rather than a deterministic action, which results in randomness in policy.

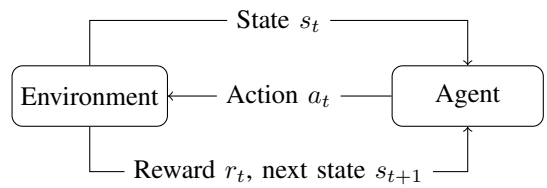


Fig. 4. Framework of RL problems in one time step. The agent observes the state of the environment s_t and makes a decision of the action a_t, which determines a reward r_t together with s_t and changes the environment state from s_t to s_{t+1} and this process starts over. The objective is to maximize the expected sum of rewards.

In our problem, we define state, action and reward as follows:

- The state should contain sufficient information to make the optimal decision of the transmission power and is defined as

$$s_t = \left(\mathbf{z}_t, \text{tr}\left(\mathbf{P}_t^{\text{IMM}}\right), \{\mu_{t-1}^i\}_{i=1}^N, \{\mu_t^i\}_{i=1}^N, \sum_{i=1}^N \mu_t^i \tilde{\mathbf{y}}_t\right). \tag{14}$$

  In (14), $\mathbf{z}_t$ is the estimated state at current time step, which is an important information because the current state determines the development in future to a large extent, $\text{tr}\left(\mathbf{P}_t^{\text{IMM}}\right)$ is the trace of the covariance matrix

[2]In practice, a stochastic policy is usually preferred than deterministic policy, where the output of π_θ is a distribution of action rather than the action itself (e.g., expectation and standard deviation of a Gaussian distribution for continuous action space and categorical probabilities for discrete action space.). The actual action is sampled from this distribution.

of the IMM estimate, which characterizes how accurate the estimate $\mathbf{z}_t$ is, $\{\mu_{t-1}^i\}_{i=1}^N$ and $\{\mu_t^i\}_{i=1}^N$ are the model probabilities in previous and current time steps, respectively. This information is important to the decision because the model probabilities and their changes impact the predictability significantly. Two examples are, 1) going straight is a much more stable state than turning, 2) drastic change of state probabilities cannot be explained by model probability mixing (35), which means an update with measurement is necessary for accurate prediction in future. $\sum_{i=1}^N \mu_t^i \tilde{\mathbf{y}}_t$ is the mean innovation weighted by the model probabilities and is the measure of how prediction deviates from measurement on average.

- The action at time step t is the transmission power P_t. The action space is therefore a one dimensional continuous space in range $(0, p_{\max})$ where $p_{\max}$ is the maximum transmission power.
- The reward at time step t is defined as

$$\begin{aligned} r_t = &- P_t - w_1 \mathbf{I}(\text{tr}(\mathbf{P}_t^{\text{IMM}}) > p_{\text{th}}) \\ &- w_2 \max(0, \text{tr}(\mathbf{P}_t^{\text{IMM}}) - p_{\text{th}}), \end{aligned} \tag{15}$$

where w_1 and w_2 are coefficients of the penalty terms. The first term is to encourage low transmission power. The second term is to ensure $\text{tr}(\mathbf{P}_t^{\text{IMM}})$ does not exceed p_{th} by setting a steep reward change at the threshold. The third term is to provide a gradient towards the correct direction.

The choice of w_1 and w_2 is empirical and should meet the following two criteria:

- The values should be sufficiently large such that the agent learns it should not save transmit power at the cost of violating the precision constraint.
- The values should not be unnecessarily large in order to avoid numerical instability.

Note that the RL formulation (13) and the reward definition (15) do not explicitly implement the constraint in (7). Instead, it issues a penalty when the constraint is violated. When the penalty is considerably higher than the saved transmit power (the first term in (15)), the agent will learn not to save energy and violate the constraint. This fact will be illustrated in Section V.

Three important terms of RL are value, Q-value and advantage, which are briefly elaborated as follows:

- The value function is defined as the expected discounted sum of rewards beginning from the given state s and following policy π_θ, i.e.,

$$V^{\pi_\theta}(s) = \mathbb{E}\left(\sum_{t=1}^{T} \gamma^t r(s_t, a_t) \middle| s_1 = s, a_t = \pi_\theta(s_t) \right), \tag{16}$$

  where $\gamma \in [0, 1)$ is the discounting factor. The value function measures how good policy π_θ is given state s.
- The Q-value is defined as the expected discounted sum of rewards beginning from the given state s and choosing action a at the current time step, then following policy π_θ, i.e.,

$$\begin{aligned} Q^{\pi_\theta}(s, a) = \mathbb{E}\Bigg(\sum_{t=1}^{T} \gamma^t r(s_t, a_t) \Bigg| s_1 = s, a_1 = a, \\ a_t = \pi_\theta(s_t) \text{ for } t > 1 \Bigg). \end{aligned} \tag{17}$$

  For algorithms in Q-learning style, the Q-value is used to determine the policy, e.g., $\pi_\theta(s) = \arg\max_a Q^{\pi_\theta}(s, a)$. For policy optimization algorithms, the difference between value $V^{\pi_\theta}(s)$ and Q-value $Q^{\pi_\theta}(s, a)$ provides a comparison between policy π_θ and action a because the only difference between value and Q-value is whether to choose action at first time step according to policy π_θ (value) or using the given action a (Q-value).
- The advantage function is defined as the difference between Q-value and value:

$$A^{\pi_\theta}(s, a) = Q^{\pi_\theta}(s, a) - V^{\pi_\theta}(s). \tag{18}$$

  Intuitively, a positive advantage indicates action a is better than the action return by the policy $\pi_\theta(s)$ and we should optimize the policy such that action a appears more often given state s. The same holds true vice versa.

B. Proximal Policy Optimization and Generalized Advantage Estimation

We briefly elaborate PPO and GAE in this section. The readers are referred to [39] and [40] for more details.

PPO uses stochastic policy, i.e., the policy determines a distribution rather than a deterministic value of the action and the actual action is sampled from the distribution. For continuous actions (our case), we usually use the Gaussian distribution, where the expectation is the output of the policy network and the variance is a constant. In each iteration, data samples $\{(s_t, a_t, r_t, s_{t+1})\}_t$ are collected with interaction with the environment. The policy is updated in such a way, that probabilities of actions with positive advantages are increased and probabilities of actions with negative advantages are decreased, i.e., the expectation given the state moves towards actions with positive advantages and away from actions with negative advantages. This can be realized with the following objective:

$$\max_\theta \sum_{s,a} \frac{\pi_\theta(a|s)}{\pi_{\theta_{\text{old}}}(a|s)} A^{\pi_{\theta_{\text{old}}}}. \tag{19}$$

The policy optimization is subject to the constraint that the KL-divergence between old and new distributions is smaller than a threshold in order to avoid dramatic changes of policy and make the training stable. Since the direct constraint optimization is difficult (implemented by trust region policy optimization [41]), a cleverly defined objective function is applied such that the gradient is nonzero only when $\pi_\theta(a|s)/\pi_{\theta_{\text{old}}}(a|s)$ is between $1 - \epsilon$ and $1 + \epsilon$, where ϵ is a positive number controlling the region size, in which the policy optimization can be performed.

The advantage is estimated with the GAE, which realizes an optimal bias-variance trade-off. Using the Bellman equation, the Q-value can be expressed as

$$A^{\pi_\theta}(s_t, a_t) = -V^{\pi_\theta}(s_t) + \sum_{i=0}^{I-1} \gamma^i r_{t+i} + \gamma^I V^{\pi_\theta}(s_{t+I}), \quad (20)$$

where $V^{\pi_\theta}(s)$ is estimated with the value network[3], r_t are sampled with the interaction with the environment[4], I is an integer tuning the bias-variance trade-off. If $I = 1$, $A^{\pi_\theta}(s_t, a_t) = -V^{\pi_\theta}(s_t) + r_t + \gamma V^{\pi_\theta}(s_{t+1})$ has high bias and low variance. If $I \to \infty$, $A^{\pi_\theta}(s_t, a_t) = -V^{\pi_\theta}(s_t) + \sum_{i=0}^{\infty} \gamma^i r_{t+i}$ has high variance and low bias. GAE introduces a parameter λ realizing a compromise between the two extreme cases. The estimated advantages are used by PPO for the policy optimization.

C. Soft Actor-Critic

We briefly elaborate SAC in this section. The readers are referred to [42] for more details.

SAC also uses stochastic policy. However, unlike PPO, which has a constant variance of action distribution, the variance of the distribution is also an output of the policy network in SAC. As a result, SAC is able to control whether to explore (high variance) or to do fine tuning (low variance). Due to possible local optima in policy and difficulty in exploration, SAC maximizes weighted sum of rewards and policy entropy, which increases with higher variance of action. In this way, the action space is more actively explored until a policy of significant advantage is found. Following this idea, the definition of the state value (14) is modified as

$$V^{\pi_\theta}(s) = \mathbb{E}\left(\sum_{t=1}^{T} \gamma^t r(s_t, a_t) + \alpha H(\pi(\cdot|s_t)) \middle| \; s_1 = s, a_t = \pi_\theta(s_t) \right), \quad (21)$$

where α is the coefficient of the entropy, $H(\pi(\cdot|s_t))$ is the entropy of policy π in state s_t, which is higher when the action variance is higher. The definition of the Q-value is similarly changed compared to (17). Since the Q-values are often dramatically overestimated, SAC maintains two independent Q-networks and uses the smaller value as the Q-value to learn the policy.

With the learned Q-values, the policy is learned such that the KL-divergence between the Q-value and the policy is minimized, i.e.,

$$\pi_{\text{new}} = \arg\min_{\pi' \in \Pi} \mathrm{D}_{\mathrm{KL}} \left(\pi'(\cdot|s) \middle\| \frac{\exp\left(\frac{1}{\alpha} Q^{\pi_{\text{old}}}(s, \cdot)\right)}{Z^{\pi_{\text{old}}}(s)} \right), \quad (22)$$

where $Z^{\pi_{\text{old}}}(s_t)$ is the normalization factor since $\exp\left(\frac{1}{\alpha} Q^{\pi_{\text{old}}}(s_t, \cdot)\right)$ is not a distribution, but it can be safely ignored because a constant does not contribute to the gradient. With a small KL-divergence, the action probability is high where the Q-value is high given each state.

The reparametrization trick is applied such that the expectation of the Q-value is not over the distribution of actions parameterized by θ but over distribution over the action noise, which is independent from θ.

[3]It is deterministic with given s but can be biased because it is estimated, therefore it has high bias and low variance.

[4]They are unbiased because they are true rewards returned from the environment and have high variance because they depend on stochastic actions and randomness in environment.

V. Training and Evaluation Results

In this section, we present the training and evaluation results. We use the open source implementation of RL algorithms Stable Baselines 3 [43]. Important environment and algorithm parameters are shown in Table I. For simplicity, we assume the measurement matrix $\mathbf{H}$ is an identity matrix $\mathbf{I}$. The Rayleigh fading channel is chosen as an example in this section because it has the lowest diversity level and therefore the worst reliability. The parameter of the Rayleigh distribution can be calibrated for individual RSUs separately such that the applied Rayleigh channel model is the worst possible channel in the responsible area of the RSU. The outage probability at time step t is

$$p_t^{\text{out}} = 1 - \exp\left(\frac{-(2^{R/W} - 1) W N_0}{2 P_t \sigma^2} \right), \quad (23)$$

where h is a Rayleigh distributed random variable with CDF of $1 - \exp(-h^2/(2\sigma^2))$ where σ tunes the mean channel gain. The other symbols in (23) is the same as (8).

When the environment is reset, it is randomly determined whether the vehicle is going straight forward or turning right (for the first scenario) or whether the vehicle is keeping lane or changing lane to left or right (for the second scenario). If the vehicle is changing lane, it is randomly determined where (i.e., the y coordinate) it should happen as well. In each step, the true vehicle state is updated with (2) (if the vehicle is going straight) or (4) and (5) (if the vehicle is turning). If the vehicle is changing to the left lane, it first turns left and then right such that the driving direction is unchanged and the vehicle position is laterally move by the lane width. The similar process applies to lane changing to the right lane. After the true vehicle state is determined, the existing histories are updated without measurement and their probabilities are multiplied by the current communication outage probability. A new history is appended with measurement (for details of this consideration, see Section III-B) and probability that the communication is successful. The expected vehicle state and the estimate variance are computed in (10) and (11), respectively.

Fig. 5 shows the improvement of the episode reward during training in the intersection scenario. Both algorithms achieve significant improvement and SAC shows significantly higher sample efficiency due to its off-policy property. SAC also realizes a higher episode reward at the end of training. Since both algorithms are successful to keep $\mathrm{tr}\left(\mathbb{E}\left(\mathbf{P}_t^{\text{IMM}}\right)\right)$ below p_{th}, the last two terms of (15) are 0 at every time step. The episode reward is simply the negative sum of transmission power. It can be observed that SAC meets the precision requirement with less transmission power than PPO. However,

TABLE I
ENVIRONMENT AND TRAINING PARAMETERS

Parameter	Value
Time resolution	0.1 s
Initial state	(0, -10, 0, 5)
Mean channel gain in Rayleigh fading channel	-50 dB
$P_{\max}$	100 mW
$\mathbf{\Pi}$ in intersection scenario	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ [5]
Initial model probabilities in intersection scenario	(0.5, 0.5)
$\mathbf{\Pi}$ in lane changing scenario	$\begin{pmatrix} 0.999 & 0.0005 & 0.0005 \\ 0.2 & 0.6 & 0.2 \\ 0.2 & 0.2 & 0.6 \end{pmatrix}$ [6]
Initial model probabilities in lane changing scenario	(0.999, 0.0005, 0.0005)
Process noise	0.0005 (m for position and m/s for velocity)
Measurement noise	0.0005 (m for position and m/s for velocity)
p_{th}	0.01 (m for position and m/s for velocity)
w_1	10
w_2	100
Learning rate	10^{-5}
Training steps	2×10^6
Entropy in SAC	0.1
ϵ (clipping range described under (19))	0.1
Batch size	1024
Number of layers in the neural network	2
Number of neurons per layer	64
Activation function	ReLU

[5] Two models in the intersection scenarios are going straight and turning right. The state transition matrix is an identity matrix because once the decision is made whether to go straight or to turn right, it cannot be changed.
[6] Three models in the lane changing scenarios are going straight, turning left and turning right. The model transition probabilities from going straight to turning left and right are small because lane changing are rare compared to going straight. On the contrary, turning left or right is not a stable state and has higher probabilities of changing to another model.

it is also to note that the computational complexity and hence the training time consumption of SAC is significantly higher than PPO.

Fig. 6 shows the true trajectory, estimated trajectory, standard deviation of estimation error as well as transmission power (depicted as red lines perpendicular to the driving direction) in the intersection scenario. With both algorithms, transmission power is low when the vehicle is outside the intersection and the transmission power increases when the vehicle enters the intersection and unveils its intended driving direction (straight or right).

As shown in Table I, the initial model probabilities in the intersection scenario are $(0.5, 0.5)$ and stay unchanged until the vehicle enters the intersection because the two models have the same prediction before the intersection. Once the vehicles enter the intersection, the two models have different predictions and $\mathbf{P}_t^{\text{IMM}}$ would be very high if the estimation is done without measurement (i.e., with prediction only). Only when the model probabilities are updated with measurements and it becomes unambiguous in which direction the vehicle is heading, $\mathbf{P}_t^{\text{IMM}}$ can be kept low again. This is the reason why

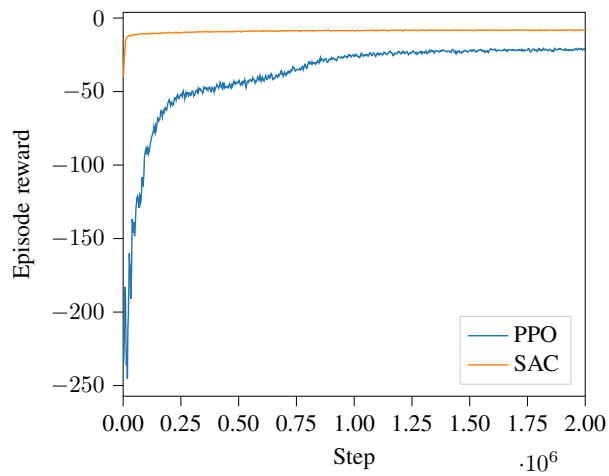


Fig. 5. Episode reward improvement in training in the intersection scenario.

the transmission power at the beginning of the intersection is high. Ater that, the transmission power can be reduced to the level before the intersection because the model transition matrix $\mathbf{\Pi}$ in this scenario is an identity matrix. Although adaptive scheduling schemes are suggested for remote sensing, e.g., in [10], the proposed methods are highly empirical and the scheme in [10] is based on the velocity, which is not suitable for the considered scenario here because the predictability is not decided by the velocity but by the vehicle position, i.e., whether the vehicle is inside or outside the intersection.

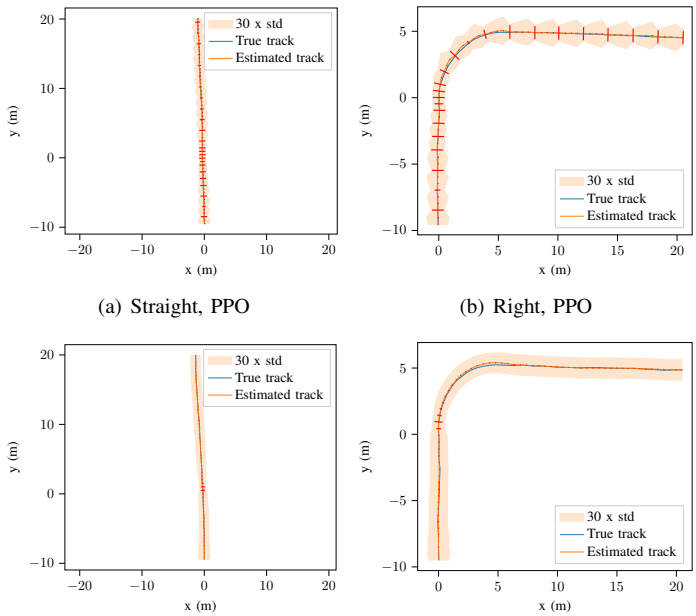
Another interesting observation is that PPO and SAC find different solutions on the straight lanes. While PPO chooses a higher transmission power and a higher interval between two transmission, SAC decides to transmit in every time step with a low transmission power. According to Fig. 5, the total transmission power realized with SAC is lower. Two possible reasons for this advantage are

- SAC is an entropy-regulated learning algorithm, which makes it less likely to get stuck in a local optimum.
- SAC tunes the variance of the action distribution, which makes fine-tuning (reducing the transmission power while keeping the estimate variance below the threshold) easier than PPO, which uses a constant action variance.

Fig. 7 depicts the improvement of the episode reward during training in the lane changing scenario. Similar to Fig. 5, SAC has a higher sample efficiency and achieves a lower total transmission power in the end of the training while keeping the estimation precision below the threshold.

Evaluation results with true trajectory, estimated trajectory, standard deviation of estimation and transmission power are presented in Fig. 8. The results are similar to Fig. 6 when the vehicle is driving straight.

Unlike in the intersection scenario, the lane changing behavior does not have a fixed trajectory. Therefore, the model transition probability matrix $\mathbf{\Pi}$ in this scenario is not an identity matrix. Instead, the models turning left and turning right in the lane changing scenario do have certain probabilities of changing to other models (in the reality, the trajectory in lane changing is not fully predictable even if you know



(a) Straight, PPO (b) Right, PPO

(c) Straight, SAC (d) Right, SAC

Fig. 6. Testing results in the intersection scenario. The orange area shows 30 times standard deviation of the estimate. A wide orange area indicates a low estimate precision.

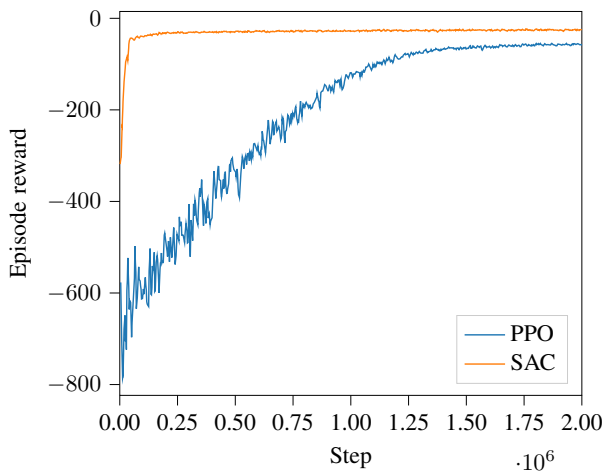


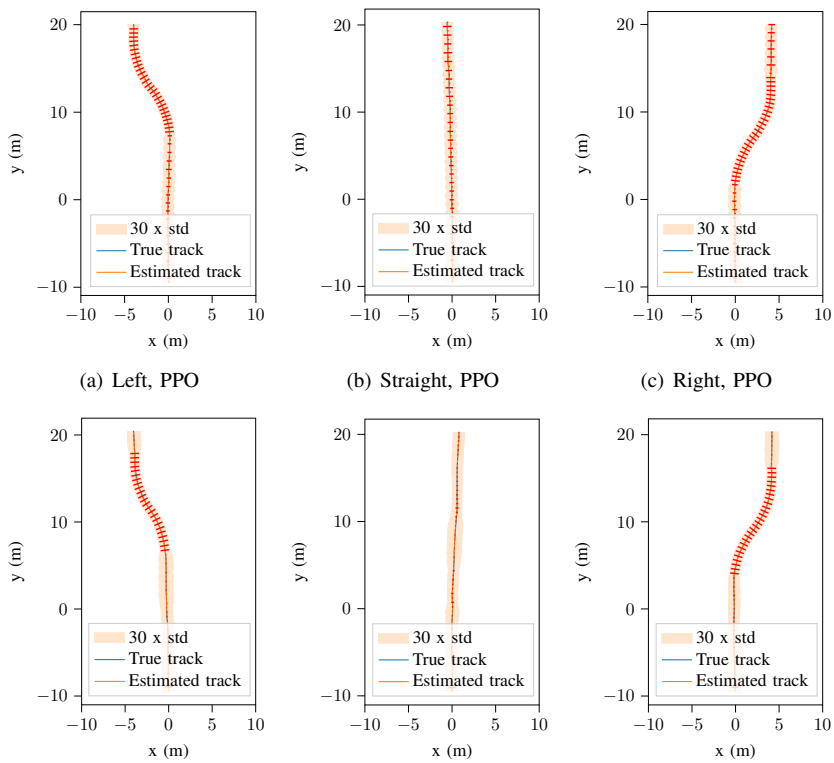
Fig. 7. Episode reward improvement in training in the lane changing scenario.

where the lane changing begins to take place). As a result, the transmission power is kept high throughout the lane changing (compared to it, the transmission power is reduced to a low level once it becomes clear where the vehicle is heading to in the intersection scenario). Only when the lane changing is finished and the model is going straight gains a high probability again, the transmission power is reduced to a low level.

As a comparison to the RL algorithms, we choose three constant transmission powers in every time step. These transmission powers are 10 mW, 50 mW and 100 mW. The evaluation results are shown in Fig. 9. With a transmission power of 10 mW, the estimate variance requirement cannot be fulfilled in both scenarios. With a transmission power of 50 mW, the estimate variance requirement is met in the intersection scenario but not in the lane changing scenario. With a transmission power of 100 mW, the estimate variance is kept below the threshold in both scenarios but the total transmission power is considerably higher than with the two RL algorithms. All the base line performances are significantly higher than the performances of PPO and SAC. A complete comparison of the base lines schemes and the RL algorithms is presented in Fig. 10.

VI. Conclusion

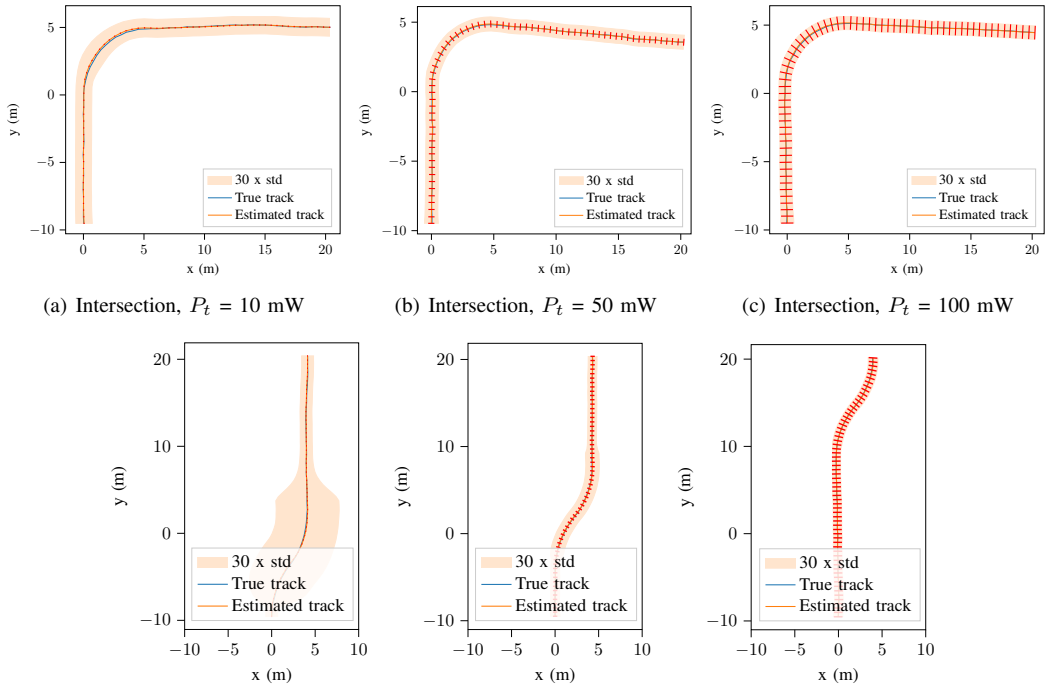
In this paper, we propose a DRL based transmission power control for remote vehicle state estimation and transmission, such that the estimation precision at the adjacent traffic participants is high enough to guarantee safety. The traffic awareness is the prerequisite of autonomous driving. Due to the limitation of precision, range and cost of the onboard sensors, remote estimation on an RSU or other vehicles is a promising solution. However, the estimate has to be transmitted to other vehicles with wireless communication, which has an inherent outage probability due to channel fading. When the communication fails, the other vehicles have to predict the vehicle state with a low precision. We propose to use KF, EKF and IMM in complicated traffic scenarios for vehicle state estimation and to use DRL to optimize the transmission power, such that the transmission power is minimized and the expected estimation variance is lower than a given threshold. Two state-of-the-art algorithms, PPO and SAC are applied. Evaluation results show that both algorithms outperform baselines with different but constant transmission power. The transmission power is increased when the vehicle behavior becomes less predictable. Compared to PPO, SAC has a higher sample efficiency and realizes a lower total transmission power. The contribution of this paper is to build a bridge between vehicular commu-



(a) Left, PPO (b) Straight, PPO (c) Right, PPO

(d) Left, SAC (e) Straight, SAC (f) Right, SAC

Fig. 8. Testing results in the lane changing scenario.



(a) Intersection, P_t = 10 mW (b) Intersection, P_t = 50 mW (c) Intersection, P_t = 100 mW

(d) Lane changing, P_t = 10 mW (e) Lane changing, P_t = 50 mW (f) Lane changing, P_t = 100 mW

Fig. 9. Baselines performances with constant transmission power.

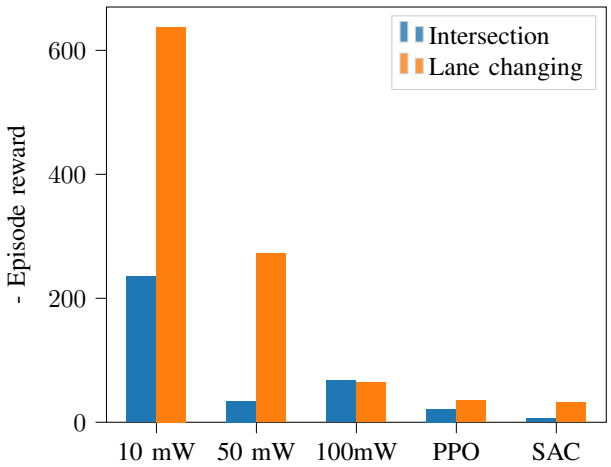


Fig. 10. Comparison of performances between base lines with constant transmission powers and the RL algorithms.

nication and traffic awareness and to solve the complicated optimization problem with the DRL algorithms.

The source code of this work is available under https://github.com/bilepeng/scheduling_remote_state_estimation_drl.

Appendix A
Formulation of Kalman Filter

Given the state $\mathbf{x}_{t-1}$ at time step $t-1$, the state $\mathbf{x}_t$ at time step t is

$$\mathbf{x}_t = F(\mathbf{x}_{t-1}) + \mathbf{w}_t, \tag{24}$$

where F is the state-transition function and $\mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{N})$ is the inherent randomness of the process and is assumed normally distributed with expectation $\mathbf{0}$ and covariance matrix $\mathbf{N}$. If F is a linear function and can be expressed as product between a matrix and $\mathbf{x}_{t-1}$, the process is described by

$$\mathbf{x}_t = \mathbf{F}\mathbf{x}_{t-1} + \mathbf{w}_t, \tag{25}$$

where $\mathbf{F}$ is the state-transition matrix.

Assuming a linear measurement model, the measurement at time t is

$$\mathbf{z}_t = \mathbf{H}\mathbf{x}_t + \mathbf{v}_t, \tag{26}$$

where $\mathbf{H}$ is the measurement matrix and $\mathbf{v}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_t)$ is the normally distributed measurement noise, with $\mathbf{R}_t$ being the covariance matrix of measurement noise at time step t.

When F is linear, we can use the Kalman filter to estimate $\mathbf{x}_t$ from the noisy measurement. Given the estimated state $\hat{\mathbf{x}}_{t-1}$ at time step $t-1$, the prior estimate of $\mathbf{x}_t$ is[5]

$$\hat{\mathbf{x}}_{t|t-1} = \mathbf{F}\hat{\mathbf{x}}_{t-1}. \tag{27}$$

The covariance matrix of the prior estimate is

$$\mathbf{P}_{t|t-1} = \mathbf{F}\mathbf{P}_{t-1}\mathbf{F}^T + \mathbf{N}. \tag{28}$$

[5]We use $\hat{\mathbf{x}}_{t|t-1}$ to specifically indicate the estimate of $\mathbf{x}_t$ is based on measurements up to time step $t-1$. If the condition is not clear, it is omitted in the notation. For example, $\hat{\mathbf{x}}_{t-1}$ is the estimate of $\mathbf{x}_{t-1}$ without specification based on which measurements the estimate is made.

The posterior estimate is computed as

$$\hat{\mathbf{x}}_{t|t} = \hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t\tilde{\mathbf{y}}_t, \tag{29}$$

where $\mathbf{K}_t$ is the Kalman gain, which is computed as

$$\mathbf{K}_t = \mathbf{P}_{t|t-1}\mathbf{H}^T[\mathbf{H}\mathbf{P}_{t|t-1}\mathbf{H}^T + \mathbf{R}_t]^{-1} \tag{30}$$

and $\tilde{\mathbf{y}}_t$ is the innovation, which is computed as

$$\tilde{\mathbf{y}}_t = \mathbf{z}_t - \mathbf{H}\hat{\mathbf{x}}_{t|t-1}. \tag{31}$$

The posterior covariance matrix is computed as

$$\mathbf{P}_{t|t} = (\mathbf{I} - \mathbf{K}_t\mathbf{H})\mathbf{P}_{t|t-1}. \tag{32}$$

In general, the covariance increases when making predictions because of the process noise and decreases after making measurement because it contains new information about $\mathbf{x}_t$. Sufficient measurements are necessary in order to maintain a small estimate covariance.

Appendix B
Formulation of Extended Kalman Filter

In some cases, the state-transition function F is nonlinear, the KF cannot be applied. Instead, we can use the EKF, which linearizes the state transition (24). The prediction (corresponding to (27)) is

$$\hat{\mathbf{x}}_{t|t-1} = f(\hat{\mathbf{x}}_{t-1}) + \mathbf{w}_t \tag{33}$$

and the linearized prediction covariance (corresponding to (28)) is

$$\mathbf{P}_{t|t-1} = \mathbf{F}(\hat{\mathbf{x}}_{t-1})\mathbf{P}_{t-1}\mathbf{F}^T(\hat{\mathbf{x}}_{t-1}) + \mathbf{N}, \tag{34}$$

where $\mathbf{F}(\hat{\mathbf{x}}_{t-1})$ is the Jacobian matrix of F at point $\hat{\mathbf{x}}_{t-1}$, i.e., let $\overline{\mathbf{x}_t} = F(\mathbf{x}_{t-1})$, the element f_{ij} in row i and column j of $\mathbf{F}(\mathbf{x}_{t-1})$ is $f_{ij}(\mathbf{x}_{t-1}) = \frac{\partial \overline{x_{t,i}}}{\partial x_{t-1,j}}$, where $\overline{x_{t,i}}$ and $x_{t-1,j}$ are ith and jth elements of $\overline{\mathbf{x}_t}$ and $\mathbf{x}_{t-1}$, respectively. Therefore, $\mathbf{F}(\hat{\mathbf{x}}_{t-1})\mathbf{P}_{t-1}\mathbf{F}^T(\hat{\mathbf{x}}_{t-1})$ is a linearized local approximation of the covariance matrix of the estimate of $\hat{\mathbf{x}}_{t|t-1}$.

Assuming the linear measurement model (26), the updating described by (29) - (32) is identical in the KF.

Appendix C
Formulation of Interacting Multiple Model

If there are multiple possible state-transition functions, it is impossible to use a single Bayesian filter to estimate the state. In this case, the IMM can be applied [44]. In IMM, a set of models is defined, where each model is a Bayesian filter and has a certain probability at each time step. Let M be the set of model indices and $i, j \in M$. A transition matrix $\mathbf{\Pi}$ defines the prior probabilities that a model is switched to another model, where the element π_{ji} is the probability of model i at the current time step given model j at the previous time step. At the beginning of each time step, the mixing probability from model j to model i is computed as

$$\mu^{ji}_{t-1|t-1} = \frac{\pi_{ji}\mu^j_{t-1}}{\sum_{l=1}^N \pi_{li}\mu^l_{t-1}}, \tag{35}$$

where N is the number of models and μ^l_{t-1} is the probability of model l at time step $t-1$. Intuitively, $\mu^{ji}_{t-1|t-1}$ is the prior

probability of model j at time step $t-1$ switching to model i at time step t given model i at time step t.

The mixed state estimates and the corresponding covariances are computed as

$$\hat{\mathbf{x}}_{t-1}^{0i} = \sum_{j=1}^{N} \mu_{t-1|t-1}^{ji} \hat{\mathbf{x}}_{t-1}^{j} \tag{36}$$

and

$$\mathbf{P}_{t-1}^{0i} = \sum_{j=1}^{N} \mu_{t-1|t-1}^{ji} \Big(\mathbf{P}_{t-1}^{j} + (\hat{\mathbf{x}}_{t-1}^{j} - \hat{\mathbf{x}}_{t-1}^{0i})(\hat{\mathbf{x}}_{t-1}^{j} - \hat{\mathbf{x}}_{t-1}^{0i})^T \Big), \tag{37}$$

where $\hat{\mathbf{x}}_{t-1}^{j}$ is the estimate of $\hat{\mathbf{x}}_{t-1}$ by model j, $\mathbf{P}_{t-1}^{j}$ is the estimate covariance by model j.

In the next step, the current state $\hat{\mathbf{x}}_{t|t-1}^{i}$ will be predicted with model i based on the mixed previous state with (27) or (33), depending on whether the state transition is linear or nonlinear, where $\hat{\mathbf{x}}_{t-1}$ is $\hat{\mathbf{x}}_{t-1}^{0i}$ for model i computed with (36). Similarly, the covariance of prediction $\mathbf{P}_{t|t-1}^{i}$ with model i is computed with (28) or (34), depending on whether the state transition is linear or nonlinear as well, where $\mathbf{P}_{t-1}$ is $\mathbf{P}_{t-1}^{0i}$ for model i computed with (37).

If measurement is available, the prediction and the covariance of model i are updated with (29) and (32), producing $\hat{\mathbf{x}}_{t|t}^{i}$ and $\mathbf{P}_{t|t}^{i}$, respectively. The model probabilities are computed as

$$\mu_{t|t}^{i} = \frac{\mathcal{N}(\mathbf{z}_t; \mathbf{H}\hat{\mathbf{x}}_{t|t}^{i}, \mathbf{P}_{t|t}^{i}) \sum_{j=1}^{N} \pi_{ji} \mu_{t-1}^{j}}{\sum_{l=1}^{N} \mathcal{N}(\mathbf{z}_t; \mathbf{H}\hat{\mathbf{x}}_{t|t}^{l}, \mathbf{P}_{t|t}^{l}) \sum_{j=1}^{N} \pi_{jl} \mu_{t-1}^{j}}, \tag{38}$$

where $\mathcal{N}(\mathbf{z}; \boldsymbol{\mu}, \boldsymbol{\sigma}^2)$ is the probability density at $\mathbf{z}$ given the normal distribution with expectation $\boldsymbol{\mu}$ and variance $\boldsymbol{\sigma}^2$. If no measurement is available, the model probabilities are computed as

$$\mu_{t|t-1}^{i} = \sum_{j=1}^{N} \pi_{ji} \mu_{t-1}^{j}. \tag{39}$$

Finally, the output estimate is computed as

$$\hat{\mathbf{x}}_{t}^{\text{IMM}} = \sum_{i=1}^{N} \mu_{t}^{i} \hat{\mathbf{x}}_{t}^{i} \tag{40}$$

and the covariance is computed as

$$\mathbf{P}_{t}^{\text{IMM}} = \sum_{i=1}^{N} \mu_{t}^{i} \Big(\mathbf{P}_{t}^{i} + (\hat{\mathbf{x}}_{t} - \hat{\mathbf{x}}_{t}^{i})^T (\hat{\mathbf{x}}_{t} - \hat{\mathbf{x}}_{t}^{i}) \Big). \tag{41}$$

References

[1] S.-W. Kim and W. Liu, "Cooperative autonomous driving: A mirror neuron inspired intention awareness and cooperative perception approach," *IEEE Intelligent Transportation Systems Magazine*, vol. 8, no. 3, pp. 23–32, 2016.

[2] D. Caveney, "Cooperative vehicular safety applications," *IEEE Control Systems Magazine*, vol. 30, no. 4, pp. 38–53, 2010.

[3] E. Steinmetz, R. Emardson, F. Brannstrom, and H. Wymeersch, "Theoretical limits on cooperative positioning in mixed traffic.," *IEEE Access*, vol. 7, pp. 49 712–49 725, 2019.

[4] H. N. Mahjoub, B. Toghi, and Y. P. Fallah, "A driver behavior modeling structure based on nonparametric Bayesian stochastic hybrid architecture," in *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*, IEEE, 2018, pp. 1–5.

[5] D. Shin, S. Yi, K. M. Park, and M. Park, "An interacting multiple model approach for target intent estimation at urban intersection for application to automated driving vehicle," *Applied Sciences (Switzerland)*, vol. 10, no. 6, 2020. DOI: 10.3390/app10062138.

[6] M. Fröhle, C. Lindberg, K. Granström, and H. Wymeersch, "Multisensor Poisson multi-Bernoulli filter for joint target–sensor state tracking," *IEEE Transactions on Intelligent Vehicles*, vol. 4, no. 4, pp. 609–621, 2019.

[7] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. I. Jordan, and S. S. Sastry, "Kalman filtering with intermittent observations," *IEEE transactions on Automatic Control*, vol. 49, no. 9, pp. 1453–1464, 2004.

[8] L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla, and S. S. Sastry, "Foundations of control and estimation over lossy networks," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 163–187, 2007. DOI: 10.1109/JPROC.2006.887306.

[9] Y. Mostofi and R. M. Murray, "To drop or not to drop: Design principles for Kalman filtering over wireless fading channels," *IEEE Transactions on Automatic Control*, vol. 54, no. 2, pp. 376–381, 2009. DOI: 10.1109/TAC.2008.2008331.

[10] Y. Liu and Z. Sun, "EKF-based adaptive sensor scheduling for target tracking," *2008 International Symposium on Information Science and Engineering, ISISE 2008*, vol. 2, pp. 171–174, 2008. DOI: 10.1109/ISISE.2008.286.

[11] W. Xiao, J. K. Wu, and L. Xie, "Adaptive sensor scheduling for target tracking in wireless sensor network," *Advanced Signal Processing Algorithms, Architectures, and Implementations XV*, vol. 5910, no. September 2005, 59100B, 2005. DOI: 10.1117/12.618124.

[12] C. Yang and L. Shi, "Deterministic Sensor Data Scheduling Under Limited Communication Resource," vol. 59, no. 10, pp. 5050–5056, 2011.

[13] P. Chavali, S. Member, and A. Nehorai, "Scheduling and Power Allocation in a Cognitive Radar Network for Multiple-Target Tracking," vol. 60, no. 2, pp. 715–729, 2012.

[14] J. Wu, Y. Li, D. E. Quevedo, V. Lau, and L. Shi, "Data-driven power control for state estimation: A bayesian inference approach," *Automatica*, vol. 54, pp. 332–339, 2015.

[15] Y. Li, J. Wu, and T. Chen, "Transmit power control and remote state estimation with sensor networks: A bayesian inference approach," *Automatica*, vol. 97, pp. 292–300, 2018.

[16] A. Arafa, K. Banawan, K. G. Seddik, and H. V. Poor, "Sample, Quantize and Encode: Timely Estimation Over Noisy Channels," pp. 1–29, 2020. arXiv: 2007.10200.

[17] T. Sui, D. Marelli, X. Sun, and M. Fu, "Multi-sensor state estimation over lossy channels using coded measurements," *Automatica*, vol. 111, p. 108 561, 2020.

[18] L. Jie, "Distributed fusion state estimation based on coded measurements," in *2020 International Conference on Computer Engineering and Application (ICCEA)*, IEEE, 2020, pp. 96–99.

[19] O. Ayan, M. Vilgelm, M. Klügel, S. Hirche, and W. Kellerer, "Age-of-information vs. Value-of-information scheduling for cellular networked control systems," *ICCPS 2019 - Proceedings of the 2019 ACM/IEEE International Conference on Cyber-Physical Systems*, pp. 109–117, 2019. DOI: 10.1145/3302509.3311050. arXiv: 1903.05356.

[20] M. Nasri, M. Kargahi, and M. Mohaqeqi, "Scheduling of accuracy-constrained real-time systems in dynamic environments," *IEEE Embedded Systems Letters*, vol. 4, no. 3, pp. 61–64, 2012. DOI: 10.1109/LES.2012.2195294.

[21] R. Talak, S. Karaman, and E. Modiano, "Minimizing age-of-information in multi-hop wireless networks," in *2017 55th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, IEEE, 2017, pp. 486–493.

[22] C. Kam, S. Kompella, G. D. Nguyen, J. E. Wieselthier, and A. Ephremides, "On the age of information with packet deadlines," *IEEE Transactions on Information Theory*, vol. 64, no. 9, pp. 6419–6428, 2018.

[23] A. M. Bedewy, Y. Sun, and N. B. Shroff, "Optimizing data freshness, throughput, and delay in multi-server information-update systems," in *2016 IEEE International Symposium on Information Theory (ISIT)*, IEEE, 2016, pp. 2569–2573.

[24] L. Dai, R. Jiao, F. Adachi, H. V. Poor, and L. Hanzo, "Deep learning for wireless communications: An emerging interdisciplinary paradigm," *IEEE Wireless Communications*, vol. 27, no. 4, pp. 133–139, 2020.

[25] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26–38, 2017.

[26] R. F. Atallah, C. M. Assi, and M. J. Khabbaz, "Scheduling the operation of a connected vehicular network using deep reinforcement learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 5, pp. 1669–1682, 2018.

[27] H. Ye, G. Y. Li, and B.-h. F. Juang, "Deep reinforcement learning based resource allocation for V2V communications," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 4, pp. 3163–3173, 2019. DOI: 10.1109/TVT.2019.2897134.

[28] B. Peng, G. Seco-Granados, E. Steinmetz, M. Fröhle, and H. W. Wymeersch, "Decentralized Scheduling for Cooperative Localization With Deep Reinforcement Learning," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 4295–4305, 2019. DOI: 10.1109/TVT.2019.2913695.

[29] J. Li and X. Zhang, "Deep Reinforcement Learning-Based Joint Scheduling of eMBB and URLLC in 5G Networks," *IEEE Wireless Communications Letters*, vol. 9, no. 9, pp. 1543–1546, 2020. DOI: 10.1109/LWC.2020.2997036.

[30] J. Song, B. Peng, C. Hager, H. Wymeersch, and A. Sahai, "Learning Physical-Layer Communication with Quantized Feedback," *IEEE Transactions on Communications*, vol. 68, no. 1, pp. 645–653, 2020. DOI: 10.1109/TCOMM.2019.2951563. arXiv: 1904.09252.

[31] Y. Liu, Z. Jiang, S. Zhang, and S. Xu, "Deep Reinforcement Learning-Based Beam Tracking for Low-Latency Services in Vehicular Networks," *IEEE International Conference on Communications*, vol. 2020-June, 2020. DOI: 10.1109/ICC40277.2020.9148759. arXiv: 2002.05564.

[32] M. Kountouris and N. Pappas, "Semantics-empowered communication for networked intelligent systems," *IEEE Communications Magazine*, vol. 59, no. 6, pp. 96–102, 2021.

[33] B. Güler, A. Yener, and A. Swami, "The semantic communication game," *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 4, pp. 787–802, 2018.

[34] A. Molin, H. Esen, and K. H. Johansson, "Scheduling networked state estimators based on value of information," *Automatica*, vol. 110, p. 108 578, 2019.

[35] A. Ndikumana, N. H. Tran, T. M. Ho, Z. Han, W. Saad, D. Niyato, and C. S. Hong, "Joint communication, computation, caching, and control in big data multi-access edge computing," *IEEE Transactions on Mobile Computing*, vol. 19, no. 6, pp. 1359–1374, 2019.

[36] G. Welch, G. Bishop, *et al.*, "An introduction to the Kalman filter," 1995.

[37] S. S. Blackman, "Multiple hypothesis tracking for multiple target tracking," *IEEE Aerospace and Electronic Systems Magazine*, vol. 19, no. 1, pp. 5–18, 2004.

[38] Available at http://www.control.isy.liu.se/student/graduate/TargetTracking/IMMderivation.pdf.

[39] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

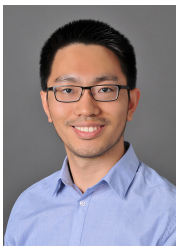
[40] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," *arXiv preprint arXiv:1506.02438*, 2015.

[41] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *International conference on machine learning*, PMLR, 2015, pp. 1889–1897.

[42] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International Conference on Machine Learning*, PMLR, 2018, pp. 1861–1870.

[43] A. Raffin, A. Hill, M. Ernestus, A. Gleave, A. Kanervisto, and N. Dormann, *Stable baselines3*, https://github.com/DLR-RM/stable-baselines3, 2019.

[44] E. Mazor, A. Averbuch, Y. Bar-Shalom, and J. Dayan, "Interacting multiple model methods in target tracking: A survey," *IEEE Transactions on aerospace and electronic systems*, vol. 34, no. 1, pp. 103–123, 1998.



Bile Peng received the Ph.D. degree with distinction from the Institut für Nachrichtentechnik, Technische Universität Braunschweig in 2018. He has been a Postdoctoral researcher in the Chalmers University of Technology, Sweden from 2018 to 2019, a development engineer at IAV GmbH, Germany from 2019 to 2020. Currently, he is a Postdoctoral researcher in the Technische Universität Braunschweig, Germany. His research interests include wireless channel modeling and estimation, Bayesian inference as well as machine learning algorithms, for signal processing and resource allocation of wireless communication systems. He received the IEEE vehicular technology society 2019 Neal Shepherd memorial best propagation paper award.

Yuhang Xie received the B.Sc. degree from Hebei University of Technology, City College, Tianjin, China, in 2015, the M.Sc. degree from the Technische Universität Braunschweig, Germany, in 2021. He has been an intern at IAV GmbH, China, Currently, he is an autonomous driving system engineer in at Bosch GmbH in Shanghai, China.

Gonzalo Seco-Granados (M'02 - SM'08) received the Ph.D. degree in Telecom. Eng. from the Univ. Politecnica de Catalunya, Spain, in 2000, and the M.B.A. degree from the IESE Business School, Spain, in 2002. From 2002 to 2005, he was a member of the European Space Agency, where he was involved in the design of the Galileo system and receivers. In 2015 and 2019, he was a Fulbright Visiting Scholar at the University of California, Irvine, USA. He is currently a Professor in the Dept. of Telecom. Eng., Univ. Autonoma de Barcelona, where he has served as Vice Dean of the Engineering School during 2011-2019. His research interests include statistical signal processing with application to GNSS and 5G localization. He is a Member of the Signal Processing for Multisensor Systems Committee of EURASIP. He has been serving as a member of the Sensor Array and Multi-channel TC for the IEEE Signal Processing Society since 2018, and as President of the Spanish Chapter of the IEEE Aerospace and Electronic Systems Society since 2019. He is a co-recipient of the 2021 IEEE Signal Processing Society Best Paper Award.

Henk Wymeersch obtained the Ph.D. degree in Electrical Engineering / Applied Sciences in 2005 from Ghent University, Belgium. He is currently a Professor of Communication Systems with the Department of Electrical Engineering at Chalmers University of Technology, Sweden. He is also a Distinguished Research Associate with Eindhoven University of Technology. Prior to joining Chalmers, he was a postdoctoral researcher from 2005 until 2009 with the Laboratory for Information and Decision Systems at the Massachusetts Institute of Technology. Prof. Wymeersch served as Associate Editor for IEEE Communication Letters (2009-2013), IEEE Transactions on Wireless Communications (since 2013), and IEEE Transactions on Communications (2016-2018). During 2019-2021, he is a IEEE Distinguished Lecturer with the Vehicular Technology Society. His current research interests include the convergence of communication and sensing, in a 5G and Beyond 5G context.

Eduard A. Jorswieck was born in 1975 in Berlin, Germany. He is managing director of the Institute of Communications Technology and the head of the Chair for Communications Systems and Full Professor at Technische Universitaet Braunschweig, Brunswick, Germany. From 2008 until 2019, he was the head of the Chair of Communications Theory and Full Professor at Dresden University of Technology (TUD), Germany. Eduard's main research interests are in the broad area of communications. He has published more than 140 journal papers, 15 book chapters, 3 monographs, and some 280 conference papers on these topics. Dr. Jorswieck is IEEE Fellow. He serves as Editor-in-Chief of the EURASIP Journal on Wireless Communications and Networking. In 2006, he received the IEEE Signal Processing Society Best Paper Award.