Hybrid Optimal Traffic Control: Combining Model-Based and Data-Driven Approaches

Urs Baumgart and Michael Burger
Fraunhofer Institute for Industrial Mathematics ITWM, Fraunhofer-Platz 1, D-67663 Kaiserslautern, Germany

Keywords: Traffic Control, Model Predictive Control, Optimal Control, Imitation Learning.

Abstract: We study different approaches to use real-time communication between vehicles, in order to improve and to optimize traffic flow in the future. A leading example in this contribution is a virtual version of the prominent ring road experiment in which realistic, human-like driving generates stop-and-go waves.
To simulate human driving behavior, we consider microscopic traffic models in which single cars and their longitudinal dynamics are modeled via coupled systems of ordinary differential equations. Whereas most cars are set up to behave like human drivers, we assume that one car has an additional intelligent controller that obtains real-time information from other vehicles. Based on this example, we analyze different control methods including a nonlinear model predictive control (MPC) approach with the overall goal to improve traffic flow for all vehicles in the considered system.
We show that this nonlinear controller may outperform other control approaches for the ring road scenario but intensive computational effort may prevent it from being real-time capable. We therefore propose an imitation learning approach to substitute the MPC controller. Numerical results show that, with this approach, we maintain the high performance of the nonlinear MPC controller, even in set-ups that differ from the original training scenarios, and also drastically reduce the computing time for online application.

1 INTRODUCTION

Increasing traffic volumes, accompanied by traffic jams and negative environmental aspects, have led to a high demand for intelligent mobility solutions that should guarantee efficiency, safety, and sustainability.

Intelligent vehicles that communicate with other road users may contribute to these solutions if they not only maximize their own goals (e.g. minimizing travel time) but aim to increase and stabilize traffic flow for all vehicles in certain traffic situations, for instance, in stop-and-go waves and congestion that emerge because of inefficient human driving behavior. With intelligent vehicle control these inefficiencies may be outbalanced if we can directly affect other vehicles' driving behavior.

A crucial point is the design of such controllers. While models that describe traffic dynamics and human driving behavior have been studied since the last century (Lighthill and Whitham, 1955; Gazis et al., 1961), nowadays, an increasing amount of data is being collected by vehicles and infrastructure systems, such that a huge amount of real-time traffic data is available. That means, besides approaches that require, at least partially, understanding of the vehicle dynamics, like optimal control theory and model predictive control (MPC), also purely data-driven approaches may be applied in traffic control.

1.1 Related Work

For most of the time, traffic flow has been controlled by infrastructure objects like speed limits (Hegyi et al., 2008) or switching through traffic light phases (McNeil, 1968; De Schutter and De Moor, 1998). But since driver assistance systems as well as (semi-) autonomous vehicles have been developed, new ways to control traffic flows have emerged. By communicating with other vehicles such intelligent vehicle controllers may increase traffic flow, e.g., with cruise control systems (Orosz et al., 2010; Orosz, 2016).

One specific scenario, in which these controllers may be applied, tested, and optimized is the artificial ring road scenario based on the experiment described in (Sugiyama et al., 2008). Here, human-like driving behavior leads to stop-and-go waves without the presence of bottleneck situations. Further, it has been

Baumgart, U. and Burger, M.
Hybrid Optimal Traffic Control: Combining Model-Based and Data-Driven Approaches.
DOI: 10.5220/0011838000003479
In *Proceedings of the 9th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS 2023)*, pages 85-94
ISBN: 978-989-758-652-1; ISSN: 2184-495X

shown in another real-world experiment that with one intelligently controlled vehicle the traffic flow can be stabilized (Stern et al., 2018). Different modeling set-ups and control strategies have been proposed to analyze the stability and controllability of the ring road network (Cui et al., 2017; Wang et al., 2020; Zheng et al., 2020).

While in these works, the underlying dynamical system that describes the dynamics of all vehicles in the system is linearized, we present an MPC approach that preserves the nonlinear vehicle dynamics and show that a linearization may decrease the controller's performance.

Further, we apply an imitation learning approach that mimics the nonlinear MPC controller with the goal to reduce the computing time for online applications while keeping the high performance of the nonlinear MPC controller. We compare our approach with several other controllers and show that it drastically reduces the required training time in comparison to other ML-based approaches that have been applied to this problem, like reinforcement learning (Wu et al., 2017; Baumgart and Burger, 2021).

Accordingly, the remaining part of this contribution is organized as follows. In Section 2, we introduce microscopic traffic models and set up the specific traffic control problem for the ring road. Then, in Section 3, we present solution approaches for the control problem and compare their results in Section 4. We also analyze the robustness of the imitation learning approach in different scenarios and close the contribution in Section 5 with a short summary and potential open problems.

2 TRAFFIC MODELING AND CONTROL

To set up the specific traffic control problem, we model human-like driving behavior with microscopic traffic models - and, more precisely, car-following (or follow-the-leader) models (Helbing, 1997; Kessels, 2019). With these models, we can describe traffic flows of individual driver-vehicle units on single lanes. To reflect human decision making, the acceleration behavior of each vehicle i depends on the headway h_i towards its leading vehicle $i+1$ and both the own speed v_i and the leader's speed v_{i+1}.
We define the headway between the two vehicles in terms of the bumper-to-bumper distance as follows

$$h_i = s_{i+1} - s_i - l_{i+1}$$

with s_i and s_{i+1} being the (front bumper) lane positions of both vehicles and l_{i+1} the length of the leading vehicle. Then, the dynamics of each vehicle i can be described by a system of ordinary differential equations (ODEs):

$$\begin{aligned} \dot{h}_i(t) &= v_{i+1}(t) - v_i(t), \\ \dot{v}_i(t) &= f(h_i(t), v_i(t), v_{i+1}(t)). \end{aligned} \tag{1}$$

The right hand side f, that determines the acceleration, depends on the specific car-following model. One choice is the Intelligent Driver Model (IDM) (Treiber and Kesting, 2013), in which the right hand side is given by

$$\begin{aligned} &f(h_i, v_i, v_{i+1}) \\ &= a_{\max}\left[1 - \left(\frac{v_i}{v_{\text{des}}}\right)^{\delta} - \left(\frac{h_i^*(v_i, \Delta v)}{h_i}\right)^2\right], \\ &h_i^*(v_i, \Delta v) = h_{\min} + \max\left(0, v_i T + \frac{v_i \Delta v}{2\sqrt{a_{\max} b}}\right), \end{aligned} \tag{2}$$

with $\Delta v = v_i - v_{i+1}$. The model depends on a set of parameters

$$\beta_{\text{IDM}} = [v_{\text{des}}, T, h_{\min}, \delta, a_{\max}, b] \tag{3}$$

which may be fitted to real-world traffic data or be used to model different driver types.

Remark 1. In our set-up, other car-following models may be used to describe the acceleration behavior of the human-driven vehicles as well. We choose the IDM because it realistically describes human driving behavior, including its inefficiencies which makes it an adequate choice for the comparison of different control approaches. Further, it avoids crashes and we can introduce heterogeneous driving behavior by assigning each vehicle i an individual set of parameters β_{IDM}^i (cf. Section 4). Typical parameter values as well as other traffic models can be found in (Treiber and Kesting, 2013).

2.1 Ring Road Control

In real-world experiments of the ring road (Sugiyama et al., 2008; Stern et al., 2018), it has been shown that human-driving behavior can be inefficient and that connected autonomous vehicles (AVs) may be able to improve traffic flow for all vehicles in the system. As it is a closed scenario, the number of vehicles is fixed which facilitates the mathematical description of the system. However, despite being an artificial scenario, similar inefficiencies (congestion and stop-and-go-waves) as in real-world scenarios occur. Therefore, the ring road experiment has proven to be an effective benchmark scenario for developing and testing new traffic control strategies.

In this work, we focus on scenarios with a total number of N vehicles including one AV and $N-1$ human drivers (HDs). We assume that the dynamics of

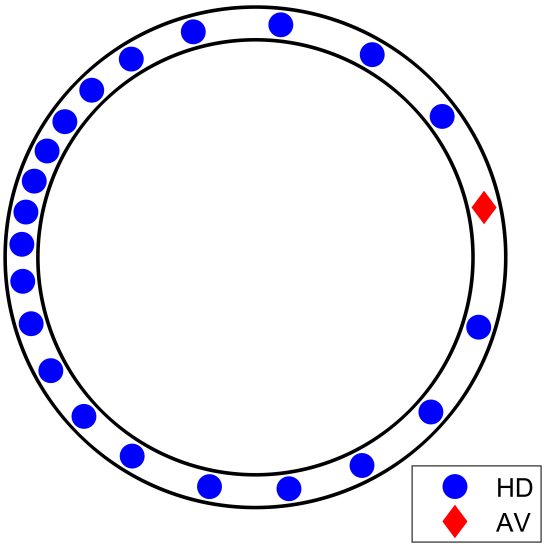

Figure 1: Visualization of the Ring Road in *Matlab* (The MathWorks, Inc., 2021) with 21 HDs and one AV.

the HDs ($i = 1,\dots,N-1$) are given by a car-following model (cf. (1)) and cannot be controlled directly. But for the AV ($i = N$), we can set the acceleration behavior with controls u such that the dynamics are given by

$$\begin{aligned} \dot{h}_N(t) &= v_1(t) - v_N(t), \\ \dot{v}_N(t) &= u(t). \end{aligned}$$

Then, we define the system's state as

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix}, x_i = \begin{bmatrix} h_i \\ v_i \end{bmatrix}$$

and the dynamics can be described as an ODE initial value problem (IVP) problem:

$$\begin{aligned} \dot{x}(t) = F(t,x(t),u(t)) &= \begin{bmatrix} v_2(t) - v_1(t) \\ f(h_1(t), v_1(t), v_2(t)) \\ \vdots \\ v_1(t) - v_N(t) \\ u(t) \end{bmatrix}, \\ x(t_0) &= x^0, t \in [t_0; t_f] \end{aligned} \tag{4}$$

with initial value x^0. To set up the control problem, first, we define two constraints for the AV such that certain safety criteria are satisfied.
A box constraint to include physical limitations for acceleration and deceleration of the AV,

$$u(t) \in [a^N_{\min}, a^N_{\max}] =: \mathcal{U}, \tag{5}$$

and a safety distance, induced by Gipps' model (Gipps, 1981; Treiber and Kesting, 2013), to keep the AV from crashing into its leading vehicle

$$h_N(t) \geq h_{\min} + v_N \Delta t^N + \frac{v_N^2}{2a^N_{\min}} - \frac{v_1^2,}{2a^N_{\min}} \tag{6}$$

with AV reaction time Δt^N and deceleration bound $a^N_{\min}$. For the other vehicles, the IDM (cf. (2)) is known to prevent crashes for all HDs (Treiber and Kesting, 2013).
While keeping these constraints satisfied, we aim to find such controls for the AV, that the traffic flow for *all* vehicles in the system is maximized. One approach to measure the traffic flow in the system is the average speed over all vehicles. We therefore define the running cost l as

$$l(x,u) = r \cdot u^2 + q \cdot \left(v^* - \frac{1}{N} \sum_{i=1}^{N} v_i \right)^2 \tag{7}$$

with $r, q \geq 0$ to penalize high actuations u of the AV and steer the average speed of the system to an equilibrium speed v^*. We can summarize Eqs. (4)-(7) to state the following nonlinear ODE optimal control problem (OCP)

$$\begin{aligned} \min_{x,u} \quad & J(x,u) = \int_{t_0}^{t_f} l(x(t),u(t))dt \\ \text{s.t.} \quad & \dot{x}(t) = F(t,x(t),u(t)), \qquad x(t_0) = x^0, \\ & u(t) \in \mathcal{U},\ x(t) \in \mathcal{X}, \qquad t \in [t_0,t_f], \end{aligned} \tag{8}$$

in which $\mathcal{X} \subset \mathbb{R}^{2N}$ denotes the set of states x such that constraint (6) is satisfied.

3 CONTROL METHODS

To find optimal controls u for the AV, in this section, we present different methods. First, we present a model predictive control approach that iteratively computes optimal controls for certain time periods. As a second model-based controller, we also set up a linear quadratic regulator.

Then, we present an approach to substitute the MPC controller by an imitation learning approach that mimics the MPC controller's behavior, but requires much less computing time. Further, we compare this approach with another ML-based techniques, reinforcement learning.

3.1 Model Predictive Control

To set up the MPC scheme, first, we define a time horizon T, a time shift τ such that $\tau < T < t_f$, and a time grid

$$t_k = \{t_0, \dots, t_f\},\ t_{k+1} - t_k = \tau.$$

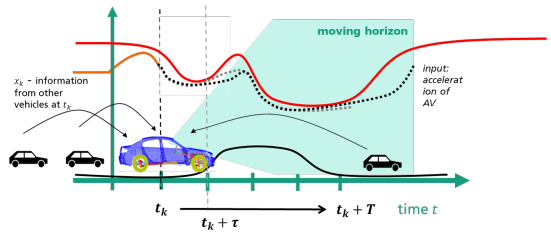

Figure 2: Visualization of the MPC scheme.

The MPC idea is visualized in Figure 2 and can be summarized as follows: at each sampling step k, we observe the current state x_k and optimize the system over time interval $[t_k, t_k+T]$. The resulting optimal control sequence $u_{|[t_k,t_k+\tau]}$ up to time τ is then used as a feedback control for the next sampling interval (Grüne and Pannek, 2011). That means, at each step, we have to solve an updated version of the OCP stated in (8):

$$\begin{aligned} \min_{x,u} \quad & J(x,u) = \int_{t_k}^{t_k+T} l(x(t),u(t))dt \\ \text{s.t.} \quad & \dot{x}(t) = F(t,x(t),u(t)), && x(t_k) = x_k, \\ & u(t) \in \mathcal{U},\ x(t) \in \mathcal{X}, && t \in [t_k, t_k+T]. \end{aligned} \tag{9}$$

Typically, the resulting OCP (9) has to be solved numerically at each time instance which may be time consuming in dependence of time horizon T. In this work, we apply a reduced discretization approach, cf. (Gerdts, 2011).

3.2 Linear Quadratic Regulator

Instead of solving OCP (9) numerically, another approach is to transform the problem to a linear quadratic problem. For these problems, we can solve the resulting differential Riccati equation analytically (Sontag, 1998; Locatelli, 2001). Here, the running cost

$$l(x,u) = u^\top R u + x^\top Q x$$

with $R \in \mathbb{R}$ and $Q \in \mathbb{R}^{2N\times 2N}$ is quadratic and the right hand side of the ODE is linear

$$\dot{x}(t) = Ax(t) + Bu(t)$$

with $A \in \mathbb{R}^{2N\times 2N}$ and $B \in \mathbb{R}^{2N}$.

In the following, we set up a linear quadratic regulator (LQR) similar to (Cui et al., 2017; Wang et al., 2020). First, we define an equilibrium state in which all vehicles travel with constant speed v^*. Depending on the set of parameters β^i_{IDM} (cf. (3)), the constant equilibrium headways h_i^* may differ for each HD. As all vehicles i travel with constant speed at the equilibrium state, the accelerations equal zero:

$$\dot{v}_i = f(h_i, v_i, v_{i+1}) = f(h_i^*, v^*, v^*) = 0,\ i = 1,\dots,N.$$

Then, we define a state that indicates the deviation from (h_i^*, v^*),

$$x_i = \begin{bmatrix} \tilde{h}_i \\ \tilde{v}_i \end{bmatrix} = \begin{bmatrix} h_i - h_i^* \\ v_i - v^* \end{bmatrix}$$

and apply a first-order Taylor expansion around $f(h_i^*, v^*, v^*)$

$$\begin{aligned} & f(h_i, v_i, v_{i+1}) \\ & \approx \frac{\partial f}{\partial h_i}(h_i^*)(\tilde{h}_i) + \frac{\partial f}{\partial v_i}(v^*)(\tilde{v}_i) + \frac{\partial f}{\partial v_{i+1}}(v^*)(\tilde{v}_{i+1}) \\ & = \alpha_{i1}\,\tilde{h}_i + \alpha_{i2}\,\tilde{v}_i + \alpha_{i3}\,\tilde{v}_{i+1}. \end{aligned}$$

Here, $\alpha_{i1}, \alpha_{i2}, \alpha_{i3}$ denote the directional derivatives of the IDM function (cf. (2)) at the equilibrium state for each vehicle and, similar to (1), we can describe the dynamics of the HDs as follows

$$\begin{aligned} \dot{\tilde{h}}_i &= \tilde{v}_{i+1} - \tilde{v}_i, \\ \dot{\tilde{v}}_i &= \alpha_{i1}\,\tilde{h}_i + \alpha_{i2}\,\tilde{v}_i + \alpha_{i3}\,\tilde{v}_{i+1}, \end{aligned}$$

for $i = 1,\dots,N-1$. Again, the acceleration behavior of the AV ($i = N$) is determined by the control input u:

$$\begin{aligned} \dot{\tilde{h}}_N &= \tilde{v}_1 - \tilde{v}_N, \\ \dot{\tilde{v}}_N &= u. \end{aligned}$$

Finally, the linear dynamics for all vehicles are given by

$$\dot{x}(t) = Ax + Bu,$$

with matrices A and B defined as

$$A = \begin{bmatrix} C_1 & D_1 & 0 & \dots & 0 \\ 0 & C_2 & D_2 & 0 & \vdots \\ \vdots & 0 & \ddots & \ddots & 0 \\ 0 & \vdots & 0 & C_{N-1} & D_{N-1} \\ D_N & 0 & \dots & 0 & C_N \end{bmatrix}, B = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ B_N \end{bmatrix},$$

and where the submatrices C_i and D_i are given by

$$\begin{aligned} C_i &= \begin{bmatrix} 0 & -1 \\ \alpha_{i1} & \alpha_{i2} \end{bmatrix}, D_i = \begin{bmatrix} 0 & 1 \\ 0 & \alpha_{i3} \end{bmatrix}, i = 1,\dots,N-1, \\ C_N &= \begin{bmatrix} 0 & -1 \\ 0 & 0 \end{bmatrix}, D_N = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, B_N = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \end{aligned}$$

Still, we aim on keeping low control values and a high average speed. By setting

$$R = r \in \mathbb{R}, Q = \text{diag}\,[0, q, 0, q, \dots, 0, q] \in \mathbb{R}^{2N\times 2N}$$

we obtain the same running cost as in the previous section (cf. (7)). The LQR is then defined as follows:

$$\begin{aligned} \min_{x,u} \quad & J(x,u) = \int_{t_k}^{t_k+T} u(t)^\top R u(t) \quad + x(t)^\top Q x(t) dt \\ \text{s.t.} \quad & \dot{x}(t) = Ax(t) + Bu(t), && x(t_k) = x_k, \\ & && t \in [t_k, t_k+T]. \end{aligned}$$

Remark 2. For the LQR, the optimal control u can be computed analytically by solving the differential Riccati equation for the tuple (A,B,Q,R), which, in general, is much faster than solving the nonlinear OCP (9) numerically. However, due to the linearization, the description of the vehicle dynamics lacks in accuracy which may lead to an unsatisfying performance of the computed controls.
Further, for the LQR set-up, constraints as (5) and (6) have to be neglected such that infeasible solutions may be computed. There are, however, approaches for constrained LQRs based on constrained quadratic programming (Nocedal and Wright, 2006). But, in general, this requires additional constraint handling and typically leads to much higher computational loads as well. In this work, we combine the LQR approach with a so-called safe speed controller based on Gipps' safe speed model (Gipps, 1981; Treiber and Kesting, 2013), such that at each time step constraint (6) is satisfied and crashes are avoided.

3.3 Imitation Learning

In this section, we introduce a controller based on imitation learning (IL) with the goal to keep the high performance of the nonlinear MPC approach but to decrease the required computing time. The general idea of IL is to *imitate* (or mimic) another controller at a certain task. This controller, for the ring road scenario, could be either an intelligent human driver or, in our applications, a controller that has been realized using MPC.

In particular, by solving the control problem over a certain timer period $[t_0,t_f]$ numerically, we obtain for each time step t of a time grid $[t_0,\dots,t_f]$ a state $x_t \in \mathcal{X}$ of state space $\mathcal{X}$ and a corresponding optimal control $u_t \in \mathcal{U}$ of control space $\mathcal{U}$. Now, we aim to find a function

$$\pi_\theta \colon \mathcal{X} \to \mathcal{U}$$

with parameters θ that maps *optimally* from state space to control space. Optimally, in this context, means as similar as possible to the other controller. This can, e.g., be achieved by optimizing θ such that

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \sum_{t=t_0}^{t_f} \|\pi_\theta(x_t) - u_t\|^2 . \tag{10}$$

That is, for all states x_t, the resulting function output $\pi_\theta(x_t)$ should be as close as possible to the MPC-based controls u_t.

3.3.1 Neural Nets

A typical choice for π_θ are multi-layer feed-forward neural nets (NNs) because they are well known for their capability of approximating nonlinear functions (Cybenko, 1989; Hornik et al., 1989; Bishop, 2006). In general, they can be described by a composition of K layers, $k = 1,\dots K$, that all consist of M_k neurons. Additionally, an activation function $\sigma^k \colon \mathbb{R} \to \mathbb{R}$ for each layer has to be specified. All of these quantities, the number of layers K, the number of neurons for each layer M_k and the activation functions σ^k are hyperparameters that are chosen preliminary. Further, each layer has a weight matrix W^k and bias vector b^k that have to be optimized by a so-called *training* procedure. That means, if we choose a NN for π_θ in the optimization problem (10), then θ consists of $W^1, b^1, \dots, W^K, b^K$.

While the optimization (or training) may require significant computing time, the NN outputs of a trained net, i.e., the controls, can be computed very fast in comparison to solving nonlinear control problems (cf. Section 4).

3.4 Extended Imitation Learning

So far, we have introduced an IL controller that shall mimic the MPC controller for one specific scenario. However, we aim to find a controller that performs well in different situations. Consequently, in the following, we assume that we have solved L different problems leading to a data set of states and controls $\mathcal{D} = \{(x_t^l, u_t^l)\}$ over time steps $t \in \{t_0,\dots,t_f\}$ and trajectories $l = 1,\dots,L$. These problems may differ in their initial values x^0 and set-ups of the dynamical system (4) induced by car-following parameters β_{IDM}, cf. (3). Then, similar to (10), we aim to optimize the parameters of a function π_θ such that

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \sum_{l=1}^{L} \sum_{t=t_0}^{t_f} \left\| \pi_\theta(x_t^l) - u_t^l \right\|^2 . \tag{11}$$

Again, the function outputs $\pi_\theta(x_t)$ should be as similar as possible to the controls u_t. But, by summing not only over time steps t but also trajectories l, different set-ups of the dynamical system can be taken into account. The goal of our approach is to be able to compute controls u even for states x that have not been observed exactly during training. To achieve this, it is therefore crucial for the controller's performance to define a diverse and substantial dataset $\mathcal{D}$ induced by the L different scenarios.

3.5 Reinforcement Learning

As an additional alternative to the approaches presented here, we have applied a model-free reinforcement learning (RL) approach to the ring road control problem. In RL, we optimize a control

mapping (often called *policy* in the RL context) π_θ which could be, e.g., a neural net. The main difference to the other control approaches is that, in general, the dynamical system, induced by the ODE of (1), is *unknown*. Thus, several simulations of the dynamical system are needed such that the controller can observe different states and apply different controls. By interacting with the dynamical system, i.e., by observing the system's response, the goal is to find such policy parameters θ that a certain objective function, called *reward*, is maximized.

Remark 3. In our implementation, we have optimized the RL agent with the Matlab Reinforcement Learning Toolbox and an actor-critic algorithm (The MathWorks, Inc., 2021). The policy π_θ is represented by a neural net with two hidden layers, 32 neurons for each layer and the *tanh* activation function.

For a more detailed description of RL in general, some common algorithms, and its application for traffic control, we refer to, e.g., (Sutton and Barto, 2018), (Duan et al., 2016), and (Wu et al., 2017; Baumgart and Burger, 2021), respectively.

4 RESULTS

In this section, we compare the different control approaches of Section 3 in terms of accuracy, robustness, performance, and computational efficiency. We show that the MPC controller performs well even if the dynamical system is not completely known and we study for the extended IL controller how many vehicles have to send data to achieve desirable results.

General Set-Up

For all scenarios, we let the vehicles start with equal headways and slightly vary each HD's driving behavior such that over time a stop-and-go wave occurs. To obtain different driving behavior, first, we introduce a nominal IDM parameter set $\beta_{\text{IDM}}^{\text{nom}} = [16, 1, 2, 4, 1, 1.5]$ (cf. (3)). Then, before each simulation, we draw samples from a Gaussian distribution with the entries of β_{IDM} as mean values and standard deviation σ. Thus, each HD's i current characteristics can be described by the corresponding IDM parameter set β_{IDM}^i defined as

$$\beta_{\text{IDM}}^i \sim \mathcal{N}(\beta_{\text{IDM}}^{\text{nom}}, \sigma). \tag{12}$$

For each simulation l, we draw new parameters which may be summarized by

$$\mathcal{B}_l = \left[\beta_{\text{IDM}}^{1,l}, \beta_{\text{IDM}}^{2,l}, \ldots, \beta_{\text{IDM}}^{22,l}\right]. \tag{13}$$

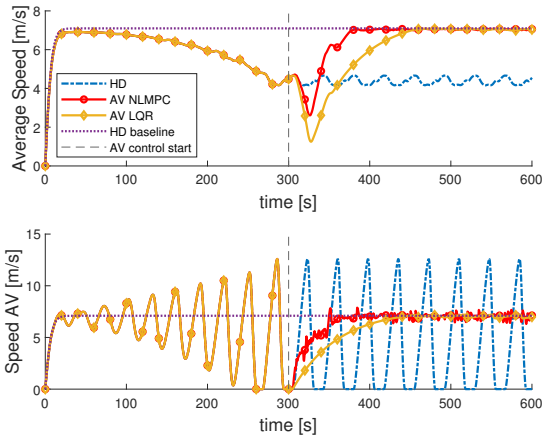

Figure 3: Comparison of the MPC solution approaches in terms of the average speed over all vehicles and the AV's speed.

At the beginning, all vehicles are controlled by HDs with their corresponding parameters β_{IDM}^i to observe the emerging stop-and-go wave caused by heterogeneous driving behavior. Then, at $t = 300$s the AV control for vehicle $i = N$ is switched on with the goal to stabilize the traffic flow.

4.1 Comparison of Controllers

At first, we compare the model-based solution approaches of Sections 3.1 and 3.2. For both, the nonlinear MPC (NLMPC) and the LQR approach, the resulting OCP is solved for time horizon $T = 30$s and then the resulting controls are applied with time shift $\tau = 2$s. In Figure 3, results are shown in terms of average speeds over all vehicles and the AV's speeds for the following controllers for one set of parameter $\mathcal{B}_l$:

- **HD Baseline:** First, all vehicles are controlled by HDs with homogeneous parameters $\beta_{\text{IDM}}^{\text{nom}}$ such that no stop-and-go wave occurs.
- **HD:** All vehicles are controlled by HDs but with heterogeneous parameters β_{IDM}^i. Without AV control the the stop-and-go wave evolves and does not vanish over time.
- **NLMPC and LQR:** For both controllers, the same stop-and-go wave evolves until the AV control start. Then, the controllers are switched on to stabilize the traffic flow for the rest of the time period. For the running cost of (7), we set $q = 1$ and $r = 5$.

The results show that both controllers, after the AV control start, slow down at the beginning, but then increase their speed which stabilizes the traffic flow and leads to an average speed similar to the HD baseline scenario. However, the controllers differ in their

performance (cf. Figure 3 and Table 1): while the NLMPC approach reaches a high average speed after the control starts and stabilizes the traffic flow earlier, the LQR approach has much faster computing times.

Further, for the NLMPC controller, the computing time clearly exceeds the time shift of $\tau = 2$s. That means, at least with our implementation of the controller, the NLMPC approach is not capable of being applied as a real-time feedback controller.

Extended Imitation Learning Controller

To achieve a similar high performance as the NLMPC approach while reducing the computing time for each control, in Section 3.4, we have presented an approach to imitate the MPC controller. Thus, we generate a data set of $L = 10$ trajectories in which the control problem is solved with the NLMPC approach. For all trajectories l, we draw parameter sets $\mathcal{B}^{\text{train}} = \{\mathcal{B}_1^{\text{train}}, \dots, \mathcal{B}_{10}^{\text{train}}\}$ according to (13) such that the HD driving behavior varies in each scenario and the controller is confronted with different stop-and-go situation at the control start. Then, we optimize the parameters θ of a NN according to (11).

The resulting NN of the extended IL controller is then applied to five test scenarios with parameter sets $\mathcal{B}^{\text{test}} = \{\mathcal{B}_1^{\text{test}}, \dots, \mathcal{B}_5^{\text{test}}\}$ (that all differ from the parameter sets used to generate the training data). Additionally to the controllers of Figure 3, in Table 1, we compare the following controllers for test scenarios $\mathcal{B}^{\text{test}}$:

- **extIL:** Extended IL controller of Section 3.4. Here, we train a NN that consists of two layers $K = 2$, $M_1 = M_2 = 10$ neurons for each layer, and the *tanh* activation function (cf. Section 3.3.1).
- **RL:** RL controller of Section 3.5 with hyperparameters and training algorithm as described in Remark 3. To achieve desirable results, experiments have shown, that we require at least a number of 1000 episodes (system simulations).
- **PI sat** PI with saturation controller of (Stern et al., 2018).

We compare the average speed and the standard deviation (SD) of the speed after the control start as well as average times for stabilization and for computing the next control over all test scenarios $\mathcal{B}^{\text{test}}$. For the NLMPC approach the next control is computed online, whereas the ML-based techniques consist of an offline optimization (training) and an online computation of the controls.

We stress, that our extended IL approach, that has been trained without direct knowledge of the dynamical system and training data that differs from the test data, performs almost as good as the NLMPC approach at least in terms of the average speed. However, in contrast to the NLMPC approach, it is real-time capable, because we only have to feedforward the current state through the NN to obtain the next control in contrast to solving an OCP numerically. Further, it can be observed that satisfying results can already be achieved with 10 training trajectories, thus, it requires much less forward runs than, e.g., RL. The latter is due to the fact that, with the used MPC trajectory data, expert system knowledge is induced into the training procedure.

Table 1: Comparison of different controllers at the ring road scenario in terms of average values over test scenarios $\mathcal{B}^{\text{test}}$. The average time to stabilize the system is defined as the time until all speeds $v_1, \dots, v_N$ are within ± 0.3m/s of the equilibrium speed.

Control-	Av. speed [m/s]	SD speed [m/s]	Av. time for stab. [s]	Av. comp. time [s]
HD	4.60	3.98	-	-
NLMPC	6.61	1.35	72.1	8.1
LQR	5.82	2.02	246.5	0.053
extIL	6.57	1.40	131.2	0.0041
RL	5.65	2.10	191.1	0.0026
PI Sat	6.41	1.60	181.5	0.0007

In Figure 4, we show resulting trajectories of the two ML-based techniques, extIL and RL, as well as of the NLMPC approach for the same set of parameters $\mathcal{B}_l$ as in Figure 3 that is part of the test data set $\mathcal{B}^{\text{test}}$.

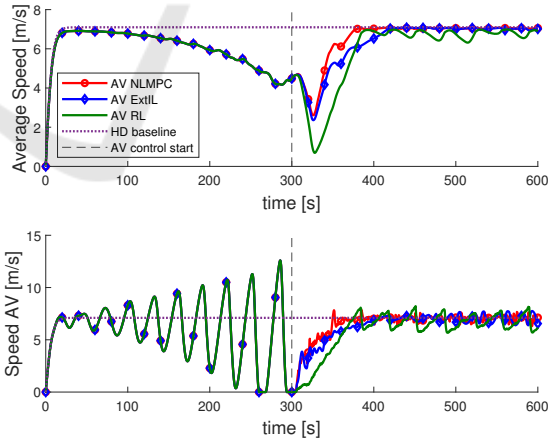

Figure 4: Comparison of the NLMPC approach with extIL and RL in terms of the average speed over all vehicles and the AV's speed.

4.2 Analysis of Robustness

In this section, we analyze the robustness of the extended IL approach in terms of the training data and

the observed data of other vehicles that is required for an online application.

4.2.1 Incorrect Description of the Underlying Dynamical System

For the NLMPC approach, we require to know the underlying dynamics of OCP (9). However, in real-world situations, it is a challenging task to find car-following parameters β^i_{IDM} that exactly describe the driving behavior of HDs. If we have enough observation data, we can calibrate the parameters of the car-following model (Kesting and Treiber, 2008). But even then, their driving behavior may change over time.

To simulate this situation, we let all HDs travel with one of the parameters sets $\mathcal{B}^{\text{test}}_l$ as in Section 4.1 such that the same stop-and-go wave occurs at each simulation. But now, we draw new parameters sets $\mathcal{B}^{\sigma} = \{\mathcal{B}^{\sigma}_1, \ldots, \mathcal{B}^{\sigma}_{10}\}$ for different standard deviations σ according to (12) and (13). That means, at each instance, the controller optimizes a system with parameters $\mathcal{B}^{\sigma}_l$ that differs from the simulated system induced by parameters $\mathcal{B}^{\text{test}}_l$. By increasing the standard deviation $\sigma \in \{0, 0.1, 0.2, 0.3, 0.4, 0.5\}$, we analyze how much the optimized system is allowed to differ from the simulated system. Table 2 shows average values over all vehicles and parameter sets $\mathcal{B}^{\sigma}$.

Desirable results up to $\sigma = 0.3$ indicate a certain robustness of the NLMPC approach against incorrect description of the underlying dynamics. This may be explained by the fact, that the MPC scheme allows to observe the real state at each sampling step and to take this real state as initial value for the next optimization.

Table 2: Influence of wrong parameters $\mathcal{B}^{\sigma}$ on the performance of the NLMPC approach.

SD σ for $\hat{\beta}^i_{\text{IDM}}$	Av. speed [m/s]	SD speed [m/s]	Av. time for stab [s]
$\sigma = 0$	6.52	1.72	97.5
$\sigma = 0.1$	6.40	1.69	90.1
$\sigma = 0.2$	6.23	1.65	96.8
$\sigma = 0.3$	6.24	1.69	102.5
$\sigma = 0.4$	5.76	1.83	140.9
$\sigma = 0.5$	4.92	1.91	210.6

4.2.2 Data-Availability

In a real-world application, we may not obtain real-time data from *all* other vehicles, for instance, because they are unwilling or physically unable to send data. But even if (in the near future) we overcome these problems, it is not guaranteed to observe all vehicles at a certain traffic situation. We therefore analyze the impact of data-availability of preceding vehicles for the extended IL controller.

In the following, we use the same training data set obtained with parameters $\mathcal{B}^{\text{train}}$ and same hyperparameters for the NNs as in Section 4.1, but now the controller only obtains data from a limited number of leading vehicles during the training. That means, additionally to the NN in which data from all leaders is available, further NNs with only input data from a limited number of leading vehicles are trained. Then, all of them are applied to the test scenarios, induced by $\mathcal{B}^{\text{test}}$. In Table 3, we compare average values over all scenarios for different numbers of leading vehicles.

The results show that, indeed, the number of vehicles sending data may affect the controller's performance. Up to four leaders the controller's performance in terms of the average speed increases and the highest performance is reached when all data is available.

For the time to stabilize the system, the interpretation is not completely clear because low values are not always achieved for the NNs with high average speed. One explanation is the fact, that in some scenarios the controller dampens the stop-and-go wave relatively fast which directly increases the averages speed, but may require some significant time to reach a completely stable traffic flow.

Consequently, as for a small number of leaders the results are already satisfying, it seems that not all data is necessary for an application of the extended IL controller.

Table 3: Influence of the number of the number of leading vehicles sending data in terms of average values over test scenarios $\mathcal{B}^{\text{test}}$.

Num. of cons. leaders	Av. speed [m/s]	SD speed [m/s]	Av. time for stab [s]
1	6.25	1.52	155.8
2	6.42	1.40	156.3
3	6.43	1.44	178.3
4	6.55	1.48	124.6
6	6.44	1.43	122.6
8	6.41	1.32	99.1
10	6.38	1.36	115.1
16	6.30	1.54	177.2
all	6.57	1.40	131.2

Remark 4. To obtain the results of Table 3, we choose the same hyperparameters for each NN with $K = 2$ layers, $M_1 = M_2 = 10$ neurons for each layer, and the *tanh* activation function. But, feedforward networks are typically initialized with random weights and biases. Depending on the parameter initialization, typ-

ical training algorithms may lead to locally instead of globally optimal solution for the objective function (11). We therefore use a Monte-Carlo type approach with 50 parameter initializations for each of the NNs to make them more comparable.

5 SUMMARY AND CONCLUSION

We have presented different model-based and data-driven solution approaches to solve the ring road control problem. While the highest performance is reached for the NLMPC approach and it performs well even if the dynamical system differs from the real system (cf. Section 4.2.1), its computing times may prevent it from being real-time applicable. Thus, we have replaced the MPC controller by an imitation learning approach which can be extended to include several different training scenarios. With this extension, the controller can perform well even in scenarios that have not been observed exactly during the training and may not even require data from all vehicles in the system (cf. Section 4.2.2). By imitating another controller, expert knowledge about the dynamical system is induced which leads to much faster training than for other ML techniques like RL.

For the training, we require an, at least basic, understanding of the HD dynamics induced by the car-following model. But to run it in real-world application, we need real-time data from other vehicles. We expect that current developments in communication technologies will enable the realization of such controllers in real-world traffic situations in the near future.

Although its performance depends on the observed training data, there are advantages of applying data-driven controllers such as the extIL approach in real-world situations. That is, while the presented NLMPC controller is only valid for the closed ring road in which we are able to describe the dynamics, the extIL controller may be applied in all set-ups in which data of one or several leading vehicles is available. As the occurring stop-and-go waves and traffic jams in, for instance, city traffic are similar to the ones at the ring road, we expect our extIL controller to perform well in these situations, too.

Like in other applications, in which ML techniques show promising results, there are still open questions regarding the application of NN-based controllers in safety critical situations like steering a vehicle. While we have shown a certain robustness of our approach, guaranteed accuracy and stability are still hard to prove and, at least partially, open tasks. Further, the tuning of hyperparameters and the reproducibility of training results of Neural Nets are critical as well and may be overcome by using other structures like radial basis function (RBF) networks (Bishop, 2006).

In this work, we have focused on scenarios with only one intelligently controlled vehicles to emphasize that already one of these suffices to outbalance the emerging stop-and-go wave. However, taking into account several AVs may further improve the results and may lead to a more efficient outbalancing effect, cf. (Chou et al., 2022).
In future work, we aim to further analyze the robustness and stability of our approach and other AI-based techniques. Especially, to guarantee certain safety criteria such that the controller can be applied not only in artificial but also realistic real-world traffic situations.

REFERENCES

Baumgart, U. and Burger, M. (2021). A reinforcement learning approach for traffic control. In *Proceedings of the 7th International Conference on Vehicle Technology and Intelligent Transport Systems - VEHITS,*, pages 133–141. INSTICC, SciTePress.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer-Verlag New York.

Chou, F.-C., Bagabaldo, A. R., and Bayen, A. M. (2022). The Lord of the Ring Road: A Review and Evaluation of Autonomous Control Policies for Traffic in a Ring Road. *ACM Transactions on Cyber-Physical Systems (TCPS)*, 6(1):1–25.

Cui, S., Seibold, B., Stern, R., and Work, D. B. (2017). Stabilizing traffic flow via a single autonomous vehicle: Possibilities and limitations. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 1336–1341.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4):303–314.

De Schutter, B. and De Moor, B. (1998). Optimal Traffic Light Control for a Single Intersection. *European Journal of Control*, 4(3):260 – 276.

Duan, Y., Chen, X., Houthooft, R., Schulman, J., and Abbeel, P. (2016). Benchmarking Deep Reinforcement Learning for Continuous Control. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1329–1338.

Gazis, D. C., Herman, R., and Rothery, R. W. (1961). Nonlinear Follow-the-Leader Models of Traffic Flow. *Operations Research*, 9(4):545–567.

Gerdts, M. (2011). *Optimal Control of ODEs and DAEs*. De Gruyter.

Gipps, P. (1981). A behavioural car-following model for computer simulation. *Transportation Research Part B: Methodological*, 15(2):105–111.

Grüne, L. and Pannek, J. (2011). *Nonlinear Model Predictive Control*. Springer-Verlag, London.

Hegyi, A., Hoogendoorn, S., Schreuder, M., Stoelhorst, H., and Viti, F. (2008). SPECIALIST: A dynamic speed limit control algorithm based on shock wave theory. In *2008 11th International IEEE Conference on Intelligent Transportation Systems*, pages 827–832.

Helbing, D. (1997). *Verkehrsdynamik*. Springer-Verlag Berlin Heidelberg.

Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366.

Kessels, F. (2019). *Traffic Flow Modelling*. Springer International Publishing.

Kesting, A. and Treiber, M. (2008). Calibrating car-following models by using trajectory data: Methodological study. *Transportation Research Record*, 2088(1):148–156.

Lighthill, M. J. and Whitham, G. B. (1955). On kinematic waves. II. A theory of traffic flow on long crowded roads. *Proc. R. Soc. Lond. A*, 229:317–345.

Locatelli, A. (2001). *Optimal Control: An Introduction*. Birkhäuser Verlag AG Basel.

McNeil, D. R. (1968). A Solution to the Fixed-Cycle Traffic Light Problem for Compound Poisson Arrivals. *Journal of Applied Probability*, 5(3):624–635.

Nocedal, J. and Wright, S. (2006). *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer New York.

Orosz, G. (2016). Connected cruise control: modelling, delay effects, and nonlinear behaviour. *Vehicle System Dynamics*, 54(8):1147–1176.

Orosz, G., Wilson, R. E., and Stépán, G. (2010). Traffic jams: dynamics and control. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 368(1928):4455–4479.

Sontag, E. D. (1998). *Mathematical Control Theory: Deterministic Finite-Dimensional Systems*. Springer-Verlag New York.

Stern, R. E., Cui, S., Delle Monache, M. L., Bhadani, R., Bunting, M., Churchill, M., Hamilton, N., Haulcy, R., Pohlmann, H., Wu, F., Piccoli, B., Seibold, B., Sprinkle, J., and Work, D. B. (2018). Dissipation of stop-and-go waves via control of autonomous vehicles: Field experiments. *Transportation Research Part C: Emerging Technologies*, 89:205 – 221.

Sugiyama, Y., Fukui, M., Kikuchi, M., Hasebe, K., Nakayama, A., Nishinari, K., Tadaki, S.-i., and Yukawa, S. (2008). Traffic jams without bottlenecks—experimental evidence for the physical mechanism of the formation of a jam. *New Journal of Physics*, 10(3):033001.

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement Learning: An Introduction*. Adaptive Computation and Machine Learning. MIT Press, 2nd edition.

The MathWorks, Inc. (2021). *Matlab R2021b*. Natick, Massachusetts, United State.

Treiber, M. and Kesting, A. (2013). *Traffic Flow Dynamics*. Springer-Verlag Berlin Heidelberg.

Wang, J., Zheng, Y., Xu, Q., Wang, J., and Li, K. (2020). Controllability Analysis and Optimal Control of Mixed Traffic Flow With Human-Driven and Autonomous Vehicles. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–15.

Wu, C., Kreidieh, A., Parvate, K., Vinitsky, E., and Bayen, A. M. (2017). Flow: Architecture and Benchmarking for Reinforcement Learning in Traffic Control. *arXiv: 1710.05465*.

Zheng, Y., Wang, J., and Li, K. (2020). Smoothing Traffic Flow via Control of Autonomous Vehicles. *IEEE Internet of Things Journal*, 7(5):3882–3896.