Array 15 (2022) 100235

Contents lists available at ScienceDirect

Array

journal homepage: www.elsevier.com/locate/array

Deep reinforcement learning for gearshift controllers in automatic transmissions

Gerd Gaiselmann [a,1], Stefan Altenburg [a,*,1], Stefan Studer [a], Steven Peters [b]

[a] Mercedes-Benz Group AG, Germany
[b] TU Darmstadt, Institute of Automotive Engineering (FZD), Germany

ARTICLE INFO

Keywords:
Deep reinforcement learning
Sim-to-real transfer
Domain adaption
Automotive
Automatic transmission
Gear shifting

ABSTRACT

Control design for gearshifts in modern automotive automatic transmissions constitutes a challenging, time consuming task performed by highly trained experts. This is due to the fact that a variety of non-linear and partially observable systems need to be actuated, such that a comfortable shifting behavior is achieved within an sufficiently low shifting time. The presented approach leverages deep reinforcement learning (DRL) to control gear shifts, outperforming current state of the art controller performance. This requires formulating the shifting task as a Markov decision process by designing suitable action and observation spaces as well as a meaningful reward function. Due to the sample complexity of DRL methods, the control agents are trained in simulation and are subsequently transferred to a real transmission on a test bench. To successfully transfer DRL agents from simulation to reality, methods such as domain randomization and domain adaption leveraging evolutionary optimization are applied. To the best of the authors' knowledge, this work is the first to successfully apply DRL for the closed loop control of a real world automotive automatic transmission of realistic complexity.

1. Introduction

Modern automatic transmissions are a key component in the drivetrain of premium vehicles, crucial for meeting the high demands on driving performance and energy efficiency. The interplay of electrical, hydraulic and mechanical components enables fast yet comfortable gearshifts. To perform gearshifts in the considered type of transmissions, the power flow through the transmission is changed by connecting or disconnecting shafts of different planetary gear sets using hydraulically actuated multi plate clutches. Consequently, gearshift controllers need to actuate these clutches, such that the power flow is diverted fast but comfortably with a high reliability. In this work, we focus on the synchronization process to connect shafts.

The variety of nonlinear subsytems involved as well as partial observability makes the design of those controllers a complex and laborious task. Classic control approaches require tedious tuning of parameter maps by highly trained experts in a multitude of physical experiments on test benches and during vehicle operation. Recent successes in the control of difficult tasks such as game play [1,2], robotics [3–6] and plant control [7,8] motivate the application of reinforcement learning (RL) techniques to the gearshift controller synthesis. Our proposed approach depicted in Fig. 1 enables a partly automated gearshift controller synthesis by means of deep reinforcement learning (DRL) methods.

The key contributions of this paper are the design of a learning framework in simulation including a Markov decision process (MDP) formulation of the shifting task and the successful transfer of DRL agents for gearshift control trained in simulation to a real transmission using two different transfer approaches. Measurements on the Mercedes-Benz 9G-TRONIC show that DRL-based gearshift controllers outperform the shifting quality of conventional control approaches and the automated process considerably reduces the effort of manual calibration for gearshift controller synthesis.

To the best of our knowledge, this work is the first to successfully apply DRL techniques for the closed loop control of a real world transmission of realistic complexity.

2. Related work

2.1. Automated function development for gear shifts

Apart from conventional manual calibration and optimization of open-loop and closed-loop controller parameter maps, several partly

* Corresponding author.
E-mail addresses: gerd.gaiselmann@mercedes-benz.com (G. Gaiselmann), stefan.altenburg@mercedes-benz.com (S. Altenburg).
[1] These authors contributed equally.

https://doi.org/10.1016/j.array.2022.100235
Received 17 December 2021; Received in revised form 15 July 2022; Accepted 15 July 2022
Available online 22 July 2022

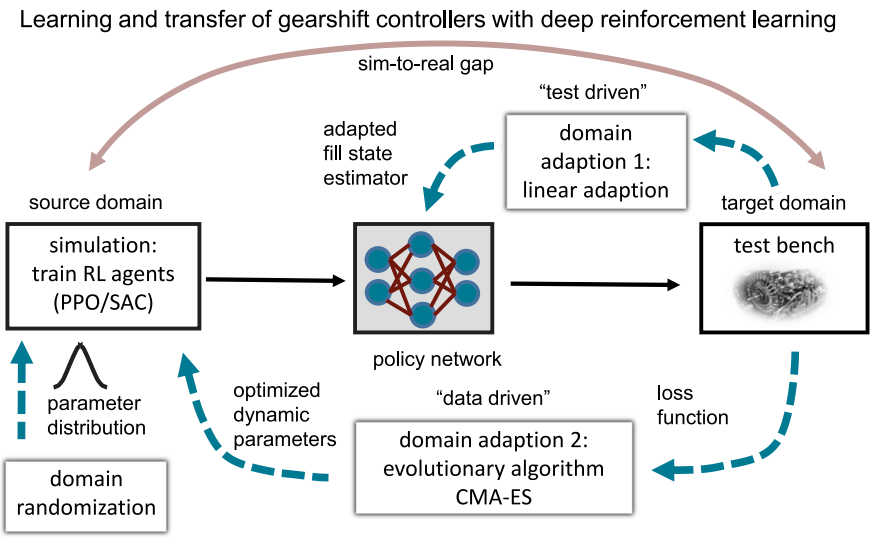

Fig. 1. Illustration of this work's contribution: Deep Reinforcement Learning (DRL) is applied to control gear shifts in an automatic transmission. Control agents are trained in simulation and then transferred to a test bench. Several approaches are shown that contribute to a successful transfer from source to target domain. Note that not all approaches are applied simultaneously.

automated methods for gear shift control exist. Manual calibration is mostly done in tedious road tests by changing the parameter maps until the resulting shifts meet the intended characteristics of the executing engineer. Many objective criteria exist to express subjective shift quality, e.g. the Vibration Dose Value (VDV) [9] or the amplitude and Root Mean Squared (RMS) value of longitudinal acceleration at the driver seat [10]. Using these objective evaluation criteria, different approaches of calibration automation have been developed and published, leveraging e.g. Fuzzy Logic and evolutionary algorithms [11,12] or gradient-based optimization over high-order polynomial functions mapping control parameters to comfort [13].

2.2. (Deep) reinforcement learning in automotive application and transmission control

Despite of the fast progress of RL in recent years, there are only few applications in the automotive industry. Common applications of RL in the automotive sector are automated parking systems [14], autonomous driving [15], engine control, e.g. idle speed control [16] or turbocharger boost control [17] as well as various hybrid energy management strategies, e.g. [18,19]. However, many current research projects are at an intermediate state in terms of robustness, safety and quality and are far from being used in series production. In transmission control, working fields are optimal gear selection [20,21] as well as controlling the shifting process itself. Becsi et al. [22,23] apply Deep Double Q-Networks and Policy Gradient (PG) for discrete-action control of a solenoid valve for a double-acting floating-piston cylinder for automated manual transmission of heavy duty vehicles. They control the synchronization using the synchromesh and the gear change. After training purely in simulation, they find better results on a test bench using Monte-Carlo Tree Search (MCTS). Best results and real time performance are achieved with a PG-MCTS agent. They also show that a PID-controller can achieve better performance under nominal conditions but PG-controllers work better with a wider range of environment settings.

Sommer Obando [24] applies tabular Q-Learning and SARSA algorithms to control normal force on a friction clutch to reduce clutch judder. However, this work considerably simplifies the control task neglecting several aspects of realistic automotive transmissions such as nonlinear actuator dynamics and partial observability. Furthermore, substantial loss of performance is reported when transferring the agents from simulations to the experimental transmission.

RL techniques are also used to learn control trajectories for open loop control of wet clutches in transmissions [25–28]. Both offline learning in simulation as well as online learning on a physical clutch are used. They apply RL to optimize a parametrized control profile similar to conventional control. Since they use open loop control, no real time action calculation of the agent is performed, which significantly reduces problem complexity but also makes no use of the agents control abilities.

Lampe et al. [29] use a DDPG agent to learn a closed-loop policy in a simplified simulation. They show that the resulting agent can control clutch engagement for vehicle start up in Automated Manual Transmissions (AMT) and Dual Clutch Transmissions (DCT). However, no validation on a physical transmission or car is carried out.

Previous research has shown that a model-free off-policy soft-actor–critic (SAC) agent can achieve similar results to a conventional closed-loop controller for the activation process of a wet clutch in an automated transmission [30]. The learning process and comparison have been executed in simulation.

In contrast to the beforementioned work, this paper is the first to successfully apply DRL for the closed loop control of a real world automotive automatic transmission of realistic complexity.

2.3. Sim-to-real transfer of deep reinforcement learning agents

Although DRL has generated impressive results in simulated environments in recent years, its application to real world systems remains an open question that gains increasing attention in the RL community [31]. Due to the high sample complexity of DRL approaches and their suboptimal behavior at the beginning of training, DRL agents are preferably trained in a simulated environment to limit training time and wear on the physical system. This results in the need for transferring the agents from the simulated source domain they were trained on to the physical target domain they are meant to control (see Fig. 1). This is known as sim-to-real transfer. An extensive overview of challenges for leveraging RL in real world systems can be found in [32]. The most relevant approaches for this work to overcome those challenges are domain randomization (DR) and domain adaption (DA).

Challenges in transferring DRL agents from simulation to reality arise from agents overfitting to simulation representations or exploiting modeling inaccuracies. When transferred to the physical system, those strategies show decreased performance or can even lead to complete failure. DR addresses this problem by perturbing the representation

of the environment within specified bounds, so agents are forced to deal with a variety of similar but different environments making their control strategy robust to a certain amount of representational uncertainty. This can, for instance, be achieved by randomizing representations of input data [33,34] or by varying dynamic parameters of the simulator [35].

DA aims to reduce the differences between simulation and reality by tuning the simulator, such that it behaves as similar as possible to measurements of the physical systems. Different approaches to that are reported in [36–38]. These share the underlying concept of collecting data from the physical system and tuning dynamics parameters of the simulator to fit those measurements using gradient free optimization, as simulators are usually not differentiable.

Other approaches for bridging the sim-to-real gap are increasing simulation fidelity by enhancing the simulation model [39,40], training agents able to adapt to changes in their environment [35,41–46] and training agents in an adversarial manner to enforce robustness [36,47–49].

3. Shifting control task

In an automotive drivetrain, a transmission transfers torque and rotational speed provided by the engine to the driven wheels. A variety of transmission ratios represented by different gears is required for different driving situations. In most automatic transmission, power flows over different sets of planetary gears which can be coupled and decoupled in various ways by means of electro-hydraulically actuated wet running multi plate clutches to realize different overall transmission ratios. Coupling and decoupling of planetary gear sets by engaging and disengaging those clutches represents the shifting procedures in the considered automatic transmission. The presented work focuses solely on shifting from neutral to first gear. This is achieved by engaging a single clutch of the transmission, in a stationary condition or during forwards or backwards coasting. Therefore, no torque transfer between two clutches is necessary, resulting in an easier control problem compared two dynamic shifts which require control of two clutches and the engine. However, the difference in transmission output torque from start to end of the shift is higher. Furthermore, due to lash between gear teeth and the clutch plate teeth and its carrier, a gentle touchpoint behavior is important. In this paper, constant transmission temperature is considered. In conventional control, temperature dependent parameter maps are used to allow control at low temperatures.

3.1. Clutch engagement

As depicted in Fig. 2, the functionality of an electro-hydraulically actuated wet running multi plate clutch is represented by a hydraulically actuated piston, several friction plates and a return spring for clutch disengagement. As the friction plates are connected to shaft 1 and shaft 2 in alternating order, friction between those plates allows to transfer a torque T_s from one shaft to the other.

To engage the clutch, the initial relative rotational speed $n(t_0)$ between shaft 1 and shaft 2 needs to be synchronized to zero. Assuming shaft 1 to be the input shaft and shaft 2 to be the output, the relation between relative rotational speed (slip speed) n and slipping torque T_s is given by

$$n(t) = n(t_0) - \int_0^t \frac{T_1 - T_s}{I_1} + \frac{T_s - T_2}{I_2} dt \tag{1}$$

with T_1 and T_2 being the respective torques of shaft 1 and 2, I_1 and I_2 representing the inertia of input and output. To achieve a desirable shifting behavior, T_s needs to be actuated by energizing a control solenoid valve which directs high pressure actuator oil over an oil channel into the oil chamber of the clutch. The resulting pressure in the chamber induces a force on the piston, counteracting the rebound spring force. As the hydraulic force exceeds the spring force, the piston

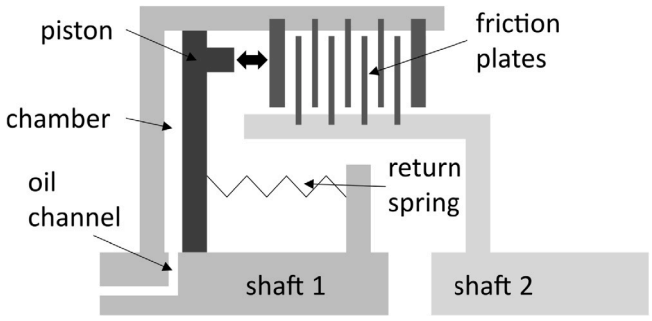

Fig. 2. Schematic cross section of a wet clutch, based on [50].

moves towards the friction plates. As soon as the touchpoint is reached, the piston touches the friction plates and applies a normal force F_n allowing them to transfer the slipping torque

$$T_s = F_n \cdot z \cdot r_m \cdot \mu \tag{2}$$

that in addition to F_n depends on the number of friction plate pairings z, their mean friction radius r_m and their friction coefficient μ. The constant factors z and r_m depend on the design of the clutch, whereas

$$F_n = f(p, I_c, H_h, H_s) \tag{3}$$

is a nonlinear function on the system pressure p of the hydraulic supply, the control current of the solenoid valve I_c, the transfer function of the hydraulic line to the piston H_h and the transfer function from the oil chamber pressure p_{ch} to the effective clutch force H_s, with all inputs being subject to uncertainties. The friction coefficient

$$\mu = f(p_c, T_{fp}, q_{oil}, n(t_0), n, x) \tag{4}$$

is a nonlinear function of the surface pressure of the friction plates p_c, their temperature T_{fp}, the coolant oil volume flow q_{oil} as well as the initial $n(t_0)$ and the current relative rotational speed n and other non-dominant factors grouped together as x. Both F_n and μ are noisy and cannot be measured in situ. The listed dependencies, uncertain influence factors and error aggregation over time make controlling gear shifts complicated. For an overview of the hydraulic control path the reader is referred to Appendix A.1.

3.2. Conventional shifting control

In the considered case, and in general for most gear shifts, the conventional actuation logic consists of an open loop control and a subordinate PI-controller in combination with adaptive schemes to achieve the desired trajectory of the clutches slip speed. The shift process can be subdivided into two major phases: The filling of the piston until the piston touches the friction plates and the synchronization phase. During filling, no change of variables can be measured. Therefore the open loop controller follows a predefined current trajectory that uses parameter maps designed for a range of operation points like temperature and torque demand. This open loop control is adaptively improved based on initial changes of the speed from previous shifts. During synchronization, in addition to the feed-forward controller, the current slip speed is calculated from sensor signals and compared to a predefined trajectory. The deviation is used for feedback-control of the actuation current. The controller can increase or decrease the transferable clutch torque and therefore the slip speed and avoid undesirable jerks. However, this conventional control strategy has several drawbacks, e.g. the limited allowed actuation range, to ensure corrections with a small jerk, and the impossibility to predict and counteract future offsets, since PID-control can only react to errors already occurred.

4. Methodology

The shifting control task introduced in Section 3 is aimed to be solved by DRL algorithms. In particular, two state of the art (SOTA) model-free DRL approaches are considered, namely Proximal Policy Optimization (PPO) [51] and Soft Actor Critic (SAC) [52] are applied to the control problem. A brief introduction to the general concept of reinforcement learning (RL) and the leveraged algorithms is given in Section 4.1.

To efficiently learn good policies, a suitable formulation of the control task as an MDP, including the careful design of observation and action spaces as well as the formulation of a meaningful reward function, is crucial. The solution is presented in Section 4.2. The measures taken for transferring the learned agents from simulation to reality are presented in Section 4.3. DR is applied to make agents robust against measurement noise from the test bench as well as parameter uncertainty. Furthermore, two DA methods, one depending on test driven updates, the other on data driven optimization are presented to reduce the sim-to-real gap.

4.1. Deep reinforcement learning

4.1.1. Introduction to reinforcement learning

In RL, an agent learns a certain behavior from interaction with an environment. This formalism builds upon the MDP framework. In this paper, a fully observable MDP defined by the tuple (S, A, T, r) with a continuous state space $S \in \mathbb{R}^n$ for $0 < n < \infty$ and action space $A \in \mathbb{R}$ is considered. The transition function of the environment $T(s, a, s')$, with $a \in A$ and $s, s' \in S$ gives the probability of a transition to the subsequent state s' when action a is performed in state s. Since in our case the state space is continuous, the transition function specifies a probability density function (PDF), such that

$$\int_{S'} T(s, a, s')ds' = \mathbb{P}(S_{s_{t+1}} \in S' | s_t = s, a_t = a)$$

denotes the probability that action a in state s results in a transition to a state in the region $S' \subseteq S$. At each state transition, the MDP emits a bounded reward $r : S \times A \times S' \to \mathbb{R}$. A RL agent is represented by a policy $\pi(a|s) : S \to \mathbb{P}(A)$ mapping the state to a probability density function over the continuous action space A, where each query to the policy given a particular state s samples an action a from the conditional distribution. Based on the described setting, a trajectory or rollout $\tau = (s_0, a_0, s_1, a_1, \dots, s_{N-1}, a_{N-1}, s_N)$ is defined as sequences of states and actions sampled from an arbitrary policy π. More precisely, the likelihood $\mathbb{P}(\tau|\pi)$ of a trajectory $\tau = (s_0, a_0, s_1, a_1, \dots, s_{N-1}, a_{N-1}, s_N)$ under the policy π is described by

$$\mathbb{P}(\tau|\pi) = \mathbb{P}(s_0) \prod_{t=t_0}^{N-1} T(s_t, a_t, s_{t+1})\pi(a_t|s_t).$$

The goal of the agent is to maximize the discounted return $R = \sum_{t=t_0}^{N} \gamma^t r(s_t, a_t)$, where $\gamma \in [0, 1]$ is a discount factor and $N \leq \infty$ is the horizon of each episode. The objective during learning is to find an optimal policy π^* that maximizes the expected return $J(\pi) = \mathbb{E}[R|\pi]$:

$$\pi^* = \text{argmax}_\pi J(\pi)$$

4.1.2. Deep reinforcement learning algorithms

Two SOTA model-free DRL approaches, namely the PPO algorithm [51] and the SAC algorithm [52] are applied to the shifting control task.

PPO represents an on-policy DRL algorithm, based on the policy gradient formulation [53]. Consequently, PPO can solely use trajectories observed under the current policy to perform policy updates. To stabilize learning, PPO leverages the concept of clipped gradient updates, which inspired by the idea of having trust regions to bound gradient updates [54], while being computationally less demanding. Furthermore, the implementation used in this work leverages the generalized advantage estimation formulation [55] to trade off between bias and variance of the advantage estimate required for policy updates.

SAC is an actor-critic Q-learning off-policy DRL algorithm. Hence, state transitions observed under any policy can be utilized for policy learning. Besides other important concepts of continuous deep Q-learning, such as a replay buffer and target Q-networks [1], the actor-critic architecture [56] and clipped double Q-Learning [57], SAC incorporates the soft MDP formulation introduced in [58]. Consequently, SAC learns a stochastic policy that maximizes the expected return as well as the entropy of the policy. Our implementation utilizes the improved architecture of SAC proposed in [52].

4.2. Markov decision process formulation of the shifting task

The application of RL techniques to the shifting control task requires to formulate the shifting process as an MDP. Therefore, the agent environment interaction needs to be designed with the definition of an appropriate state and action space formulation. Furthermore, a reward function that communicates desirable behavior to the agent needs to be designed.

The shifting procedure is structured according to an auxiliary measure, the shifting progress

$$\text{sp}(t) = \max((n(t_0) - n(t))/n(t_0), 0)$$

describing the portion of initial relative rotational speed that has already been synchronized. Based on this formulation, the shifting procedure can be divided into three sub phases ph:

- $ph = 1$: Filling phase: 0%–10% shifting progress
- $ph = 2$: Power conversion phase: 10%–90% shifting progress
- $ph = 3$: Harmonization phase: 90%–100% shifting progress.

During the filling phase, oil flows into the piston chamber and the piston moves towards the friction plates, however no significant normal force is applied to the friction plates resulting in a slipping torque close to zero. The threshold of 10% is an engineering decision based on expert knowledge, making the phase definition robust against sensory noise in the determination of the current relative rotational speed. The phase corresponding to a shifting progress from 10% up to 90% is referred to as power conversion phase. During this phase the piston applies a significant normal force on the clutch and the main portion of the synchronization is achieved. The harmonization phase corresponds to a shifting progress from 90% to 100%. During this phase, the last portion of relative rotational speed is synchronized. Empirical research shows, that the agent benefits from this subdivision of the synchronization phase, since the additional change of input state leads to a smoother final synchronization.

4.2.1. State formulation

The state s_t of the environment is defined by

- Elapsed time (in ms) since initialization of the shifting procedure: $t - t_0$
- Shifting progress $sp(t)$
- Phase $ph(t)$
- Control current applied to the solenoid hydraulic valve $I_c(t)$
- Fill state $fs(t)$

giving a compact, yet expressive formulation of the point of operation of the transmission. Most of the signals used for the state formulation are introduced above, whereas the fill state signal requires further explanation.

In order to fulfill the high requirements on cost efficient design, the transmission only provides sensors for the most relevant measures of the shifting process, leading to partial observability of the shifting

process, which is a known challenge in the deployment of RL approaches [32]. In particular, there is no expressive physical measure available for the state of the system during filling phase. For high performance shifting it is however crucial to know precisely when the piston starts to touch the friction plates and a slipping torque is transferred. Consequently, a fill state estimation model to overcome the partial observability during filling is trained.

The fill state estimator is implemented as a feed-forward neural network (FNN) trained in a supervised learning setup (see Appendix A.4). To generate labeled training data, I_c actuation trajectories of random length are generated. Half of the trajectories are generated completely random, whereas the other half of the trajectories are randomized and noisy versions of the conventional actuation strategy. This takes care of the fact that the fill state estimator is required to perform sufficiently accurate for any actuation but a more precise prediction is desired for reasonable trajectories. After the trajectories of random length are applied, the control current is kept constant at its maximum level. As soon as the phase transition from filling phase to power conversion phase is reached, indicated by an shifting progress of 10%, the run is stopped and the corresponding data is stored. For each run, the random trajectory of control currents is the feature and the number of maximum current time steps until filling is completed is the label. Consequently, the FNN is trained to predict how many time steps of maximum level current actuation are left until filling is completed, given the history of control currents applied to the system so far. During the filling phase, the estimated fill state fs is provided to the agent, whereas in the subsequent phases this value is set to -1 to indicate the fill state estimator is inactive.

4.2.2. Action formulation

The policy provides the environment not with the control current I_c as action a_t directly but with a differential control current applied to the hydraulic pressure control valve as an action, i.e. $I_{c,t+1} = I_{c,t} + a_t$. The action is scaled according to the phase of the shifting procedure. Thus, the action $a \in [-1, 1]$ is modified by following formula:

$$a_t = \begin{cases} 100a_t, & \text{filling phase} \\ 10a_t, & \text{power conversion and harmonization phase} \end{cases}.$$

This is motivated by the fact that the agent needs to vary the actuation in a wide range during the filling phase, whereas a careful actuation is required during the subsequent phases as a change in the control current strongly impacts the slipping torque. In this way, the space of possible solutions is restricted by removing some implausible ones. This clearly speeds up the learning process.

4.2.3. Reward formulation

The main objective of the shifting control task are fast, yet comfortable gear shifts. Fast corresponds to a short time from initialization of the shift procedure until complete synchronization of relative rotational speed, whereas comfortable corresponds to a low longitudinal jerk during shifting.

As a proxi-measure for the longitudinal jerk, we use the gradient of the output torque of the transmission $j = \frac{\mathrm{d}T_{out}}{\mathrm{d}t}$ that correlates to the gradient of the relative rotational speed of the synchronized clutch.

Consequently, fast and comfortable are competing goals of the shifting procedure as a high gradient in relative rotational speed leads to fast, uncomfortable shifts and vice versa. The reward formulation

$$r = F_1 * p_{jerk} + F_2 * p_{time} + F_3 * p_{decr} + F_4 * p_{ft} + F_5 * p_{fc} + F_6 * p_{end} \quad (5)$$

includes both goals in an additive manner.

To enforce fast gear shifts, a constant penalty p_{time} is given in each simulation time step. To penalize non comfortable shifts, a penalty p_{jerk} based on the cubic absolute value of j is applied.

Moreover, various expert knowledge based penalties are added: A dense penalty p_{decr} is given for a decrease in shifting progress, to enforce a monotone reduction of relative rotational speed. At the end of the filling phase, two optional rewards are added: One penalty p_{ft} for exceeding maximal acceptable filling time and another penalty p_{fc}, based on how much the control current at end of the filling phase exceeds a predefined maximum, above which no comfortable shift can be achieved. If the agent cannot complete a full shifting process within a given time limit, an additional dense penalty p_{end} is given, based on the achieved shifting progress.

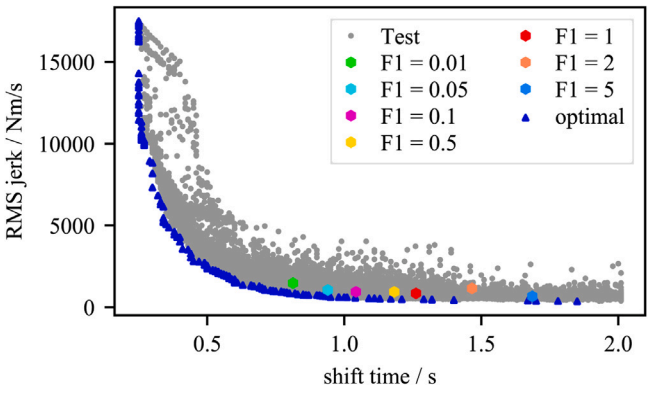

Fig. 3. All solutions of agents during testing (gray), mean convergence points of agents with settings $F_1 \in [0.01, 0.05, 0.1, 0.5, 1, 2, 5]$ (colored hexagons) and pareto optimal solutions (triangles). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The expert reward terms should eventually reach zero, once the agent converges towards an optimal solution. Through different weighting of the competing goals through F_1 and F_2, agents for different pareto optimal solutions can be trained yielding either comfortable slow shifts or sportive shifts with higher jerk peaks. Fig. 3 depicts various combinations of the root mean square (RMS) value of the transmission output jerk j and synchronization time of agents with weighting of the p_{jerk} with the values $F_1 \in [0.01, 0.05, 0.1, 0.5, 1, 2, 5]$ during testing in simulation. $F_1 = 1$ is giving a balanced weighting of jerk and shifting time reward, comparable to conventional control of series production. For each setting, 20 agents are trained for 2 million time steps, which roughly corresponds to 40.000 gearshifts. All agents converge to a stable policy within the training setup. For each setting the convergence point is displayed in color. Each test point in Fig. 3 corresponds to a found solution by an agent. From this set, the pareto optimal solutions are determined and depicted as blue triangles. The lineup of the convergence points meet the expected order with longer and smoother shifts as $F1$ increases, thereby verifying the selected reward formulation. During the initial exploration phase the agents find many solutions with a shifting time under 0.75s and a undesirable high jerk.

It is up to the calibration engineer to determine the objective along the pareto front. However, it is increasingly complicated to achieve robust solutions close to the pareto front for short shifting times since there are only few interactions possible as well for long shifting times where long term dependencies increase the difficulty of the control task.

4.3. Transfer from simulator to transmission

Validation of the agents is performed on a specialized transmission test bench using a production series transmission. The test bench consists of electrical machines to model the combustion engine and the car's driving resistance. Both electrical machines as well as the transmission are connected to a highly accurate simulator, imitating the rest of the drivetrain and all necessary ECUs. Only the agent's policy networks are used to compute control inputs for the solenoid hydraulic valve. When deploying to the real transmission, the control performance of the agents potentially suffers from the sim-to-real gap

introduced in Section 2.3. To enable a successful transfer DR and two DA methods are leveraged. For evaluation of the shifting quality, it is best practice to define key performance indicators (KPIs) [59]. The mean μ and standard deviation σ of the total shift time t and the root-mean-square (RMS) value of the jerk j_{RMS} are used as KPIs, based on the competing goals defined in Section 4.2.3.

4.3.1. Domain randomization

To make DRL agents robust against a limited amount of uncertainty about the dynamic behavior of the synchronization process, different simulation parameters that are known to be crucial for the clutch dynamic are sampled from a Gaussian distribution at the initialization of each shift during training (see Fig. 1). Those parameters mainly influence μ, H_h and H_s and force the agent to learn a robust policy in face of unknown behavior of its environment. The mean values of the distributions are either expectations from expert knowledge or optimized to fit the dynamics of the real transmission. The variance is chosen as an engineering decision yielding sufficiently robust but not overly conservative agents. Additionally the policies have to be robust against noise from sensors. Therefore, Gaussian noise is added to the relative rotational speed of the simulator before calculating the shifting process for the state. For a full list of randomization parameters the reader is referred to Appendix A.2.

4.3.2. Domain adaption approach: Test driven

A major challenge in the hydraulic control path Appendix A.1 is the delayed system response during the filling phase, where no change in relative rotational speed is measurable until the filling is completed. If the chamber gets overfilled, the piston reaches the touchpoint with a high velocity, resulting in a high initial jerk. Due to the partial observability, the agent is guided by the fill state estimator, that has a high effect on its behavior during filling. In this test driven domain adaption approach (DA1), the policies are trained on the standard setting of the simulator, where the dynamic parameters are chosen as to be assumed on the test bench. DR is active during training to create robust agents, that can deal with small differences between simulation and test bench parameters. By transferring those trained policies to the test bench, multiple gear shifts are executed with the standard policy. Then, the fill state estimator is adapted to the observed behavior of this gear shift, namely a linear correction is applied to the fill state fs such that it matches the behavior of the test bench transmission (see Fig. 1). Directly afterwards, the identical policies with adapted fill state estimator are tested on the transmission test bench.

4.3.3. Domain adaption approach: Data driven

The data driven domain adaption approach (DA2) aims to reduce model inaccuracies, that result in deviations between the simulator dynamics and the real system (see Fig. 1).

Therefore, relevant dynamic parameters of the simulator are determined by experts and tuned such that it acts as similar as possible to measurements from the real system. In order to optimize those parameters, a batch of shifts on the test bench is collected. The action trajectories observed from the test bench are applied to the simulator and a loss function is formulated to minimize deviations in the trajectories of relative rotational speed n and hydraulic system pressures between simulator and real system. As the simulator is not differentiable, the covariance matrix adaption evolutionary strategy (CMA-ES) optimizer [60] is utilized to tune the dynamic parameters. CMA-ES is a state of the art gradient free optimizer that is an especially suitable choice for black box optimization problems in the continuous domain with a high dimensional search space [61]. The optimized values serve as mean values of the DR distributions.

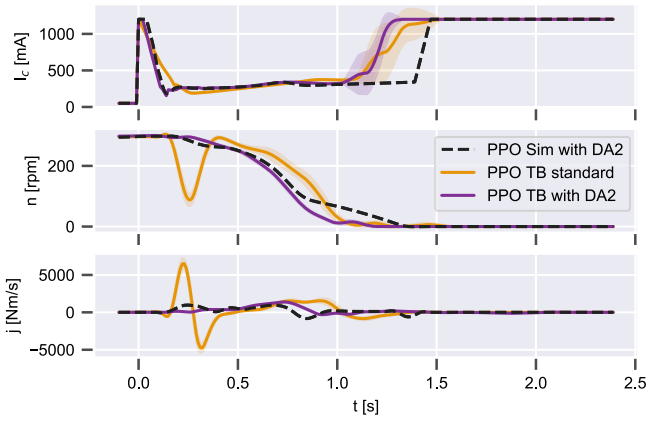

Fig. 4. Gear shifting procedures conducted by PPO agents. PPO learns a suitable actuation of the control current I_c after purely training in simulation, that triggers a fairly quick reduction of the relative rotational speed n keeping the maximal jerk j low (PPO Sim with DA2, mean of 10 simulation runs). When transferred to the transmission test bench, the PPO policy produces a high and uncomfortable jerk j (PPO TB standard, mean and standard deviation of 50 test bench runs). Optimizing the simulator according to the DA2 approach (see Section 4.3.3) and training a new agent in this environment enables a successful sim-to-real transfer (PPO TB with DA2, mean and standard deviation of 50 test bench runs). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

5. Results

The results of the methods introduced above are presented in the following. Shifts conducted by PPO agents are depicted in Fig. 4, whereas shifts conducted by SAC agents are shown in Fig. 5. Both figures illustrate the three most relevant measures of the shifting process, namely the control current on the solenoid valve that depends on the agent's actions, the relative rotational speed over the clutch that needs to be synchronized to zero and the jerk produced by the shift that needs to be kept low for comfortable shifting. Both SAC and PPO are able to learn suitable policies to conduct high performance shifts in simulation. This can be seen from the black dotted lines in Figs. 4 and 5. Those depict the mean of 10 shifting procedures of one agent trained in simulation leveraging DR. For both algorithms, five agents are trained in simulation (training hyperparameters are listed in Appendix A.3) with DR (noise hyperparameters are listed in Appendix A.2) and the best performing agent is used to conduct the plotted shifting procedures. Both algorithms learn actuation strategies to reduce the relative rotational speed to zero quickly, while producing a low amount of jerk. This is achieved by a high control current at the beginning of the shift that fills the chamber with oil quickly. By the time the piston starts to touch the friction plates, the control current is reduced such that normal force on the friction plates and hence slipping torque is built gradually, resulting in a gentle reduction of relative rotational speed causing a low amount of jerk. During synchronization, the control current is increased slowly but constantly resulting in an continuously rising transferable slipping torque. As soon as the relative rotational speed reaches zero, a hard programmed logic ramps up the control current to its maximum value to lock the clutch. This ramp up process is not considered part of the shifting process subsequently, as the DRL agents are not in control during this procedure.

When transferring those policies directly to the test bench without applying DA, shifting quality suffers as expected. The orange plots in Figs. 4 and 5 show the mean and standard deviation of 50 shifts on the test bench conducted by the same agent as was used for shifting in simulation. As the piston of the real transmission touches the friction plates earlier than in simulation, control current is not reduced on time. This causes an abrupt rise in transferable slipping torque resulting in high peaks of jerk. Shifting quality of both SAC and PPO agents are

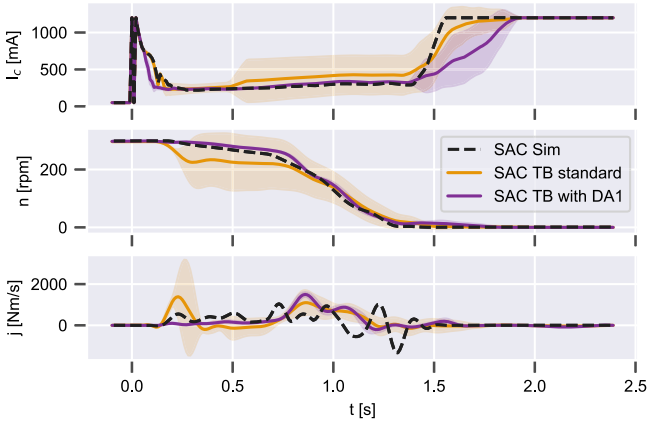

Fig. 5. Gear shifting procedures conducted by SAC agents. SAC learns a suitable actuation of the control current I_c after purely training in simulation, that triggers a quick reduction of the relative rotational speed n keeping the jerk j at low levels (SAC Sim, mean of 10 simulation runs). When transferred to the transmission test bench, the SAC policy produces a high variance in the synchronization trajectories with degradation of the key performance indicators (SAC TB standard, mean and standard deviation of 50 test bench runs). However, when adapting the fill state estimation model according to the DA1 (see Section 4.3.2) approach, successful sim-to-real transfer can be reported (SAC TB with DA1, mean and standard deviation of 50 test bench runs). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

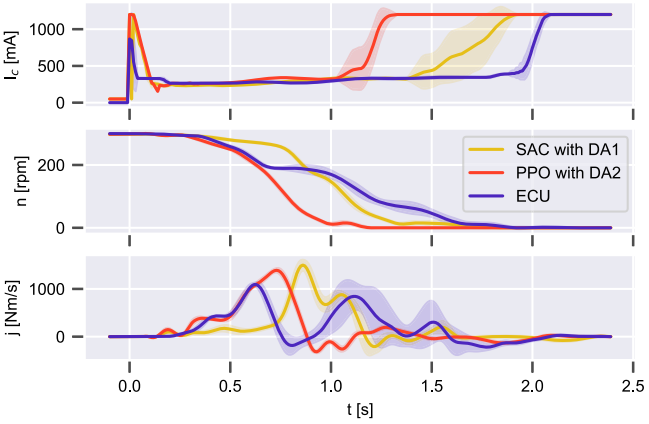

Fig. 6. Overview of gear shifting procedures conducted by two DRL agents compared to the ECU controller. The gear shift is actuated by the control current I_c that produces a reduction of the relative rotational speed n over time and causes a jerk j. It can be stated, that both DRL approaches show a high quality actuation in terms of time t and jerk j_{RMS} while successfully bridging the sim-to-real gap. SAC with DA1 shows a slightly improvement of shifting time and jerk compared to the ECU controller while keeping variance low. PPO with DA2 shows a major improvement in the shifting time at the cost of a slightly increased jerk. Note that time and jerk are competing goals and that the presented approach allows for an automated synthesis of new DRL control strategies by changing weights in the reward function (Eq. (5)).

unacceptable. Consequently, DA methods are required for successful transfer of the trained agents. The best results for PPO when evaluated on the transmission test bench can be achieved with the data driven DA2 approach (see Section 4.3.3), whereas best results of SAC are observed applying the test driven DA1 approach (see Section 4.3.2).

The violet plots in Figs. 4 and 5 show the mean and standard deviation of 50 shifts using agents leveraging the corresponding DA approach. It can be observed that both DA agents show a considerably reduced standard deviation region as compared to the directly transferred agents.

The PPO agent shows a very good performance that however differs from the behavior of the agent shifting in simulation. This is, as the PPO agent leveraging DA2 is a new agent trained from scratch in the adapted simulator. The SAC agent leveraging the DA1 approach instead, shows very similar behavior as in simulation as the agent is the same in both cases and solely the fill state estimator is updated to fit the test bench dynamic. Finally, the performance of the DRL agents with their respective adaption method are compared to the conventional gearshift controller, including open loop control, closed loop control and adaptive schemes (see Section 3.2), of the ECU, see Fig. 6. Tests are carried out using a production version of the Mercedes-Benz 9G-TRONIC automatic transmission. As should be recovered from Fig. 3, low shifting time and low jerks are competing quality criteria, with lower shifting times resulting in a higher jerk. The PPO agent significantly reduces shifting time, while producing only slightly more jerk during shifting as compared to the ECU shifts. The SAC agent instead improves both quality criteria slightly as compared to the standard shifting controller.

Table 1 summarizes the achieved results using the aforementioned mean μ and standard deviation σ of the KPIs total shift time t and RMS value of the jerk j_{RMS}.

Table 1
Summary of test bench results consisting of the mean μ and standard deviation σ of the KPIs total shift time t and RMS value of the jerk j_{RMS} for SAC, PPO and ECU.

Adaption	SAC		PPO		ECU
	Standard	With DA1	Standard	With DA2	
μ_t [s]	1.41	1.69	1.22	1.18	1.83
σ_t [s]	0.33	0.13	0.09	0.03	0.11
$\mu_{j,RMS}$ [N m/s]	566.04	396.06	1324.27	416.70	412.81
$\sigma_{j,RMS}$ [N m/s]	219.82	19.90	133.79	8.10	48.80

6. Discussion

Even though direct transfer of DRL agents trained in the standard simulator is found to be infeasible, both of the DRL agents leveraging DA can be reported to outperform the current ECU control approach on a testbench. Shifting time can be reduced significantly while producing no or negligible additional jerk. This proves the potential of DRL methods to improve upon the state of the art in automotive applications such as learning gear shift controllers.

The proposed DA methods show different strengths and weaknesses. DA1 represents a method for quick adaption to the physical system which is computationally cheap, as the test driven adaption of the fill state estimator allows to reuse agents trained in the standard simulator. However, this approach did not lead to satisfactory results when applying to PPO agents, as their actuation strategy seems to account less for the fill state input. DA2 on the other hand, provides a structured way to enhance simulator accuracy, leading to the best shifting results achieved in the investigated setup when combined with PPO agents. This comes at the cost of retraining agents in the adapted simulator after observing dynamic behavior of the physical system. It should be noted that the test bench measurements need not necessarily to be recorded under a DRL control policy, but other control approaches such as linear controllers can serve as an observation policy as well. Results of DA2 in combination with SAC agents did not show as good performance as could be reported for PPO agents. It appears that SAC agents do not generalize as well to the adapted simulator dynamics and tend to show noisy actuation at the end of the filling phase. Consequently, further research to stabilize all combinations of proposed approaches is required. Nevertheless, both DA approaches show that successful sim-to-real transfer of DRL agents can enhance shifting quality considerably.

It should be noted that the evaluation is done using a fixed setup and no extrapolation to different oil temperatures or other starting conditions occurring for the transmission in reality are evaluated. No

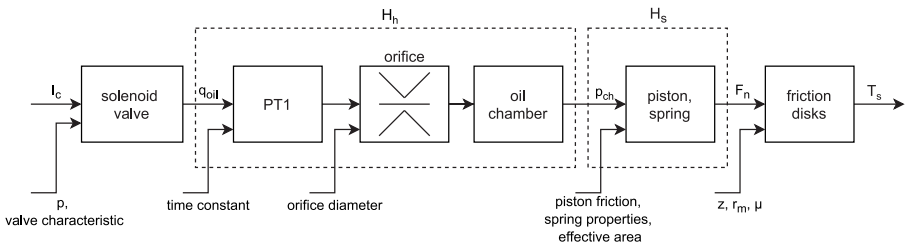

Fig. A.7. Schematic control path from the control current I_c to the slipping torque T_s.

statement can be made on the performance for test scenarios with different conditions so far. But research in that direction shows promising results.

7. Conclusion

The presented methodology, namely the application of SOTA DRL algorithms to the shifting control task in a modern automatic transmission shows the tremendous potential of such techniques for automotive applications. To the best of the authors knowledge, this work is the first to successfully apply DRL techniques for the closed loop control of a real world transmission of realistic complexity. It can be reported that the control performance of classical control approaches can be outperformed, while reducing the required effort to fine tune the control system. Furthermore, the proposed reward formulation allows to explicitly trade off between the main quality criteria of a gear shift, namely shifting time and shifting comfort.

Designing a suitable formulation of the shifting task in the MDP framework as well as the sim-to-real transfer of the DRL agents were found to be the main challenges for a successful application.

Subsequently, the presented approach will be transferred to gear shifts with multiple active clutches. In particular, this includes some changes to the described training framework and the formulated MDP, such as the expansion of the action state to control two clutches and the requested engine torque, and constructing a new reward function, based on the mentioned criteria. First simulation runs show promising results. Additionally, future research focuses on transferring the approach to new automatic transmissions and to further enhance automatization, simulation fidelity and the capabilities of our approach.

CRediT authorship contribution statement

Gerd Gaiselmann: Conceptualization, Methodology, Software, Writing – original draft. **Stefan Altenburg:** Methodology, Software, Data curation, Investigation, Test bench validation, Visualization, Writing – original draft, Review & editing. **Stefan Studer:** Validation, Visualization, Writing – original draft. **Steven Peters:** Supervision, Project administration, Writing – review.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to thank Bernd Frauenknecht for implementing the PPO algorithm and the DA2 approach and Thomas Doster and Katja Deuschl for their creativity and commitment to promote the novel techniques for transmission control. Specials thanks are forwarded to Emmanuel Chrisofakis and his colleagues to push the limits of the physics simulator. Moreover, the foresight of our management to support this novel technique from the project start until today is highly appreciated.

Appendix

A.1. Hydraulic control path

The hydraulic control path from the control current I_c to the slipping torque T_s is displayed in Fig. A.7. The transfer function H_h from the oil volume flow q_{oil} from solenoid valve to the oil pressure in the chamber p_{ch} consists of a first order plus dead time block, an orifice and the oil chamber. The subsequent transfer function H_s includes the piston friction and the return spring, depending on the effective area of the piston, its kinetic friction, and the spring properties, namely the return spring preload, its deflection at the touchpoint and its characteristic. Finally, the slipping torque T_s of the friction disks control the relative rotational speed, see Eq. (1).

A.2. Domain randomization setup

Parameter	Noise application
Hydraulic orifice diameters	Per episode
Clutch friction coefficient offset	Per episode
Piston friction coefficient	Per episode
PT1 time constant	Per episode
Clutch friction coefficient	Per timestep
Relative rotational speed	Per timestep

A.3. Training parameters SAC and PPO

	SAC	PPO
NN type	MLP	MLP
Policy architecture (net_arch)	[128, 64, 64]	[128, 64, 64]
Learning rate	0.0003	0.0003
Buffer size	1e6	–
Batch size	128	128
Max shift time	1.4 s	1.4 s
Number of timesteps	2.5e6	5.0e6
Timesteps training until test run	5e4	5e4

A.4. Training parameters fill state estimator

	Fill state estimator
NN type	MLP
NN architecture	[16, 16, 1]
Activation	Relu
Loss	Mean squared error
Optimizer	Adam
Epochs	100
Batch size	128

References

[1] Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, Graves A, Riedmiller M, Fidjeland AK, Ostrovski G, Petersen S, Beattie C, Sadik A, Antonoglou I, King H, Kumaran D, Wierstra D, Legg S, Hassabis D. Human-level control through deep reinforcement learning. Nature 2015;518(7540):529–33. http://dx.doi.org/10.1038/nature14236.

[2] Silver D, Huang A, Maddison CJ, Guez A, Sifre L, van den Driessche G, Schrittwieser J, Antonoglou I, Panneershelvam V, Lanctot M, Dieleman S, Grewe D, Nham J, Kalchbrenner N, Sutskever I, Lillicrap T, Leach M, Kavukcuoglu K, Graepel T, Hassabis D. Mastering the game of go with deep neural networks and tree search. Nature 2016;529(7587):484–9. http://dx.doi.org/10.1038/nature16961.

[3] Lin L-J. Reinforcement learning for robots using neural networks (Ph.D. thesis), Carnegie Mellon University; 1993.

[4] Ha S, Xu P, Tan Z, Levine S, Tan J. Learning to walk in the real world with minimal human effort. In: CoRL 2020. 2020.

[5] Hwangbo J, Lee J, Dosovitskiy A, Bellicoso D, Tsounis V, Koltun V, Hutter M. Learning agile and dynamic motor skills for legged robots. Science Robotics 2019;4(26):eaau5872. http://dx.doi.org/10.1126/scirobotics.aau5872.

[6] Raziei Z, Moghaddam M. Adaptable automation with modular deep reinforcement learning and policy transfer. Eng Appl Artif Intell 2021;103:104296. http://dx.doi.org/10.1016/j.engappai.2021.104296.

[7] Schaefer AM, Schneegass D, Sterzing V, Udluft S. A neural reinforcement learning approach to gas turbine control. In: IJCNN. 2007, p. 1691–6. http://dx.doi.org/10.1109/IJCNN.2007.4371212.

[8] Li Y, Wen Y, Tao D, Guan K. Transforming cooling optimization for green data center via deep reinforcement learning. IEEE Trans Cybern 2020;50(5):2002–13. http://dx.doi.org/10.1109/TCYB.2019.2927410.

[9] Horste K. Objective measurement of automatic transmission shift feel using vibration dose value. In: SAE technical paper series. SAE technical paper series, 1995, http://dx.doi.org/10.4271/951373.

[10] Maier P. Entwicklung einer methode zur objektivierung der subjektiven wahrnehmung von antriebsstrangerregten fahrzeugschwingungen (Ph.D. thesis), Karlsruher Institut für Technologie; 2011, URL https://d-nb.info/1021178756/34.

[11] Bagot B, Bek M, Vollmar R. Grundlegende innovationen im applikationsprozess von automatikgetrieben. In: Optimierung komplexer antriebsstränge - die herausforderung der zukunft. ISDM; 2007, p. 147.

[12] Kahlbau S. Mehrkriterielle optimierung des schaltablaufs von automatikgetrieben (Ph.D. thesis), BTU Cottbus-Senftenberg; 2013.

[13] Küçükay F, Kassel T, Alvermann G, Gartung T. Effiziente abstimmung von automatisch schaltenden getrieben auf dem rollenprüfstand. ATZ Automobiltech Z 2009;111(3):216–23. http://dx.doi.org/10.1007/BF03222062.

[14] Du Z, Miao Q, Zong C. Trajectory planning for automated parking systems using deep reinforcement learning. Int J Automot Technol 2020;21(4):881–7. http://dx.doi.org/10.1007/s12239-020-0085-9.

[15] Sallab A, Abdou M, Perot E, Yogamani S. Deep reinforcement learning framework for autonomous driving. Electron Imaging 2017;2017(19):70–6. http://dx.doi.org/10.2352/ISSN.2470-1173.2017.19.AVM-023.

[16] Xue J, Gao Q, Ju W. Reinforcement learning for engine idle speed control. In: ICMTMA. Piscataway, NJ: IEEE; 2010, p. 1008–11. http://dx.doi.org/10.1109/ICMTMA.2010.249.

[17] Hu B, Li J. Shifting deep reinforcement learning algorithm towards training directly in transient real-world environment: a case study in powertrain control. IEEE Trans Ind Inf 2021;17(12):8198–206. http://dx.doi.org/10.1109/TII.2021.3063489.

[18] Biswas A, Anselma PG, Emadi A. Real-time optimal energy management of electrified powertrains with reinforcement learning. In: ITEC. IEEE; 2019.

[19] Heimrath A, Froeschl J, Baumgarten U. Reflex-augmented reinforcement learning for electrical energy management in vehicles. In: ICAI. 2018, p. 429–30.

[20] Li G, Gorges D. Ecological adaptive cruise control for vehicles with step-gear transmission based on reinforcement learning. IEEE Trans Intell Transp Syst 2020;21(11):4895–905. http://dx.doi.org/10.1109/TITS.2019.2947756.

[21] Ngo DV, Hofman T, Steinbuch M, Serrarens A, Merkx L. Improvement of fuel economy in power-shift automated manual transmission through shift strategy optimization - an experimental study. In: VPPC. IEEE; 2010, p. 1–5. http://dx.doi.org/10.1109/VPPC.2010.5729123.

[22] Bécsi T, Aradi S, Szabó A, Gáspár P. Policy gradient based reinforcement learning control design of an electro-pneumatic gearbox actuator. IFAC-PapersOnLine 2018;51(22):405–11. http://dx.doi.org/10.1016/j.ifacol.2018.11.577.

[23] Bécsi T, Szabó A, Kovari B, Aradi S, Gáspár P. Reinforcement learning based control design for a floating piston pneumatic gearbox actuator. IEEE Access 2020;8:147295–312. http://dx.doi.org/10.1109/ACCESS.2020.3015576.

[24] Sommer Obando H. Reinforcement learning framework for the self-learning suppression of clutch judder in automotive drive trains (Ph.D. thesis), Karlsruher Institut für Technologie (KIT); 2016, http://dx.doi.org/10.5445/IR/1000061436.

[25] Gagliolo M, van Vaerenbergh K, Rodríguez A, Nowé A, Goossens S, Pinte G, Symens W. Policy search reinforcement learning for automatic wet clutch engagement. In: ICSTCC. 2011, p. 1–6.

[26] van Vaerenbergh K, Rodríguez A, Gagliolo M, Vrancx P, Nowé A, Stoev J, Goossens S, Pinte G, Symens W. Improving wet clutch engagement with reinforcement learning. In: IJCNN. 2012, p. 1–8. http://dx.doi.org/10.1109/IJCNN.2012.6252825.

[27] Pinte G, Stoev J, Symens W, Dutta A, Zhong Y, Wyns B, de Keyser R, Depraetere B, Swevers J, Gagliolo M, Nowe A. Learning strategies for wet clutch control. In: ICSTCC. 2011, p. 438–45.

[28] Dutta A, Zhong Y, Depraetere B, van Vaerenbergh K, Ionescu C, Wyns B, Pinte G, Nowe A, Swevers J, de Keyser R. Model-based and model-free learning strategies for wet clutch control. Mechatronics 2014;24(8):1008–20. http://dx.doi.org/10.1016/j.mechatronics.2014.03.006.

[29] Lampe A, Serway R, Siestrup L, Guehmann C. Artificial intelligence in transmission control – Clutch engagement with reinforcement learning. 2019, p. I–113. http://dx.doi.org/10.51202/9783181023549-I-113.

[30] Deuschl K, Doster T, Gaiselmann G, Studer S. Automated functional development for automatic transmissions using deep reinforcement learning. ATZ Electron Worldw 2020;15:8–13. http://dx.doi.org/10.1007/s38314-020-0252-9.

[31] Höfer S, Bekris K, Handa A, Gamboa JC, Golemo F, Mozifian M, Atkeson C, Fox D, Goldberg K, Leonard J, Liu CK, Peters J, Song S, Welinder P, White M. Perspectives on Sim2Real Transfer for Robotics: A Summary of the R:SS 2020 Workshop.

[32] Dulac-Arnold G, Levine N, Mankowitz DJ, Li J, Paduraru C, Gowal S, Hester T. Challenges of real-world reinforcement learning: definitions, benchmarks and analysis. Mach Learn 2021;110:2419–68. http://dx.doi.org/10.1007/s10994-021-05961-4.

[33] Sadeghi F, Levine S. CAD2RL: Real single-image flight without a single real image. In: RSS. 2017, http://dx.doi.org/10.15607/RSS.2017.XIII.034.

[34] Tobin J, Fong R, Ray A, Schneider J, Zaremba W, Abbeel P. Domain randomization for transferring deep neural networks from simulation to the real world. In: IROS. 2017, p. 23–30. http://dx.doi.org/10.1109/IROS.2017.8202133.

[35] OpenAI, Akkaya I, Andrychowicz M, Chociej M, Litwin M, McGrew B, Petron A, Paino A, Plappert M, Powell G, Ribas R, Schneider J, Tezak N, Tworek J, Welinder P, Weng L, Yuan Q, Zaremba W, Zhang L. Solving rubik's cube with a robot hand. 2019, CoRR, abs/1910.07113.

[36] Rajeswaran A, Ghotra S, Levine S, Ravindran B. EPOpt: LEarning robust neural network policies using model ensembles. 2016, CoRR, abs/1610.01283 URL http://arxiv.org/abs/1610.01283.

[37] Chebotar Y, Handa A, Makoviychuk V, Macklin M, Issac J, Ratliff N, Fox D. Closing the sim-to-real loop: Adapting simulation randomization with real world experience. In: ICRA. 2019, p. 8973–9. http://dx.doi.org/10.1109/ICRA.2019.8793789.

[38] Vuong Q, Vikram S, Su H, Gao S, Christensen HI. How to pick the domain randomization parameters for sim-to-real transfer of reinforcement learning policies?, URL http://arxiv.org/pdf/1903.11774v1.

[39] Tan J, Zhang T, Coumans E, Iscen A, Bai Y, Hafner D, Bohez S, Vanhoucke V. Sim-to-real: Learning agile locomotion for quadruped robots. Robot: Sci Syst XIV 2018.

[40] Hwangbo J, Lee J, Dosovitskiy A, Bellicoso D, Tsounis V, Koltun V, Hutter M. Learning agile and dynamic motor skills for legged robots. Science Robotics 2019;4(26):eaau5872. http://dx.doi.org/10.1126/scirobotics.aau5872.

[41] Christiano P, Shah Z, Mordatch I, Schneider J, Blackwell T, Tobin J, Abbeel P, Zaremba W. Transfer from simulation to real world through learning deep inverse dynamics model. In: ICRA. 2019.

[42] Yu W, Tan J, Liu CK, Turk G. Preparing for the unknown: learning a universal policy with online system identification. 2017, CoRR, abs/1702.02453 URL http://arxiv.org/abs/1702.02453.

[43] Yu W, Liu CK, Turk G. Policy transfer with strategy optimization. 2018, CoRR, abs/1810.05751 URL http://arxiv.org/abs/1810.05751.

[44] Zhang C, Yu Y, Zhou Z-H. Learning environmental calibration actions for policy self-evolution. In: Rosenschein JS, Lang J, editors. IJCAI. California: IJCAI; 2018, p. 3061–7. http://dx.doi.org/10.24963/ijcai.2018/425.

[45] Peng XB, Andrychowicz M, Zaremba W, Abbeel P. Sim-to-real transfer of robotic control with dynamics randomization. In: Lynch K, editor. ICRA. Piscataway, NJ: IEEE; 2018.

[46] OpenAI, Andrychowicz M, Baker B, Chociej M, Józefowicz R, McGrew B, Pachocki J, Petron A, Plappert M, Powell G, Ray A, Schneider J, Sidor S, Tobin J, Welinder P, Weng L, Zaremba W. Learning dexterous in-hand manipulation. Int J Robot Res 2020;39(1):3–20. http://dx.doi.org/10.1177/0278364919887447.

[47] Pinto L, Davidson J, Sukthankar R, Gupta A. Robust adversarial reinforcement learning. In: ICML. ICML'17, 2017, p. 2817–26.

[48] Narayanaswami SK, Sudarsanam N, Ravindran B. An active learning framework for efficient robust policy search, URL http://arxiv.org/pdf/1901.00117v1.

[49] Mehta B, Diaz M, Golemo F, Pal CJ, Paull L. Active domain randomization. In: CORL. 2019.

[50] Depraetere B, Pinte G, Van den Broeck L, Swevers J. A two-level optimization based learning control strategy for wet clutches. In: IFAC, Vol. 1. 2010, http://dx.doi.org/10.3182/20100826-3-TR-4015.00013.

[51] Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O. Proximal policy optimization algorithms. 2017, CoRR, abs/1707.06347 URL http://arxiv.org/abs/1707.06347.

[52] Haarnoja T, Zhou A, Hartikainen K, Tucker G, Ha S, Tan J, Kumar V, Zhu H, Gupta A, Abbeel P, Levine S. Soft actor-critic algorithms and applications. 2018, CoRR, abs/1812.05905 URL http://arxiv.org/abs/1812.05905.

[53] Williams RJ. Simple statistical gradient-following algorithms for connectionist reinforcement learning. Mach Learn 1992;8(3–4):229–56. http://dx.doi.org/10.1007/BF00992696.

[54] Schulman J, Levine S, Moritz P, Jordan MI, Abbeel P. Trust region policy optimization. In: ICML. 2015.

[55] Schulman J, Moritz P, Levine S, Jordan M, Abbeel P. High-dimensional continuous control using generalized advantage estimation. In: ICLR. 2016.

[56] Lillicrap TP, Hunt JJ, Pritzel A, Heess N, Erez T, Tassa Y, Silver D, Wierstra D. Continuous control with deep reinforcement learning. In: ICLR. 2016.

[57] Fujimoto S, van Hoof H, Meger D. Addressing function approximation error in actor-critic methods. In: ICML. 2018, p. 1587–96.

[58] Haarnoja T, Tang H, Abbeel P, Levine S. Reinforcement learning with deep energy-based policies. In: ICML. 2017, p. 1352–61.

[59] Studer S, Bui TB, Drescher C, Hanuschkin A, Winkler L, Peters S, Müller KR. Towards CRISP-ML(Q): a machine learning process model with quality assurance methodology. Mach Learn Knowl Extr 2021;3(2):392–413. http://dx.doi.org/10.3390/make3020020.

[60] Hansen N, Ostermeier A. Completely derandomized self-adaptation in evolution strategies. Evol Comput 2001;9(2):159–95. http://dx.doi.org/10.1162/106365601750190398.

[61] Hansen N, Auger A, Ros R, Finck S, Pošík P. Comparing results of 31 algorithms from the black-box optimization benchmarking BBOB-2009. In: GECCO. 2010, p. 1689. http://dx.doi.org/10.1145/1830761.1830790.