2024 European Control Conference (ECC)
June 25-28, 2024. Stockholm, Sweden

# Energy-Aware Speed Regulation in Electrical Drives: A Load-Agnostic Motor Control Approach via Reinforcement Learning

Steven Klotz[1], Thorsten Bucksch[1], Dip Goswami[2] and Daniel Mueller-Gritschneder[3]

*Abstract*— **Robotic and automotive platforms are rapidly expanding in features and are incorporating more and more electric motor components. Consequently, the energy efficiency of motor control systems emerges as a major design challenge. The process of formulating and fine-tuning specialized speed regulation strategies for each application becomes progressively more laborious and expensive. A reinforcement learning agent specialized in electrical motor dynamics, capable of generalizing across a wide range of possible end-use applications, presents a promising and convenient solution.**

**In this article, we introduce a novel design of a reinforcement learning agent, grounded in time series analysis, intended for application-agnostic electric motor control that optimizes both speed regulation and energy efficiency. Trained on the motor's internal dynamics, the agent provides operating point-specific control inputs, eliminating the need for manual tuning and application system-identification. Compared to application tuned classical control methods, the agent exhibited on-par or improved speed regulation performance and demonstrated advanced capability to save energy, showcasing its potential for future applications.**

## I. Introduction

The continued increase in demand for energy poses a critical challenge to sustainable development. As such energy-efficient control strategies are of interest across multiple application domains. Especially in mobile and battery-operated applications, such as in the robotic and automotive sector, energy conservation plays a key role to extend the range or operation time. *Model-based* control approaches, relying on the mathematical understanding of the system plant model and control problem, offer strong design choices.

Model-predictive control, one of the most promising methodologies, calculates future system states and optimizes the control input with respect to a cost function [1]. The approach has been successfully applied to multiple domains like robotic tasks and electrical motor applications [2] and also showed capability to conserve energy [3]. Extended state observer and adaptive controller are other potent model-based approaches for motor control problems, offering significant resilience to external disturbances [4].

In practice, for electrical motor applications, the hardware configuration in terms of compute device, bridge driver and motor are often determined before the mechanical dynamics of the end-use application are fully understood or identified at all. Therefore, limiting the applicability of model-based closed-loop control approaches for fast prototyping and deployment. In this context, *model-free* control strategies are desirable to effortlessly and effectively shape the control quality and energy consumption of any system, without the requirement for manual tuning or expert domain knowledge. As such, developing a universal motor control policy, which is capable of generalizing to the most complex end-use dynamics while offering high performance, would simplify the optimisation and deployment process.

**Contributions:** In this work, we set out to derive a control policy suitable for brushless direct current (BLDC) and permanent magnet synchronous motor (PMSM) applications, by analysing patterns stemming from current and speed data of the underlying motor dynamics.

We introduce an end-application agnostic motor control policy focusing on energy-aware speed regulation that uses a deep learning-based reinforcement learning agent to overcome the presented challenges. Based on time series information the agent evaluates the recent control history and deduces the optimal control, input balancing speed control and energy demand. During training, the agent is exposed to different operating regimens, that mirror potential application scenarios. Through this exposure, it identifies the internal motor dynamics and autonomously learns a control policy that generalizes to possible end-use applications. To the best of our knowledge, this is the first reinforcement learning based approach that optimizes for speed regulation and energy consumption simultaneously in a comprehensive manner for motor applications.

We compare the presented control policy with classical model-free control schemes for two practical end-use applications – an automotive compressor system and a ratcheting application. The results show that significant energy savings can be obtained depending on the application – In the compressor application, the agent demonstrates improved speed control compared to the PI baseline, but only utilizes $46\%$ of the energy. In the ratcheting application, it shows enhanced disturbance rejection and an $8\%$ reduction in energy demand.

## II. Related Work

*Model-free* reinforcement learning builds on the idea of an agent learning optimal actions through interactions with the environment, optimizing a reward signal [5]. In the domain of continuous-time control, two algorithms have garnered significant attention due to their efficacy: the Deep Deterministic Policy Gradient (DDPG) [6] and the Soft Actor-Critic (SAC) [7] methodologies. Leveraging reinforcement learning to improve the control performance of motor control

[1] Automotive, Infineon Technologies AG, Munich, Germany (steven.klotz,thorsten.bucksch)@infineon.com
[2] Electronic Systems, TU Eindhoven, Eindhoven, Netherlands d.goswami@tue.nl
[3] Electronic Design Automation, TU Munich, Munich, Germany daniel.mueller@tum.de

tasks has largely focused on reducing tracking error, either by tuning traditional control parameters [8], augmenting the control inputs [9] or directly applying the agents calculated action [10]. Additional work [11] focused on enabling plant independent reinforcement learning to remove the requirement for individual retraining by meta-reinforcement learning, streamlining the deployment process for motors that share similar features.

## III. PMSM Motor Model

Permanent Magnet Synchronous Motors are commonly installed in industrial applications requiring high efficiency, power density and a long-life span. Operating on the principles of magnetic field interaction and alignment, an electrical motor converts electrical to mechanical energy. A regulated voltage applied to the stator windings, gives rise to electric current which, in turn, generates a magnetic field that interacts with the magnetic field emanated by the permanent magnets on the rotor. The consequential electromagnetic interaction generates a torque that propels the rotor to rotate synchronously with the stator's magnetic field.

### A. System model

In field-oriented control (FOC) the generated electromagnetic torque $T_e(t)$ in the dq-frame is described by the dq-currents $(i_d, i_q)$, inductances $(L_d, L_q)$, flux linkage $\psi_{\mathrm{pm}}$ and pole pairs $Z_p$:

$$T_e(t) = \frac{3}{2} Z_p [(L_d - L_q) i_d(t) i_q(t) + i_q(t)\psi_{\mathrm{pm}}]. \tag{1}$$

The *dq*-frame PMSM plant model is described by [12]:

$$\begin{bmatrix} \frac{di_d(t)}{dt} \\ \frac{di_q(t)}{dt} \end{bmatrix} = \begin{bmatrix} \frac{1}{L_d}(u_d(t) - R_s i_d(t) + wL_q i_q(t)) \\ \frac{1}{L_q}(u_q(t) - R_s i_q(t) - w(L_d i_d(t) + \psi_{\mathrm{pm}})) \end{bmatrix}, \tag{2}$$

with the additional control voltage input variables $(u_d, u_q)$ and the stator resistance $R_s$. The total energy conversion in a motor system, or the power conversion at a specific point in time, can be described by:

$$E_{\mathrm{elec}} = \int P_{\mathrm{elec}}\ dt = \int P_{\mathrm{mech}} + P_{\mathrm{copper}}\ dt, \tag{3}$$

whereby $E_{\mathrm{elec}}$ is the electrical source input energy in the dq-frame, with the power described as $P_{\mathrm{elec}} = \frac{3}{2}(u_d i_d + u_q i_q)$. $P_{\mathrm{mech}}$ represents the mechanical power converted in the rotor, and $P_{\mathrm{copper}}$ represents the electrical copper losses. The mechanical power converted by the motor $P_{\mathrm{motor}} = T_e \times w$ is given by the product of torque $T_e$ and speed $w$. Meanwhile, the copper loss in the dq-frame is calculated as $P_{\mathrm{copper}} = R_s(i_d^2 + i_q^2)$.

### B. Control problem

Accurately tracking a reference speed $w^*$ is the primary objective of the motor control loop with the additional objective for energy efficient execution. Commonly, the external mechanical load moment $T_{\mathrm{m}}$ acting on the motor depends on the orientation $\theta$ and angular velocity $w$ of the rotor, which results in the following control equation with inertia $J$:

$$T_e(t) - T_{\mathrm{m}}(\theta, w, t) = J\frac{dw}{dt}. \tag{4}$$

Consequently, a controller's task is to produce electrical torque $T_e$ by applying the optimal input voltages $u^*_{dq_t}$ subject to its state observation and objective prioritization. Designing the feedback function and parameters $\theta$ for the cascading speed control loop with reference speed error $w_{\mathrm{err}}$:

$$\begin{aligned} w_{\mathrm{err}}(t) &= w^*(t) - w(t) \\ i^*_{dq}(t) &= f(w_{\mathrm{err}}(t), \theta), \end{aligned} \tag{5}$$

as well as inner current control loop with current error $i^*_{dq_{\mathrm{err}}}$:

$$\begin{aligned} i^*_{dq_{\mathrm{err}}}(t) &= i^*_{dq}(t) - i_{dq}(t) \\ u_{dq}(t) &= f(i^*_{dq_{\mathrm{err}}}(t), \theta), \end{aligned} \tag{6}$$

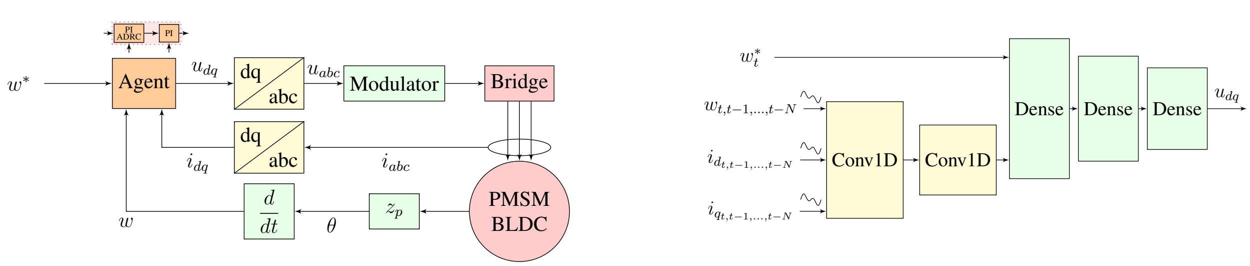
for individual end-use applications requires expert knowledge in motor and end-application dynamics.

## IV. Reinforcement Agent Control Loop

Rather than adhering to the conventional method of manually fine-tuning control parameters to design each application separately, we propose a reinforcement learning agent, which can deploy application agnostic, offering strong adaptability and optimized control performance. The goal of the soft actor-critic (SAC) agent is to learn an optimal control policy $\pi^*$, that analyses the recent time series of the speed and current trajectories and deduce an optimal control action $u^*_{dq_t}$ subject to the reference speed $w^*_t$:

$$u^*_{dq_t} = \pi^*(w^*_t, w_{t,t-1,\ldots,t-N}, i_{dq_{t,t-1,\ldots,t-N}}). \tag{7}$$

By training the SAC agent, we aim to train a deep actor neural network control policy $\pi_\theta$ that inherently captures




(a) Agent and classical control scheme for FOC control loop

(b) Control agent actor network design

Fig. 1: Reinforcement learning agent based control loop and feature design

the internal motor dynamics under varying external load conditions and speed commands. The optimal policy $\pi^*$ can be evaluated as the policy $\pi$ maximizing the cumulative reward function $J(\pi, \theta)$, driven by the policy $\pi$ and network parameters $\theta$, over the policy space $\Pi$.

$$\pi^* = \arg\max_{\pi \in \Pi} J(\pi, \theta) \tag{8}$$

With the implicit method of training an agent and embedding the motor dynamics in the network parameters $\theta$ any desirable control properties can be defined over the reward function $J$. In subfigure 1a, a restructured implementation of the field-oriented control loop is depicted, with the agent placed at the centre. The agent, supplied with sensor data, outputs the corresponding control vector $u_{dq}$. A description of the agent's design is provided in the following:

*1) Actor Network Design:* The actor network of the agent is engineered to understand the motor's current operating regime, contingent upon the present input attributes, as illustrated in Fig. 1b. In this configuration, the reference speed $w^*$ serves as the control tracking objective, and the dq-currents $(i_d, i_q)$ along with the speed attribute $w$ are represented as a time series with length $N$. Leveraging stacked convolutional filters (Conv1D), the agent proficiently extracts high-level features [13] from the control time series and employs the subsequent dense layer to deduce the optimal control input $u^*_{dq}$. In this work the agent is designed using *leaky-relu* for the activation function, with a stacked 5-3 Conv1D filter construct and three fully connected dense layers (15-10-4), which results in 1.1k parameters.

*2) Reward Function Design:* The reward function aims to achieve precise tracking of the target speed $w^*_t$, while simultaneously minimizing energy usage.. Consequently, the reward function incorporates two components, as formalized in Equation 9, each scaled by factors $\alpha$ and $\beta$:

$$r_t = \alpha r_{\Delta w_t} - \beta r_{E_t}. \tag{9}$$

The speed deviation reward $r_{\Delta w}$ at a given time $t$ is computed based on the discrepancy between the reference speed $w^*_t$ and the motor speed $w_t$ or awarded based on reference speed proximity as follows:

$$r_{\Delta w_t} = \begin{cases} 1 & |\dfrac{w^*_t - w_t}{w^*_t}| \leqq 0.1 \\ -(\dfrac{w^*_t - w_t}{w^*_t})^2 & \text{else} \end{cases} \tag{10}$$

To enhance energy efficiency, an $r_E$ penalty term is introduced that penalizes electrical input power, with an additional emphasis on copper losses, scaled by $\gamma$ and $\delta$:

$$r_{E_t} = \gamma P_{\text{elec}_t} + \delta P_{\text{copper}_t}. \tag{11}$$

## V. Training Loop

During training the agent interacts in a feedback loop, as in Fig. 2, with the motor environment collecting experiences $(s, a, s', r)$ from the motor dynamics using the state $s$, action $a$, next state $s'$ and reward $r$. In actor-critic reinforcement learning methodologies, learning a $Q$-value function is imperative to assign an expected return value to any state-action pair $(s, a)$ under a given policy $\pi$. Specifically, in the SAC framework, the $Q$-value function estimates the future return by integrating the immediate reward $R$ obtained at the ensuing time step, the discounted future $Q$-value $Q^\pi(s', a')$, and the entropy of the policy, thereby balancing exploitation and exploration. Inspired by the SAC algorithm [7], the $Q$-value function can be mathematically defined as:

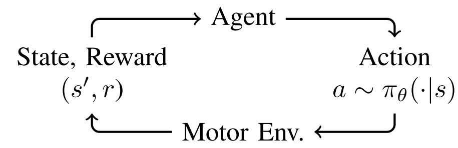


Fig. 2: Agent training loop

$$\begin{aligned} Q^\pi(s, a) = \mathbb{E}_{(s',a') \sim p(.|s,a)} \, [&R(s, a, s') \\ &+ \gamma \left( \mathbb{E}_{a' \sim \pi(.|s')} [Q^\pi(s', a')] - \alpha \log \pi(a'|s') \right) ] \end{aligned} \tag{12}$$

where $\gamma$ is the discount factor, and $\alpha$ modulates the trade-off between reward and entropy maximization. In the case of the motor agent, the state is composed of the observation $s = (w^*_t, w_{t,t-1,\ldots,t-N}, i_{dq_{t,t-1,\ldots,t-N}})$ and the $Q$-estimator calculates an expected return according to the defined reward function considering the speed and energy objective. The control policy network parameter $\theta$ are tuned to optimize for the highest return $J(\theta)$ within the estimator parameter space $\phi$. Mathematically this is laid out by considering the maximal expectation considering the $Q$-estimate and entropy.

$$\begin{aligned} \theta^* &= \arg\max_{\theta} J(\theta) \\ &= \mathbb{E}_{s \sim \rho_\pi, a \sim \pi_\theta} \left[ Q_\phi(s, a) - \alpha \log \pi_\theta(a|s) \right] \end{aligned} \tag{13}$$

Consequently, when exposed and trained on all operating conditions, the agent will adopt a control policy $\pi_\theta$ with parameters $\theta$ similar to the optimal policy $\pi^*$. The result is a control policy suitable for every end-use application acquired without manual tuning, substituting the feedback control equation and parameter definition in Equation 5 and 6 with $u_{dq}(t) = \pi_\theta(s)$. Algorithm 1 presents a high level overview of the training procedure.

**Algorithm 1** Motor agent training algorithm

```
Initialize actor and critic network parameters θ
Initialize memory replay, training policy and optimizer
for episode n to N do
    Generate representative periodical torque profile
    Set reference speed w* and starting operating point
    for step t to T do
        Observe state s and select action a ~ π_θ(·|s)
        Apply action a in the environment
        Record (s, a, s', r) and update memory replay
        SAC update [7] on actor and critic parameters θ
    end for
end for
```

## VI. Results and Evaluation

The proposed agent is analysed in a simulation for training stability and convergence but also benchmarked with classical model-free control strategies, namely, PI and Active Disturbance Rejection Control (ADRC), focusing on speed regulation and energy demand. The classical controllers are calibrated to balance speed regulation along with energy demand. To infuse realism, sensor and control compute time delay is introduced, wherein the outer speed control loop and the agent control frequency are configured at $f_{\text{control}} = 4$ kHz, with the inner current control loop set to $f_{\text{current}} = 5$ kHz, and the speed and current sensor update frequency fixed at $f_{\text{sense}} = 20$ kHz.

### A. Agent Training Results

Figure 3 illustrates the training results of the control agent, captured over several experimental runs. The average reward over the training process, for both the training policy and *greedy* evaluation policy, including the standard variation, are displayed. In all evaluated training procedures, the agent successfully learned a stable control policy and converged to the same average reward after approximately 200k training steps with no improvement thereafter.

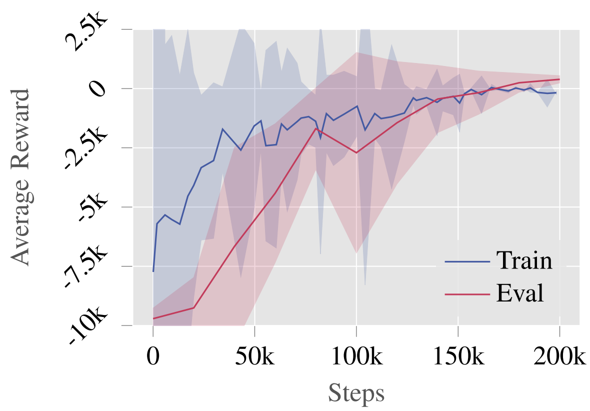



Fig. 3: Training trajectories of motor control agent

### B. Control Performance

For the control performance we investigate the effect of two highly dynamic end-use applications, whereby the external load moment depends on the rotor angle $\theta$ as shown in Fig. 4. The blue (—) profile depicts a sinusoidal load, similar to a compressor application, where the external torque causes periodic acceleration or deceleration of the rotor's rotation. A torque profile similar to a ratcheting system in red (—) marks the second evaluation scenario, in which the motor is subjected to torque step functions that elevate the force in increments, followed by an immediate release of the load torque upon completion of the rotation. The evaluation applications are chosen to represent distinct characteristics, whereby the compressor application exemplifies rapid yet smooth and continuous load changes. On the other hand, the ratcheting application illustrates abrupt load jumps followed

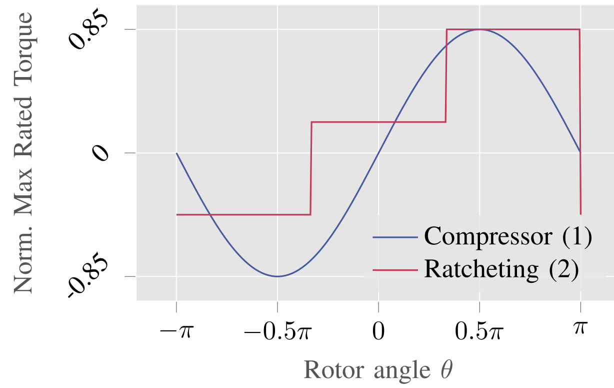
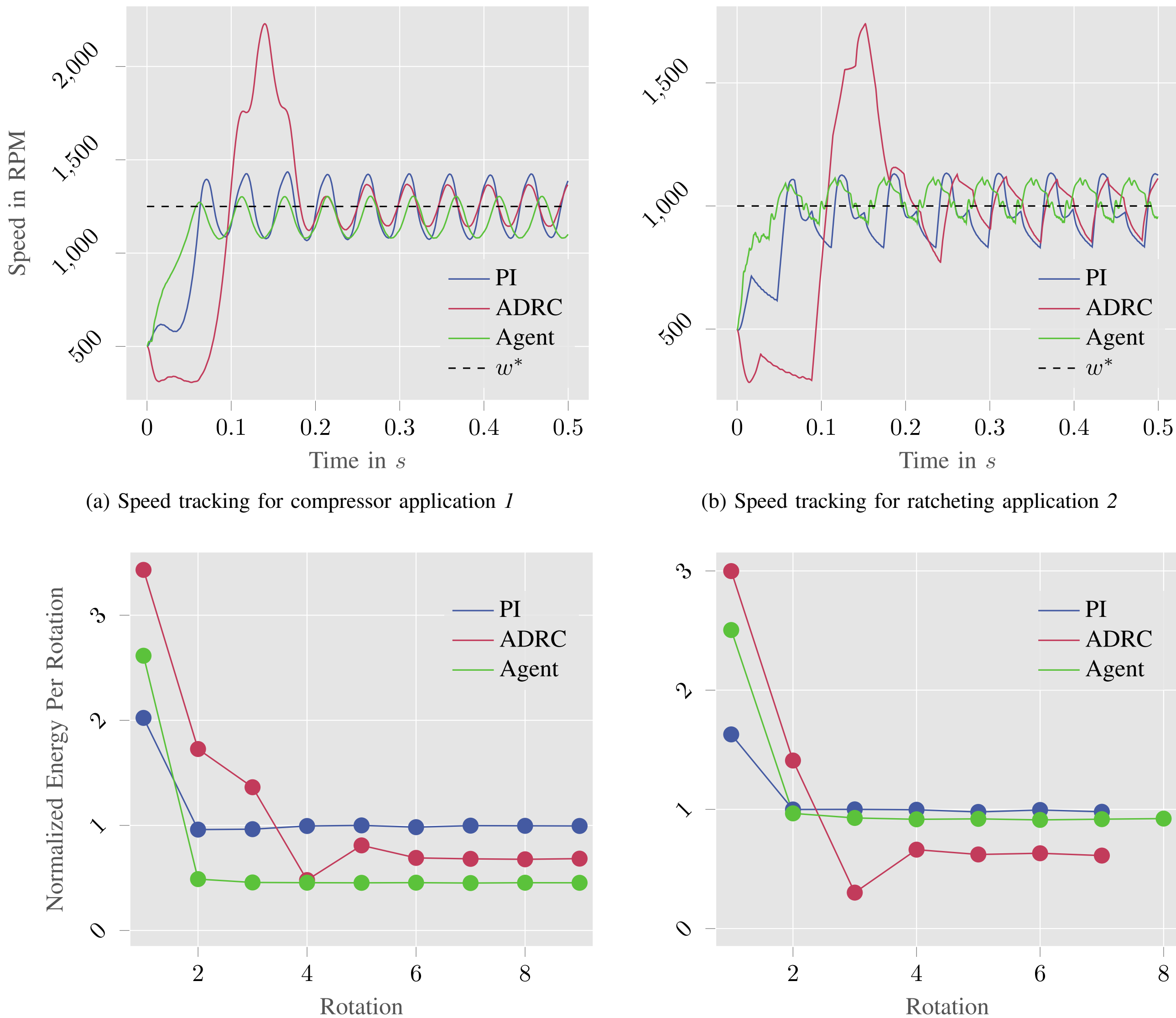



Fig. 4: Applications with periodic load disturbance profile

by a consistent load. Figure 5 with its subfigures displays results of the control setups' speed regulation and energy demand for the control strategies according to the two evaluated application scenarios. Table I lists the key values to evaluate the speed regulation performance in respect to settling time and disturbance rejection along with the energy demand per rotation.

*1) Compressor application:* All control approaches successfully operate around the reference speed, albeit with varying amplitude and different rise time as shown in subfigure 5a. In the initial stages, the agent exhibits more aggressive actions compared to the traditional controller reaching the desired reference speed earlier, a logical outcome stemming from the design of the squared speed penalty. Additionally, enabled by its filter-based design and the absence or requirement to accumulate any integral or conduct observations, the agent exhibits much faster reactions. For this reason, the agent achieves the fastest settling time of around 47 ms, compared with the 58 ms of the baseline PI and 192 ms for the ADRC. The ADRC controller, affected by the load torque on the motor, necessitates supplementary observations to determine the reference currents for acceleration, thereby extending the settling time. The extended observations combined with the rapid decrease in load leads to an overshoot of the target speed, significantly exceeding the reference speed. During the acceleration phase, the agent demonstrates the quickest response. After completion of the acceleration phase all controller exhibit the effects of the periodic disturbance torque. During the periodic disturbance rejection phase the agent prefers a slightly reduced effective speed, but rejects the disturbance on the same level as the ADRC. The remaining disturbance magnitude between the high and low extrema is identical with the ADRC control quality at 222, compared with 380 for the PI approach. Compared with the default PI strategy more steady speed regulation is achieved. In terms of speed regulation, the agent shows similar control performance, combining a fast settleing time and solid disturbance rejection.

Subfigure 5c shows the normalized energy demand per rotation, thereby putting the power demand of the control approaches in context with converted mechanical work. For




(a) Speed tracking for compressor application *1*



(b) Speed tracking for ratcheting application *2*



(c) Energy per rotation for compressor application *1*



(d) Energy per rotation for ratcheting application *2*

Fig. 5: Evaluation of motor control objectives in terms of speed and energy

the first rotations the control strategies are tasked to gain speed and momentum, which requires additional energy input compared to maintaining speed in the periodic disturbance phase. As the agent and PI method tend to converge to the reference speed fast and precisely they demand a moderate increase of electrical power input initially during the acceleration phase. Due to the substantial overshoot of the ADRC controller during the acceleration phase more energy is consumed in total for the first rotations. Upon reaching the target speed, each control approach actively counteracts the periodic disturbance torque. For the smooth and rapid torque changes of the compressor the agent requires significantly less energy to neutralize the disturbance compared to the baseline PI controller. While delivering comparable disturbance rejection to the ADRC, the agent requires only $46\%$ of the baseline PI energy demand per rotation, and still beats the energy requirement of the ADRC at $69\%$.

*Summary:* The agent demonstrated an efficient and fast acceleration phase with disturbance rejection control quality identical to the ADRC in the following. For the smooth torque profile of the compressor the agent identified control inputs, that promise significant energy savings in comparison with the classical controllers.

*2) Ratcheting application:* When evaluating the ratcheting mechanism for the disturbance scenario the agent prefers a much more interventionist control approach compared with the classical controller. As previously, the agent displays a fast acceleration process requiring 39 ms to settle in comparison with 54 ms for the PI and 244 ms for the ADRC. During the disturbance rejection phase, the remaining disturbance amplitude is significantly reduced, leading to muted high and low extremes. The PI and ADRC operate with a remaining disturbance amplitude of 295 and 252. In contrast, the agent reduces the disturbance magnitude to 157. Despite employing

TABLE I: Evaluation results contrasting speed control performance and energy demand

| | | Speed Control | | Energy per rotation | |
|---|---|---|---|---|---|
| | | Settling Time in ms | Rem.Dist. Magnitude | 1st Rotation | Avg. Rotation |
| Compressor Application | PI | 58 | 380 | 2.02 | 1.0 |
| | ADRC | 192 | 222 | 3.43 | 0.69 |
| | Agent | **47** | **222** | 2.61 | **0.46** |
| Ratcheting Application | PI | 54 | 295 | 1.64 | 1.0 |
| | ADRC | 244 | 252 | 3.01 | 0.63 |
| | Agent | **39** | **157** | 2.50 | 0.92 |

a more aggressive disturbance rejection strategy, the agent consumes $8\%$ less energy per rotation compared to the PI baseline set at 1. Actively countering the torque disturbance demands energy, which is why the ADRC consumes less energy for this application, requiring $37\%$ less than the PI controller.

*Summary:* Without prior application knowledge the agent is capable to act fast and precisely to reach steady state behaviour, similar to the PI approach, and displays the preferable disturbance rejection property of the ADRC. For this application, the agent prioritized speed regulation, significantly reducing the disturbance amplitude. Yet, it still outperformed the PI in terms of energy consumption, although it ranks behind the ADRC controller in energy demand.

## VII. Discussion and Future Work

In our experiments, particularly for applications that are smooth yet highly dynamic, the agent is able to achieve significant energy savings using time series information, even without prior knowledge of the application. Navigating the trade-off between speed control and energy consumption involves a complex interplay of various system variables. The design of the reward function serves as a proxy, markedly simplifying the process and alleviating the burden of design decision for applications. In the future we look to improve the feature representation of the agent and investigate the control decisions of the agent and explore the physical reasoning. Transitioning the focus to hardware execution: Keeping the parameter count and memory demand low is paramount to utilize the control approach in embedded systems. Investigating the effects of quantization on the control performance and reducing the memory footprint of the suggested method is a critical aspect. Additionally, overcoming the sim-to-real gap introduced by training in a synthetic environment will affect the control strategy learned by the agent. Hardware effects, such as sense and computation delay, but also a simulation-plant mismatch will affect the control performance of any control system, especially the one trained on data.

## VIII. Conclusion

The work presented a reinforcement learning control approach tailored to provide an end-application agnostic method to allow for energy-efficient speed regulation for electrical motors. Based on a filter design that analyses the current and speed data time series of a PMSM motor, the agent is trained on different operating regimes, covering a wide array of possible applications. The learned control policy blends a fast and efficient acceleration phase with advanced disturbance rejection control at low energy cost. In comparison with manually tuned classical controller the agent demonstrates advanced capabilities to conserve energy, while delivering comparable speed regulation. Because of the end-application agnostic training procedure the agent learns the internal dynamics of the motor and can therefore deliver operating point specific control inputs without the need for manual parameter tuning. As such it offers universal deployment when limited knowledge of the end-application dynamics is present, while maintaining a high level of speed regulation performance and energy efficiency.


## Acknowledgment

This work is part of the Penta project ”ECOMAI” (project number 2021028) and is funded by the German Federal Ministry of Education and Research (BMBF) within the funding program ”Penta Call 6”.


## Reproducibility

Where possible, project resources are available at: github.com/eureka-ecomai/ai-motor-control.

## References


[1] Rawlings, J., Mayne, D. & Diehl, M. Model predictive control: theory, computation, and design. (Nob Hill Publishing,2017).

[2] Abu-Ali, M., Berkel, F., Manderla, M., Reimann, S., Kennel, R. & Abdelrahem, M. Deep Learning-Based Long-Horizon MPC: Robust, High Performing, and Computationally Efficient Control for PMSM Drives. *IEEE Transactions On Power Electronics*. **37**, 12486-12501 (2022,10).

[3] Kim, S. & Kim, K. Model Predictive Control for Energy-Efficient Yaw-Stabilizing Torque Vectoring in Electric Vehicles With Four In-Wheel Motors. *IEEE Access*. **11** pp. 37665-37680 (2023).

[4] Shihua Li & Zhigang Liu Adaptive Speed Control for Permanent-Magnet Synchronous Motor System With Variations of Load Inertia. *IEEE Transactions On Industrial Electronics*. **56**, 3050-3059 (2009,8).

[5] Sutton, R. & Barto, A. Reinforcement learning: an introduction. (The MIT Press,2020)

[6] Lillicrap, T., Hunt, J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D. & Wierstra, D. Continuous control with deep reinforcement learning. (arXiv,2019,7).

[7] Haarnoja, T., Zhou, A., Abbeel, P. & Levine, S. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. (arXiv,2018,8).

[8] Lu, P., Huang, W. & Xiao, J. Speed Tracking of Brushless DC Motor Based on Deep Reinforcement Learning and PID. *2021 7th International Conference On Condition Monitoring Of Machinery In Non-Stationary Operations (CMMNO)*. pp. 130-134 (2021,6,11).

[9] Wang, Y., Fang, S., Hu, J. & Huang, D. A Novel Active Disturbance Rejection Control of PMSM Based on Deep Reinforcement Learning for More Electric Aircraft. *IEEE Transactions On Energy Conversion*. **38**, 1461-1470 (2023,6).

[10] Peng, W., Lan, Y., Chen, S., Lin, F., Chang, R. & Ho, J. Reinforcement Learning Control for Six-Phase Permanent Magnet Synchronous Motor Position Servo Drive. *2020 3rd IEEE International Conference On Knowledge Innovation And Invention (ICKII)*. pp. 332-335 (2020,8,21).

[11] Jakobeit, D., Schenke, M. & Wallscheid, O. Meta-Reinforcement-Learning-Based Current Control of Permanent Magnet Synchronous Motor Drives for a Wide Range of Power Classes. *IEEE Transactions On Power Electronics*. **38**, 8062-8074 (2023,7).

[12] Quang, N. & Dittrich, J. Vector Control of Three-Phase AC Machines: System Development in the Practice.

[13] Wang, Z., Yan, W. & Oates, T. Time Series Classification from Scratch with Deep Neural Networks: A Strong Baseline. (2016)