Max van Meer, Automated Model-Free Commutation for Coarse Pointing Actuators in Free-Space Optical Communication, Accepted for *2024 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, Boston, MA, 2024, uploaded to Zenodo May 1, 2024

Automated Model-Free Commutation for Coarse Pointing Actuators in Free-Space Optical Communication

Max van Meer[1], Kjell van Schie[1], Gert Witvoet[1,2], Tom Oomen[1,3]

***Abstract*— Switched Reluctance Motors (SRMs) are widely used for their simplicity and cost-effectiveness, for example, in coarse laser pointing for free-space optical (FSO) communication, with torque ripple being a challenge in their implementation. This paper introduces an automated, model-free approach to minimize torque ripple in SRMs despite manufacturing variations. Using an extremum-seeking approach, the commutation parameters are adapted online to mitigate the position-dependent velocity ripple through gradient descent. Simulations show that the method's effectiveness is consistent across different SRMs, rendering it applicable to mass production, and experimental results verify that torque ripple is almost entirely eliminated in several hours. The developed approach enables accurate SRM control with minimal expert knowledge, functioning as a crucial step toward their deployment in constellation projects for FSO communication.**

I. Introduction

Due to their mechanical simplicity and cost-effective production, Switched Reluctance Motors (SRMs) are favored as actuators in various applications, including coarse laser pointing for free-space optical (FSO) communication, encountering torque ripple as a significant challenge upon implementation, see Figure 1. SRMs require an electronic commutation scheme for current distribution [1], [2], [3], as depicted schematically in Figure 2.

Torque ripple, a position-dependent fluctuation of the rotor torque, is primarily caused by a mismatch between the modeled Torque-Current-Angle (TCA) relationship and the actual system, resulting in imperfect commutation. The TCA relationship of an SRM can be measured using a dedicated set-up [4] or identified from position sensor data and currents [5]. The former requires expensive equipment, and while the latter overcomes this problem, it still requires expert knowledge for successful deployment.

Indeed, manual calibration of each SRM's commutation scheme is cost-prohibitive in mass production, e.g., for constellation projects requiring thousands of FSO communication terminals. Therefore, in [6], an approach is developed to design a single, robust commutation function capable of controlling a variety of SRMs with slight differences. Considering the variance of the modeled TCA over different SRMs or rotor teeth, the average torque ripple over all SRMs or teeth is reduced. Still, as individual manufacturing defects are left unmodeled, some torque ripple inevitably remains with such a universal commutation scheme. Ideally, each SRM would ship with a unique commutation scheme compensating for individual manufacturing defects.

Simultaneously, extremum-seeking control (ESC) has gained increased attention for its ability to optimize controller parameters online without relying on an accurate system dynamics model [7]. This online scheme adjusts parameters during operation to minimize a cost function [8]. Both local variants of ESC, incorporating arbitrary continuous optimization methods with a derivative estimator [9], and global variants [10], are well-documented.

The identification approach presented in [5] yields accurate SRM models without extra sensors. Still, it requires expert knowledge for successful deployment, and thus, it does not necessarily scale to a mass-production setting. On the other hand, [6] aims to tackle this problem of scale by clever design of a universal commutation function that achieves low average torque ripple on a range of SRMs, but individual manufacturing variations are uncompensated.

Although recent advances have improved SRM's applicability to mass production, machine-specific tuning remains sub-optimal. This paper aims to develop a fully automated and model-free method for commutation design using extremum-seeking control. The developed algorithm scales universally across SRMs, adapting the commutation function to specific manufacturing defects using data. In short, the contributions of this paper are:

C1: An automated, model-free approach to commutation design is presented.
C2: Simulations validate the effectiveness of the method.
C3: Experimental results show that the approach significantly reduces torque ripple on a real setup.

This paper is structured as follows. Section II describes the problem, followed by the approach in Section III and implementation in Section IV. Subsequently, Sections V and VI provide simulation and experimental results, respectively. Finally, Section VII concludes the paper.

II. Problem formulation

We formalize the problem of online commutation design for Switched Reluctance Motors. First, we describe SRM dynamics. Next, the control of SRMs through commutation functions is addressed. Subsequently, the need for online

[1]Max van Meer (e-mail: m.v.meer@tue.nl), Kjell van Schie, Gert Witvoet, and Tom Oomen are with the Control Systems Technology section, Department of Mechanical Engineering, Eindhoven University of Technology, The Netherlands. This work is part of the research program VIDI with project number 15698, which is (partly) financed by the Netherlands Organisation for Scientific Research (NWO). In addition, this research has received funding from the ECSEL Joint Undertaking under grant agreement 101007311 (IMOCO4.E). The Joint Undertaking receives support from the European Union's Horizon 2020 research and innovation program.

[2]Gert Witvoet is also with the Department of Optomechatronics, TNO, Delft, The Netherlands.

[3]Tom Oomen is also with the Delft Center for Systems and Control, Delft University of Technology, Delft, The Netherlands.

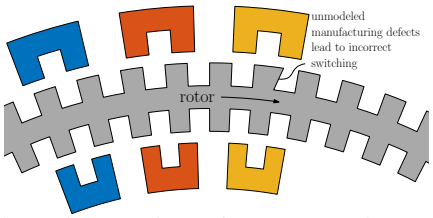


Fig. 1: Schematic overview of an SRM, where magnetized stators attract rotor teeth to generate torque. Incorrect current switching leads to torque ripple.

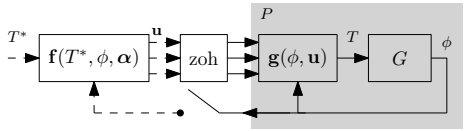


Fig. 2: Block scheme of commutation for an SRM. When commutation function $\mathbf{f}$ is designed such that $\mathbf{gf} \approx 1$, the system is linearized, achieving $T \approx T^*$.

optimization of commutation functions is addressed, and finally, the problem is formalized.

A. Switched Reluctance Motor dynamics

Switched Reluctance Motors (SRMs) exhibit a nonlinear relationship between torque, current, and rotor angle; see Figure 1. Neglecting magnetic saturation, an SRM m with n_t teeth and n_c produces a torque [2]

$$T(t) = \mathbf{g}_m(\phi(t))\mathbf{u}(t), \tag{1}$$

at time $t \in \mathbb{R}$, where $\mathbf{g}_m(\phi) : \mathbb{R} \to \mathbb{R}^{1\times n_c}$ is determined by the phase inductance, which varies periodically with the mechanical rotor angle ϕ, having a spatial period of $\frac{2\pi}{n_t}$. The torque and inductances are unmeasured. Moreover, $\mathbf{u}(t) = [i_1^2(t), \dots, i_{n_c}^2(t)]^\top$ is a vector of squared coil currents. The following section explains how a desired torque T^* is realized, using a model $\hat{\mathbf{g}}_m \approx \mathbf{g}_m$.

B. Controlling SRMs via commutation functions

To achieve a desired torque T^* in SRMs, we invert the nonlinear TCA relationship (1) with a commutation function $\mathbf{u} = \bar{\mathbf{f}}_m(\phi, T^*)$ as follows. First, $\bar{\mathbf{f}}_m$ is structured as

$$\bar{\mathbf{f}}_m(\phi, T^*) := \begin{cases} \mathbf{f}_m^+(\phi)T^* & T^* \geq 0, \\ -\mathbf{f}_m^-(\phi)T^* & T^* < 0, \end{cases} \tag{2}$$

where $\mathbf{f}_m^+(\phi), \mathbf{f}_m^-(\phi) : \mathbb{R} \to \mathbb{R}^{n_c}$ are functions that are designed to satisfy

$$\hat{\mathbf{g}}_m(\phi)\mathbf{f}_m^+(\phi) \approx 1, \qquad \hat{\mathbf{g}}_m(\phi)\mathbf{f}_m^-(\phi) \approx -1, \tag{3}$$

such that when $\mathbf{f}_m^+$ or $\mathbf{f}_m^-$ is multiplied with a desired torque T^*, the resulting currents lead to $T \approx T^*$, see (1). For ease of explanation, but without loss of generality, this paper restricts the focus to commutation functions in the forward direction, i.e., $\mathbf{f}_m(\phi) = \mathbf{f}_m^+(\phi)$. Moreover, since $\mathbf{f}$ produces squared currents, it must be ensured that

$$\mathbf{f}_m(\phi) \geq \mathbf{0}. \tag{4}$$

Multiple functions $\mathbf{f}_m(\phi)$ satisfy these requirements due to the vector nature of $\mathbf{g}_m$ and $\mathbf{f}_m$. By applying the control law

$$\mathbf{u}(t) = \mathbf{f}_m(\phi(t))T^*(t), \tag{5}$$

the squared currents for each coil to achieve the desired torque T^* are determined, as visualized in Figure 2. Combined with (1), the resulting torque is:

$$\begin{aligned} T(\phi(t)) &= b_m(\phi(t))T^*(t), \\ \text{with } b_m(\phi(t)) &= \mathbf{g}_m(\phi(t))\mathbf{f}_m(\phi(t)). \end{aligned} \tag{6}$$

Here, $b(\phi)$ denotes the torque ratio T/T^*. Ideally, if the model is perfect, i.e., $\hat{\mathbf{g}}_m(\phi) = \mathbf{g}_m(\phi)$, and $\mathbf{f}_m$ meets the requirements in (3), the realized torque matches the desired torque for all rotor positions ϕ. However, if $\hat{\mathbf{g}}_m(\phi) \neq \mathbf{g}_m(\phi)$, torque ripple occurs, degrading the tracking performance [11]. The next section explains why this poses a challenge in FSO communication.

C. Commutation challenges in satellite communication

Future communication networks may involve thousands of satellites with optical communication terminals, each featuring a unique coarse-pointing actuator due to manufacturing variations. Consequently, each SRM m has a unique TCA relationship $\mathbf{g}_m$. When a single commutation function $\mathbf{f}_{\text{avg}}$ is designed based on some average model $\hat{\mathbf{g}}_{\text{avg}}$ such that $\hat{\mathbf{g}}_{\text{avg}}\mathbf{f}_{\text{avg}} = 1$, but all SRMs feature slightly different dynamics $\mathbf{g}_m$, torque ripple occurs, see (6). At the same time, in mass production, it is extremely costly to obtain good models $\hat{\mathbf{g}}_m \approx \mathbf{g}_m$ of all individual SRMs.

Moreover, even if individual models $\hat{\mathbf{g}}_m$ are available, the SRM dynamics $\mathbf{g}_m$ in orbit might differ significantly from its dynamics in the factory. The reason for this is twofold: (i) during launch, extreme forces cause slip in mechanical components, slightly altering the location of the coils, and (ii), ambient temperatures during operation vary from those in the factory, range from -30 °C to 60 °C, affecting magnetic properties and tooth geometry. The following section addresses these issues in the problem definition.

D. Problem definition

The problem considered in this paper is to develop a model-free, data-driven commutation design framework for a class $\mathcal{M}$ of SRMs, with a high degree of automation, such that it can be cost-effectively applied to individual SRMs $m \in \mathcal{M}$, potentially even allowing for easy recalibration in space. Specifically, the approach must yield a commutation function $\mathbf{f}_m(\phi)$ such that $\mathbf{g}_m\mathbf{f}_m(\phi) \approx 1$, without relying on an accurate model $\hat{\mathbf{g}}_m$. Instead, only measurements of the rotor positions ϕ and the applied squared currents $\mathbf{u}$ are available. The following section explains how this is achieved using extremum-seeking control.

III. Automated Model-Free Commutation

This section addresses the automated design of commutation functions, starting with an overview of the method.

A. Extremum-seeking control scheme

Extremum-seeking control is a data-driven approach to the online optimization of controller parameters; see Figure 3. This iterative method relies on closed-loop experiments with parameters $\boldsymbol{\theta}_k \in \mathbb{R}^n$ to obtain data $\phi \in \mathbb{R}^N$. A cost function $\mathcal{J}(\boldsymbol{\phi})$ is evaluated to assess the quality of parameters $\boldsymbol{\theta}_k$.

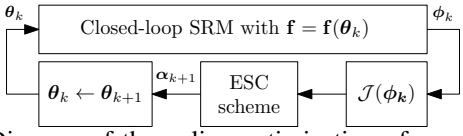


Fig. 3: Diagram of the online optimization of commutation function parameters $\boldsymbol{\theta}$ to minimize the cost $\mathcal{J}$.

Subsequently, the extremum-seeking controller decides on the next parameters $\boldsymbol{\theta}_{k+1}$ to apply experimentally. The choice of these parameters is based on either an estimation of the cost function gradient or an update towards a lower cost function. We emphasize that this approach is online, i.e., the experiments immediately follow each other without offline processing. Next, we detail the cost function design.

B. Design of the cost function

The cost function $\mathcal{J}(\boldsymbol{\phi}(\boldsymbol{\theta}))$ must reflect the desire to eliminate torque ripple, see (6); however, this torque cannot be measured. Instead, we make the following assumption to derive the degree of torque ripple from measurements of $\boldsymbol{\phi}$.

Assumption 1. *The mechanical system $G(s) := \phi(s)/T(s)$ satisfies $\lim_{s\to 0} sG(s) = K$, with $K \in \mathbb{R}$.*

Under this assumption of damping, which is satisfied for many systems that experience viscous friction, a constant velocity $\omega = \mathrm{d}/\mathrm{d}t\ \phi(t)$ in Figure 2 implies a constant torque $T(t)$. When $T^*(t)$ is constant over time, then ripples in the velocity are caused by either (i) external torque disturbances or (ii) imperfect commutation. The following section explains how these two sources of velocity ripples are distinguished.

1) Averaging over rotor teeth: A key requirement in extremum-seeking control is that the cost $\mathcal{J}(\boldsymbol{\phi}(\boldsymbol{\theta}))$ is a function of the controller parameters $\boldsymbol{\theta}$, i.e., the same $\boldsymbol{\theta}$ always results in the same cost. External disturbances that affect ϕ challenge this requirement since these cause identical experiments to lead to different position signals $\boldsymbol{\phi}$. To mitigate the effect of these disturbances on the cost function, we rely on the following assumption.

Assumption 2. *External disturbances that affect $\phi(t)$ are not spatially periodic with the period $\frac{2\pi}{n_t}$ of a rotor tooth.*

Hence, there is a distinction between commutation faults, which are spatially periodic with $\frac{2\pi}{n_t}$, and external disturbances. The effect of external disturbances is mitigated by averaging the rotor velocity over the rotor teeth, as follows.

First, in the discrete-time domain, the rotor velocity is estimated from the rotor position as

$$\omega(\tau) = \frac{\alpha q - \alpha}{q - e^{\alpha}}\phi(\tau), \tag{7}$$

where q is the forward-shift operator such that $q\phi(\tau) = \phi(\tau + 1)$, $\tau \in \mathbb{N}$ is the sample number and $\alpha \in \mathbb{R}^+$ is the cutoff frequency of the lowpass filter. Next, to average out these rotor velocities over the teeth, a tooth of width $\frac{2\pi}{n_t}$ is divided into n_b bins. The velocity samples $\omega(\tau)$ are grouped in these bins B_i as sets

$$B_i = \left\{ \omega(\tau) \mid (i-1)\frac{2\pi}{n_t n_b} \le \tilde{\phi}(\tau) < i\frac{2\pi}{n_t n_b} \right\}, \tag{8}$$

where $\tilde{\phi} := \mathrm{mod}\,(\phi, 2\pi/n_t)$. The average for each bin is

$$\overline{\omega}_i = \frac{1}{|B_i|}\sum_{\omega \in B_i} \omega, \tag{9}$$

where $|B_i|$ denotes the number of samples $\omega(\tau)$ in set B_i. This grid of averaged velocities enables a cost function that penalizes the commutation-induced velocity ripple.

2) Definition of the cost function: The cost function must reflect the desire to minimize the velocity ripple. At the same time, the commutation parameters also affect the magnitude of the velocity, given a constant T^*. Although the velocity magnitude is secondary, it must be considered in the cost function for two reasons. First, at high velocities, the velocity ripple naturally decreases because of the rotor inertia, and it becomes harder to distinguish this ripple in the data. Second, friction may halt the rotor at low velocities. Therefore, the cost function consists of (i), a soft constraint to penalize deviations from some desired velocity ω_d, and (ii), the root-mean-squared deviation of the velocity, relating to the velocity ripple. The former is defined by

$$\mathcal{J}_\omega(\boldsymbol{\theta}) = (\overline{\omega} - \omega_d)^2, \tag{10}$$

and the velocity ripple is penalized using

$$\mathcal{J}_r(\boldsymbol{\theta}) = \sqrt{\frac{1}{n_b}\sum_{i=1}^{n_b}(\overline{\omega}_i - \tilde{\omega})^2}, \tag{11}$$

where $\tilde{\omega}$ denotes the mean

$$\tilde{\omega} = \frac{1}{n_b}\sum_{i=1}^{n_b}\overline{\omega}_i, \tag{12}$$

such that the total cost function is

$$\mathcal{J}(\boldsymbol{\theta}) = \mathcal{J}_r(\boldsymbol{\theta}) + \beta\mathcal{J}_\omega(\boldsymbol{\theta}). \tag{13}$$

Here, $\beta > 0$ is a parameter to be tuned; see Section IV-B.

C. Parametrization of the commutation functions

$\mathbf{f}(\phi, \boldsymbol{\theta})$ should feature a minimal number of parameters for fast convergence. Moreover, requirements (4) and (3) must be satisfied for any $\boldsymbol{\theta}$. To satisfy these criteria, we parametrize $\mathbf{f}$ using torque sharing functions [1] as follows.

Consider the requirement $\mathbf{g}(\phi)\mathbf{f}(\phi) \approx 1$. Even though $\mathbf{g}(\phi)$ is unknown, it is known to be approximately sinusoidal with period $\frac{2\pi}{n_t}$. Therefore, we parametrize $\mathbf{g}$ in terms of $\boldsymbol{\theta}$ with a Fourier basis, and subsequently parametrize $\mathbf{f}$ in terms of $\mathbf{g}(\phi, \boldsymbol{\theta})$. First, parametrize the elements of $\hat{\mathbf{g}}$ as

$$\begin{aligned}\hat{g}_c(\phi, \boldsymbol{\theta}) &= \theta_{s+1}\sin\left(\phi - \theta_{s+2}\right), \\ &\quad + \sum_{i=2}^{n_h}\theta_{s+2i-1}\sin\left(i\left(\phi - \theta_{s+2}\right) - \theta_{s+2i}\right), \\ \text{with } s &:= 2n_h(c-1), \qquad c \in \{1, \ldots, n_c\},\end{aligned} \tag{14}$$

such that each even element of $\boldsymbol{\theta}$ relates to the amplitude of a sinusoid, and each odd element corresponds to its phase shift. The phase θ_{s+2} of the first harmonic is included in the sum to ensure that the phases of higher harmonics align with those of the first when the corresponding θ_{s+2i} is zero.

Subsequently, $\mathbf{f}$ is parametrized as

$$\begin{aligned} f_c\left(\phi, T^*, \boldsymbol{\theta}\right) =& \mathbf{f}_{\mathrm{TSF},c}\left(\phi - \theta_{s+2}, T^*\right) \\ & \cdot \operatorname{sat}\left(1/\hat{g}_c\left(\phi, \boldsymbol{\theta}\right), \overline{g}\right) T^*, \end{aligned} \tag{15}$$

where f_c denotes the c^{th} element of $\mathbf{f}$ and $\operatorname{sat}(x, x_{\max})$ saturates x to a maximum $x_{\max}$ as

$$\operatorname{sat}\left(x, x_{\max}\right) \triangleq \operatorname{sgn}(x) \min\left\{|x|, x_{\max}\right\}, \tag{16}$$

and $\overline{g}$ is a positive constant. In addition, the function $\mathbf{f}_{\mathrm{TSF}}(\phi, T^*) : \mathbb{R} \times \mathbb{R} \to \mathbb{R}^{n_c}$ is defined to allocate a torque across different coils. It adheres to the following conditions:

$$\sum_{c=1}^{n_c} \mathbf{f}_{\mathrm{TSF},c}(\phi, T^*) = \begin{cases} 1 & \text{if } T^* \geq 0, \\ -1 & \text{if } T^* < 0, \end{cases} \tag{17}$$

see [1] for details. It is important to note that when ϕ is such that $g_c(\phi) = 0$, the function $\mathbf{f}_{\mathrm{TSF},c}(\phi, T^*)$ is zero by design. This ensures that (15) is well-defined for all values of ϕ.

To summarize, (15) parametrizes the commutation function in terms of $\boldsymbol{\theta}$, indirectly through $\hat{\mathbf{g}}(\phi, \boldsymbol{\theta})$. This parametrization ensures that (4) is satisfied for any $\boldsymbol{\theta}$. In addition, requirement (3) is satisfied if the number of Fourier coefficients n_h is sufficiently large, and the parameters $\boldsymbol{\theta}$ are in accordance with the true dynamics $\mathbf{g}_m$.

D. Parameter update

We update the parameters $\boldsymbol{\theta}$ to minimize $\mathcal{J}(\boldsymbol{\omega}(\boldsymbol{\theta}))$ according to the scheme in Figure 3 as follows. We first estimate the gradient and then update parameters towards the negative gradient for convergence to a local minimum.

1) Gradient estimation: To estimate the gradient of the cost function, we perturb each parameter $\theta_{k,j}$ in separate experiments j, in immediate succession. Given a parameter vector $\boldsymbol{\theta}_k$, $\boldsymbol{\delta} \in \mathbb{R}^{n_\theta} > 0$ defines a perturbation vector that is small with respect to $\boldsymbol{\theta}$. Next, n_θ pairs of experiments j are conducted using $\boldsymbol{\theta}_k = \boldsymbol{\theta}_{k,j}^+$ and $\boldsymbol{\theta} = \boldsymbol{\theta}_{k,j}^-$, with

$$\boldsymbol{\theta}_{k,j}^+ = \boldsymbol{\theta}_{k,j} + \mathbf{A}_{jj}\boldsymbol{\delta}, \quad \boldsymbol{\theta}_{k,j}^- = \boldsymbol{\theta}_{k,j} - \mathbf{A}_{jj}\boldsymbol{\delta}, \tag{18}$$

where $\mathbf{A}_{jj}$ is a unit matrix that has only zero entries except for the j^{th} element on the diagonal, which is unity. Subsequently, the gradient is estimated as

$$\begin{aligned} \nabla\mathcal{J}(\boldsymbol{\theta}_k) &:= \left[\frac{\partial\mathcal{J}(\boldsymbol{\theta}_k)}{\partial\theta_{k,1}}, \ldots, \frac{\partial\mathcal{J}(\boldsymbol{\theta}_k)}{\partial\theta_{k,n_\theta}}\right], \\ \frac{\partial\mathcal{J}(\boldsymbol{\theta}_k)}{\partial\theta_{k,j}} &\approx \frac{\mathcal{J}(\boldsymbol{\theta}_{k,j}^+) - \mathcal{J}(\boldsymbol{\theta}_{k,j}^-)}{2\delta_j}, \end{aligned} \tag{19}$$

where $\theta_{k,j}$ denotes the j^{th} element of vector $\boldsymbol{\theta}_{k,j}$.

2) Updating towards the minimum: With the gradient estimate (19) computed, the commutation function parameters are updated towards the estimated minimum as

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \gamma\nabla\mathcal{J}(\boldsymbol{\theta}_k), \tag{20}$$

with $\gamma \in \mathbb{R}$ a step-size. The choice of this parameter is addressed in the next section. The developed approach is summarized in Algorithm 1.

IV. Implementation

This section describes the design considerations in choosing parameters of Algorithm 1, experimental considerations, and computational aspects.

A. Parameter tuning

First consider the extremum-seeking parameters γ and $\boldsymbol{\delta}$. The step size γ should be sufficiently large for fast convergence. However, when γ is chosen too large, the optimization scheme diverges; see [8] for details. On the other hand, the perturbation vector $\boldsymbol{\delta}$ for gradient estimation relates to the accuracy of the gradient estimate. Its value relates to the smoothness of the cost landscape: when $\boldsymbol{\delta}$ is too large compared to the smoothness of $\mathcal{J}(\boldsymbol{\theta})$, the gradient estimates are inaccurate. When $\boldsymbol{\delta}$ is chosen too small, the accuracy of the gradient estimate is limited by external disturbances that remain after averaging with (9).

Moreover, the number of bins n_b depends on the smoothness of the velocity ripple. When n_b is chosen too small, there may be insufficient data in each bin for adequate averaging. On the other hand, when it is too large, the averaged velocities $\overline{\boldsymbol{\omega}}_i$ may insufficiently reflect sharp transitions in the velocity ripple. We recommend choosing the experiment length N as small as possible for convergence speed but sufficiently large such that each bin B_i contains sufficient data to average external disturbances.

B. Experimental considerations

The most important experimental tuning parameters are the desired torque T^* and the desired velocity ω_d, which relate to each other. Indeed, the ratio between ω and T is given by the damping d of the sytem, and T relates to T^* through $T = \mathbf{g}(\phi)\mathbf{f}(\phi, \boldsymbol{\theta})T^*$. Therefore, $\hat{d} \approx d$ must be estimated, for example, using experimental data for different values of T^*. This estimate determines the desired torque $T^* = \hat{d}\omega_d$, such that ω_d can be chosen sufficiently large for the rotor not to be forced to a standstill by friction, and not so high that the rotor inertia obfuscates the velocity ripple from the data. Given ω_d, we recommend setting β as small as possible to primarily target the velocity ripple in (13). If the average velocity drifts too far from ω_d, β must be increased.

A second consideration is that when $\boldsymbol{\theta}_k$ is updated to $\boldsymbol{\theta}_{k+1}$, $\omega(t)$ experiences a transient, before returning to its steady-state velocity ripple which is periodic with the rotor teeth. This transient is unrelated to torque ripple, and thus, the first N_{tr} samples of $\omega(\tau)$ of each experiment are discarded. Moreover, the choice of cutoff frequency α must be sufficiently large not to smoothen out the commutation-induced torque ripple in (7) and small enough to mitigate amplified sensor noise from differentiation of the position measurement.

Finally, when a commutation function for backward movements is also desired, the experiments need only be extended to include a backward movement, i.e., flipping the sign of T^*, approximately doubling the experimental time. All other aspects of the approach remain unchanged, including the number of parameters $\boldsymbol{\theta}$.

C. Computational aspects

The following measures minimize the computational time between experiments to reduce the total experimental time of Algorithm 1.

Algorithm 1 Online optimization of commutation functions

Require: Step size γ, perturbation vector $\boldsymbol{\delta}$, initial $\boldsymbol{\theta}_0$.

Initialize $k \leftarrow 0$, start experiment with $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}_0$.
while $k < k_{\max}$ **do**
  for $j \in \{1, \dots, n_\theta\}$ **do**
    for $\iota \in \{+, -\}$ **do**
      Set $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}_j^\iota$, see (18).
      Wait N samples and compute $\mathcal{J}(\boldsymbol{\omega}(\boldsymbol{\theta}_j^\iota))$.
    end for
  end for
  Compute gradient $\nabla\mathcal{J}(\boldsymbol{\theta}_k)$ with (19).
  Compute $\boldsymbol{\theta}_{k+1}$ with (20).
  Set $k \leftarrow k + 1$.
end while

1) During the experiments, evaluate (7) every time step and immediately sort the samples into the bins in (8).
2) Similarly, update the summations in (9), (12) and (13) at every time step.

The next section demonstrates the effectiveness of the approach in a simulation.

V. Simulation Results

This section presents a simulation study to demonstrate that the developed approach effectively minimizes the torque ripple of SRMs, even without model knowledge.

A. Simulation setup

A series of $M = 100$ SRMs is considered, each featuring $n_t = 131$ identical teeth and $n_c = 3$ coils. The following equation describes the true TCA relationship of the SRMs:

$$\mathbf{g}_m^\top(\phi, \boldsymbol{\theta}_m) = \boldsymbol{\psi}_g(\phi)\boldsymbol{\theta}_m, \quad m \in \{1, \dots, M\}, \tag{21}$$

where the variation across the SRMs follows from

$$\boldsymbol{\theta}_m \sim \mathcal{N}(\boldsymbol{\theta}^\circ, \boldsymbol{\Sigma}_{\boldsymbol{\theta}}), \tag{22}$$

with $\boldsymbol{\theta} \in \mathbb{R}^{90}$, and the structure $\boldsymbol{\psi}_g$ comprises of radial basis functions, see Figure 4. This simulation study considers the true systems $\mathbf{g}_m$ to be unknown. Moreover, the linear dynamics are governed by

$$G(s) := \frac{\phi(s)}{T(s)} = \frac{273.97}{s(s + 8.9014)}, \tag{23}$$

and measurements of ϕ are corrupted by i.i.d. additive white noise with variance $7{\cdot}10^{-15}$ rad. Three approaches are compared. First, a single commutation function $\mathbf{f}_{\text{conv}}$ is designed using (15) based on the average model $\hat{\mathbf{g}}_{\text{avg}}(\phi, \boldsymbol{\theta}^\circ)$. Second, the method of [6] is applied to develop a single commutation function $\mathbf{f}_{\text{robust}}$ that exhibits robustness through gradual switching of currents. Finally, we follow the approach of this paper. A single low-order model $\hat{\mathbf{g}}_{\text{lo}}(\phi, \boldsymbol{\theta}_{\text{lo}}) \approx \hat{\mathbf{g}}(\phi, \boldsymbol{\theta}^\circ)$ with a Fourier structure and $n_h = 5$ harmonics is applied to all SRMs, but Algorithm 1 is applied to update the parameters $\boldsymbol{\theta}_{\text{lo}}$ online, such that each SRM ends with its own, unique commutation function $\mathbf{f}_m(\phi, \boldsymbol{\theta}_{\text{lo},m})$. The parameters of Algorithm 1 are chosen as $\gamma = 0.2$, $\omega_d = 0.3$ rad/s, $F_s = 1$ kHz and $\boldsymbol{\delta} = \mathbf{1}_{n_c n_h} \otimes [0.1, \pi/180]^\top$, with $\otimes$ the Kronecker product. The N_{tr} samples corresponding to the first five teeth of each iteration are discarded to account for transients. Finally, $n_b = 201$ bins are used to average the velocity across sixteen rotor teeth.

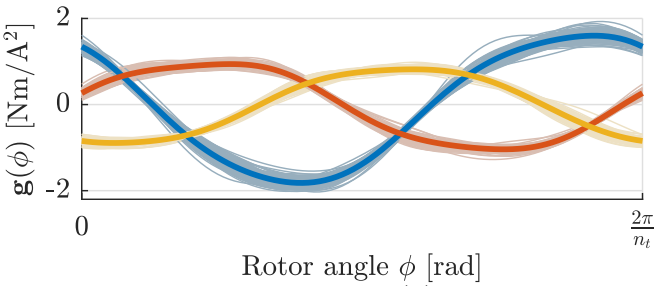


Fig. 4: The TCA relationships ($\mathbf{g}_m(\phi)$) for 100 simulated SRMs show slight variations. The average is highlighted as $\mathbf{g}_{\text{avg}}(\phi)$ in bold.

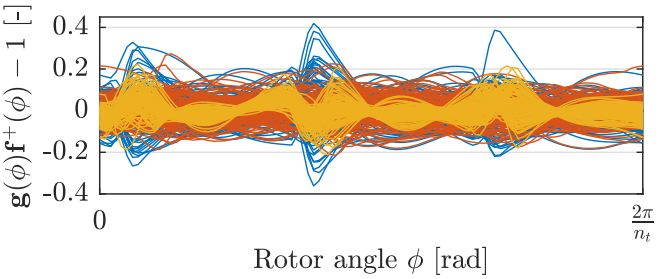


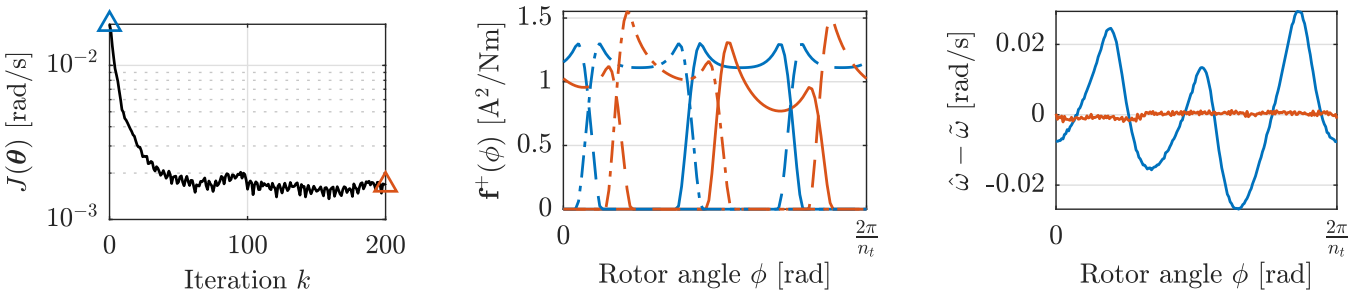
Fig. 5: Torque ripple in SRMs: Conventional $\mathbf{f}_{\text{conv}}$ (—) incurs significant ripple due to model mismatch. $\mathbf{f}_{\text{robust}}$ (—) reduces this, but not entirely. Our $\mathbf{f}_{\text{opt},m}$ (—) further minimizes torque ripple by tailoring to each SRM m online.

B. Results and analysis

Algorithm 1 significantly reduces torque ripple in 100 SRM simulations over $k_{\max} = 100$ iterations, see Figure 5. This reduction is primarily attributed to implementing custom-tailored commutation functions for each SRM, designed to adapt to their specific dynamics, an approach not feasible with traditional, labor-intensive methods. Each SRM simulation represents a total real-time span of three hours, yet accelerated computing allows faster completion.

The average root-mean-squared deviation b_{RMSD} of b_m in (6) is 0.073 when using using $\mathbf{f}_{\text{conv}}$. The robust commutation functions of [6] lead to $b_{\text{RMSD}} = 0.062$ (-14%) and Algorithm 1 further reduces this to 0.040 (-44%), showing a considerable reduction in torque ripple. Moreover, the robustness of the approach is confirmed: the same algorithm parameters, applied to 100 significantly different SRMs, lead to convergence of the cost function on each SRM without manual intervention across SRMs.

Figure 5 also suggests room for further improvement. The current limitation stems from $\mathbf{f}(\phi, \boldsymbol{\theta})$ being parameterized with only five harmonics, see (15). This means the model structure is not sufficiently rich to achieve $\mathbf{g}_m(\phi)\mathbf{f}(\phi, \boldsymbol{\theta}) = 1$ at angles ϕ where the current transitions between coils. Augmenting the number of harmonics would further improve the performance, albeit with an increase in convergence time.



(a) (b) (c)

Fig. 6: Experimental results. (a) Algorithm 1 reduces the cost significantly, (b) gradually updating the commutation function from $\mathbf{f}(\boldsymbol{\theta}_0)$ (△ , —) to $\mathbf{f}(\boldsymbol{\theta}_{200})$ (△ , —). Consequently, (c) the average velocity ripple per tooth is removed almost entirely.

VI. Experimental results

This section shows that the developed method for automated commutation design effectively eliminates torque ripple in an experimental study.

A. Experimental setup

The experiment uses a single-axis coarse-pointing assembly from TNO [4], equipped with an SRM with $n_t = 131$ teeth and $n_c = 3$ coils. The initial commutation scheme is defined by (15), where each $\hat{g}_c$ simply consists of a single sinusoid. These sinusoids are rudimentarily shifted in phase to achieve a movement given a constant T^*. Algorithm 1 is applied with $T^* = 15$ mNm, $\gamma = 2.5$, $F_s = 10$ kHz, $\boldsymbol{\delta} = \mathbf{1}_{n_c n_h} \otimes [0.1, 0.01]^\top$, and $\hat{\mathbf{g}}$ parametrized using (14) with $n_h = 3$ harmonics, for a duration of $k_{\max} = 200$ iterations. We define soft constraints with a damping constant estimated at $\hat{d} = 0.03$ by measuring the average angular velocity for multiple values of T^*. Finally, we use $\beta = 0.05$ and average the velocity using $n_b = 201$ bins over 61 teeth, with the transient samples N_{tr} spanning the first ten teeth. The total experimental time is twelve hours for 200 iterations.

B. Results and analysis

Figure 6 presents the results. The developed algorithm reduces the cost function by an order of magnitude after approximately 100 iterations, eliminating the velocity ripple almost entirely. The resulting commutation function is shifted in phase compared to the initial function, and it assigns more currents to some coils than others, which is consistent with independent results [5] on the same setup.

From Figure 6a, it becomes clear that most of the performance gain is achieved in the first 50 iterations, i.e., three hours of experimental time. We stress that the algorithm requires no expert intervention during this time. The same software can be executed on various SRMs to achieve similar results despite individual manufacturing variations.

VII. Conclusion

The developed algorithm enables accurate control of Switched Reluctance Motors, even in the face of unmodeled manufacturing variations. Experiments and simulations demonstrate a significant reduction of torque ripple, which is crucial for high-precision applications such as FSO satellite communication, where CPAs require micro-radian pointing accuracy across a 360-degree range. The model-free nature of the algorithm renders it particularly well-suited for the mass production of SRMs. Hence, accurate tracking is possible with minimal development time, and commutation functions can be readjusted automatically when needed. Future research is aimed at considering magnetic saturation, such that the torque ripple is reduced even for large velocities.

VIII. Acknowledgements

The authors thank TNO, especially Lukas Kramer and Emre Deniz, for their contributions to the experiments.

References

[1] J. J. Wang, "A common sharing method for current and flux-linkage control of switched reluctance motor," *Electric Power Systems Research*, vol. 131, pp. 19–30, 2016.

[2] M. Ilić-Spong, T. J. Miller, S. R. Macminn, and J. S. Thorp, "Instantaneous Torque Control of Electric Motor Drives," *IEEE Transactions on Power Electronics*, vol. PE-2, no. 1, pp. 55–61, 1987.

[3] G. Fang, F. Pinarello Scalcon, D. Xiao, R. Vieira, H. Grundling, and A. Emadi, "Advanced Control of Switched Reluctance Motors (SRMs): A Review on Current Regulation, Torque Control and Vibration Suppression," *IEEE Open Journal of the Industrial Electronics Society*, vol. 2, no. April, pp. 280–301, 2021.

[4] L. Kramer, J. Peters, R. Voorhoeve, G. Witvoet, and S. Kuiper, "Novel motorization axis for a Coarse Pointing Assembly in Optical Communication Systems," in *IFAC PapersOnLine*, vol. 53, no. 2. Elsevier Ltd, 2020, pp. 8426–8431.

[5] M. van Meer, R. A. González, G. Witvoet, and T. Oomen, "Nonlinear Bayesian Identification for Motor Commutation: Applied to Switched Reluctance Motors," in *62nd IEEE Conference on Decision and Control (CDC)*. Singapore: IEEE, dec 2023, pp. 5494–5499.

[6] M. van Meer, G. Witvoet, and T. Oomen, "Robust commutation design: Applied to switched reluctance motors," in *Proceedings of the European Control Conference ECC24*. Stockholm: IEEE, 2024.

[7] Y. Tan, W. H. Moase, C. Manzie, D. Nešić, and I. M. Mareels, "Extremum seeking from 1922 to 20101," *Proceedings of the 29th Chinese Control Conference, CCC'10*, pp. 14–26, 2010.

[8] C. Zhang and R. Ordóñez, *Extremum-Seeking Control and Applications*, ser. Advances in Industrial Control. London: Springer London, 2012.

[9] D. Nešić, Y. Tan, W. H. Moase, and C. Manzie, "A unifying approach to extremum seeking: Adaptive schemes based on estimation of derivatives," *Proceedings of the IEEE Conference on Decision and Control*, pp. 4625–4630, 2010.

[10] S. Z. Khong, D. Nešić, C. Manzie, and Y. Tan, "Multidimensional global extremum seeking via the DIRECT optimisation algorithm," *Automatica*, vol. 49, no. 7, pp. 1970–1978, 2013.

[11] C. Gan, J. Wu, Q. Sun, W. Kong, H. Li, and Y. Hu, "A Review on Machine Topologies and Control Techniques for Low-Noise Switched Reluctance Motors in Electric Vehicle Applications," *IEEE Access*, vol. 6, pp. 31 430–31 443, 2018.