

Comparative Analysis of State Estimation Techniques for a DC Motor

1st Author Adel Abdelsamed
akmab@kth.se

2nd Project Partner: Daniel Pei Purroy
danpp@kth.se

Abstract—For modern robotics and industrial automation, it is very important to achieve precise state estimation in DC motors. In order increase the precision of state estimation methods, static friction models must be included. This study addresses the challenges posed by static friction nonlinearity. We present a comprehensive analysis of five state estimation techniques, that includes the Kalman Filter, Extended Kalman Filter, Unscented Kalman Filter, and Marginalized Particle Filter. The accuracy, robustness, and computational efficiency of the state estimators are evaluated using both simulation and experimental data. Our results highlight the superiority of nonlinear estimators in capturing the nonlinear rotor speed dynamics. This effect is particularly visible in low-speed regions where static friction dominates.

I. INTRODUCTION

Permanent magnet DC motors are among the most widely used types of actuators in servomechanical systems with applications ranging from mobile robots, robotic manipulators, machine tools and industrial machines [1]. Their popularity stems from their compact size, relatively low cost and the efficacy of precise control [2]. The ability to achieve precise control is particularly facilitated by the relatively simple mathematical model that describes the DC motor's dynamics, making it amenable to various control strategies [3].

Precise control of the rotor's position or speed is essential in many applications. This is typically achieved through feedback control strategies that rely on accurate measurements of the armature current, rotor position, and speed. However, increasing the number of sensors in electromechanical systems is undesirable due to the associated increase in hardware complexity and motor weight [5]. This has driven research towards sensorless control strategies that leverage state estimation techniques to reduce the reliance on physical sensors [4].

The primary challenge in state estimation for DC motors lies in the nonlinearities introduced by static friction, which are often neglected in linear models. These nonlinearities, particularly in low-speed regions, can lead to significant modeling errors and deteriorate the performance of control systems [2]. While linear estimation techniques, such as the Kalman filter (KF), have been widely used for state estimation in DC motors, they often fail to capture the full complexity of the system's dynamics, especially when static friction is present [2]. Nonlinear estimation techniques, such as the Extended Kalman Filter (EKF), Unscented Kalman Filter (UKF), and Particle Filter (PF), have been proposed to address these limitations,

but their comparative performance in the presence of static friction has not been investigated [4].

This project aims to address this gap by conducting a comprehensive comparative analysis of state estimation techniques for a DC motor, with a particular focus on the impact of static friction. The main contributions of this project are

- 1) Incorporation of static friction nonlinearity into state estimation,
- 2) Comprehensive comparative analysis of state estimation techniques including but not limited to the unscented Kalman filter and the marginalized particle filter and
- 3) Dual validation framework using simulation data and experimental data.

By incorporating the nonlinearities from static friction and addressing the challenges posed by the discontinuities in the model, the analysis explores the impact of including this complexity on the accuracy, robustness of state estimation and computational demand. Furthermore, while previous works often focus on implementing one or at most two techniques, this project provides a broader evaluation by considering five different state estimation techniques among which is the marginalized particle filter (MPF), which to the best of our knowledge have not been used before in friction estimation in DC motors, making this a novel contribution to the field. For all filter types reasoning is provided behind the choice of the filter. Lastly, the proposed filtering techniques are validated in two stages: first using an identified Simulink model, and subsequently with experimental data from a MAXON S 2326 permanent magnet DC motor to ensure practical applicability.

The results of this study demonstrate that nonlinear estimation techniques, particularly the UKF and MPF, outperform traditional linear methods like the KF, especially in the presence of static friction. The MPF, in particular, achieves the lowest mean absolute error (MAE). However, superior performance comes at the cost of increased computational complexity. Experimental results suggest that the added computational complexity of nonlinear estimators is justified.

This report is structured as follows. The context of this project is established through a discussion of related work in Section II. In Section III, the DC motor is modeled and the parameter identification framework is described. This step facilitates the understanding of the system's dynamics and the source of nonlinearity. Since the project mainly focuses on the state estimation task, the result of the identification step is already presented at that point. In Section IV, the different

state estimation techniques are introduced and their application. In Section V, the simulation and experimental results are presented. Lastly, the key findings and the implications of the study are summarized in Section VI.

II. RELATED WORK

Since the KF, was introduced by Rudolph E. Kalman in 1960 [6], it became a fundamental tool for state estimation in control systems. Its application to DC motor control has been explored extensively, particularly in sensorless control and noise reduction, driven by the need to reduce the amount of sensors in physical systems. For instance, the work by Praesomboon et al. [7] applies a linear model of the motor with a standard KF to achieve speed control and mitigate sensor noise, using only current measurements. By integrating the KF with a discrete-time PI controller, the paper demonstrates superior closed-loop performance experimentally using a digital signal processor. This study highlights the KF's effectiveness in reducing the number of sensors and rejecting sensor noise, while neglecting friction making it particularly relevant to our work.

However, relying solely on linear estimation techniques can limit the accuracy of state estimation, as actual DC motors exhibit nonlinear behaviors [2]. Nonlinear estimation techniques, such as the EKF, UKF, and PF, address this limitation by incorporating additional effects into the estimation models, thereby achieving greater accuracy and robustness [13].

The comparison between nonlinear estimation techniques is also well-documented in literature. The work by Aydogmus et al. [4] focuses on EKF and PF for a DC motor system with a nonlinear model. Unlike our setup, where nonlinearity arises from static friction, the nonlinearity here is due to an inverted pendulum load torque. Furthermore, the study utilizes current measurements as opposed to our approach, where position is measured using an encoder. The study concludes that the PF provides better accuracy and robustness under varying load conditions, though at the cost of higher computational complexity. While closed-loop control with the PF outperforms the EKF, the results are limited to simulations rather than experimental validation.

In the work by Srey et al. [1], the authors implement an EKF alongside a PD controller with dynamic compensation to estimate both states (armature current and rotor speed) and system parameters online. Here, the EKF assumes a linear model, and nonlinearity is introduced by estimating model parameters as static variables. In contrast, we focus on the interplay between state estimation and system nonlinearity caused by static friction.

Further exploration of PF in DC motor systems is seen in the work carried out by Rgiatos [3]. This study addresses state estimation using encoder measurements to estimate rotor angle, rotor speed, and armature current, which closely parallels our work. However, nonlinearity in the model arises from armature reaction. Results show enhanced tracking accuracy of sinusoidal and ramp signals but at the expense of computational load, a common theme in PF. While state feedback

linearization is used for control, the results are also limited to simulations, a contrast to our use of experimental data for validation.

Recent advancements in PF techniques have introduced the MPF for systems that combine linear and nonlinear components as proposed in the seminal work by Schön et al. [15]. Building on this foundation, the paper by Šmíd and Peroutka applied the MPF to permanent magnet synchronous motor drives (PMSM) [9]. The reduced-order model of the PMSM is nonlinear, even with friction effects being neglected, and the MPF demonstrated reliable state estimation even in challenging conditions such as standstill and low speeds. The approach was validated using both simulation and experimental results, showing effective performance in open-loop and closed-loop control with low computational overhead. Although our motor type is the brushed DC motor, these contributions inspire the use of MPF for estimating nonlinear friction effects.

The UKF, introduced by Simon J. Julier and Jeffrey K. Uhlmann in the early 2000s [8], has also been applied to motor control. For instance, in the work by Lv et al. [5], the UKF is used to estimate the states (armature currents and rotor speed) of a brushless DC motor under varying load conditions. This state estimator is relevant to our brushed DC motor system, since, by leveraging the unscented transform, the UKF avoids the challenges posed by the nonlinearities.

Historically, nonlinear estimation techniques have been heavily utilized for motor control systems, having evolved from the linear Kalman filter to more advanced methods such as the EKF, UKF, PF and MPF. However, a review of the literature reveals a lack of studies addressing the incorporation of static friction into state estimation models. Our work builds on this foundation by applying the KF, EKF, UKF, PF and MPF to a brushed DC motor while considering static friction-induced nonlinearity in a comparative analysis. The analysis provides insights how well the consideration of friction improves the accuracy of the state estimates - if at all - and how the different filters compare with respect to robustness and computational load. Lastly unlike many of the referenced studies, our contribution includes experimental validation.

III. MODELING AND IDENTIFICATION

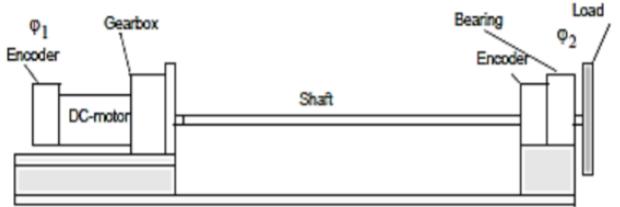


Fig. 1. Setup of the permanent magnet DC motor.

The configuration of the permanent magnet DC motor MAXON S 2326 is illustrated in Figure 1. This setup represents a typical application of DC motors, where an encoder is mounted on the motor side, and a gearbox connects the motor

via a rigid shaft to a flywheel load. The gearbox increases the torque on the load side, and due to the rigidity of the shaft, a secondary encoder is unnecessary on the load side.

Modeling and parameter identification of nonlinear DC motors have been extensively discussed in the literature. Here, we adopt the approach detailed in [2]. The electrical subsystem is modeled via Kirchoff's voltage law yielding

$$u = R \cdot i_a + L \cdot \frac{di_a}{dt} + k_t \cdot \omega_r, \quad (1)$$

where u is the applied armature voltage, i_a the applied armature current, ω_r the rotor speed, R the armature resistance, L the armature inductance and k_t the back-emf coefficient.

The mechanical subsystem is modeled via Newton's third law resulting in

$$\dot{\omega}_r \cdot (J_r + J_g) = \tau_m - d_m \cdot \omega_r - \tau_s - \frac{J_l}{n^2} \cdot \dot{\omega}_r, \quad (2)$$

where J_r is the rotor inertia, J_g the gearbox inertia, J_l the load inertia, d_m is the viscous friction coefficient, n the gear ratio, $\tau_m = k_t \cdot i_a$ the applied torque and τ_s is the static friction torque.

Since the main objective of this project is not the modeling part, in modeling static friction, the aim is to keep the model as simple as possible, while capturing the essence of the static friction phenomena [10]. The Karnop stick-slip model was used for implementing the static friction because of its simplicity and numerical stability while still achieving great accuracy [11]. The static friction τ_s is given by

$$\tau_s = \begin{cases} \tau_a = k_t \cdot i_a, & |\omega_r| \leq d_v \\ \tau_c \cdot \text{sgn}(\omega_r), & |\omega_r| > d_v \end{cases}, \quad (3)$$

where τ_c is the Coloumb friction torque, τ_a the applied torque and d_v is the velocity deadband. For the models used in state estimation, only the case $|\omega_r| > d_v$ is used to simplify the analysis. Using Eq. 1-3, a state space model of the system is obtained

$$\frac{d}{dt} \begin{bmatrix} i_a \\ \phi_r \\ \omega_r \end{bmatrix} = \underbrace{\begin{bmatrix} -\frac{R}{L} \cdot i_a - \frac{k_t}{L} \cdot \omega_r + \frac{1}{L} \cdot u \\ \omega_r \\ \frac{k_t}{J_{tot}} \cdot i_a - \frac{d_m}{J_{tot}} \cdot \omega_r - \frac{\tau_c}{J_{tot}} \cdot \text{sgn}(\omega_r) \end{bmatrix}}_{\mathbf{f}(i_a, \phi_r, \omega_r, u)}, \quad (4)$$

where $J_{tot} = J_r + J_g + \frac{J_l}{n^2}$ is the total inertia as seen from the rotor's side.

Some model parameters are obtained from the manufacturer's datasheet, while others — such as the viscous friciton coefficient d_m , Coloumb friction τ_c , velocity deadband d_v and the total inertia J_{tot} — require experimental identification. The identification follows the Strejc method [3]. Several step responses were sent to the motor and recorded. By neglecting the armature inductance, the transfer function from applied

TABLE I
IDENTIFIED AND MANUFACTURER'S PARAMETERS OF THE DC MOTOR
MAXON S 2326.

Model parameter	Parameter Value
Armature Resistance R	1.637×10^{-5} Nm/rad sec
Armature Inductance L	2.091×10^{-5} kgm ²
Torque-current coefficient k_t	69.7×10^{-3} N m A ⁻¹
Velocity deadband d_v	1×10^{-3} rad s ⁻¹
Total inertia J_{tot}	2.091×10^{-5} kgm ²
Viscous friction coefficient d_m	1.28×10^{-5} Nm/rad sec
Coloumb friction τ_c	9×10^{-4} N m

armature current to rotor speed is first-order and given by (see Appendix VII-A for the derivation)

$$G(s) = \frac{\omega_r(s)}{u(s)} = \frac{K}{s \cdot T + 1}, \quad (5)$$

$$T = \frac{J_{tot}}{d_m + \frac{k_t^2}{R}}, \quad K = \frac{k_t}{R \cdot d_m + k_t^2}.$$

The time constant T is defined as the time it takes to reach 63.2% of the final value of the rotor speed, while the final value of the step response is the gain K multiplied by the applied voltage u . Using the recorded step responses, d_m and J_{tot} can be computed across a range of speed profiles through the definition of the gain K and time constant T in Eq. 5. It is noted that d_m , although dependent on the rotor speed- is assumed constant [3]. Furthermore, the Coloumb friction is determined by considering Eq. 2 at steady-state

$$k_t \cdot i_{a,ss} = d_m \cdot \omega_{r,ss} + \text{sgn}(\omega_{r,ss}) \cdot \tau_c. \quad (6)$$

Since Eq. 6 is linear in τ_c , it can be estimated by least-squares. The velocity deadband d_v is estimated by measuring the input voltage and the current generated in the armature at which the motor starts to move [3]. Finally, the parameters are manually tuned to enhance the model's accuracy with respect to the real-time rotor speed measurements. The identified parameters, along with those provided in the manufacturer's datasheet, are summarized in Table I.

In order to perform state estimation, a discrete-time model is required. The state space equation in Eq. 4 is discretized using Euler method and the following discrete-time model is obtained

$$\begin{bmatrix} i_a^k \\ \phi_r^k \\ \omega_r^k \end{bmatrix} = \begin{bmatrix} i_a^{k-1} \\ \phi_r^{k-1} \\ \omega_r^{k-1} \end{bmatrix} + \delta_t \cdot \mathbf{f}(i_a^{k-1}, \phi_r^{k-1}, \omega_r^{k-1}, u_k) \quad (7)$$

$$y_k = \underbrace{\begin{bmatrix} 0 & 1 & 0 \end{bmatrix}}_{C_d} \cdot \begin{bmatrix} i_a^k \\ \phi_r^k \\ \omega_r^k \end{bmatrix},$$

with δ_t being the sample time. The measurement function is assumed to be linear neglecting the quantization effects of the encoder. In reality, the used encoder has a resolution of 3600 pulse per revolution.

A. Choice of sample time

The choice of sample time is not trivial, since a too large step size will lead to instability of the difference equations in Eq. 7. Therefore, the sample time must guarantee stability of the discrete-time system. Since the stability formulation for discontinuous nonlinear systems is much more difficult to formalize, the nonlinear term $-\frac{\tau_c}{J_{tot}} \cdot sgn(\omega_r)$ resulting from the static friction in Eq. 4 is neglected and the following linearized system from Eqn. 7 is obtained

$$\begin{aligned} \begin{bmatrix} i_a^k \\ \phi_r^k \\ \omega_r^k \end{bmatrix} &= \underbrace{\begin{bmatrix} 1 - \frac{R \cdot \delta_t}{L} & 0 & -\frac{k_t \cdot \delta_t}{L} \\ 0 & 1 & \delta_t \\ \frac{k_t \cdot \delta_t}{J_{tot}} & 0 & 1 - \frac{d_m \cdot \delta_t}{J_{tot}} \end{bmatrix}}_{\mathbf{A}_d} \cdot \begin{bmatrix} i_a^{k-1} \\ \phi_r^{k-1} \\ \omega_r^{k-1} \end{bmatrix} + \underbrace{\begin{bmatrix} \delta_t \\ 0 \\ 0 \end{bmatrix}}_{\mathbf{B}_d} \cdot u_k \\ y^k &= \underbrace{\begin{bmatrix} 0 & 1 & 0 \end{bmatrix}}_{\mathbf{C}_d} \cdot \begin{bmatrix} i_a^k \\ \phi_r^k \\ \omega_r^k \end{bmatrix} \end{aligned} \quad (8)$$

The linear system is stable, if the eigenvalues satisfy $|\lambda_i(\mathbf{A}_d)| \leq 1$, $\forall i$. MATLAB's symbolic toolbox is used to solve the characteristic equation

$$p(\lambda) = \left(\lambda - 1 + \frac{R \cdot \delta_t}{L} \right) \cdot \left(\lambda - 1 + \frac{d_m \cdot \delta_t}{J_{tot}} \right) - \frac{k_t^2 \cdot \delta_t^2}{J_{tot} \cdot L}. \quad (9)$$

For a sample time $\delta_t \leq 0.2$ ms, the system is stable. In the following, the sample time is fixed at $\delta_t = 0.1$ ms.

B. Observability

A necessary and sufficient condition for a state estimator to exist is that the system is observable. However, observability for nonlinear systems can only be defined locally [12]. Therefore, we conduct the analysis again using the linearized model in Eq. 8. The linearized system is observable, since the observability matrix \mathcal{O} is full rank

$$rank(\mathcal{O}) = rank \begin{bmatrix} \mathbf{C}_d \\ \mathbf{C}_d \cdot \mathbf{A}_d \\ \mathbf{C}_d \cdot \mathbf{A}_d^2 \end{bmatrix} = 3 = n_x, \quad (10)$$

with n_x being the state dimension.

IV. STATE ESTIMATION

A. Kalman Filter

An advantage of neglecting static friction is the resulting linear system in Eq. 8, for which the standard KF can be employed. The KF is the optimal state estimator under the assumption of a linear system and Gaussian noise [13]. The linear model in Eq. 8 is superimposed with white Gaussian noise

$$\begin{aligned} x_k &= \mathbf{A}_d \cdot x_{k-1} + \mathbf{B}_d \cdot u_k + \epsilon_k \\ y_k &= \mathbf{C}_d \cdot x_k + \eta_k, \end{aligned} \quad (11)$$

where $\epsilon_k \sim \mathcal{N}(0, R_k)$ denotes the process noise and $\eta_k \sim \mathcal{N}(0, Q_k)$ the measurement noise. The process and measurement noise are assumed to be uncorrelated. The Kalman filter

operates in two steps - prediction and an update step - in order to obtain the state estimate at each sampling instant [13].

$$\begin{aligned} \text{Prediction Step} \quad &\begin{cases} \bar{\mu}_k &= \mathbf{A}_d \cdot \bar{\mu}_{k-1} + \mathbf{B}_d \cdot u_k, \\ \bar{\Sigma}_k &= \mathbf{A}_d \Sigma_{k-1} \mathbf{A}_d^T + R_k, \end{cases} \\ \text{Update Step} \quad &\begin{cases} K_k &= \bar{\Sigma}_k \mathbf{C}_d \cdot (\mathbf{C}_d \bar{\Sigma}_k \mathbf{C}_d^T + Q_k)^{-1}, \\ \mu_k &= \bar{\mu}_k + K_k(y_k - \mathbf{C}_d \bar{\mu}_k). \end{cases} \end{aligned} \quad (12)$$

The performance of the Kalman filter depends on the tuning of the noise covariance matrices Q and R and the initial estimate of the state μ_0 as well as the initial covariance matrix Σ_0 . Guidelines for selecting μ_0 and Σ_0 are given in the Appendix VII-C.

B. Extended Kalman Filter

In order to incorporate the nonlinear term resulting from static friction, the state-space model becomes nonlinear as shown in Eq. 7. Hence, the EKF can be used to overcome the linearity assumption. The EKF still assumes uncorrelated, white, zero-mean Gaussian process and measurement noise.

$$\begin{aligned} x_k &= \mathbf{g}(x_{k-1}, u_k) + \epsilon_k \\ y_k &= \mathbf{h}(x_k) + \eta_k, \end{aligned} \quad (13)$$

with $\mathbf{g}(x_{k-1}, u_k) = x_{k-1} + \delta_t \cdot \mathbf{f}(x_{k-1}, u_k)$ and $\mathbf{h}(x_k) = \mathbf{C}_d \cdot x_k$ as shown in Eqn.7.

EKF is based on approximating the nonlinear system by linearizing the system using Taylor expansion. Once the system equations are linearized, the EKF results in the same prediction and update step as in the standard KF.

$$\begin{aligned} \text{Prediction Step} \quad &\begin{cases} \bar{\mu}_k &= \mathbf{g}(\bar{\mu}_{k-1}, u_k) \\ \bar{\Sigma}_k &= \mathbf{G}_k \Sigma_{k-1} \mathbf{G}_k^T + R_k, \end{cases} \\ \text{Update Step} \quad &\begin{cases} K_k &= \bar{\Sigma}_k \mathbf{H}_k \cdot (\mathbf{H}_k \bar{\Sigma}_k \mathbf{H}_k^T + Q_k)^{-1}, \\ \mu_k &= \bar{\mu}_k + K_k(y_k - \mathbf{H}_k \bar{\mu}_k), \end{cases} \end{aligned} \quad (14)$$

with the the following Jacobians

$$\begin{aligned} \mathbf{G}_k &= \left. \frac{\partial \mathbf{g}(x_{k-1}, u_k)}{\partial x_{k-1}} \right|_{x_{k-1}=\mu_{k-1}}, \\ \mathbf{H}_k &= \left. \frac{\partial \mathbf{h}(x_k)}{\partial x_k} \right|_{x_k=\bar{\mu}_k} = \mathbf{C}_d \cdot \bar{\mu}_k. \end{aligned} \quad (15)$$

The measurement function remains linear, since the only source of nonlinearity stems from the static friction in the rotor speed update equation. A challenge in using the EKF is the discontinuity resulting from the signum function. Hence, in the following the signum function is approximated using a continuous function in order to be able to form the Jacobians. We make use of the following lemma [11]

$$\lim_{\xi \rightarrow \infty} \frac{2}{\pi} \cdot \tan^{-1}(\xi x) = sgn(x). \quad (16)$$

The parameter ξ affects the steepness with which the signum function is approximated and thereby influences greatly the

performance of the filter. Choosing ξ too high will result in the filter being sensitive to noise in near-zero operating regions and hence might induce divergence. In practice, it is better to keep ξ in the range $[10, 10^3]$ and compensate for modeling inaccuracies with a higher process noise variance for the rotor speed. Using Eq. 16, the Jacobian becomes

$$G_k = \begin{bmatrix} 1 - \frac{R \cdot \delta_t}{L} & 0 & -\frac{k_t \cdot \delta_t}{L} \\ 0 & 1 & \frac{\delta_t}{J_{tot}} \\ \frac{k_t \cdot \delta_t}{J_{tot}} & 0 & 1 - \frac{d_m \cdot \delta_t}{J_{tot}} - \frac{2\delta_t \tau_c}{\pi J_{tot}} \cdot \frac{\xi}{1 + (\xi \omega_r^{k-1})^2} \end{bmatrix}. \quad (17)$$

C. Unscented Kalman Filter

Similar to the EKF, the UKF is based on approximating the nonlinear transformations. However, the UKF utilizes a derivative-free approximation called the unscented transform, where $2n_x + 1 = 7$ points, called the Sigma points, represent the mean and covariance of the stochastic variable. These are then passed through the nonlinear transformation, and the weighted outcome of this is used to estimate the posterior mean and covariance [12]. In this way, the UKF offers a deterministic sampling approach to approximate the posterior distribution. The UKF still assumes Gaussian priors and Gaussian noise and approximates the posterior by a Gaussian distribution. Since it does not require computing Jacobians, it does not form a challenge to the discontinuity of the static friction and yields acceptable computational time for our real-time system due to the fixed number of Sigma points.

The UKF algorithm that runs at each sampling instant is shown in Algorithm 1 with explanation of each step. The UKF consists also of a prediction step (steps 1-3) and an update step (steps 4-9). The weights of the mean $w_m^{[i]}$ and the covariance $w_c^{[i]}$ are constant throughout all the iterations and are computed as follows

$$\begin{aligned} w_m^{[0]} &= \frac{\lambda}{n_x + \lambda}, \quad w_c^{[0]} = \frac{\lambda}{n_x + \lambda} + (1 - \alpha^2 + \beta), \\ w_m^{[i]} &= w_c^{[i]} = \frac{1}{2(n_x + \lambda)} \quad \text{for } i = 1, \dots, 2n, \end{aligned}$$

with $\lambda = \alpha^2(n_x + \kappa) - n_x$. The UKF is greatly influenced by the design parameters α , κ and β , which are usually chosen ad-hoc [14]. In choosing those design parameters, we follow the approach detailed in [14]. The scaling parameter α controls the spread of the SPs around the mean, with larger values spreading the points further and smaller values pulling them closer. The parameter β is used to tune the capture of higher-order moments, particularly the kurtosis of the distribution, and it is typically set to 2 for Gaussian distributions. The parameter κ adjusts the scaling to minimize covariance estimation errors, balancing the SPs' spread while maintaining numerical stability. Special care must be taken when tuning α and κ as there is a redundancy in both parameters. To initialize the selection of these parameters the following guidelines are chosen [14]

- If a decrease in the spread of the SPs is desired, use $\kappa = 0$ and $\alpha < 1$.

- If an increase in the spread of the SPs is desired, use $\kappa > 0$ and $\alpha = 1$.

Algorithm 1 UKF [13]

1. Generate sigma points:

$$\mathcal{X}_{k-1} = [\mu_{k-1} \quad \mu_{k-1} + \gamma \sqrt{\Sigma_{k-1}} \quad \mu_{k-1} - \gamma \sqrt{\Sigma_{k-1}}]$$

2. Propagate sigma points through the process model:

$$\mathcal{X}_k^* = g(\mathcal{X}_{k-1}, u_k)$$

3. Compute predicted mean and covariance:

$$\bar{\mu}_k = \sum_{i=0}^{2n} w_m^{[i]} \mathcal{X}_k^{*[i]}$$

$$\bar{\Sigma}_k = \sum_{i=0}^{2n} w_c^{[i]} (\mathcal{X}_k^{*[i]} - \bar{\mu}_k)(\mathcal{X}_k^{*[i]} - \bar{\mu}_k)^T + R_k$$

4. Generate predicted sigma points:

$$\hat{\mathcal{X}}_k = [\bar{\mu}_k \quad \bar{\mu}_k + \gamma \sqrt{\bar{\Sigma}_k} \quad \bar{\mu}_k - \gamma \sqrt{\bar{\Sigma}_k}]$$

5. Map sigma points through the measurement model:

$$\mathcal{Y}_t = h(\hat{\mathcal{X}}_k)$$

6. Compute predicted measurement mean and covariance:

$$\hat{y}_t = \sum_{i=0}^{2n} w_m^{[i]} \mathcal{Y}_t^{[i]},$$

$$S_k = \sum_{i=0}^{2n} w_c^{[i]} (\mathcal{Y}_t^{[i]} - \hat{y}_t)(\mathcal{Y}_t^{[i]} - \hat{y}_t)^T + Q_k$$

7. Compute cross-covariance:

$$\Sigma_k^{x,y} = \sum_{i=0}^{2n} w_c^{[i]} (\hat{\mathcal{X}}_k^{[i]} - \bar{\mu}_k)(\mathcal{Y}_t^{[i]} - \hat{z}_k)^T$$

8. Compute Kalman gain:

$$K_k = \Sigma_k^{x,y} S_k^{-1}$$

9. Update mean and covariance:

$$\mu_k = \bar{\mu}_k + K_k(y_k - \hat{y}_k), \quad \Sigma_k = \bar{\Sigma}_k - K_k S_k K_k^T$$

D. Particle Filter

All previous filters assume Gaussian priors and noise resulting in a Gaussian posterior distribution. The PF relaxes this assumption and approximates any posterior distribution. Furthermore, PF also do not require forming Jacobians and hence the discontinuity of the static friction term in the DC motor model can be included without further consideration [13]. These two properties: ability to handle nonlinearity and handling multi-modal distributions make it suitable for the DC motor application.

The main idea behind PF is to represent the posterior distribution $p(x_k|y_{1:t}, u_{1:t})$ by a set of N particles $\mathcal{X}_k = \{x_k^{[i]}, \forall i = 1, \dots, N\}$. The algorithm for the particle filter is represented in Algorithm 2, if steps 4.a and 4.c are neglected [15].

Each particle $x_k^{[i]}$ represents a possible state of the system. Since, we do not marginalize over any states in the standard particle, $x_k^{n,(i)}$ represents the entire state vector for the i -th particle at time step k . After initializing the N particles from the prior distribution $p_{x_0^n}(x_0^n)$, the unnormalized importance weights $w_k^{(i)}$ are computed from the measurement model, which simplifies using our model in Eq.7 to $p(y_k|x_k^{n,(i)}) = \mathcal{N}\left(C_d x_k^{n,(i)}, C_k \bar{P}_k C_k^T + Q_k\right)$.

To avoid particle degeneracy, low-variance resampling with replacement is performed according to the particles' normalized weights $\hat{w}_k^{(i)}$. Low-variance sampling offers a more systematic approach for resampling, ensuring that particles with higher weights are more likely to be included in the particle set after resampling.

$$r \sim \mathcal{U}\left(0, \frac{1}{N}\right), \quad \text{and} \quad r_j = r + \frac{j-1}{N}, \quad \forall j = 1, \dots, N,$$

where N particles are then selected based on their cumulative sum of their normalized weights [13].

Lastly, new particles are predicted for the next time step $k+1$. The prediction is based on sampling from the system dynamics in Eq. 13 yielding $p\left(x_{k+1}^n|x_k^{n,(i)}, Y_k\right) = \mathcal{N}\left(\mathbf{g}(x_k^{n,(i)}), R_k\right)$.

E. Marginalized Particle Filter

Directly applying a PF to all three states leads to an exponential increase in the number of particles needed to adequately represent the posterior distribution, as the volume to be covered increases. Since, the nonlinearity manifests itself only in one state, the rotor speed state update, marginalizing over the linear states (current and angle) and using a PF only for the rotor speed equation, lowers the incurred computational costs while increasing the accuracy of the filter given the same number of particles. This is the key idea of the MPF, also called Rao-Blackwellized particle filter [15].

The MPF partitions the state vector as

$$x_k = \begin{bmatrix} x_k^l \\ x_k^n \end{bmatrix}$$

with x_k^l denoting the variables with conditionally linear dynamics and x_k^n denoting the nonlinear dynamics. The trick here is to use Bayes theorem to marginalize out the conditionally linear states from the desired posterior distribution

$$p(x_k|u_{1:k}, y_{1:k}) = p(x_k^l, \mathcal{X}_k^n|u_{1:k}, y_{1:k}) = \underbrace{p(x_k^l|\mathcal{X}_k^n, u_{1:k}, y_{1:k})}_{KF} \underbrace{p(\mathcal{X}_k^n|u_{1:k}, y_{1:k})}_{PF} \quad (18)$$

and use Kalman Filter to update the conditionally linear states for each particle [15].

Algorithm 2 PF and MPF [15]

1. Initialization:

for $i = 1$ to N **do**

 Initialize particles $x_0^{n,(i)} \sim p_{x_0^n}(x_0^n)$

 Set $\{x_0^{l,(i)}, P_0^{l,(i)}\} = \{\bar{x}_0, \bar{P}_0\}$

end for

2. Importance Weights Evaluation:

for $i = 1$ to N **do**

 Evaluate $w_k^{(i)} = p(y_k|x_k^{n,(i)}, Y_{k-1}) =$

$\mathcal{N}\left(\mathbf{h}_k + \mathbf{C}_k \bar{x}_k^l, \mathbf{C}_k \bar{P}_k \mathbf{C}_k^T + Q_k\right)$

 Normalize $\hat{w}_k^{(i)} = \frac{w_k^{(i)}}{\sum_{j=1}^N w_k^{(j)}}$

end for

3. Particle Filter Measurement Update (Resampling):

Resample N particles with replacement

$\Pr\left(x_k^{n,(i)} = \bar{x}_k^{n,(j)}\right) = \hat{w}_k^{(j)}$

4. Particle Filter Update and Kalman Filter:

4.a Kalman Filter Measurement Update:

$$M_k = \mathbf{C}_k \bar{P}_k \mathbf{C}_k^T + Q_k$$

$$K_k = \bar{P}_k \mathbf{C}_k^T M_k^{-1}$$

$$x_k^l = \bar{x}_k^l + K_k (y_k - h_k - \mathbf{C}_k \bar{x}_k^l)$$

$$P_k = \bar{P}_k - K_k M_k K_k^T$$

4.b Particle Filter Update (Prediction):

for $i = 1$ to N **do**

 Predict new particles $x_{k+1}^{n,(i)} \sim p\left(x_{k+1}^n|x_k^{n,(i)}, Y_k\right) =$

$\mathcal{N}\left(\mathbf{f}_k^n + \mathbf{A}_k^n x_k^l, \mathbf{A}_k^n P_k (\mathbf{A}_k^n)^T + \mathbf{G}_k^n R_k^n (\mathbf{G}_k^n)^T\right)$

end for

4.c Kalman Filter Update:

$$z_k = x_{k+1}^n - \mathbf{f}_k^n$$

$$\bar{A}_k^l = \mathbf{A}_k^l - \mathbf{G}_k^l (R_k^{ln})^T (\mathbf{G}_k^n R_k^n)^{-1} \mathbf{A}_k^n$$

$$\bar{R}_k^l = R_k^l - (R_k^{ln})^T (R_k^n)^{-1} R_k^{ln}$$

$$N_k = \bar{A}_k^l P_k (\mathbf{A}_k^n)^T + \mathbf{G}_k^n R_k^n (\mathbf{G}_k^n)^T$$

$$L_k = \bar{A}_k^l P_k (\mathbf{A}_k^n)^T N_k^{-1}$$

$$\bar{x}_{k+1}^l = \bar{A}_k^l x_k^l + \mathbf{B}_k^l (x_k^n) u + \mathbf{G}_k^l (R_k^{ln})^T (\mathbf{G}_k^n R_k^n)^{-1} z_k + \mathbf{f}_k^l + L_k (z_k - \mathbf{A}_k^n x_k^l)$$

$$\bar{P}_{k+1} = \bar{A}_k^l P_k (\bar{A}_k^l)^T + \mathbf{G}_k^l R_k^l (\mathbf{G}_k^l)^T - L_k N_k L_k^T$$

The MPF utilizes a general system structure to decompose the states

$$x_{k+1}^n = \mathbf{f}_k^n (x_k^n) + \mathbf{A}_k^n (x_k^n) x_k^l + \mathbf{G}_k^n (x_k^n) \epsilon_k^n$$

$$x_{k+1}^l = \mathbf{f}_k^l (x_k^n) + \mathbf{A}_k^l (x_k^n) x_k^l + \mathbf{B}_k^l (x_k^n) u + \mathbf{G}_k^l (x_k^n) \epsilon_k^l \quad (19)$$

$$y_k = \mathbf{h}_k (x_k^n) + \mathbf{C}_k (x_k^n) x_k^l + \eta_k,$$

where the state noise is assumed white and Gaussian distributed with

$$\epsilon_k = \begin{bmatrix} \epsilon_k^l \\ \epsilon_k^n \end{bmatrix} \sim \mathcal{N}(0, R_k), \quad R_k = \begin{bmatrix} R_k^l & R_k^{ln} \\ (R_k^{ln})^T & R_k^n \end{bmatrix}. \quad (20)$$

The measurement noise is assumed white and Gaussian distributed according to $\eta_k \sim \mathcal{N}(0, Q_k)$. It is noted that we extend the decomposition presented in the original paper [15] to include a control term $\mathbf{B}_k^l (x_k^n) u$ acting on the linear state

variables x_k^l . These assumptions, although not a requirement for the MPF, are needed to obtain a closed-form analytical solution for the conditionally linear states updates. The exact expressions for the functions appearing in Eq. 19 are provided in the Appendix VII-B for our model.

The algorithm for the MPF is presented in Algorithm 2. The use of the KF adds the steps 4.a and 4.c to the standard PF algorithm, where the KF update and prediction step are performed, respectively for each particle [15].

V. RESULTS

The estimation algorithms were validated through a two-step approach: first using simulation data generated from the identified model in Simulink, and then using experimental data from the real motor.

A. Simulation Results

The model in Eq. 4, whose identified parameters are presented in Table I is implemented in Simulink to generate data. To ensure the generated data closely resembles the real motor's behavior, all captured effects, including static friction and the dead-zone nonlinearity (though neglected in the estimation models), were included.

To simulate realistic conditions and test the robustness of the algorithms, a 10 % uncertainty was introduced to all identified parameters, accounting for inherent uncertainties in parameter estimation. Although in reality the resistance to armature R_a slightly varies with temperature [4], its impact on the simulation results is minimal and was thus assumed constant. Furthermore, given that the employed encoder has a standard deviation of 10 quantization intervals, the measured rotor angle is affected by additive Gaussian noise with zero mean and a standard deviation of $\frac{20\pi}{3600}$.

The performance of all filters depend on the design parameters. Thus, all filters underwent an extensive manual tuning process to optimize performance. Furthermore, for the PF and MPF filters, the maximum number of particles N is fixed at 1000 particles to avoid excessive computational loads. 1000 particles are selected for the PF, however the MPF utilizes only 500 particles. Guidelines for how the initial estimates are selected are presented in Appendix VII-C.

The algorithms were tested using different 4-second waveforms, including sine, sawtooth, and square waves. These waveforms were chosen to evaluate the filters' tracking performance across varying signal characteristics, sine waves for smooth signals, ramps to assess the ability to track linear changes, and square waves for sharp discontinuities. Additionally, two frequencies were used: 1 Hz for low-frequency signals and 50 Hz for high-frequency signals.

Waveform amplitudes were also varied to assess performance across operating ranges. With a maximum applied voltage of 24 V, three amplitudes were tested: low (3 V), medium (10 V), and high (24 V). The low-speed region, where static friction dominates, is particularly challenging for estimation, making the amplitude variation crucial for evaluation [2]. An overview of the test signals is shown in Table II.

TABLE II
OVERVIEW OF TEST SIGNALS USED IN SIMULATION.

Waveform	Frequency (Hz)	Amplitude (V)
Sine	1, 50	3, 10, 24
Sawtooth	1, 50	3, 10, 24
Square	1, 50	3, 10, 24

The accuracy of the state estimators is evaluated using the mean absolute error (MAE) between true and estimated states

$$\left[\begin{array}{c} \bar{e}_{i_a} \\ \bar{e}_{\phi_r} \\ \bar{e}_{\omega_r} \end{array} \right] = \frac{\sum_{k=0}^{N_{sig}} \left| \begin{bmatrix} i_a^k \\ \phi_r^k \\ \omega_r^k \end{bmatrix} - \begin{bmatrix} \hat{i}_a^k \\ \hat{\phi}_r^k \\ \hat{\omega}_r^k \end{bmatrix} \right|}{N_{sig}}, \quad (21)$$

across all test signals for each state respectively. N_{sig} denotes the length of the signal.

The MAE values for the five filters — KF, EKF, UKF, PF, and MPF—are summarized in Table III.

For the estimation of i_a , the MPF achieves the lowest MAE at 4.664×10^{-4} , outperforming the other methods. Still, the differences among the filters are minor for this state.

In the estimation of ϕ_r , all filters perform similarly, with the KF, EKF, and MPF yielding identical errors of 0.0016. This suggests that for this state, even simpler methods like the KF can be effective due to its linear dynamics in nature. Additionally, since ϕ_r is directly measured, the updates rely less on modeled dynamics.

For ω_r , the UKF demonstrates the best performance, achieving the lowest MAE of 0.4420. The MPF and EKF also perform well with a slightly higher error of 0.4458 and 0.4533, respectively. The KF, while computationally efficient, shows the highest error at 0.5074, highlighting its limitations in accurately capturing the nonlinear dynamics of the rotor speed.

The MPF demonstrates the lowest estimation errors for both the armature current and rotor position, while achieving the second lowest error in rotor speed estimation, marginally trailing the UKF. A consistent trend is evident in Table III, where the use of nonlinear estimators reduces errors in capturing the nonlinear dynamics of the rotor speed.

The computational performance of the algorithms is also evaluated and compared. Each filter implementation was optimized for vectorization, taking into account the specific dynamics of the system to ensure a fair and consistent comparison. The filters were executed in MATLAB on a 16 GB RAM, Intel i7-1260P 2.10 GHz core (64 bit) processor. To mitigate noise and the occasional impact of limited cache availability, the results were averaged over 10 runs. The average and maximum computational times per iteration for each filter are presented in Table IV.

Overall, the KF and EKF demonstrate the lowest maximum and average computational times, highlighting their efficiency. However, the improved accuracy provided by the nonlinear estimators comes at the cost of increased computational complexity. The UKF, due to its deterministic sampling approach, exhibits a moderate average computational time of 0.011 ms,

TABLE III
MEAN ABSOLUTE ERROR (MAE) OF THE STATES ACROSS THE SIGNALS IN TABLE II USING SIMULATION DATA.

Filter Type	\bar{e}_{i_a} (A)	\bar{e}_{ϕ_r} (rad)	\bar{e}_{ω_r} ($\frac{\text{rad}}{\text{s}}$)
KF	5.713×10^{-4}	0.0016	0.5074
EKF	5.767×10^{-4}	0.0016	0.4533
UKF	6.036×10^{-4}	0.0027	0.4420
PF	6.055×10^{-4}	0.0017	0.4881
MPF	4.664×10^{-4}	0.0016	0.4458

TABLE IV
COMPARISON OF THE COMPUTATIONAL TIME OF EACH ITERATION FOR THE ESTIMATORS USING SIMULATION DATA.

Comput. Time	KF	EKF	UKF	PF	MPF
Max Time t_{max} [ms]	0.51	0.56	1.04	5.59	4.97
Avg Time t_{avg} [ms]	0.002	0.003	0.011	0.651	0.088

approximately double that of the EKF. The MPF, while still more computationally demanding than the KF and EKF, shows a significant improvement over the PF in terms of computational efficiency.

Given the constraint that the sample time should not exceed 0.1 ms, the MPF's average computational time falls below this threshold, making it suitable for practical applications. However, its maximum computational time is considerably higher.

A comparison between the KF and the best-performing nonlinear state estimator, the MPF, is presented in the left column of Figure 2 for a low-amplitude sinusoidal input signal and in Figure 3 for a square wave input. This is close to the near-zero operating region, where static friction is most dominant.

The simulation results show that although the inclusion of nonlinear static friction reduces the MAE in rotor speed estimation, its impact is not substantial enough to justify its inclusion and the increased computational demand. This finding demonstrates the robustness of the KF, even in the presence of parametric uncertainties in simulation.

B. Experimental Results

While the simulation results are based on a model that closely aligns with the one used in the estimation algorithms, further validation using real-world data is conducted to ensure the soundness of the results and their applicability in practical scenarios.

1) *Experimental setup:* The target hardware used in this project is the LAUNCHXL-F28379D microcontroller board, a real-time control MCU from Texas Instruments. The board features dual CPUs operating at a frequency of 200 MHz. The microcontroller is interfaced with the MAXON S 2326 motor via USB, and an encoder is mounted on the rotor side to measure the angular position. To evaluate the performance of the estimation algorithms, current measurements from the motor driver and tachometer readings from the motor shaft are logged. The measured data are assumed to correspond

TABLE V
MEAN ABSOLUTE ERROR (MAE) OF THE STATES ACROSS THE SIGNALS IN TABLE II USING EXPERIMENTAL DATA.

Filter Type	\bar{e}_{i_a} (A)	\bar{e}_{ϕ_r} (rad)	\bar{e}_{ω_r} ($\frac{\text{rad}}{\text{s}}$)
KF	0.0052	8.27×10^{-6}	8.3357
EKF	0.0033	8.27×10^{-6}	5.2838
UKF	0.0016	8.23×10^{-6}	2.576
PF	0.0044	0.0114	4.2475
MPF	0.0016	1.99×10^{-7}	2.6062

to the true states of the systems. Data logging is performed using MATLAB's external mode at a frequency of 10 kHz. The logged data is then used as ground truth and the estimation algorithms are run offline on the same device used for simulation.

2) *Results:* Similar to the simulation results, the filters' tracking performance is evaluated using the same test signals shown in Table II. Two new frequencies are selected to represent high (50π) and low frequencies (π), and the signal duration is extended to 11 s. The filters are retuned to account for different noise characteristics in the experimental setup. Table V summarizes the MAE between the true and estimated states for all filter types.

For ϕ_r , all filters yield similar errors. Notably, the PF has the highest error of $0.0014 \text{ rad s}^{-1}$, primarily due to the limited number of particles. Increasing the particle count improves accuracy. Furthermore, a higher measurement noise covariance is chosen to avoid particle deprivation and therefore allow the tracking of multiple hypotheses, particularly during the initial phase of localization. The MPF achieves the lowest MAE of 1.00×10^{-7} .

For ω_r , nonlinear estimators generally outperform the linear KF. The KF shows the highest MAE at $8.3357 \text{ rad s}^{-1}$, reflecting its limitations in capturing nonlinear rotor speed dynamics. The UKF exhibits the lowest MAE of 2.576 rad s^{-1} , followed closely by the MPF with a MAE of $2.6062 \text{ rad s}^{-1}$.

Since the current is linked to the rotor speed using the back-emf constant, the same trend as with the rotor speed can be seen. The MPF and UKF both have the lowest MAE of 0.0016 rad with the KF showing the highest MAE of 0.0052 rad.

Figure VI displays the mean and maximum computational times per iteration for each filter. The KF and EKF have the lowest average computational time at 0.0017 ms, while the UKF and MPF exhibit moderate computational times, averaging 0.011 ms and 0.084 ms, respectively.

Again, a comparison between the KF and the best-performing nonlinear state estimator, the MPF, using the experimental data is presented in the right column of Figure 2 for a low-amplitude sinusoidal input signal and in Figure 3 for a square wave input.

Comparing the MAE results from the simulation data (Table III) and experimental data (Table V) reveals generally lower errors in the simulation results. Furthermore, the performance of the KF degrades significantly more compared to the nonlinear state estimators using the experimental data.

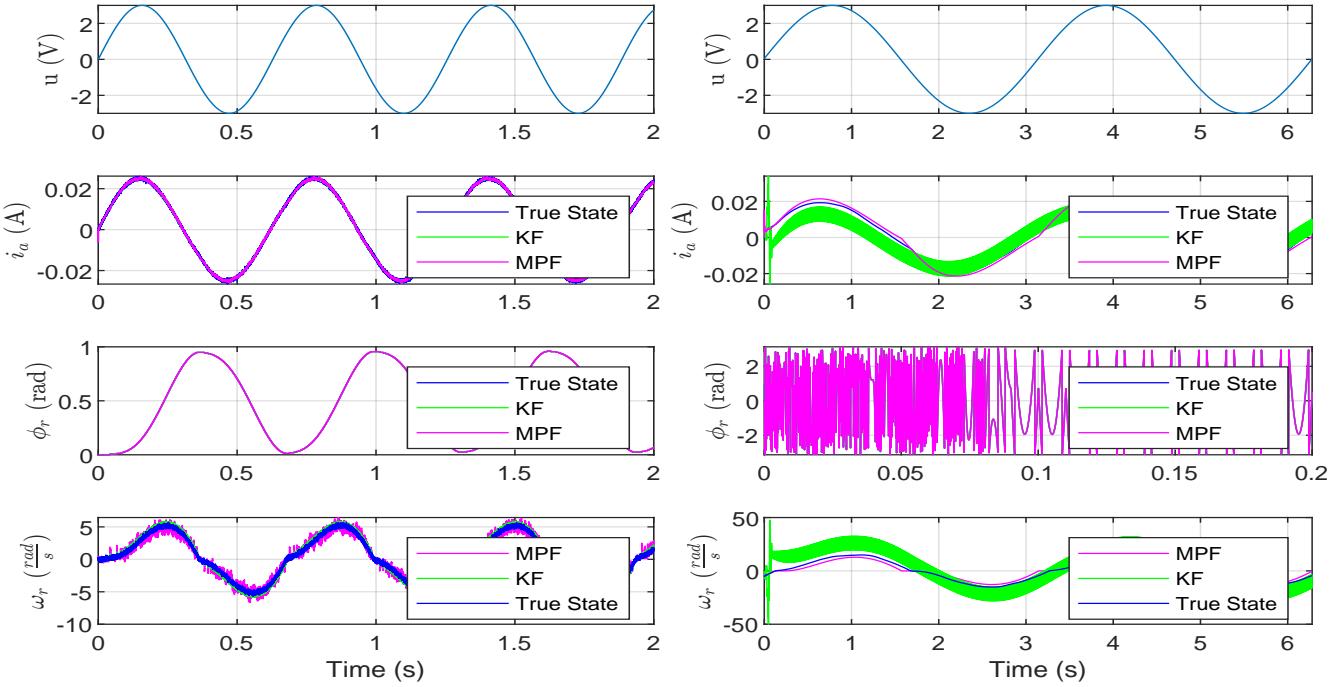


Fig. 2. Comparison of KF and the best-performing nonlinear estimator, MPF, performance in tracking the true state over time. The input voltage is a low amplitude sine signal. The left plots represent simulation data, while the right plots show experimental data. The rotor angle is wrapped between $[-\pi, \pi]$. The performance of the KF using simulation data is close to the MPF. However, this is insight does not extend using experimental data.

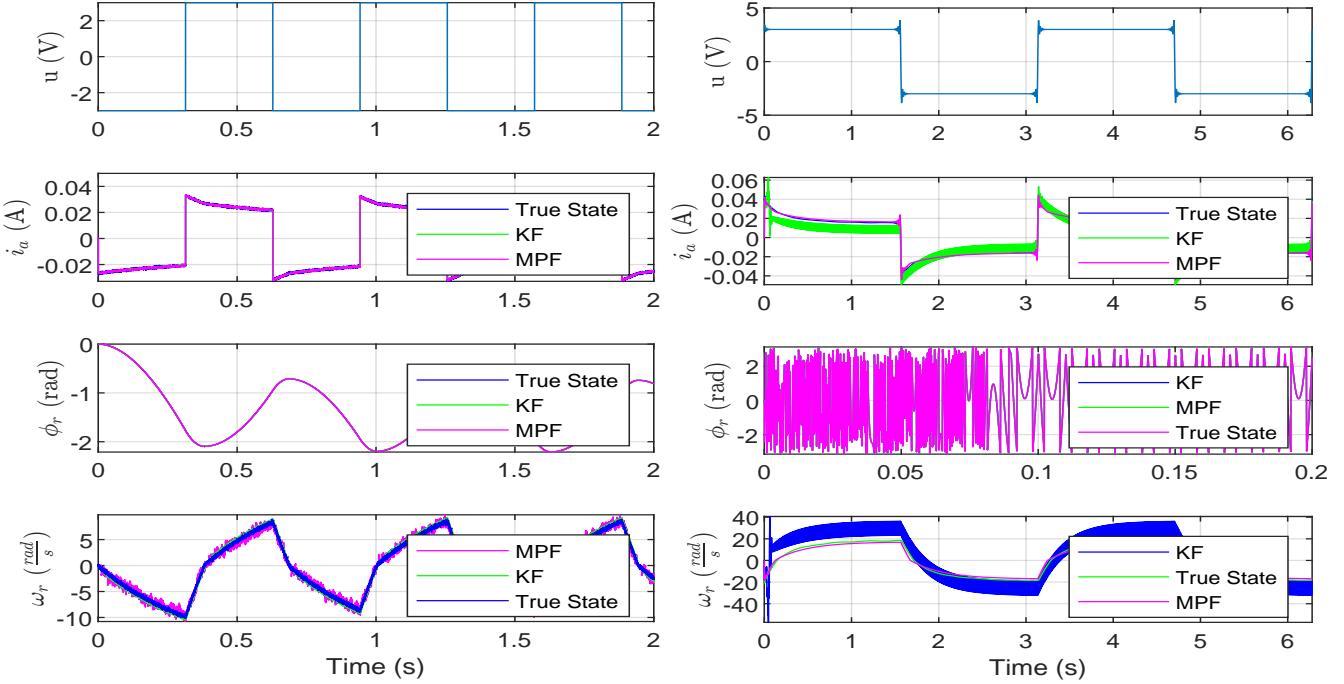


Fig. 3. Comparison of KF and the best-performing nonlinear estimator, MPF, performance in tracking the true state over time. The input voltage is a low amplitude square signal. The left plots represent simulation data, while the right plots show experimental data. The rotor angle is wrapped between $[-\pi, \pi]$. The performance of the KF using simulation data is close to the MPF. However, this is insight does not extend using experimental data.

Furthermore, in both datasets, the MPF consistently delivers the highest accuracy averaged over all states, closely followed by the UKF. Contrary to the simulation results, the incorporation of static friction using the experimental data achieves

higher accuracy, particularly in the rotor speed equation with less noise. The increased noise observed in the KF's performance can be attributed to its neglect of static friction. To maintain a low estimation error, the process uncertainty is

TABLE VI
COMPARISON OF THE COMPUTATIONAL TIME OF EACH ITERATION FOR
THE ESTIMATORS USING EXPERIMENTAL DATA.

Comput. Time	KF	EKF	UKF	PF	MPF
Max Time t_{max} [ms]	0.36	1.23	2.41	5.21	4.09
Avg Time t_{avg} [ms]	0.0017	0.0029	0.011	0.146	0.084

increased for the current and rotor speed, which consequently amplifies the noise in the estimates.

The key finding of our project can also be seen in Figures 2 and 3. While in simulation, the KF performance is similar to the MPF, the MPF clearly outperforms the KF using experimental data. This is attributed to neglected nonlinear dynamics between the simulation model in Eq. 7 generating the simulation data and the real motor. In reality, DC motors experience hysteresis and thus friction is not memory-less as assumed in our model generating the simulation data.

Based on the experimental results, the added computational complexity introduced by the UKF or MPF is favored as opposed to the KF.

VI. SUMMARY

In conclusion, a comparative analysis of state estimation techniques for a DC motor was conducted in this project. The analysis focuses on the impact of static friction on the precision, robustness and computational load. Five methods—KF, EKF, UKF, PF, and MPF—are evaluated using both simulation and experimental data from a MAXON S 2326 motor. Our key contributions are the incorporation of static friction into state estimation, a comprehensive comparison of estimation techniques, and a dual validation framework combining simulation and experimental results.

Simulation results show that the improved accuracy of nonlinear estimators (EKF, UKF, MPF) over the linear KF did not justify the increased computational complexity. However, experimental results demonstrated that nonlinear estimators, particularly the UKF and MPF, outperformed the KF in capturing rotor speed dynamics, justifying their computational cost in real-world applications. These findings confirm the importance of incorporating static friction, especially in low-speed regions and position control applications, where precise tracking is of significant importance.

For future work, we suggest extending the friction model to dynamic models (e.g. LuGre, Dahl) for increased accuracy. In addition, further studies could explore whether the insights gained from this study extend to other electromechanical systems such as robotic manipulators.

REFERENCES

- [1] S. Srey, V. Chhour and S. Srang, "Lumped Parameter Estimation of a Low Cost DC Motor for Position Controller Design," 2021 International Conference on Advanced Mechatronics, Intelligent Manufacture and Industrial Automation (ICAMIMIA), Surabaya, Indonesia, 2021, pp. 1-6, doi: 10.1109/ICAMIMIA54022.2021.9807810.
- [2] C. A. Pérez-Gómez, J. U. Liceaga-Castro and I. I. Siller-Alcalá, "Non-Linear Modeling and Identification of a Permanent Magnet DC Motor," 2020 24th International Conference on Circuits, Systems, Communications and Computers (CSCC), Chania, Greece, 2020, pp. 189-194, doi: 10.1109/CSCC49995.2020.000041.
- [3] Gerasimos G. Rigatos, Particle and Kalman filtering for state estimation and control of DC motors, ISA Transactions, Volume 48, Issue 1, 2009, Pages 62-72, ISSN 0019-0578, <https://doi.org/10.1016/j.isatra.2008.10.005>.
- [4] O. Aydogmus and M. F. Talu, "Comparison of Extended-Kalman- and Particle-Filter-Based Sensorless Speed Control," in IEEE Transactions on Instrumentation and Measurement, vol. 61, no. 2, pp. 402-410, Feb. 2012, doi: 10.1109/TIM.2011.2164851.
- [5] H. Lv, G. Wei and Z. Ding, "UKF — Based for sensorless brushless DC motor control," 2014 International Conference on Mechatronics and Control (ICMC), Jinzhou, China, 2014, pp. 199-203, doi: 10.1109/ICMC.2014.7231547.
- [6] Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. Transactions of the ASME-Journal of Basic Engineering, 82(Series D), 35-45.
- [7] S. Praesomboon, S. Athaphaisal, S. Yimman, R. Boontawan and K. Dejhan, "Sensorless speed control of DC servo motor using Kalman filter," 2009 7th International Conference on Information, Communications and Signal Processing (ICICS), Macau, China, 2009, pp. 1-5, doi: 10.1109/ICICS.2009.5397682.
- [8] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," in Proceedings of the IEEE, vol. 92, no. 3, pp. 401-422, March 2004, doi: 10.1109/JPROC.2003.823141.
- [9] V. Šmídl and Z. Peroutka, "Marginalized particle filter for sensorless control of PMSM drives," IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society, Montreal, QC, Canada, 2012, pp. 1877-1882, doi: 10.1109/IECON.2012.6388915.
- [10] J. C. Martinez-Rosas and L. Alvarez-Icaza, "Adaptive compensation of dynamic friction in an industrial robot," 2008 IEEE International Conference on Control Applications, San Antonio, TX, USA, 2008, pp. 1145-1150, doi: 10.1109/CCA.2008.4629664.
- [11] Papageorgiou, D., Blanke, M., Niemann, H. H., & Richter, J. H. (2020). Online friction parameter estimation for machine tools. Advanced Control for Applications: Engineering and Industrial Systems, Article e28. <https://doi.org/10.1002/adc.2.28>
- [12] Dan Simon. 2006. Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches. Wiley-Interscience, USA.
- [13] Thrun, S., Burgard, W., & Fox, D. (2005). Probabilistic robotics (Intelligent robotics and autonomous agents). The MIT Press.
- [14] K. Nielsen, C. Svahn, H. Rodriguez-Deniz and G. Hendeby, "UKF Parameter Tuning for Local Variation Smoothing," 2021 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), Karlsruhe, Germany, 2021, pp. 1-8, doi: 10.1109/MFI52462.2021.9591188.
- [15] T. Schon, F. Gustafsson and P. . -J. Nordlund, "Marginalized particle filters for mixed linear/nonlinear state-space models," in IEEE Transactions on Signal Processing, vol. 53, no. 7, pp. 2279-2289, July 2005, doi: 10.1109/TSP.2005.849151.

VII. APPENDIX

A. Derivation of transfer function from applied voltage u to rotor speed ω_r

Neglecting the armature inductance in Eq.1 leads to

$$u = R \cdot i_a + k_t \cdot \omega_r. \quad (22)$$

Solving Eq. 22 for i_a and substituting the expression in Eq. 1 while neglecting the static friction term yields

$$\dot{\omega}_r \cdot J_{tot} = k_t \cdot \frac{u - k_t \cdot \omega_r}{R} - d_m \cdot \omega_r. \quad (23)$$

Applying the Laplace transformation to the above equation and collecting the terms on each side yields

$$\begin{aligned} \omega_r(s) \cdot \left(s \cdot J_{tot} + \left(d_m + \frac{k_t^2}{R} \right) \right) &= \frac{k_t}{R} \cdot u(s), \\ G(s) = \frac{\omega_r(s)}{u(s)} &= \frac{k_t/R}{s \cdot J_{tot} + \left(d_m + \frac{k_t^2}{R} \right)} = \frac{\frac{k_t}{R \cdot d_m + k_t^2}}{s \cdot \frac{J_{tot}}{d_m + \frac{k_t^2}{R}} + 1}, \end{aligned} \quad (24)$$

which corresponds to the transfer function in Eq. 5.

B. Marginalized Particle Filter System Decomposition

The exact system matrices for the mixed nonlinear and linear state space decomposition shown in Eq. 19 are obtained by equating the expressions with Eq. 13 and utilizing the following state decomposition

$$x_k^l = \begin{bmatrix} i_a^k \\ \phi_r^k \end{bmatrix}, \quad x_k^n = \omega_r^k. \quad (25)$$

The system matrices are

$$\begin{aligned} \mathbf{f}_k^n(x_k^n) &= \left(1 - \frac{d_m \cdot \delta_t}{J_{tot}} \right) \cdot x_k^n - \frac{\text{sgn}(x_k^n) \cdot \delta_t \cdot \tau_c}{J_{tot}} \\ \mathbf{A}_k^n(x_k^n) &= \begin{bmatrix} \frac{k_t \cdot \delta_t}{J_{tot}} & 0 \end{bmatrix} \\ \mathbf{G}_k^n(x_k^n) &= 1 \\ \mathbf{f}_k^l(x_k^n) &= \begin{bmatrix} -\frac{k_t \cdot \delta_t}{L} \cdot x_k^n \\ \delta_t \cdot x_k^n \end{bmatrix} \\ \mathbf{A}_k^l(x_k^n) &= \begin{bmatrix} 1 - \frac{R \cdot \delta_t}{L} & 0 \\ 0 & 1 \end{bmatrix} \\ \mathbf{G}_k^l(x_k^n) &= I_{2 \times 2} \\ \mathbf{B}(x_k^n) &= \begin{bmatrix} \frac{\delta_t}{L} \\ 0 \end{bmatrix} \\ \mathbf{h}_k(x_k^n) &= 0 \\ \mathbf{C}_k(x_k^n) &= [0 \quad 1] \end{aligned} \quad (26)$$

C. Guidelines for Initial States Selection

All filters require prior knowledge of the states. Physical insight of the system along with the initial measurement of the position and initial applied voltage can be used to provide good initial estimates.

For i_a , steady-state conditions are assumed, leading to the estimate $i_a^0 = \frac{u}{R_o}$. The motor is presumed stationary; however, if the motor is rotating, i_a will be lower. To account for this uncertainty, the variance of the initial state is adjusted accordingly.

For ϕ_r , the initial rotor angle can be directly estimated from the position measurement, with its variance corresponding to the sensor variance.

Finally, for ω_r , steady-state conditions are again applied, resulting in $\omega_r^0 = \frac{k_t i_a^0}{d_m}$. This state is typically the least accurate due to its dependency on the current estimate. As a result, a higher variance is assigned to this initial state to reflect the uncertainty.