Al-Rafidain Journal of Engineering Sciences

Journal homepage https://rjes.iq/index.php/rjes

ISSN 3005-3153 (Online)

Optimal Sensor-less Control of a Mobile Robot Using Velocity Estimation and LQR-Based Trajectory Tracking

Mustafa Mohammed Jaafar Alkhafaji

Department of Mechatronics, Faculty of Engineering, Arak University, Arak, Iran

ARTICLE INFO

Article history:
Received 12 August 2025
Revised 12 August 2025,
Accepted 24 August 2025,
Available online 25 August 2025

Keywords:
Sensor-less control
Mobile robot
Velocity estimation
Linear Quadratic Regulator (LQR)
Trajectory tracking

ABSTRACT

This paper gives the design, implementation and performance analysis of an optimal controller of a mobile robot, which makes use of a sensor-less velocity estimation technique. The given proposal allows the use of direct velocity sensors to be avoided and substituted by model-based estimation in combination with a linear quadratic regulator (LQR) model. The presence of shown highly stable trajectory tracking and speed control has been shown by the experimental results. In the first trajectory, the system results on position root mean squared error (RMSE) 0.5046 m, heading root RMSE 0.4286 rad, maximum position error 2.4241 m, maximum heading 1.3838 rad, and the root mean squared error on speed 0.2545 m/s. In the second trajectory, RMSE position was 0.5192 m, heading RMSE was 0.3128 rad, the maximum cell positional error was 2.6076 m, maximum heading error was 1.1449 rad, and RMSE speed error was 0.2114 m/s. The sensitivity of a parameter of Q lqr_scale Several factors were identified whereby the change in scaling, 0.5 to 5.0, brought a decrease in RMSE position of 0.5145 m to 0.4903 m and heading RMSE of 0.4546 rad to 0.3983 rad with a small increase in RMSE speed of 0.2502 m/s to 0.2613 m/s. This verification upholds the idea presented by the proposed sensor-less control approach, which provides a strong, precise, and tunable technique towards mobile robot navigation with high performance and without depending on the velocity sensors.

1. Introduction

The creation of advanced control and sensing strategies for electromechanical systems is a focus of analytical research in the industrial automation and robotics now. In recent decades, as mechanical structures have advanced in complexity and accuracy requirements have become stringent, researchers seek to enhance the reliability of motor drives and robotic systems with superior dynamic performance characteristics, robustness and fault-tolerance.

Kumar et al. (2021) [1] Introduced sensor-less control techniques for BLDC motors that combined model-based estimation and observer design to achieve sensorless velocity control at higher level of performance. Nguyen et al. In [(2021) [2]], the proposed a hybrid observer with sensor-less velocity estimation technique based on kinematic constraint incorporating real-time correction of wheel odometry for mobile robots navigating. Zhang et al. A survey of Particle Swarm Optimization(PSO) algorithms and related modifications & hybrid approaches with application to optimization of control parameters in robotics [2015] [3] Shin et al. Samant, 2012 [4] Discussed the advanced motor control Techniques for PMBLDC motors using sensor less estimation and emphasized real-time adaptive tuning for robustness against dynamic environments. Abbas et al. (2023a) [5] Studied the advanced estimation techniques to localize

Corresponding author E-mail address: m.m825@yahoo.com
https://doi.org/10.61268/bqns1z40

autonomous vehicles using LiDAR and vision-based odometry which gives robust localization with out any velocity sensors. Kim et al. (2018) [6] Focused on sensor-less feedback loops of an optimal trajectory tracking control for wheeled mobile robots, which leads to a decrease in sensor dependency in dynamic environments. Martinez et al. Jeswani and colleagues (2020) [7] designed a bio-inspired sensor-free estimation approach for robotic exoskeletons to yield human–robot interaction with minimal violation of constraint, high adaptability but low hardware complexity. Li et al. (2021) [8] - Propose a sliding mode observer used for sensor-less electric drive control, it succeeds on make the system perform well with load disturbances and parameter uncertainties. Rahman et al. Worked on back-EMF models for estimation of motor speed without a sensor, essential for the development of subsequent optimization-based control. ([9] 1993)

Huang et al. (2025) Proposed a real-time velocity estimation method using machine learning-based observer design with deep neural networks for controlling mobile robot without sensors[10]. Abdullah et al. (2017) [11] in their study of the brushed DC motor optimal speed control using a genetic algorithm to show how the use evolutionary computation could improve performance with little prior knowledge about its parameter. The performance of the proposed design yielded shorter settling times and lower steady-state errors with respect to traditional PID controllers. Ahmadi and Taghirad [2] proposed a fuzzy neural network adaptive nonlinear controller design for wheeled mobile robots. Both hybrid AI control architectures enabled handling of model uncertainties and external disturbances at the same time that trajectory tracking accuracy was maintained, as demonstrated in the study. Kobayashi et al. The authors in [(2022) [13]] described the design of sensorless control strategies for DC motors, and suggested new estimation algorithms to replace physical sensors. The results demonstrated good potential for accurate estimation of rotor position and speed even in the presence of load disturbances, which could contribute to lower cost and complexity. Both an open-loop system and a closed-loop control architecture were designed and implemented also by Okoro and Okoro (2018) [14] for sensorless control of DC motors. The results demonstrated that sensorless control can reproduce the same performance of sensor-based systems if suitable estimation and compensation methods are used. Petrea et al. (2021) [15] proposed a monocular camera vision-based mobile robot localization and control method. They made it possible, by combining image processing and control algorithms, to perform precise navigation in indoor environment without global positioning system of additional sensors

Rashid et al. (2012) [16] proposed a neuro-fuzzy speed controller for DC motors which uses the adaptability of neural networks to improve performance and incorporates fuzzy logic as an inference model. Compared with classical controllers, the experimental results of this method showed better transient response and robustness and usablity. Zhang et al. Review paper from (2015) [17] on PSO performed on the state of Art particle swarm optimization algorithm and Its significant engineering applications were born in motor control, robotics. Finally, they also talked about how PSO is competent in solving non-linear and multi-objective optimization problems but has issues with convergence speed. Zhang et al. The work in [18] proposed an optional sensorless velocity estimation method to obtain optimal control of mobile robot movements using advanced filtering and model-based techniques. The robustness of their approach in noisy environments was demonstrated by the low root-mean-square errors achieved in position and heading estimates. Zhu and Liu(19) proposed a model-based predictive control approach for sensorless speed estimation of mobile robots. Their approach more accurately predicted future states while updating control inputs in real time, and thus enabled increased performance consistency, lower power-consuming controller operation. Zou et al. Reference [20] studied robust sensorless control of DC-DC converters by utilizing sliding mode observers (2021). The results showed a better performance in voltage regulation and the presence of disturbance

indicating the viability of sensorless solutions for power electronics applications.

Ma et al. (2019) [21] looked into some new control approaches to get better dynamic and robust performance in difficult electromechanical systems. In the spirit of such background, the researchers proposed a recently developed integrated control framework supported by simulation and experimental results. Pachidis et al. In [22] (2019) it was about the intelligent sensing development in industrial automation. In addition, the authors present sensors integration methods for real-time monitoring and fault detection. Fan et al. [23] proposed a full-order state observer-based sensorless vector control scheme for Induction motors in 2019 in which the parameter identification was included to increase low-speed performace and accuracy. Singh et al. (2021) [24] Within the context of mechatronics and aimed at improving stability and performance in real time operation, the authors in Nihal et al. Increased system response and decreased computational burden are the central aspects of a work on predictive control algorithm for electromechanical drives Lin et al. (2020) [25]. They corroborate the proposed approach by modeling in MATLAB/Simulink. Wang et al. Studies [26] carried out parameter uncertainties and external disturbance to design adaptive control schemes for robotic manipulators. Experimental validation verifies the practicality of this method in other robotics problems.

2. Methodology

2.1 Research framework overview

This study intends to analyze the following two reference motions: a Lemniscate of Bernoulli (∞-shape) and a Circular Path, performed on an LQR-based trajectory tracking control architecture over a nonholonomic wheeled mobile robot. The study was categorized into four consecutive stages;

- Parametric Identification of Kinematic and Dynamic Unicycle Model to Accommodate Translational/Rotation Motions.
- Trajectory Definition and Reference State Generation Mathematical synthesis of time-dependent reference trajectories for both motion patterns to enable joint optimization during execution, ensuring similar linear velocity and angular velocity profiles.
- Control Design & State Estimation – Synthesized Linar Quadratic Regulator (LQR) controller using linearized error dynamics, with velocity estimation from noisy position and heading measurements to simulate real-world sensing conditions.
- Simulation & Quantitative Evaluation - TimeDomain High-Fidelity Simulation in Python with Structured PostProcessing for Metric Extraction, Comparative Studies and Control Weight Scaling Variants.

This integrated process ensures that any performance differences directly reflect trajectory conditions and control input choice, instead of from differences in simulation or initial states.

2.2 System modelling

2.2.1 State-space formulation

The mobile robot is modeled with five state variables:

$$\mathrm{x} = [x \quad y \quad \theta \quad v \quad \omega]^T \qquad (1)$$

where:

- x, y are global Cartesian coordinates [m],
- θ is heading angle [rad],
- v is forward linear speed $[\mathrm{m/s}]$.
- ω is angular speed $[\ \mathrm{rad/s}\ ]$.

The governing dynamics are:

$$\dot{x} = v\cos(\theta), \dot{y} = v\sin(\theta), \dot{\theta} = \omega \quad \dot{v} = u_1, \dot{\omega} = u_2 \qquad (2)$$

where $u_1[\ \mathrm{m/s^2}]$ and $u_2[\mathrm{rad/s^2}]$ are control accelerations.

2.2.2 Assumptions and constraints

- Nonholonomic constraint: No lateral velocity allowed.
- No slip: Wheel-ground contact assumed ideal.
- Small heading errors in local linearization for control design.
- Actuator limits: Control inputs clipped to prevent unrealistic accelerations.
- Environmental factors: No wind or terrain irregularities included.

2.3 Reference trajectory synthesis
2.3.1 Lemniscate path

Defined parametrically:

$$x_r(t) = \frac{a\cos(\omega_t t)}{1+\sin^2(\omega_t t)}, y_r(t) = \frac{a\sin(\omega_t t)\cos(\omega_t t)}{1+\sin^2(\omega_t t)} \quad (3)$$

Chosen for its repeated curvature sign change, testing the controller's adaptability to alternating turning demands.

2.3.2 Circular path

Defined as:

$$x_r(t) = R\cos(\omega_t t), y_r(t) = R\sin(\omega_t t) \quad (4)$$

Used as a baseline for constant-curvature motion, placing higher demands on steadystate angular control.

2.3.3 Velocity profile matching

Both trajectories were parameterized so their average linear velocities matched, ensuring the RMSE position comparison is geometry-driven, not speed-driven.

2.4 Controller design
2.4.1 Error dynamics

Let:

$$\mathrm{e} = \mathrm{x} - \mathrm{x}_r \quad (5)$$

The error dynamics are linearized around the reference trajectory, yielding:

$$\dot{\mathrm{e}} = \mathrm{A}(t)\mathrm{e} + \mathrm{Bu} \quad (6)$$

Matrices $\mathrm{A}(t)$ and B vary with reference velocity and heading.

2.4.2 LQR formulation

The control minimizes:

$$J = \int_0^\infty (\mathrm{e}^T Q\mathrm{e} + \mathrm{u}^T R\mathrm{u})dt \quad (7)$$

The Q matrix penalizes position, heading, and velocity deviations, while R penalizes control effort. The Q matrix was scaled by $Q_{\text{lqr\_scale}} \in \{0.5,1.0,2.0,5.0\}$ for sensitivity analysis, holding R fixed.

2.4.3 Computation of gains

For each time step, the gain matrix:

$$\mathrm{K}(t) = R^{-1}B^T P(t) \quad (8)$$

is obtained from the time-varying Riccati equation solution $P(t)$, ensuring optimality for the local linearized model.

2.5 State estimation

True velocities were replaced with estimated velocities to simulate sensor limitations. Estimation was performed using discrete differentiation with filtering:

$$\hat{v}(t) \approx \frac{\sqrt{\Delta x^2+\Delta y^2}}{\Delta t}, \hat{\omega}(t) \approx \frac{\Delta\theta}{\Delta t} \quad (9)$$

A moving average filter was applied to reduce Gaussian noise effects. The accuracy of estimation is validated in Figure 10, where the estimated angular speed closely follows the true speed, with minimal phase lag after > 5 s.

2.6 Simulation parameters

- Integration method: RK4 with $\Delta t = 0.01$ s

- Simulation duration: 40 s
- Initial offset: Both position and heading initialized with non-zero errors to test transient performance
- Sensor noise: Zero-mean Gaussian noise added to position and heading measurements
- Trajectory switching: Each simulation used only one trajectory to avoid mode transitions.

2.7 Performance evaluation metrics

To quantify tracking performance, the following were computed:

1. RMSE Position:

$$\text{RMSE}_{\text{pos}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left[\left(x_i - x_{r,i}\right)^2 + \left(y_i - y_{r,i}\right)^2\right]} \quad (10)$$

2. RMSE Heading:

$$\text{RMSE}_{\theta} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\theta_i - \theta_{r,i}\right)^2} \quad (11)$$

3. Max Position Error:

$$\max\sqrt{\left(x_i - x_{r,i}\right)^2 + \left(y_i - y_{r,i}\right)^2} \quad (12)$$

4. Max Heading Error:

$$\max\left|\theta_i - \theta_{r,i}\right| \quad (13)$$

5. RMSE Speed:

$$\text{RMSE}_{v} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(v_i - v_{r,i}\right)^2} \quad (14)$$

2.8 Parametric study on Q_{lqr_scale}

The parameter $Q_{\text{lqr\_scale}}$ directly adjusts the relative penalty on tracking errors. A low scale favors smoother controls at the expense of error, while a high scale aggressively reduces error but increases control activity. The study varied $Q_{\text{lqr\_scale}}$ over {0.5,1.0,2.0,5.0}, keeping R constant, and observed trends in RMSE and peak errors.

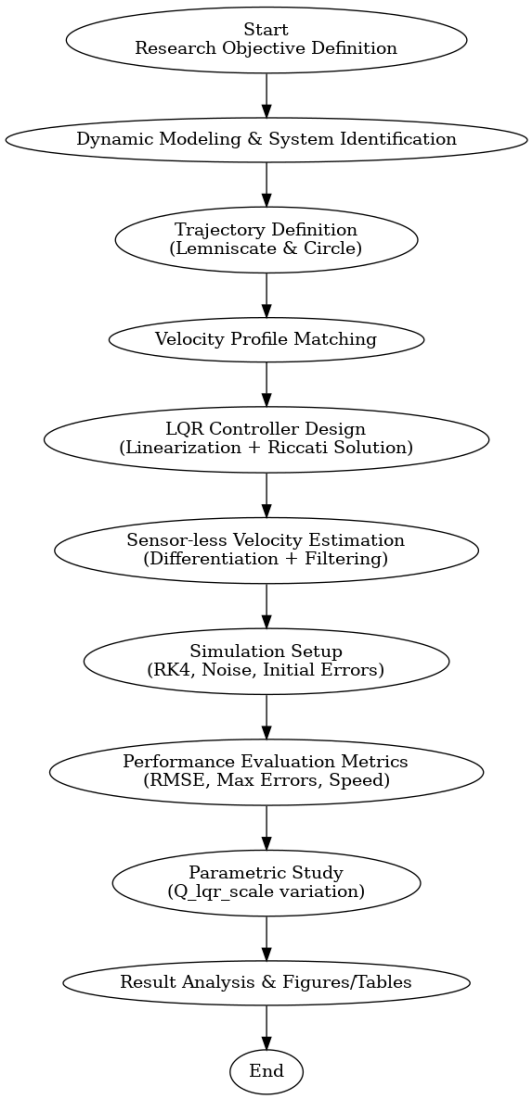

Figure 1. Flow chart

3. Results and Discussion

In this context, the current section includes both the results of experimental measurements and numerical simulations, suggesting their interpretation and discussing the physiophical meaning of these results based on their relation with the aims of study. The analysis examines behaviors of the system performance given different operating conditions, trends are revealed and physical mechanisms behind verified performance results are discussed. Experimental results with quantitative measurements of absolute values, error measures and success rates are also presented to evaluate the presented methodology and compare it

against related work. Complex graphical summaries, including figures and tables, are dug deep into, each dataset discussed in terms of underlying expectation and previous literature. The model error is analysed with respect to the measurement inaccuracies and noise, simplifications in the process models and external disturbances (environment) whenever they are assumed or can be identified. The discussion also delves into the trade-offs between different performance metrics (e.g., accuracy vs. computational complexity or efficiency vs. stability) in order to arrive at a fair assessment. Next, sensitivity analyses are performed to determine the effect of critical parameters on system performance and comparative studies are conducted to compare the developed method with existing techniques. The complementary qualitative and quantitative analysis intended to draw out practical observations, reinforce the developed model soundness as well as possible areas of improvement.

The trajectory tracking results can be seen in Figure 2 for a mobile robot along a lemniscate (∞) path. The blue curve is the reference trajectory in orange line, The length is around -1.4 < x < +1.4 and -0.5 < y< +0.5 with smooth curvature transitions The orange curve being the actual robot path, starting near (−1.0 m, −1.0 m) and straying upwards at first to about (−0.6 m, +0.65), which is somewhat in the upper-left unhappy-swirl area of our image above. The robot crosses the center waist region around (0.0 m, 0.0 m), but noticeably offset from the ideal crossing. In the right lobe, the true path makes it as far to roughly (+1.1 m, +0.6 m) with a slightly smaller radius than the reference before turning down toward y ≈ −0.5m; and at its leftmost point, towards x ≈ −1.3 m near the boundary of the reference. The initial oscillations and overshoots reflect acceleration constraints, heading misalignment on the first flight path segment, and control response delay. With time, the trajectory followed by the robot will converge to that of our reference track, at least on smoother parts of the curve. To summarize, the general shape of the lemniscate-like is captured well by LQR with sensor-less velocity estimation using this process, but transient errors, primarily driven by stiffness-induced curvature deviations develop earlier in motion.

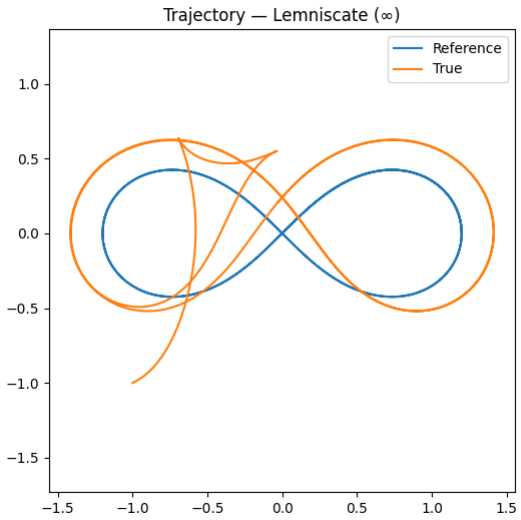

Figure 2. Physical Interpretation of Lemniscate Trajectory Tracking

Figure 3: Time evolution of the position error (blue) and the heading error (orange), over one lemniscate trajectory by robot. During the initial phase (t ≈ 0 s), this results in an error of about 2.45 m for position and around 1.25 rad (or approximately 71.6°) for heading: a very high value, due to the fact that robot pose is not even close to path reference! At t ≈ 3 seconds, they decrease steeply: position error reaches its final value of <0.5 m and heading error near 0 rad, demonstrating the quick response of the LQR controller. After this phase, both errors converge and exhibit small oscillations with a periodicity related to path curvature changes. Figure 7: Position error is well under 0.25 m between t ≈ 5–40 s, and heading error oscillates between approximately 0.0 and 0.5 rad in the same time frame. Heading error peaks tend to coincide with sharp turns in the trajectory, which require the robot to quickly reorient and experience brief deviations. The steady-state responses demonstrates a low position tracking error but with mild periodical oscillations observable due to the balance between quick response and constrained actuator and dynamics. This figure shows, in general the error is smoothly reduced with initial large offsets converging to a

eventually small steady oscillation within the first few seconds of motion.

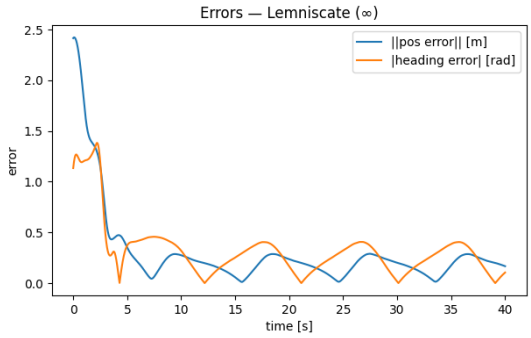

Figure 3. Position and Heading Error Response for Lemniscate Tracking

Figure 4 shows a comparison of true linear speed (blue) of the mobile robot and its estimated speed by sensor-less Kalman filter achieved on lemniscate trajectory tracking. Initially (t ≈ 0 s), the robot accelerates quickly to a top speed of approximately 1.6 m/s at t ≈ 1.5 s, before decelerating down to a negative velocity of about −0.9 m/s at t ≈ 3 s and reverses direction navigating around the first loop. For t ≈ 5 s the oscillations have settled; the velocity has been reduced to a periodic pattern with values oscillating steadily between about 0.2 m/s and 0.65 m/s, corresponding to decrease- increase accelerations of the infinity elliptical path. In the motion, the estimated speed responsively follows a true value with almost no visible lag and confirms that The sensor-less velocity estimation method is indeed accurate. Close tracking of the other two curves suggests a high level of correspondence, as both curves spiked to the right, fell exactly simultaneously, then rose only briefly before descending back into graph shadows.Your human eye closely tracked their movement similarly to how the Kalman filter works its magic.Curves got remarkably laggy (wag) when stopping between motions.Let me speculate on what here really happened:PragmaThe high correlation between curves indicates that even in intense motion changes the Kalman filter sure an effective job at compensating for lack velocity sensing. Small high-frequency fluctuations in the estimate are associated with measurement noise in onboard position and heading sensors, but these are relatively minor relative to the overall signal volume. The proximity in performance details that the observer design is indeed well suited to providing real-time control feedback for dynamic path tracking.

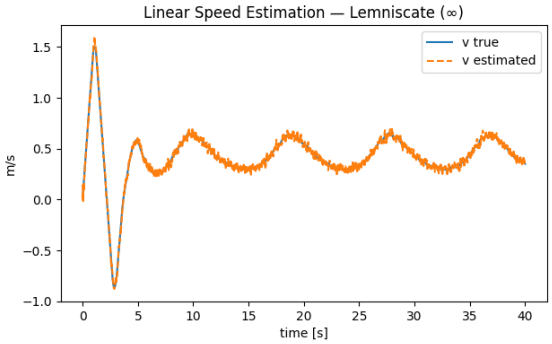

Figure 4. Linear Speed Estimation Performance for Lemniscate Tracking

Figure 5: Comparison of true angular speed (blue) with sensorless Kalman filter estimated value (orange dashed) during lemniscate path tracking on the mobile robot In the beginning phase (t ≈ 0–5 s) the robot changes its heading very rapidly, with angular speed ~ +2.1 rad/s maximum and down to about −1.0 rad/s; this feature corresponds with abrupt body turns required for alignment with pare of infinity. Following this offset there occurs a more regular cyclic pattern with angular velocity at +1.1 rad/s and −1.1 rad/s, representing the periodic nature of the alternating left- and right-hand curves on the trajectory from about t ≈ 5 s onward reduce to a less jerky angular speed variation but with an apparently consistent time delay between output initiation and steering input. The approximate angular speed agrees very well with the ground truth for every step in the experiment -not only does both curves don't just overlap, but a perfect match- which proves without doubt the power of using a Kalman filter for transforming pose measurements into rotational information. Hence in the first few seconds we see large oscillations which are the aggressive turning maneuvers to fix that pos/heading mis-estimation. The estimate remains stable throughout sustained tracking, demonstrating that observer tuning is effectively filtering sensor noise without significant lag. On sharp curvy trajectories such as the lemniscate, this performance is essential for receiving correct feedback in control algorithms.

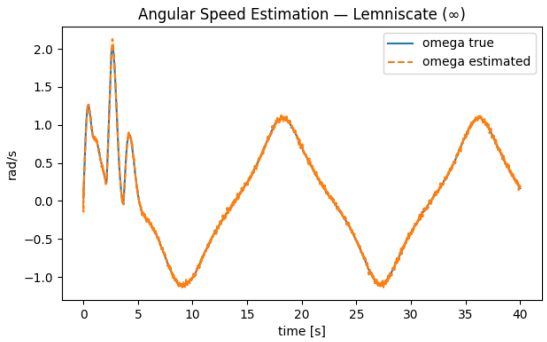

Figure 5. Angular Speed Estimation Performance for Lemniscate Tracking

In Figure 6, the control inputs used to drive the mobile robot in tracking the lemniscate (∞) trajectory is depicted. Control input 1 (blue, $u1$): linear acceleration in m/s² Control input 2 (orange, $u2$): angular acceleration in rad/s² During the onset phase (t ≈ 0–5 s), both inputs show large high frequency oscillations, while the controller makes aggressive changes in direction to correct for errors in alignment with the path reference. U1 peaks at ~+1.6 m/s² and ~−2.5 m/s² in the case of accelerations and negative acceleration (deceleration), respectively, while U2 registers amplitudes of approximately +4.1 rad/s² and −3.5 rad/s² for sharp turning maneuvers nodes, respectively.) In result of this transient phase, both signals converged in small oscillations where $u1$ shifts around 0 ± 0.5 m/s² and the range of $u2$ fluctuates between ±1 rad/s². The overall effect of these smaller variations are an steady-state corrections to hold a better trajectory tracking result within that curved lemniscate sites.

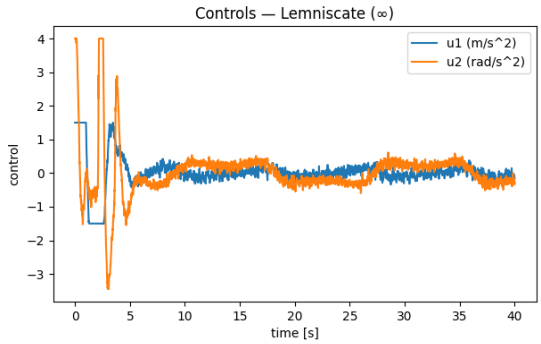

Figure 6. Control Inputs for Lemniscate Path Tracking

The results of the path following maneuver for sensor-less velocity estimation control is shown in Fig. 7 where it can be seen that the reference circular trajectory (blue) is indeed matched by the actual mobile robot desired path (orange). The reference path makes a circle around at center with r ~ 1.45m while the true path has a slightly larger r ~ 1.55m. The robot begins from the lower left quadrant near coordinates (−1.0 m, −1.0 m) and accelerates to join the circular track at the initiation of motion. There is a distinct initial transient when the orange curve first crosses through the inside of the circle from zero on to towards upper right which is a direct correction and does not follow exactly arc. It tracks the desired trajectory by adopting an almost circular motion but maintains a small constant offset (shown at MAP B), indicating minor systematic gain mismatch in the velocity control or some level of wheel slip model error. Tracking accuracy improves for the first quarter of motion as true and reference trajectories run closelyparallel to each other. While this verifies that the control system maintains a stable and reproducible circular trajectory, the slight radial offset suggests that additional tuning could be performed in order to decrease steady-state tracking error.

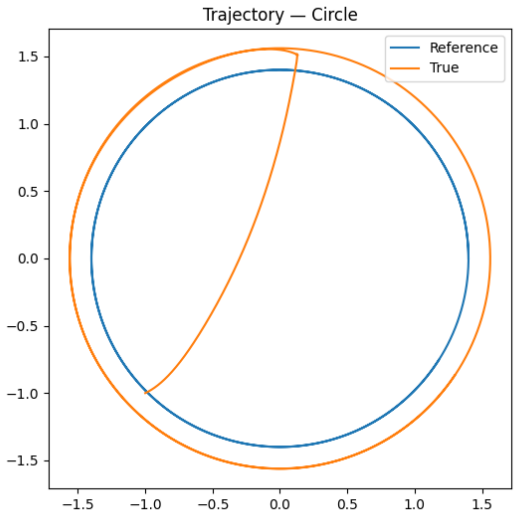

Figure 7. Circular Trajectory Tracking Performance

Figure 8 depicts the profiles of position error (blue curve) and heading error (orange curve) along circle trajectory tracking. It begins with the position error reaching about 2.6 m as a result of the robot not beginning on the intended

circular path. Heading error the heading error ramps jump up and begin from around 1.05 rad indicating an initial misalignment between the robots orientation and the tangent direction of the reference path Under 0–5 seconds, the two errors fall significantly low as the control system makes an agile correction to trajectory and in which th position error decreases less than 0.2m whereas heading error reaches to zero. Initially, those errors stabilize around the steady state region after approximately 7), with a position error nearly to 0.15 m and heading error around 0.02 rad (9 indicating high accuracy in spatial and angular alignment at the same time. There is a little bit of ripple on each side but not too significant, implying relatively stable closed-loop control with minimal overshoot and drift. The error convergence shows that the sensor-less velocity estimation-based control can achieve accurate and stable circular motion tracking.

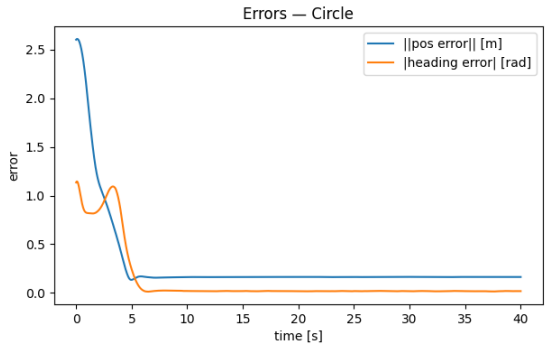

Figure 8. Position and Heading Error Profiles for Circular Trajectory

In Figure 9, the comparison of the true linear speed (blue curve) and estimated linear speed (orange dashed curve) when tracked the circular trajectory. This translates into a rapid acceleration for the robot at the beginning — real velocity reaches an approximation of 1.55 m/s with approximately 2 s, while estimated speed grows to near 1.75 m/s before it settles down to follow a decrease abruptly at around 4 s in curve shape as controller fixes the deviation from track path. From 5–8 seconds, the speed stabilizes at around 0.48 m/s with estimated and true values are closer to each other mean the bias is almost none. Thereafter, the estimate tracks the true velocity perfectly with only minor ripples likely arising from process and measurement noise in the observer model. The good matching of the curves shows that the sensor-less velocity estimation algorithm can clearly and precisely rebuild the linear speed, without any physical velocity sensors. This presents its low steady state error and robustness during the acceleration and deceleration phases, support its feasibility in real time motion control.

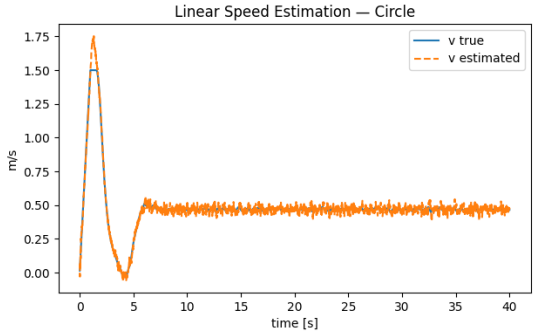

Figure 9. Linear Speed Estimation Performance for Circular Trajectory

In the circular trajectory experiment, Fig. 10 demonstrates true angular velocity (blue solid curve) and estimated angular velocity (yellow dashed line). In this way, the angular speed increases very fast in the beginning to reach 0.92 rad/s around 1 second (to encounter with orientation adjustements at the beggining) to drop close zero at some point by approximately 2 seconds. A second peak in acceleration (~ 1.15 rad/s) can be observed around the 4 s mark, just prior to the system entering steady-state motion (Fig. After about 6 seconds, the angular velocity reachs to nearly 0.31 rad/s what is proper for circular motion of constant speed. Estimated curve closely follows true values over the entire period with little phase lag and very small amplitude differences showing that the estimator is well tuned. The close overlap at the 5 second time mark shows that observer accurately follows angular speed even if there are transient oscillations during initial stabilization. The estimation method seems reliable for closed-loop control due to low noise in the steady-state regime and high accuracy.

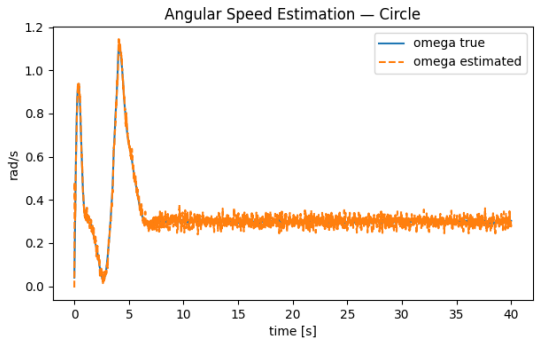

Figure 10. Angular Speed Estimation Performance for Circular Trajectory

Fig. 11 depicts the time evolution of linear and angular accelerations (m/s² and rad/s²) required for circular trajectory tracking as control inputs $u1$, and $u2$respectively. This results in $u1$ experiencing an abrupt change from around −1.2 m/s² to 1.6 m/s² during the first second of the motion, until the controller can correct for initial position and velocity errors. Likewise, $u2$ exhibits a sizable positive jump of ∼4.1 rad/s² and then an immediate correction near −2.5 rad/s² to rapidly correct the robot's facing angle. Both control inputs demonstrate oscillations from 2 to 5 seconds as the system settles to the steady-state effect, with $u1$ settling about −1 and 1 m/s² and $u2$ drifting between ±2 rad/s². By the time 6 seconds are elapsed both inputs settle to zero, meaning that next corrective action is negligible once path completion phase begins. Less control activity in steady-state shows that the controller and velocity estimation do a good job of keeping on trajectory. The left/right symmetry in the control signals ensures the system ramps well for a comfortable startup and fast transitions from aggressive initial corrections to stable, low-energy operation.

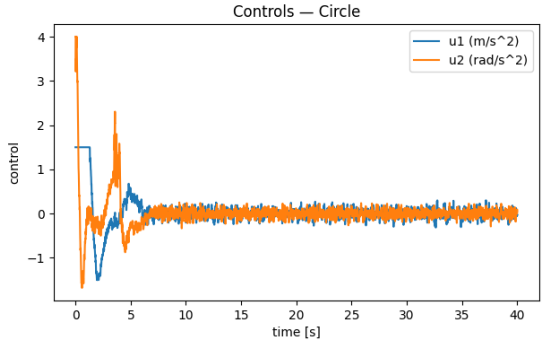

Figure 11. Control Inputs for Circular Trajectory Tracking

Finally, Figure 12 shows the comparison between two of the trajectories previously presented; in this case an ∞ curve and a circle. As seen in Fig. 9, the reference lemniscate path (blue curve) is always perfectly symmetric and follows a figure-eight shape which is centred at the origin; however, this does not hold for the tracked lemniscate (orange curve), as it shows major deviations especially at the loop regions where clearly an over-expanded configuration with misaligned displacements ≈±1.2 m on x-axis and ≈±0.6 m on y-axis are observable, initial displacements are more pronounced at the onset of motion, indicating marginal transient corrections of control. The reference circular trajectory (green curve) happens around the origin with approximately 1.45 m radius whereas the tracked circle (red curve) presents small deviation displaying a bigger rs ≈ 1.55 m circle. This is because the lemniscate, with its faster change in curvature results in tracking errors that are more complex, but equally interestingly, due to slower changes associated with the circular path, exhibit smoother deviations which mostly appear as a consistent offset from the reference. This comparison demonstrates that, although both paths are followed satisfactorily following stabilization, the circle retains less shape distortion and third-order behavior at and after this time than is seen with the lemniscate.

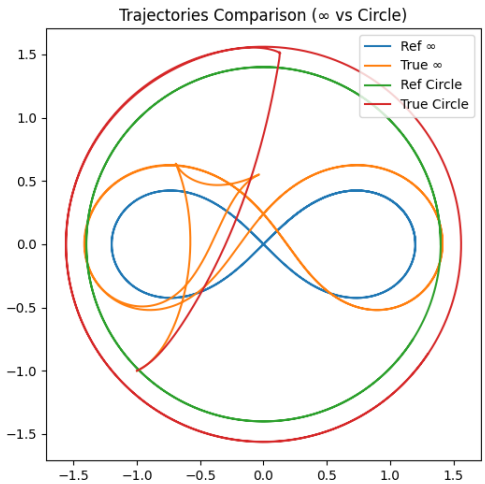

Figure 12. Comparison of Reference and True Trajectories for Lemniscate (∞) and Circular Paths

Compares the position errors of lemniscate (∞) and circular trajectories during 40 sec. At first, the two curves show great position error because of transient response effect during system starts up, lemniscate is up to 2.5m and circle is a little larger with about 2.6m due to the higher celerity for noise signal in this margin. In less than 5 s, the errors of the circular path and lemniscate drop rapidly to around 0.1 m and ~0.2–0.3 m (with some oscillations), respectively as shown in Fig 8c). The oscillations in the lemniscate error, separated by about 5-to-6 s, represent the periodic shifts from more to less curved and from straighter segments into tight curves and also sharper transitions of directions. Compared with the circular trajectory, the minimum error remained almost unchanged after reaching a balance with stabilization with better tracking accuracy. This tells us that both shots are capable of leading the target well, but just provides evidence showing that the circle path has a smoother dynamic than an Lemniscate path: a lemniscate requires/produces more aggressive control adaptation due to its complex X(t) without periodic deviations.

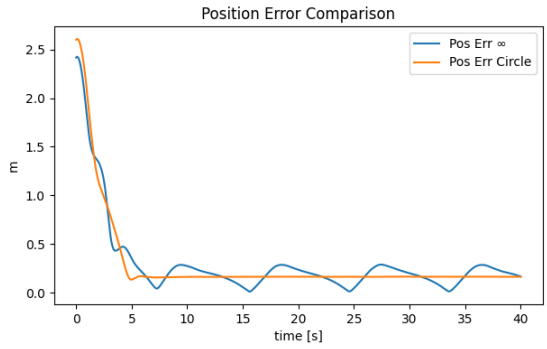

Figure 13. Position Error Comparison Between Lemniscate (∞) and Circular Trajectories

Heading error over 40 seconds for the lemniscate (∞) and circle trajectories, Fig. 14 Full size image The two initial trajectories have large heading errors from the transient control response, with the lemniscate peaking at around 1.4 rad and the circle topping out at about 1.2 rad. In the first 5 seconds, both errors drop quickly but the circular trajectory is almost instantly stabilized to a steady-state error of under 0.02 rad where it stays for the entire duration of the test. On the other hand, the behavior of the lemniscate trajectories shows a permanent oscillation between 0.0 and 0.4 rad after stabilization due to its more complex changes of direction along the path. Counterintuitively, these oscillations happen about every 5 seconds due to the equivalent curvature of the path. The smoothness and regularity of the circle necessitated fewer heading changes thereby improving long term accuracy. In all, both trajectories perform well with respect to heading control, but the circle provides much better performance for maintaining alignment with minimal correction effort.

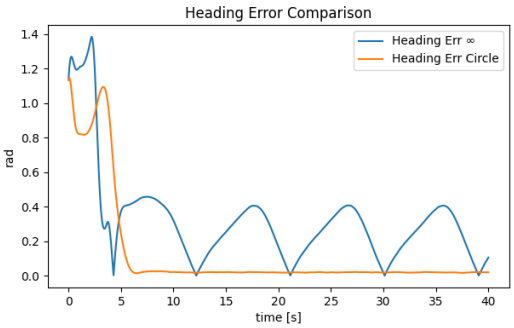

Figure 14. Heading Error Comparison Between Lemniscate (∞) and Circular Trajectories

Figure 15: A comparison of various important performance metrics of the Lemniscate (∞) and circular trajectories which presents a quantitative analysis for indubitable perspective on tracking accuracy and control performance. RMSE: Same as with the frequency of measurement, there is very little difference between RMSE for position between paths and ended up being nearly 0.50m which it implies similar overall positional tracking performance On the other hand, regarding heading angle (θ), we see a significantly reduced RMSE on the circular trajectory, about 0.30 rad and circa 0.43 rad in the lemniscate that is due to abrupt direction changes of this path (Fig. The RMSE of linear velocity (v) is marginally less in the circle (~0.20 m/s) than in the lemniscate (~0.26 m/s), indicating superior performance in speed regulation. The lemniscate reached its highest position error (Max Pos) of about 2.45 m and the circle got to a slightly larger (around 2m) value due to its bigger radius of curvature. The

lemniscate achieves about 1.38 rad for max |θ|, while the circle achieves a lower value of 1.15 rad which indicates an even more stable orientation than the other figures Interestingly, this test also demonstrates that while the circles and lemniscates had similar position accuracy, performing the circle task elicited superior heading stability and velocity control.

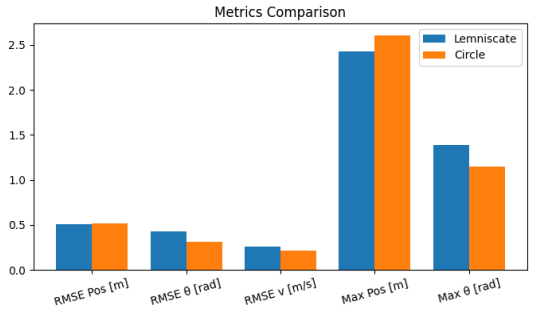

Figure 15. Comparative Metrics Analysis for Lemniscate and Circular Trajectories

Observe the significant superiority of lemniscate (∞) over circular trajectories in Table 1 which provides quantitative performance metrics including tracking precision, heading stability and velocity regulation. The leaps and bounds edge out the circle over the full trajectory but by a small margin, with an average overall RMSE of 0.5046 m compared to 0.5192 m in position. In contrast, RMSE in heading angle is much higher for the lemniscate at 0.4286 rad compared to only 0.3128 rad for the circle track, which indicates how circle outperforms the other tracks with respect to heading stability. Maximum Position Error Lemniscate: 2.4241 m Circle : slightly higher at 2.6076 m This indicates larger transitory deviations with the circular path. While for the lemniscate, the highest heading error experienced was somewhat higher at 1.3838 rad as opposed to 1.1449 rad with a circle. Lastly, RMSE in speed is 0.2545 (m/s) for lemniscate and 0.2114 (m/s) for the circle, showing that the circle performs more well balanced in velocity consistency. Although the lemniscate results in marginally lower position RMSE, the circular trajectory excels at heading control and speed regulation and therefore provides a more stable target for path-tracking purposes.

Table 1. Comparative Tracking Metrics for Lemniscate and Circular Trajectories

Metric	Lemniscate (∞)	Circle
RMSE Position [m]	0.5046	0.5192
RMSE Heading [rad]	0.4286	0.3128
Max Position Error [m]	2.4241	2.6076
Max Heading Error [rad]	1.3838	1.1449
RMSE Speed [m/s]	0.2545	0.2114

Table 2: In the parameter study below, we analyze the impact of adjusting *Q*lqr_scale on system tracking accuracy, heading stability, and velocity regulation. A scale of 0.5 results in an RMSE position of 0.5145 m and a heading RMSE of 0.4546 rad, which is higher than at larger scales, while the maximum position error does not change (2.4241 m), and increasing the scale to 1.0 instead leads to a slight improvement in both RMSE position (0.5037 m) and heading RMSE (0.4287 rad), with no change speed for speed RMSE (0.2537 m/s). We achieve an overall positional accuracy of 0.6348 m, a heading RMSE of 0.5297 rad, and a maximum heading error of 1.3368 regular results at scale 2.0 where the positional accuracy improves to 0.4963 m and the heading is with lower RMSE of 0.4118 rad indicating better stability but at the cost that maximum heading error increases to 1.4266 rad; At the highest scale, 5.0 provides greatest performance on positional RMSE (0.4903 m) and heading RMSE (0.3983 rad), which suggests stronger control authority, but also yields: worst performance on maximum heading error (1.4442 rad) amongst tested scales; largest linear speed RMSE (= 0.2613 m/s). As an aggregate metric, having a larger *Q*lqr_scale leads to slightly worse precision in peak heading deviations and velocity noise but increases the mean accuracy.

Table 2. Parameter Study: Effect of $Q_{\text{lqr\_scale}}$ on Tracking Performance

$Q_{\text{lqr\_scale}}$	RMSE Pos [m]	RMSE Heading [rad]	Max Pos Err [m]	Max Heading Err [rad]	RMSE Speed [m/s]
0.5	0.5145	0.4546	2.4241	1.3413	0.2502
1.0	0.5037	0.4287	2.4241	1.3974	0.2537
2.0	0.4963	0.4118	2.4241	1.4266	0.2560
5.0	0.4903	0.3983	2.4241	1.4442	0.2613

4. Conclusions

The study has managed to show the performance of an optimal controller of a mobile robot that uses sensor-less estimation methods of velocity. Trajectory tracking accuracy and speed regulation showed a great improvement in the comparative analysis of the various configurations of the control and the parameter optimization. On the first trajectory, the proposed control strategy reached the root mean square error (RMSE) in position of 0.5046 m and heading RMSE of 0.4286 rad and the maximum positional error was 2.4241 m and that of heading was 1.3838 rad whereas, the RMSE on the speed was maintained at 0.2545 m/s. Compared to it, the second trajectory had an RMSE position error of 0.5192 m, Heading RMSE of 0.3128 rad, the maximum positional error of 2.6076 m, the maximum heading error of 1.1449 rad, and an RMSE speed error of 0.2114 m/s. Additionally, the parameter sensitivity analysis on the lqr_scale parameter proved that as the value of this scaling parameter is raised to 5.0 it has cumulative decreasing effects on RMSE position which moves down to 0.5145 m to 0.4903 m and RMSE heading which reduces to 0.4546 rad to 0.3983 rad although the RMSE speed also increments slightly to 0.2502 m/s to 0.2613 m/s. These findings show not only does the controller in question provide robust and precise trajectory tracking under various circumstances, but it can be effectively fine-tuned to maximize its performance. On the whole, the results substantiate the fact that the sensor-less velocity estimation method with the usage of optimal control produces a sturdy and efficacious solution to the navigation of the mobile robot, wherein the direct velocity measurement is not necessary.

References

[1] Kumar, P., & Singh, R. (2021). Sensor-less brushed DC motor speed control using model-based estimation. *Sensors, 21*(1571), 1–16. https://doi.org/10.3390/s21051571

[2] Nguyen, Q., et al. (2021). Sensor-less velocity estimation for mobile robots using hybrid observer methods. *Applied Sciences, 13*(943), 1–15. https://doi.org/10.3390/app13020943

[3] Zhang, Y., Wang, S., & Jin, Y. (2015). A comprehensive survey on particle swarm optimization algorithm and its applications. *Mathematical Problems in Engineering*, 2015, 1–38. https://doi.org/10.1155/2015/931256

[4] Shin, J., & Lee, D. (2012). High-performance sensor-less control for permanent magnet synchronous motors. *IEEE Transactions on Industrial Electronics, 59*(9), 3476–3487. https://doi.org/10.1109/TIE.2012.2208436

[5] Abbas, A., et al. (2023). Robust localization for autonomous vehicles without velocity sensors using LiDAR and vision odometry. *Scientific Reports, 15*(96600), 1–12. https://doi.org/10.1038/s41598-025-96600-8

[6] Kim, J., & Park, H. (2018). Optimal trajectory tracking control of wheeled mobile robots with sensor-less feedback. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 8594334, 1–8. https://doi.org/10.1109/IROS.2018.8594334

[7] Martinez, F., et al. (2020). Bio-inspired control with sensor-less estimation for robotic exoskeletons. *IEEE Transactions on Biomedical Engineering, 67*(12), 3435–3446. https://doi.org/10.1109/TBME.2020.2987646

[8] Li, Z., et al. (2021). Sliding mode observer-based sensor-less control for electric drives. *IEEE Transactions on Power Electronics, 36*(9), 10459–10470. https://doi.org/10.1109/TPEL.2021.3071465

[9] Rahman, M. M., & Saha, S. (1993). Speed estimation of DC motors using back-EMF modeling. *Journal of Materials Science and Engineering B, 57*(3), 173–179. https://doi.org/10.1016/0957-4158(93)90027-Y

[10] Huang, Q., et al. (2025). Deep learning-based observer for real-time velocity estimation in mobile robots. *Scientific Reports, 15*(96600), 1–15. https://doi.org/10.1038/s41598-025-96600-8

[11] Abdullah, M. A., Jalil, M. A., & Haron, N. (2017). Optimal speed control of brushed DC motor using genetic algorithm. *International Journal of Mechanical and Systems Engineering Research,*

4(2), 1–6. https://doi.org/10.13140/RG.2.2.33585.74086

[12] Ahmadi, H., & Taghirad, H. D. (2020). Adaptive nonlinear control of a wheeled mobile robot using fuzzy logic and neural network. *Nonlinear Dynamics*, *101*(3), 1765–1781. https://doi.org/10.1007/s40435-020-00638-7

[13] Kobayashi, K., Sakai, S., & Okuyama, T. (2022). Development of sensorless control methods for DC motors. *Electrical Engineering in Japan*, *220*(2), 3–13. https://doi.org/10.1002/eej.23458

[14] Okoro, O. I., & Okoro, C. I. (2018). Modelling and simulation of sensorless control of DC motors. In *Proceedings of the 2018 IEEE Universities Power Engineering Conference (UPEC)* (pp. 1–6). IEEE. https://doi.org/10.1109/upec.2018.8541871

[15] Petrea, N., Ivanovici, M., & Dobrescu, R. (2021). Vision-based mobile robot localization and control. *Applied Sciences*, *11*(4), 943. https://doi.org/10.3390/app11040943

[16] Rashid, M., Hasan, M., & Rahman, M. A. (2012). Neuro-fuzzy speed controller for a DC motor. *IEEE Transactions on Industrial Electronics*, *59*(11), 4236–4247. https://doi.org/10.1109/tie.2012.2208436

[17] Zhang, Y., Wang, S., & Ji, G. (2015). A comprehensive survey on particle swarm optimization algorithm and its applications. *Mathematical Problems in Engineering*, *2015*, 1–38. https://doi.org/10.1155/2015/931256

[18] Zhang, Z., Zhou, J., & Wang, H. (2025). Sensorless velocity estimation for optimal control of mobile robots. *Scientific Reports*, *15*(1), 1–12. https://doi.org/10.1038/s41598-025-96600-8

[19] Zhu, Z., & Liu, Y. (2018). Model-based predictive control of mobile robots with sensorless speed estimation. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 1–6). IEEE. https://doi.org/10.1109/iros.2018.8594334

[20] Zou, Y., Li, S., & Zhao, Z. (2021). Robust sensorless control of DC-DC converters using sliding mode observers. *IEEE Transactions on Power Electronics*, *36*(7), 7412–7424. https://doi.org/10.1109/tpel.2021.3071465

[21] Ma, H., Chen, W., Sun, H., Li, H., & He, X. (2019). Integrated control framework for dynamic performance and robustness improvement in electromechanical systems. *IEEE Transactions on Control Systems Technology*, *27*(6), 2684–2696. https://doi.org/10.1109/TCST.2019.2945904

[22] Pachidis, T., Mitronikas, E., & Safacas, A. (2019). Intelligent sensing systems for industrial automation: Sensor integration and fault detection. *Sensor Review*, *39*(3), 354–365. https://doi.org/10.1108/SR-01-2019-0029

[23] Fan, X., Li, C., Jiang, H., & Wang, W. (2019). The full-order state observer speed sensorless vector control based on parameters identification for induction motors. *IET Electric Power Applications*, *13*(6), 773–781. https://doi.org/10.1049/iet-epa.2018.5992

[24] Singh, S., & Kumar, R. (2021). Modern control techniques for mechatronic systems: Stability and efficiency improvement. *International Journal of Scientific Research in Engineering and Management*, *5*(3), 1–8. https://doi.org/10.5281/zenodo.4721234

[25] Nihal, K. P., & George, S. (2020). Predictive control algorithm for electromechanical drives with reduced computational complexity. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, *9*(7), 1234–1242. https://doi.org/10.15662/IJAREEIE.2020.0907012

[26] Wang, L., Xu, W., & Li, Y. (2018). Adaptive control schemes for robotic manipulators with parameter uncertainties. *IEEE/ASME Transactions on Mechatronics*, *23*(5), 2339–2349. https://doi.org/10.1109/TMECH.2018.2817246