# Filtering Strategies for State Estimation of Omniwheel Robots

Benjamin M. Dyer
*School of Engineering*
*University of Guelph*
*Guelph, Canada*
*dyerb@uoguelph.ca*

Trevor Robin Smith
*Department of Engineering Physics*
*McMaster University*
*Hamilton, Canada*

S. Andrew Gadsden
*School of Engineering*
*University of Guelph*
*Guelph, Canada*

M. Biglarbegian
*School of Engineering*
*University of Guelph*
*Guelph, Canada*

***Abstract*—Various state estimation strategies are investigated using a kinematic model of a four-wheel holonomic robot with Swedish wheels. A multi-tiered filtering strategy is implemented using Kalman Filters (KF) developed to estimate wheel velocity with an Extended Kalman Filter (EKF) and a Smooth Variable Structure Filter (SVSF) developed for state estimation of the robot. The use of only KFs on the wheels, only EKF or SVSF on the robot, and KFs on the wheels with either an EKF or SVSF on the robot is tested. Simulation results show that using a KF on each wheel in conjunction with either a SVSF or EKF on the robot yields an order of magnitude better state estimation compared to other configurations allowing for increased control of the robot.**

***Index Terms*—omnidirectional, encoders, estimation theory, Kalman filtering, SVSF**

## I. Introduction

Holonomic omniwheel robots allow for high maneuverability in a variety of environments. In order to be fully autonomous, however, it is necessary to have a method of accurately measuring position which can be fed into a controller. When attempting to control land-based mobile robots, the position and velocity of the robot's center of mass are often required. State determination uses sensors that inevitably inject noise into the system, such as through imperfect sensing and robot slippage. Various filtering techniques can be applied to improve the accuracy of state estimation.

### A. Omniwheel Robots

Omniwheel robots use wheels with rollers attached at an angle from the wheel's plane [1]. By correctly configuring the wheels, the robot becomes holonomic, enabling control of all degrees of freedom. The two main types of omnidirectional wheels are the Mechanum wheel with roller positions at 45 degrees to the wheel's plane; and Swedish wheels with roller positions at 90 degrees to the wheel's plane.

Robot designs incorporating mechanum omniwheels often utilize a standard four-wheeled vehicle with the rollers of inline wheels facing opposite directions [1] [2]. The configuration enables smoother rolling and, subsequently, a less complicated control problem. Additionally, the robot's conventional design enables the ability to form trains so multiple robots can work together to accomplish tasks impossible for an individual robot [3].

Contrary to mechanum wheels, 90-degree Swedish wheels avoid this efficiency problem as the force vector acts like that of the conventional wheel [4]. In order to obtain a holonomic robot, the wheels are positioned radially about the robot, typically in three-wheel or four-wheel configurations. The disadvantage of using Swedish wheels is an increase in vibrations and slippage due to the roller design [5] [6], creating avoidable non-linearities and injecting noise in the system, which must be accounted for when designing a controller [4].

### B. State Estimation

Maintaining high accuracy measurements of the robot state observables are essential for controlling sophisticated robotic systems [7]. These states are typically velocity of the robot's center of mass, angular velocity of each robot wheel, or the torque each wheel is applying [5].

Encoders operate by counting the number of times a wheel spins, typically using either a hall effect sensor or an optical sensor. By measuring the number of encoder state changes - also referred to as ticks - over a known duration, angular position and velocity measurements can be input into a controller [8].

Integrating encoder measurements have the known issue of error accumulation. Rapid, constant stream of pulses from each encoder creates a potential for information-loss, injecting noise into the system when recursive processes depend on the uncertain measurement [8]. Experimental techniques enable velocity estimation as a function of measured ticks in a given period and measurement variance [9].

Encoders are limited to measuring the angular velocity of the corresponding wheel, causing undetected motion from slip [7]. Under the ideal rolling condition, this is not a problem since the tangential velocity of the wheel is found using $v = r \times \omega$ [10]. As the system has no way of detecting robot slip, there is a deviation from the calculated and real position [11]. Slippage effects can be controlled by applying artificial constraints on accelerations, but determining the acceptable acceleration threshold is a challenging task and slip cannot be eliminated [7].

### C. Filtering Techniques

It is common practice to apply KFs to encoder output signals to minimize unwanted behaviours [4]. Filtering becomes essential when the system model of the robot relies on accurate knowledge of each wheel's angular velocity, as errors propagate and accumulate, leading to inaccurate estimates [7]. Previous work shows that the use of a Kalman filter (KF) for encoders can increase the accuracy of position estimation up to three times, and velocity estimation up to two orders of magnitude [12]. When filtering the encoder, it is advantageous to apply an adaptive KF based on the smallest interval the encoder can measure in order to improve accuracy during slow rotations [13] [14].

Work has been done on omniwheel robots using both Swedish and mechanum style wheels, showing that the use of an extended Kalman filter (EKF) can provide acceptable state estimation without explicitly accounting for error introduced by the motors. Research also shows the use of a UKF on omniwheel robots provides worse estimation than an EKF [15]. The majority of research has focused on the use of sensor fusion often involving encoders and some form of vision, however, there is little research in state estimation using only encoders on omniwheel robots.

The smooth variable structure filter (SVSF) is a filtering strategy based on the concept of a sliding mode allowing the filter to be more robust than the EKF or UKF. The SVSF utilizes a smoothing boundary layer in order to calculate the covariance in contrast to the gains used in KFs, EKFs, and UKFs. The SVSF is able to be applied to both linear and non-linear systems, however, for this research the SVSF will be applied only to the non-linear robot kinematics [16]. The SVSF has not been applied to omniwheel robots in the past.

## II. System Description

In order to develop filters, a system model must be defined. As such, this paper develops a dynamic model of the motors and a kinematic model of the omniwheel robot.

### A. Motor Dynamics

The motor dynamics can be described based on Newton's second law and Kirchhoff's voltage law, as seen in (1) and (2), respectively.

$$J\ddot{\theta} + b\dot{\theta} = Ki \tag{1}$$

$$L\frac{di}{dt} + Ri = V - K\dot{\theta} \tag{2}$$

where:

- $J$ is the motor moment of inertia
- $b$ is the motor viscous friction
- $K$ is the motor torque constant
- $i$ is the motor current
- $L$ is the motor inductance
- $R$ is the motor resistance
- $V$ is the motor voltage
- $\dot{\theta}$ is the angular velocity of the motor shaft about the shaft axis

From these equations the full dynamics can be written out in (3) using the observer shown in (4).

$$\frac{d}{dt}\begin{bmatrix}\dot{\theta}\\ i\end{bmatrix} = \begin{bmatrix}-\frac{b}{J} & \frac{K}{J}\\ -\frac{K}{L} & -\frac{R}{L}\end{bmatrix}\begin{bmatrix}\dot{\theta}\\ i\end{bmatrix} + \begin{bmatrix}0\\ \frac{1}{L}\end{bmatrix}V \tag{3}$$

$$\mathbf{z}_{motor} = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}\dot{\theta}\\ i\end{bmatrix} \tag{4}$$

where $\mathbf{z}_{motor}$ is the measurement vector.

### B. Omniwheel Robot Kinematics

One way to describe an omniwheel robot is using its kinematics, starting with the constraint on Swedish wheels during ideal rolling, described in (5).

$$\begin{bmatrix}\sin(\alpha_i)\\ -\cos(\alpha+\beta+\gamma)\\ -l\cos(\beta+\gamma)\end{bmatrix}^T \mathbf{R}(\theta)\dot{\xi}_I - r\dot{\phi}\cos\gamma = 0 \tag{5}$$

where:

- $\alpha$ is the angle between the wheel center and robot positive x-axis
- $\beta$ is the mount angle of the wheel with respect to $\alpha$
- $\gamma$ is the angle between rollers and wheel plane
- $l$ is the distance from the robot's center of mass to the wheel center
- $\mathbf{R}(\theta)$ is a rotation matrix (6)
- $\dot{\xi}_I$ is the state velocities in the global frame (7)
- $r$ is the wheel radius
- $\dot{\phi}$ is the angular velocity of the wheel

Fig. 1 displays the coordinates in terms of the robot's frame of reference.

$$\mathbf{R}(\theta) = \begin{bmatrix}\cos\theta & \sin\theta & 0\\ -\sin\theta & \cos\theta & 0\\ 0 & 0 & 1\end{bmatrix} \tag{6}$$

$$\dot{\xi}_I = \begin{bmatrix}\dot{x} & \dot{y} & \dot{\theta}\end{bmatrix}^T \tag{7}$$

Using (5) and the robot geometry, the kinematics can be derived. The wheel model used for simulations are standard 90 degree Swedish wheels, meaning $\gamma = 0$, and due to radial mounting, $\beta = 0$. For each wheel $i$ the constraint simplifies to (8).

$$\begin{bmatrix}\sin\alpha_i\\ -\cos\alpha_i\\ -l\end{bmatrix}^T \mathbf{R}(\theta)\dot{\xi}_I - r\dot{\phi}_i = 0 \tag{8}$$

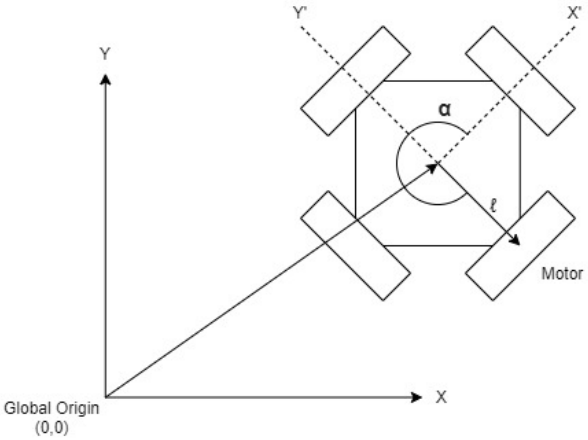


Fig. 1: Coordinate diagram of 4-wheel drive omniwheel robot

where $\alpha_i = \frac{2i\pi - \pi}{4}$ for $i = 1...4$. Making the substitution the kinematics of the robot in a global frame are described by (9).

$$\mathbf{u} = \begin{bmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{\phi}_3 \\ \dot{\phi}_4 \end{bmatrix} = \frac{1}{r}\mathbf{B}(\theta)\,\dot{\xi}_I \tag{9}$$

$$\mathbf{B}(\theta) = \frac{\sqrt{2}}{2}\begin{bmatrix} \cos\theta + \sin\theta & -\cos\theta + \sin\theta & l \\ \cos\theta - \sin\theta & \cos\theta + \sin\theta & l \\ -\cos\theta - \sin\theta & \cos\theta - \sin\theta & l \\ -\cos\theta + \sin\theta & -\cos\theta - \sin\theta & l \end{bmatrix} \tag{10}$$

where **u** contains the inputs for the system. For control system development, it is more useful to use the inverse kinematic model, described by (11).

$$\dot{\xi}_I = \mathbf{B}^{-1}(\theta)\,(r \cdot \mathbf{u}) \tag{11}$$

The inverse kinematic model is then transformed into discrete-time, as it is convenient to implement both controllers and Bayesian filters in discrete-time. The discrete-time form of the inverse kinematic model takes the form:

$$\xi_{I_{k+1}} = \xi_{I_k} + \mathbf{B}^{-1}(\theta)_k\,(r \cdot \mathbf{u}_{k+1}) \cdot T \tag{12}$$

where $k$ denotes the current time step, and $T$ denotes the length of each time step.

## III. Filtering Strategy

Several different filters enable determination of the robot position to varying degrees of accuracy. A KF is applied to the output from the encoders to gain more accurate measurements of the motors angular velocities. This filtered data is input to the kinematic model of the omniwheel robot, where an EKF or SVSF is applied to adjust for any additional noise and give a more accurate position measurement. This filtering strategy is displayed in Fig. 2b. For comparison, the EKF, SVSF are used on the unfiltered encoder inputs as seen in Fig. 2c, and the position of the robot is determined using only the filtered encoder inputs, shown in Fig. 2a.

### A. Motor Kalman Filter

In order to reduce complexity, it makes sense to have a single filter that applies to all four motors simultaneously. In order to do this we simply combine the motor dynamics from (3) and (4) and include noise terms in a standard state-space equation described in (13).

$$\mathbf{x}_{k+1} = (\mathbf{A}_m + \mathbf{I})\,\mathbf{x}_k + (\mathbf{B}_m\mathbf{V}_k + \mathbf{w}_k)\,T \tag{13}$$

$$\mathbf{z}_{k+1} = \mathbf{C}_m\mathbf{x}_{k+1} + \mathbf{v}_k \tag{14}$$

$$\mathbf{x} = \begin{bmatrix} \dot{\theta}_1 & i_1 & \dot{\theta}_2 & i_2 & \dot{\theta}_3 & i_3 & \dot{\theta}_4 & i_4 \end{bmatrix}^T \tag{15}$$

$$\mathbf{A}_m = diag\left(\begin{bmatrix} -\frac{b}{J} & \frac{K}{J} \\ -\frac{K}{L} & -\frac{R}{L} \end{bmatrix}\begin{bmatrix} -\frac{b}{J} & \frac{K}{J} \\ -\frac{K}{L} & -\frac{R}{L} \end{bmatrix}\begin{bmatrix} -\frac{b}{J} & \frac{K}{J} \\ -\frac{K}{L} & -\frac{R}{L} \end{bmatrix}\begin{bmatrix} -\frac{b}{J} & \frac{K}{J} \\ -\frac{K}{L} & -\frac{R}{L} \end{bmatrix}\right) \tag{16}$$

$$\mathbf{B}_m = diag\left(\begin{bmatrix} 0 & \frac{1}{L} & 0 & \frac{1}{L} & 0 & \frac{1}{L} & 0 & \frac{1}{L} \end{bmatrix}\right) \tag{17}$$

$$\mathbf{V} = \begin{bmatrix} V_1 & V_2 & V_3 & V_4 \end{bmatrix}^T \tag{18}$$

$$\mathbf{C}_m = diag\left(\begin{bmatrix} 1 & 0 \end{bmatrix}\begin{bmatrix} 1 & 0 \end{bmatrix}\begin{bmatrix} 1 & 0 \end{bmatrix}\begin{bmatrix} 1 & 0 \end{bmatrix}\right) \tag{19}$$

where $\mathbf{w}_k$ is a vector for each motors input noise defined as $\mathbf{w}_k = \mathcal{N}(0, \mathbf{Q}_k)$ with $\mathbf{Q}_k$ being the input noise co-variance matrix, $\mathbf{v}$ is a vector composed of the measurement noise for each motor defined as $\mathbf{v}_k = \mathcal{N}(0, \mathbf{R}_k)$ with $\mathbf{R}_k$ being the measurement noise co-variance matrix, and $T$ is the time step of the system. Implementation of the Kalman filter was completed in a standard form, using a time update and a measurement update defined in (20)-(21) and (22)-(24) respectively.

$$\hat{\mathbf{x}}_{k+1|k} = \mathbf{A}_m\hat{\mathbf{x}}_{k|k} \tag{20}$$

$$\mathbf{P}_{k+1|k} = \mathbf{A}_m\mathbf{P}_{k|k}\mathbf{A}_m^T + \mathbf{Q} \tag{21}$$

$$\hat{\mathbf{x}}_{k+1|k+1} = \hat{\mathbf{x}}_{k+1|k} + \mathbf{K}_{k+1}[\mathbf{z}_{k+1} - \mathbf{C}_m\left(\hat{\mathbf{x}}_{k+1|k}\right)] \tag{22}$$

$$\mathbf{K}_{k+1} = \mathbf{P}_{k+1|k}\mathbf{C}_m^T\left[\mathbf{C}_m\mathbf{P}_{k+1|k}\mathbf{C}_m^T + \mathbf{R}\right] \tag{23}$$

$$\mathbf{P}_{k+1|k+1} = (\mathbf{I} - \mathbf{K}_{k+1}\mathbf{C}_m)\,\mathbf{P}_{k+1|k} \tag{24}$$

where $\hat{\mathbf{x}}_{k+1|k+1}$ is the estimation of the state vector, $\hat{\mathbf{x}}_{k+1|k}$ is the prediction of the state vector, $\mathbf{K}_{k+1}$ is the Kalman gain, $\mathbf{P}_{k+1|k}$ is the prediction of the error covariance, and $\mathbf{P}_{k+1|k+1}$ is the updated error co-variance.

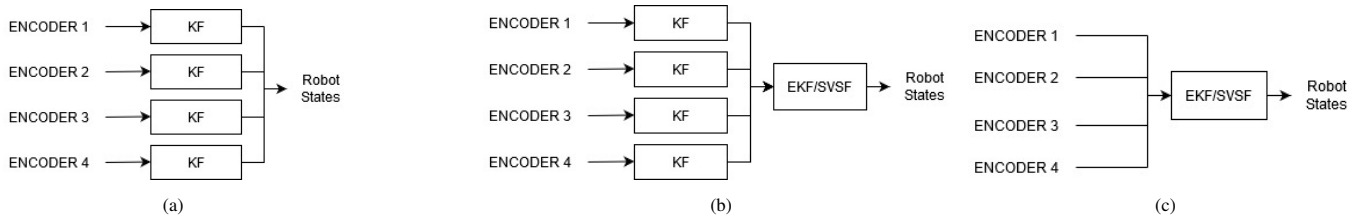


Fig. 2: Structures of each filtering strategy where (a) depicts using only KFs on each wheel and no filter on the robot, (b) depicts KFs on each wheel and either an EKF or SVSF on the robot, and (c) depicts using only an EKF or SVSF on the robot

### *B. Extended Kalman Filter for the Robot*

While the KF is sufficient for estimation of the motors, the non-linearity of the omniwheel kinematics would cause a KF to be insufficient. An alternative implementation is with an EKF, using both the non-linear and linearized kinematics. First the discrete version of the system from (12) is redefined and non-linear functions are created shown in (25)-(28).

$$\mathbf{x}_{k+1} = f\left(\mathbf{x}_k, \mathbf{u}_{k+1}\right) + \mathbf{w}_{k+1} \tag{25}$$

$$\mathbf{z}_{k+1} = h\left(\mathbf{x}_{k+1}\right) + \mathbf{v}_{k+1} \tag{26}$$

$$f\left(\mathbf{x}_k, \mathbf{u}_{k+1}\right) = \xi_{I_i} + \mathbf{B}^{-1}\left(\theta\right)_i \left(r \cdot \mathbf{u}_{i+1} + \mathbf{v}_{s_i}\right) \cdot T \tag{27}$$

$$h\left(\mathbf{x}_{k+1}\right) = \mathbf{x}_{k+1} \tag{28}$$

With the state-space kinematics redefined in terms of non-linear functions the EKF can be applied. The method first predicts the state and then updates the state estimates as seen in equations (29)-(30) and (31)-(35) respectively.

$$\hat{\mathbf{x}}_{k+1|k} = f\left(\hat{\mathbf{x}}_{k|k}, \mathbf{u}_{k+1}\right) \tag{29}$$

$$\mathbf{P}_{k+1|k} = \mathbf{F}_{k+1}\mathbf{P}_{k|k}\mathbf{F}_{k+1}^T + \mathbf{Q}_{k+1} \tag{30}$$

$$\widetilde{\mathbf{y}}_{k+1} = \mathbf{z}_{k+1} - h\left(\hat{\mathbf{x}}_{k+1|k}\right) \tag{31}$$

$$\mathbf{S}_{k+1} = \mathbf{H}_{k+1}\mathbf{P}_{k+1|k}\mathbf{H}_{k+1}^T + \mathbf{R}_{k+1} \tag{32}$$

$$\mathbf{K}_{k+1} = \mathbf{P}_{k+1|k}\mathbf{H}_{k+1}^T\mathbf{S}^{-1}{}_{k+1} \tag{33}$$

$$\hat{\mathbf{x}}_{k+1|k+1} = \hat{\mathbf{x}}_{k+1|k} + \mathbf{K}_{k+1}\widetilde{\mathbf{y}}_{k+1} \tag{34}$$

$$\mathbf{P}_{k+1|k+1} = \left(\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H}_{k+1}\right)\mathbf{P}_{k+1|k} \tag{35}$$

Where $\hat{\mathbf{x}}_{k+1|k}$ is the predicted state estimate, $\mathbf{P}_{k+1|k}$ is the predicted covariance estimate, $\widetilde{\mathbf{y}}_{k+1}$ is the innovation residual, $\mathbf{S}_{k+1}$ is the innovation covariance, $\mathbf{K}_{k+1}$ is the near optimal Kalman gain, $\hat{\mathbf{x}}_{k+1|k+1}$ is the updated state estimate, $\mathbf{P}_{k+1|k+1}$ is the updated covariance estimate, $\mathbf{F}_{k+1}$ is the Jacobian of $f$ defined in (36), and $\mathbf{H}_{k+1}$ is the Jacobian of $h$ defined in (37).

$$\mathbf{F}_{k+1} = \left.\frac{\partial f}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}_{k|k}, \mathbf{u}_k} \tag{36}$$

$$\mathbf{H}_{k+1} = \left.\frac{\partial h}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}_{k+1|k}} \tag{37}$$

### *C. Smooth Variable Structure Filter for the Robot*

Another method of estimating the state of a non-linear system is the SVSF. The SVSF begins by using the predicted state estimates, predicted measurements, and measurement error to calculate the SVSF gain, as shown in (38)-(41) respectively.

$$\hat{\mathbf{x}}_{k+1|k} = \hat{f}\left(\hat{\mathbf{x}}_{k|k}, \mathbf{u_k}\right) \tag{38}$$

$$\hat{\mathbf{x}}_{k+1|k} = \mathbf{C}\hat{\mathbf{x}}_{k+1|k} \tag{39}$$

$$\mathbf{e}_{z,k+1|k} = \mathbf{z}_{k+1} - \hat{\mathbf{z}}_{k+1|k} \tag{40}$$

$$\mathbf{K}_{k+1}^{SVSF} = \mathbf{C}^{+}\left(\left|\mathbf{e}_{z,k+1|k}\right|_{Abs} + \gamma\left|\mathbf{e}_{z,k|k}\right|_{Abs}\right)\left(\frac{\mathbf{e}_{z,k+1|k}}{\psi}\right) \tag{41}$$

Where $\gamma$ is the convergence rate, and $\psi$ is the smoothing boundary layer widths. The SVSF gain is used to refine the state estimate after which the measurement and error are updated, as shown in (42)-(44):

$$\hat{\mathbf{x}}_{k+1|k+1} = \hat{\mathbf{x}}_{k+1|k} + \mathbf{K}_{k+1}^{SVSF} \tag{42}$$

$$\hat{\mathbf{x}}_{k+1|k+1} = \mathbf{C}\hat{\mathbf{x}}_{k+1|k+1} \tag{43}$$

$$\mathbf{e}_{z,k+1|k+1} = \mathbf{z_{k+1}} - \hat{\mathbf{z}}_{k+1|k+1} \tag{44}$$

## IV. Simulations

All of the simulations result from MATLAB and compare the following six filters:

- No filters applied
- KFs on motors
- EKF on robot
- SVSF on robot
- EKF on robot and KFs on motors
- SVSF on robot and KFs on motors

The robot follows an arbitrary path defined in (51) allowing for long simulations without repeating the same path multiple time.

Several physical constants must be defined and input to the kinematic and dynamic models. These values are listed in Table I. The values for the robot kinematics are distances measured using a set of calipers. The values for the motor

TABLE I
LIST OF PHYSICAL CONSTANTS NEEDED TO DESCRIBE SYSTEM KINEMATICS AND DYNAMICS

| Variable | Symbol | Value | Units |
|---|---|---|---|
| Wheel radius | $r$ | 3.275 | cm |
| Robot Radius | $l$ | 19.5 | cm |
| Motor Resistance | $R$ | 1 | $\Omega$ |
| Motor Inductance | $L$ | 0.01 | $H$ |
| Motor Moment of Inertia | $J$ | 0.01 | $N \cdot m^2$ |
| Motor Viscous Friction | $b$ | 0.1 | $N \cdot m \cdot s$ |
| Motor Constant | $K$ | 0.01 | $N \cdot m/A$ |
| Time step | $T$ | 0.02 | $s$ |

dynamics were determined experimentally using a multimeter, oscilloscope, NE555 based wave generator, and a Maxwell bridge. The $\gamma$ and $\psi$ terms were tuned manually and are shown in (45) and (46). The input and measurement noise covariances of each motor are defined in (47) and (48), and the robot's covariances are defined in (49) and (50) respectively. Both cases assume noise to be dependent only on the respective state resulting in diagonal matrices.

$$\gamma = 0.5 \tag{45}$$

$$\psi = \begin{bmatrix} 1.1 \times 10^{-4} & 0.4 \times 10^{-4} & 0.5 \times 10^{-4} \end{bmatrix} \tag{46}$$

$$\mathbf{Q}_{motor} = \begin{bmatrix} 10 & 0 \\ 0 & 0.5 \end{bmatrix} \tag{47}$$

$$\mathbf{R}_{motor} = \begin{bmatrix} 0.01 \end{bmatrix} \tag{48}$$

$$\mathbf{Q}_{robot} = diag(10^{-3}, 10^{-3}, 10^{-3}) \tag{49}$$

$$\mathbf{R}_{robot} = diag(10^{-9}, 10^{-9}, 10^{-9}) \tag{50}$$

### A. Results

The robot follows a reference trajectory defined using the motor input voltages, as seen in (51) for 600 seconds. Simulations were run 300 times, with the resulting state RMSE for each filter displayed in table II.

$$\mathbf{u} = 12 \begin{bmatrix} \sin\left(\frac{\pi}{10}t\right) \\ \cos\left(\frac{\pi}{5}t\right) \\ -\sin\left(\frac{\pi}{10}t\right) \\ -\cos\left(\frac{\pi}{5}t\right) \end{bmatrix} + \sin\left(\frac{\pi}{20}t\right)\cos\left(\frac{\pi}{20}t\right) \tag{51}$$

The results show that using only KFs on the wheels gives some increase in estimation accuracy over having no filter. The use of an SVSF or EKF with no filtering performed on the wheels, however, results in a doubling of estimation accuracy. The EKF and SVSF show near-identical performance, with the SVSF being slightly more accurate. Using KFs on the wheels in conjunction with either an SVSF or EKF decreases the RMSE two-fold with an average accuracy of below 2 cm. Once again, the EKF and SVSF show near-identical performance.

TABLE II
AVERAGE STATE RMSES OVER 300 TEST RUNS USING THE PATH DESCRIBED IN (51) OVER 600 SECONDS

| Filter | x RMSE (cm) | y RMSE (cm) | $\theta$ RMSE (rad) |
|---|---|---|---|
| No Filter | 10.02 | 11.83 | 0.0869 |
| KF | 9.28 | 10.98 | 0.0719 |
| EKF | 4.63 | 5.89 | 0.0511 |
| SVSF | 4.63 | 5.88 | 0.0511 |
| KF+EKF | 1.63 | 1.98 | 0.0168 |
| KF+SVSF | 1.63 | 1.97 | 0.0168 |

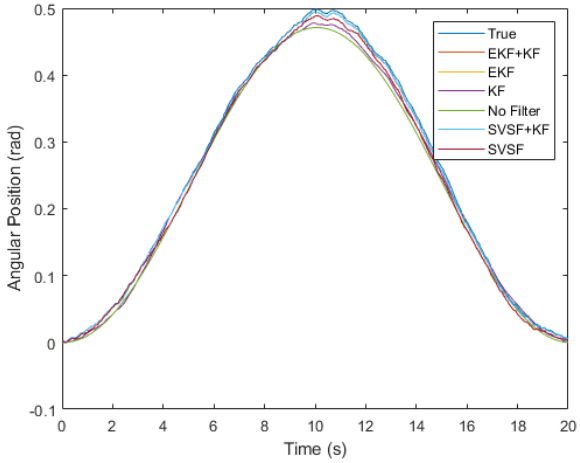


Fig. 3: Estimates of the robot's angular position over the course of a 20 second path using each filter combinations

Results from running the robot using (51) for 20 seconds can be seen in Fig. 3 and Fig. 4. The error in the x-position estimate, defined in Fig. 5, clearly shows the KF+EKF and KF+SVSF filters accumulate the least error and, therefore, will cause the least drift. Minimizing error is crucial since the estimates are fed into a controller in order to accurately move the robot. Over 30 minutes, the robot drifts approximately 20 cm when using the best filtering strategy and over a meter when using no filtering strategy. These results show the necessity of using high accuracy filters when running a robot over long trajectories.

## V. CONCLUSION

Using encoders on each wheel enables robot position estimation using the dynamic and kinematic models described. The implementation of filtering strategies shows distinct improvement in the accuracy of position estimation. Using KFs on each wheel without further filtering allows for increased accuracy of the robot's position, but with an RMSE of approximately 10 cm. Such low accuracy would cause controllers to have difficulty in accurately moving the robot along some desired path. Using an EKF or SVSF on the robot

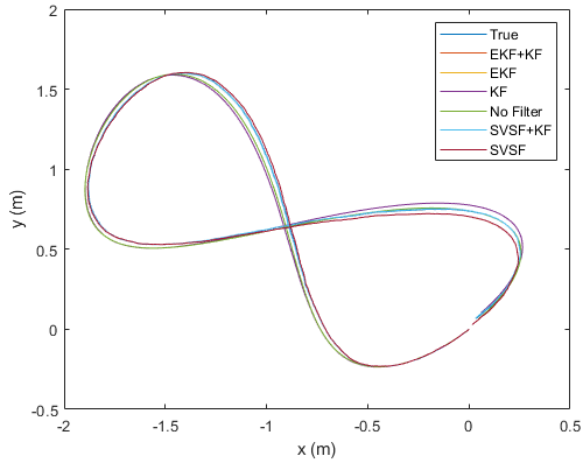


Fig. 4: Estimates of the robot's position over the course of a 20 second path using each filter combinations

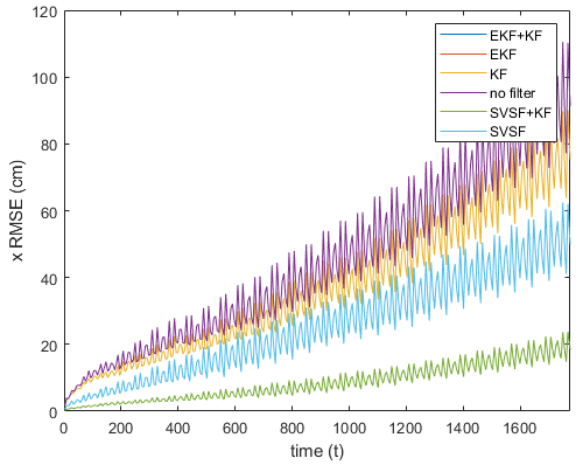


Fig. 5: X position RMSE at all timesteps for the robot following the path in (51)

shows estimation improvements by a factor of two, however, the best filtering strategy is found to be making use of an EKF or SVSF on the system with KFs on wheels. This result is confirmed by significantly lower estimation drift compared to other filtering strategies.

## References


[1] A. V. Borisov, A. A. Kilin, and I. S. Mamaev, "Dynamics and control of an omniwheel vehicle," *Regular and Chaotic Dynamics*, vol. 20, pp. 153–172, 2015.

[2] A. Kilin, P. Bozek, Y. Karavaev, A. Klekovkin, and V. Shestakov, "Experimental investigations of a highly maneuverable mobile omniwheel robot," *International Journal of Advanced Robotic Systems*, vol. 14, no. 6, p. 1729881417744570, 2017. [Online]. Available: https://doi.org/10.1177/1729881417744570

[3] L. Lin and H. Shih, "Modeling and adaptive control of an omni-mecanum-wheeled robot," *Intelligent Control and Automation*, vol. 4, no. 166-179, 2013.

[4] Y. Liu, R. L. Williams, and J. J. Zhu, "Integrated control and navigation for omni-directional mobile robot based on trajectory linearization," in *2007 American Control Conference*, July 2007, pp. 2153–2158.

[5] D. Stonier, S. Cho, S. Choi, N. S. Kuppuswamy, and J. Kim, "Non-linear slip dynamics for an omniwheel mobile robot platform," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, April 2007, pp. 2367–2372.

[6] A. Bramanta, A. Virgono, and R. E. Saputra, "Control system implementation and analysis for omniwheel vehicle," in *2017 International Conference on Control, Electronics, Renewable Energy and Communications (ICCREC)*, Sep. 2017, pp. 265–270.

[7] R. Rojas and A. Förster, "Holonomic control of a robot with an omnidirectional drive ." 2006.

[8] S. C. Venema, "A kalman filter calibration method for analog quadrature position encoders," Master of Science in Electrical Engineering, University of Washington, 1994.

[9] R. Kavanagh, "Shaft encoder characterisation through analysis of the mean-squared errors in nonideal quantised systems," *IEE Proceedings-Science, Measurement and Technology*, vol. 149, no. 2, pp. 99 – 104, 2002/03/. [Online]. Available: http://dx.doi.org/10.1049/ip-smt:20020304

[10] M. Sotnikova, E. Veremey, and N. Zhabko, "Wheel angular velocity stabilization using rough encoder data," in *2014 14th International Conference on Control, Automation and Systems (ICCAS 2014)*, Oct 2014, pp. 1345–1350.

[11] M. Boisvert and P. Micheau, "Estimators of wheel slip for electric vehicles using torque and encoder measurements," *Mechanical Systems and Signal Processing*, vol. 76-77, pp. 665 – 676, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0888327016000686

[12] Y. Zimmerman, A. Brandes, and Y. Oshman, "Improving the accuracy of analog encoders via kalman filtering," *IFAC Proceedings Volumes*, vol. 36, no. 12, pp. 279 – 284, 2003, 5th IFAC International Symposium on Intelligent Components and Instruments for Control Applications 2003, Aveiro, Portugal, 9-11 July 2003. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1474667017325478

[13] D. Luong-Van and J. Katupitiya, "An adaptive kalman filter with quadrature encoder quantisation compensation," *IFAC Proceedings Volumes*, vol. 37, no. 14, pp. 109 – 114, 2004, 3rd IFAC Symposium on Mechatronic Systems 2004, Sydney, Australia, 6-8 September, 2004. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1474667017310893

[14] W. Shaowei and W. Shanming, "Velocity and acceleration computations by single-dimensional kalman filter with adaptive noise variance," *Przeglad Elektrotechniczny*, vol. 88, 01 2012.

[15] L. Armesto and J. Tornero, "Slam based on kalman filter for multi-rate fusion of laser and encoder measurements," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 2, Sep. 2004, pp. 1860–1865 vol.2.

[16] S. A. Gadsden, "Smooth variable structure filtering: Theory and applications," PhD, McMaster University, oct 2011.