Omnidirectional Mobile Robot Motion Control

Pavéz Fabriani Eduardo Martín

edumarpav@yahoo.com.ar

Control Systems, Faculty of Engineering,

Cuyo National University,

Mendoza, Argentina

February 2023

Abstract

This project presents the development of a control system for an omnidirectional mobile robot designed for surveillance applications. The control system includes a state feedback control and a Kalman filter to improve the accuracy of the robot's movement and reduce the effect of noise in the process and measurements. The design of the state observer and the sensor model are also discussed.

The report presents noise and performance analyses. The system is capable of accurately tracking references, with a robust response to disturbances.

Overall, the developed control system provides a reliable and efficient solution for omnidirectional mobile robots designed for surveillance applications.

Contents

1. Introduction

The use of mobile robots in various applications has become increasingly popular in recent years. One application area that has gained particular interest is surveillance, where mobile robots can be used to monitor and collect data in harsh environments with spatial limitations. In this project, a control system is developed for an omnidirectional mobile robot designed for surveillance purposes.

The control system includes a state feedback control and a Kalman filter to improve the accuracy of the robot's movement and reduce the effect of noise in the measurements. The robot is equipped with mecanum wheels, which allow it to move in any direction without the need for complex steering mechanisms.

This report outlines the development and simulation of the control system, including the kinematics model, and state space model of the plant. It is also discussed the design of the state observer and the sensor model.

The results section of the report presents the noise analysis, observer estimations, and operation limits of the system. Outputs and references are compared to evaluate system’s performance. It is expected to achieve a relative error to the maximum velocity of at most 5%, a response time of half a second to follow references, and smooth operations in order to capture proper image quality.

2. Development

2.1 Omnidirectional mobile robot.

An omnidirectional mobile robot consists of a wheeled platform that can be controlled to move in any horizontal direction without changing its orientation.

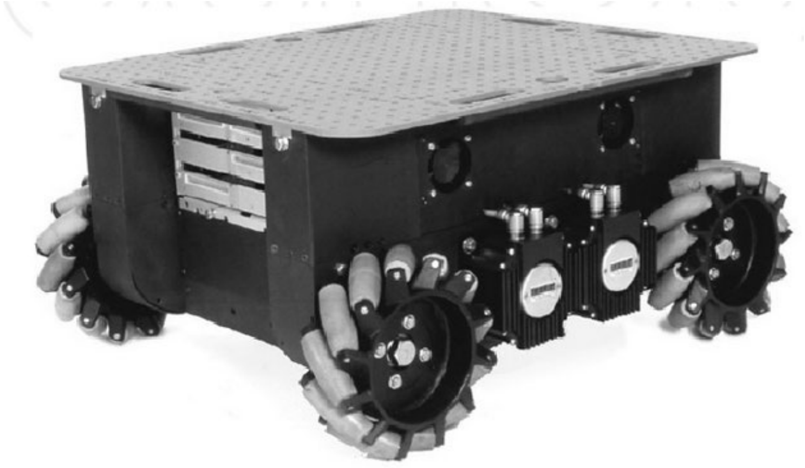

Figure 1. Omnidirectional mobile platform

This system strongly relies in special wheels that allow, through their velocities combination, the resulting movement in any direction. This study makes use of the mecanum wheels configuration of Figure 1.

2.1.1 Mecanum wheels

Mecanum wheels are a type of wheel designed for omnidirectional mobility, allowing a mobile robot to move in any direction without changing its orientation. Unlike conventional wheels, which have a single plane of movement, mecanum wheels have multiple rollers around the circumference that allow for movement in multiple directions. The rollers are angled at 45 degrees in relation to the wheel's plane of movement, creating a unique rolling motion. This allows the robot to roll in any direction, including diagonal movement, by controlling the speeds and directions of the individual wheels.

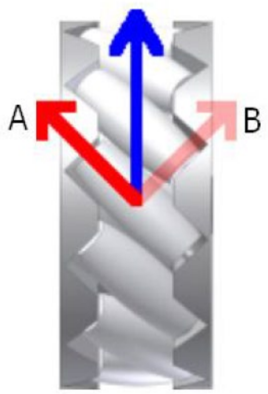

Figure 2. Bottom view of the force exerted by the motor on the mecanum wheel. [1]

2.1.2 Resulting movement

As mentioned previously, depending on the velocity and rotating direction of its four wheels, the robot will rotate or move in a different linear direction. That can be appreciated in the Figure 3.

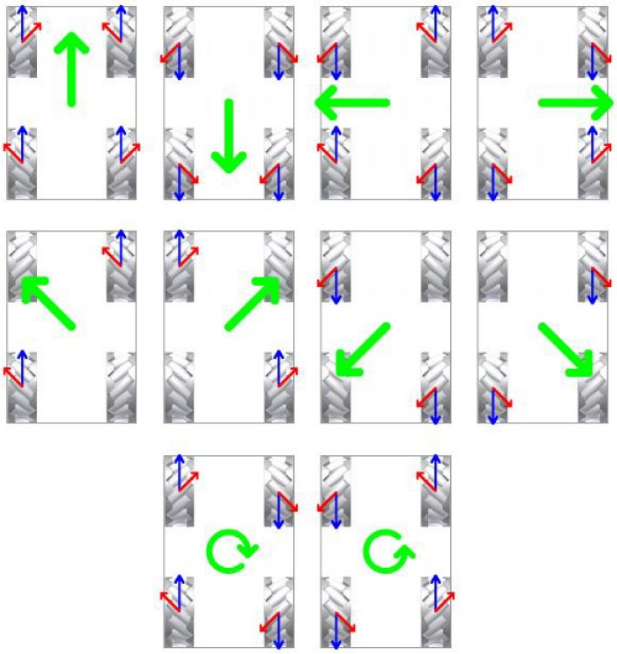

Figure 3. Robot displacement direction depending on wheels rotating direction [1]

2.1.3 Kinematics Model

For mobile robots, the forward kinematics model is used to determine robot position and orientation from its wheels' rotations. The derivation of the model for the omnidirectional mobile robot is obtained from [1].

In Figure 2, the arrangement of wheels and axes Σ_0, Σ_{iW} $(i = 1,2,3,4)$ can be seen. $V_{iW}(i = 1,2,3,4) \in \mathbb{R}$ is the velocity vector corresponding to the revolutions of the wheel, where $V_{iW} = R_w \times \omega_{iW}$, R_w is the wheel radius, ω_{iW} is the angular velocity of the wheel, and $V_{ir}(i = 1,2,3,4) \in \mathbb{R}$ is the tangential velocity vector of the roller in contact with the floor.

Wheel radius (R_w)	$10\ cm$
Platform width (l_r)	$30\ cm$
Platform length (L_r)	$45\ cm$

Table 1. Structural parameters of omnidirectional mobile platform

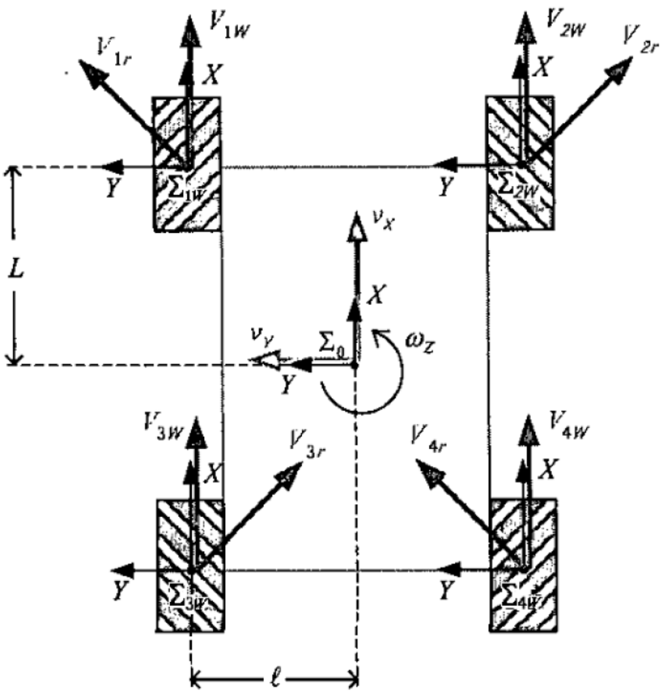

Figure 4. Mecanum wheels configuration and coordinated systems.

V_{iX} is the component that contributes the linear velocity of the wheel V_{iW} plus the velocity contributed in the direction of the free rollers $V_{ir}.cos\ 45º$. Similarly, V_{iY} is $V_{ir}.\sin 45º$ from Figure 1.

$$V_{1X} = V_{1W} + \frac{V_{1r}}{\sqrt{2}} \quad , \quad V_{1Y} = \frac{V_{1r}}{\sqrt{2}}$$

$$V_{2X} = V_{2W} + \frac{V_{2r}}{\sqrt{2}} \quad , \quad V_{2Y} = -\frac{V_{2r}}{\sqrt{2}}$$

$$V_{3X} = V_{3W} + \frac{V_{3r}}{\sqrt{2}} \quad , \quad V_{3Y} = -\frac{V_{3r}}{\sqrt{2}}$$

$$V_{4X} = V_{4W} + \frac{V_{4r}}{\sqrt{2}} \quad , \quad V_{4Y} = \frac{V_{4r}}{\sqrt{2}}$$

$$V_{1X} = v_X - l_r\omega_Z \quad , \quad V_{1Y} = v_Y + L_r\omega_Z$$

$$V_{2X} = v_X + l_r\omega_Z \quad , \quad V_{2Y} = v_Y + L_r\omega_Z$$

$$V_{3X} = v_X - l_r\omega_Z \quad , \quad V_{3Y} = v_Y - L_r\omega_Z$$

$$V_{4X} = v_X + l_r\omega_Z \quad , \quad V_{4Y} = v_Y - L_r\omega_Z$$

Where v_X, v_Y, w_Z represent linear and angular velocities of the vehicle with respect to X and Y axis.

Considering that the radius of the wheels is R_w, their velocity is $\dot{\theta}_i$ and combining all previous equations, the inverse kinematics model is obtained:

$$\dot{\theta}_1 = \frac{1}{R_w}(v_X - v_Y - (L_r + l_r)\omega_Z)$$

$$\dot{\theta}_2 = \frac{1}{R_w}(v_X + v_Y + (L_r + l_r)\omega_Z)$$

$$\dot{\theta}_3 = \frac{1}{R_w}(v_X + v_Y - (L_r + l_r)\omega_Z)$$

$$\dot{\theta}_4 = \frac{1}{R_w}(v_X - v_Y + (L_r + l_r)\omega_Z)$$

Equation 1. Inverse Kinematics

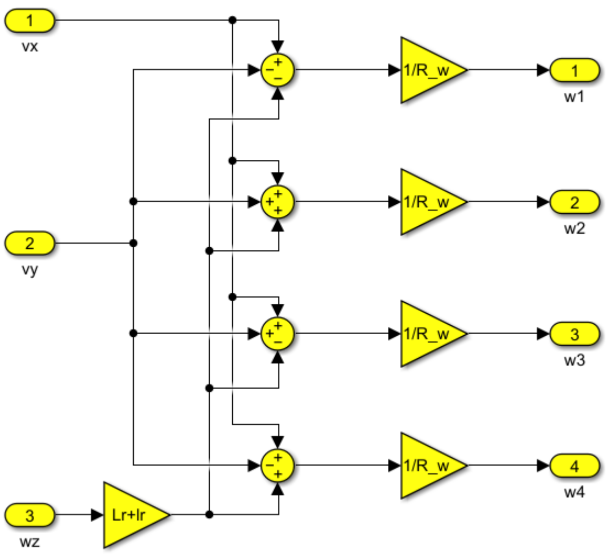

Figure 5. Inverse Kinematics Simulink model

Adding all four equations in such a way that the desired variable to obtain is always positive, the forward kinematics model results:

$$v_X = \frac{R_w}{4}(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3 + \dot{\theta}_4)$$

$$v_Y = \frac{R_w}{4}(-\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3 - \dot{\theta}_4)$$

$$\omega_z = \frac{R_w}{4(L_r + l_r)}(-\dot{\theta}_1 + \dot{\theta}_2 - \dot{\theta}_3 + \dot{\theta}_4)$$

Equation 2. Forward Kinematics

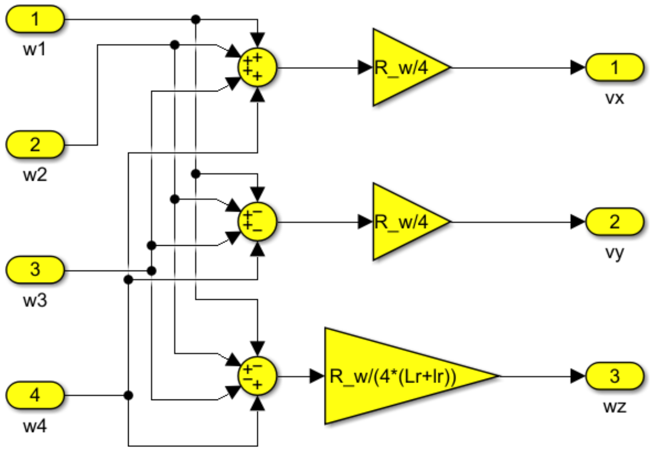

Figure 6. Forward Kinematics Simulink model

2.1.4 Motors

DC motors were chosen as the actuators for the robot, and in order to control their behavior, a linear mathematical model is obtained from [2]. The mathematical model provides a foundation for understanding and predicting the behavior of the DC motors in the omnidirectional mobile robot, and is an essential component for the control and motion planning algorithms. By using this model, the system can be optimized and refined to ensure smooth and reliable motion of the robot.

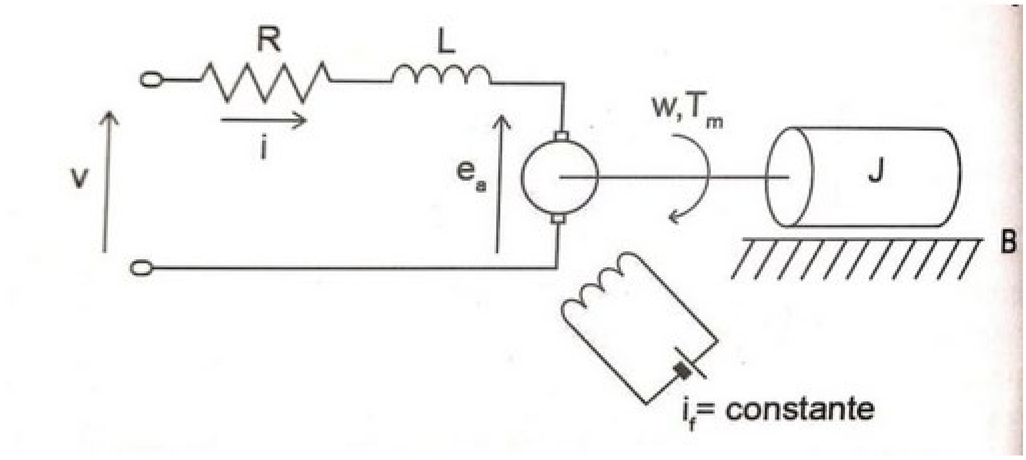

Figure 7. DC motor physical model

The equations that describe the model are:

$$L\frac{di(t)}{dt} = v(t) - R.i(t) - e_a(t)$$

$$J\frac{d\omega(t)}{dt} = T_m(t) - B.\omega(t)$$

$$e_a(t) = K_a.\omega(t)$$

$$T_m(t) = K_m.i(t)$$

Equation 3. DC motor mathematical model

The motor parameters were extracted from the BSG23-28AA-03 datasheet [3]:

Parameter	Value
Moment of inertia (J)	$0.01\ Kg.m2$
Viscous friction (B)	$0.002\ N.m.s$
Electromotive force constant (K_a)	$0.145\,\frac{V}{rad.s}$
Torque constant (K_m)	$0.145\ \frac{Nm}{A}$
Resistance (R)	$0.93\ \Omega$
Inductance (L)	$1.872\ mH$
Currents limit (A)	$5.5\ A$
Voltage limit (V)	$48\ V$
Angular velocity limit (w)	$243\,\frac{rad}{s}$
Gearbox ratio $(adim)$	30

Table 2. Motor parameters

A gearbox is coupled to each motor in order to increase their angular velocities. This is to allow for more precise measurements and a torque increase.

Considering that the project makes use of a linearized mathematical model of the plant, the controller should be robust enough to be efficient when implemented in the real system, that might vary slightly from the dynamics proposed.

2.2 State Space Model

In order to get the state space model of one motor, the inputs, state variables and outputs are defined as:

$$u(t) = v(t) \qquad ; \qquad x(t) = \begin{bmatrix} \omega(t) \\ i(t) \end{bmatrix} \qquad ; \qquad y(t) = \omega(t)$$

Then the state space model is:

$$\dot{x}(t) = \begin{bmatrix} -\frac{B}{J} & \frac{K_m}{J} \\ -\frac{K_a}{L} & -\frac{R}{L} \end{bmatrix}.x(t) + \begin{bmatrix} 0 \\ \frac{1}{L} \end{bmatrix} u(t)$$

$$y(t) = [1 \quad 0].x(t)$$

And considering there are 4 motors the state space model of the whole system results in:

$$u = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} \quad x = \begin{bmatrix} w_1 \\ i_1 \\ w_2 \\ i_2 \\ w_3 \\ i_3 \\ w_4 \\ i_4 \end{bmatrix} \quad y = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix} \quad A = \begin{bmatrix} -\frac{B_1}{J_1} & \frac{K_{m1}}{J_1} & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{K_{a1}}{L_1} & -\frac{R_1}{L_1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{B_2}{J_2} & \frac{K_{m2}}{J_2} & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{K_{a2}}{L_2} & -\frac{R_2}{L_2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{B_3}{J_3} & \frac{K_{m3}}{J_3} & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{K_{a3}}{L_3} & -\frac{R_3}{L_3} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\frac{B_4}{J_4} & \frac{K_{m4}}{J_4} \\ 0 & 0 & 0 & 0 & 0 & 0 & -\frac{K_{a4}}{L_4} & -\frac{R_4}{L_4} \end{bmatrix}$$

$$B = \begin{vmatrix} 0 & 0 & 0 & 0 \\ \frac{1}{L_1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & \frac{1}{L_2} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{L_3} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{L_4} \end{vmatrix} \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\dot{x}(t) = A\,.\,x(t) + B.u(t)$$
$$y(t) = C.x(t)$$

Equation 4. State space model of the system

A and B matrices are simplified by assuming that all the motors have identical parameters, and are those stated in Table 1.

Process disturbance is added to the state space model of Equation 4. The resulting system is

$$\dot{x}(t) = A\,.\,x(t) + B.u(t) + v(t)$$
$$y(t) = C.x(t)$$

Equation 5. State space model with process disturbance

Where $v(t)$ is a 8x1 vector of variables with zero mean and variance of 1.

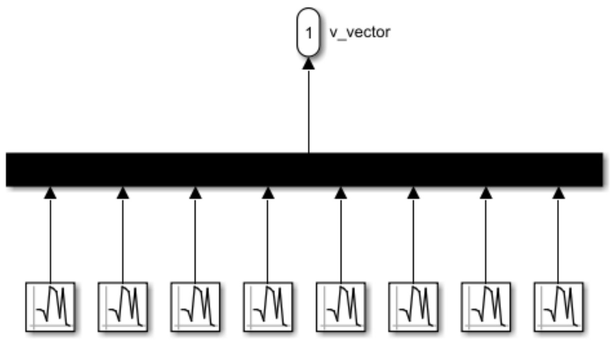

Figure 8. Process disturbance Simulink model

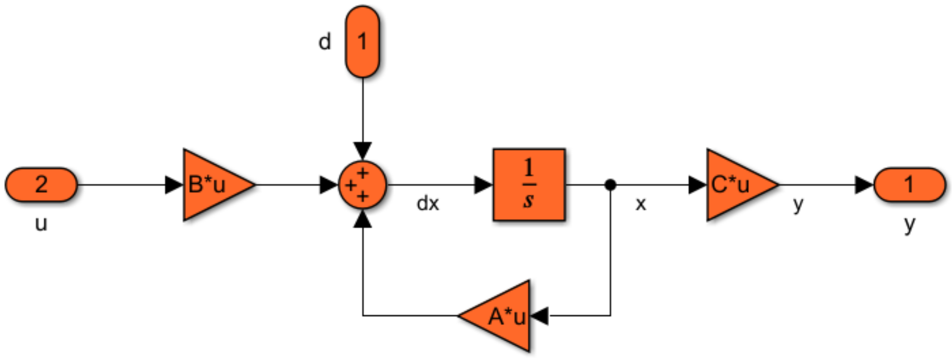

Figure 9. Plant Simulink model

2.2.1 Observability

It is desired to design a state feedback control system, and not accessing all the states directly with sensors would be a cheaper alternative. Then, a state observer can be designed in the controller to estimate some of them. Before doing so, it is important to check for system's observability, which will help on the decision of choosing the right sensors.

Observability refers to the ability of inferring the internal states of a system from its inputs and outputs. In this particular case, by observing the velocity of the 4 motors, it is possible to estimate the current being drawn by each of them.

To be precise, observability's definition is [4]:

"A linear system is observable if for every T>0 it is possible to determine the state of the system $x(T)$ through the measurements of $y(t)$ and $u(t)$ on the interval $[0, T]$"

According to the observability rank condition's theorem:

"A linear system is observable if and only if the observability matrix reachability W_o

$$W_o = \begin{bmatrix} C \\ CA \\ \dots \\ CA^{n-1} \end{bmatrix}$$

has full row rank."

It is calculated in MATLAB that the matrix rank is 8, validating that with the use of four velocity sensors it is possible to estimate all the states:

Observability

```
O=obsv(A,C);
rankobs=rank(O);
disp(['The observability matrix rank is: ',num2str(rankobs)])
```

```
The observability matrix rank is: 8
```

2.2.2 Reachability

Now, in order to decide whether a state feedback controller would be effective, it is necessary to check before for system's reachability.

Reachability refers to the ability of driving the system from one state to another using the available inputs. Checking for reachability is important because it provides valuable information about the controllability of the system. If the system is not reachable, it means that the control inputs cannot affect the state of the system, making it difficult or even impossible to achieve the desired control behavior. On the other hand, if the system is reachable, the control inputs can be used effectively to drive the system to the desired states, leading to improved control performance. [5]

By definition:

"A linear system is reachable if for any $x_0, x_f \in \mathbb{R}^n$ there exists a $T > 0$ and $u: [0, T] \to \mathbb{R}$ such that if $x(0) = x_0$ then the corresponding solution satisfies $x(T) = x_f$."

According to the Reachability rank condition theorem:

"A linear system is reachable if and only if the reachability matrix W_r

$$W_r = [B \quad AB \quad \dots \quad A^{n-1}B]$$

is invertible (has full rank)."

It is calculated in MATLAB that the matrix rank is 8:

Controllability

```
Co=ctrb(A,B);
rankco=rank(Co);
disp(['The controllability matrix rank is: ',num2str(rankco)])
```

```
The controllability matrix rank is: 8
```

2.3 State Observer

In order to estimate the internal states of the system, an observer has been designed for the control system. The observer uses the velocity of the motors and the control signal as inputs to estimate the current being drawn by each motor. The purpose of the observer is to provide accurate information about the internal states of the system, even in the presence of measurement noise or unmodeled dynamics.

In the closed loop observer in Figure 9 the lower system replication receives the feedback of the output estimation error. It helps the observer estimation of the system to correct its states estimation until the output error dynamics tend to zero.

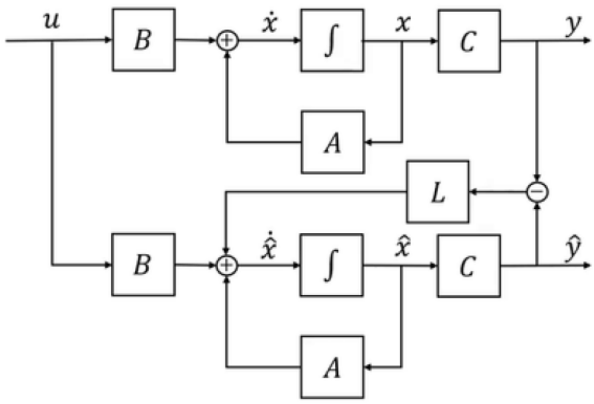

Figure 10.Generic closed loop observer block diagram [5]

The resulting observer system equations are: [5]

$$\dot{\hat{x}} = A\hat{x} + Bu + L\tilde{y}$$
$$\hat{y} = C\hat{x}$$

Equation 6

It is a task for the control designer to choose the proper L gain value to make the error estimation dynamics as fast as possible. A useful approach to choose that gain is the use of the Ackerman method to place the estimator system poles where desired. In this project they are placed 6 or more times to the left than the least dominant poles in the closed-loop system that will be later explained.

2.3.1 Discrete State Observer

When it comes to the observer implementation it is important to consider that the controller is a digital system. An alternative discrete implementation of the observer obtained via Euler method is [5]

$$\hat{x}(t_{k+1}) = \hat{x}(t_k) + (t_{k+1} - t_k)(A\hat{x}(t_k) + Bu(t_k) + L(y(t_k) - C\hat{x}(t_k)),$$

Equation 7

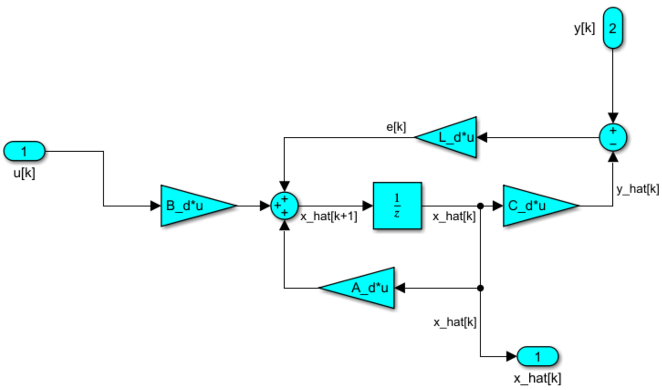

Figure 11. Simulink model of discrete state observer

The discrete system matrices representing the original system can be obtained combining Equation 6 and Equation 7. The matrices result in:

$$A_d = I + A.dt \qquad B_d = B.dt \qquad C_d = C.dt \qquad L_d = L.dt$$

Equation 8. Discrete system matrices

2.3.2 Discrete Kalman Filter

An improvement of a discrete state observer is a discrete Kalman Filter. A Kalman Filter considers measurement error and process error in its estimations, allowing more robust results.

Both the previous discrete state observer and the discrete Kalman Filter estimations will be compared in the results section.

The discrete Kalman Filter considers the following system equations [6]

$$x[k+1] = Ax[k] + Bu[k] + v[k]$$
$$y[k] = Cx[k] + w[k]$$

where v represents the process disturbance and w the measurement noise. Both are supposed to be zero mean and Gaussian. The design parameters R_v and R_w must be chosen and they represent the process disturbance covariance matrix and the measurement noise covariance matrix respectively.

Then the covariance of the estimation error $P_{\tilde{x}}[k]$ can be obtained as

$$P_{\tilde{x}}[k+1] = (A - L[k]C)P_{\tilde{x}}[k](A - L[k]C)^T + R_v + L[k]R_w L^T[k]$$

And the optimal observer gain $L[k]$

$$L[k] = AP_{\tilde{x}}[k]\, C^T\left(R_w + CP_{\tilde{x}}[k]\, C^T\right)^{-1}$$

It is important to note that the Discrete Kalman Filter is a recursive filter. This implies that for each time step the state estimation x and optimal gain L are calculated.

There are different algorithmic approaches for the filter implementation, and for the current project the Simulink Kalman Filter block is chosen.

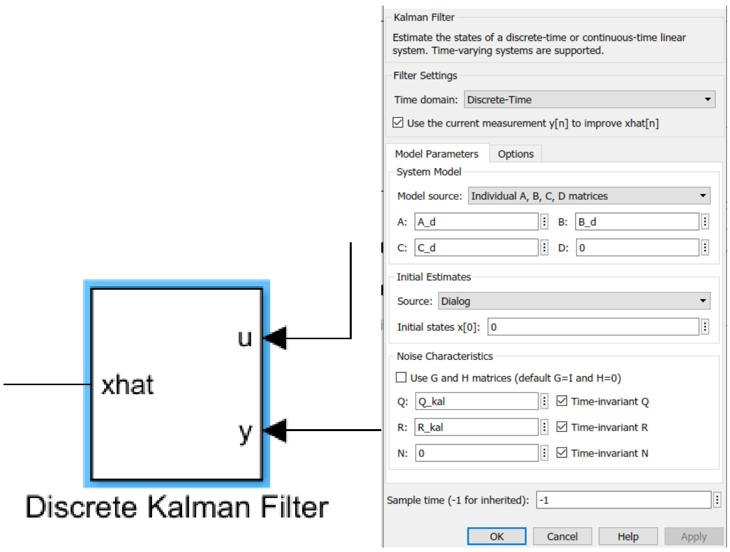

Figure 12. Simulink Kalman Filter block and its parameters configuration

R_v can be obtained directly from the process disturbance assumed for the system in Equation 5.

It will be an 8x8 identity matrix. R_w matrix will be obtained in the section 2.5 Sensor Modelling

2.4 State Feedback Control

Finally, once the observer has been designed, the entire state estimation can be accessed to implement a state feedback controller. [7]

The control law implemented is

$$u(t) = -K.x(t) + k_r.r(t)$$

Where K is the feedback gain, k_r is the steady-state reference gain and $r(t)$ is the reference input.

The closed-loop system dynamics becomes

$$\dot{x}(t) = (A - BK)x(t) + Bk_r r(t)$$

The control objective is to choose K such that the closed loop system dynamics gets a maximum overshoot of 4% and a settling time of 0.5 seconds in order to obtain a stable and fast enough response to references.

$$\xi = -\frac{\ln(M_p)}{\sqrt{\pi^2 + \ln(M_p)^2}} \approx \frac{1}{\sqrt{2}}$$

$$\omega_n = \frac{4}{\xi T_s} \approx 12$$

The characteristic equation is

$$s^2 + 2\xi\omega_n s + \omega_n^2$$

Operating in MATLAB the dominant poles are obtained to be $p_{1,2} = -8.4853 \pm 8.4853i$

The other 6 poles are placed to have a real part 5 and 7.5 times to the left.

2.5 Sensor Modelling

An incremental encoder is used to measure rotational speed. Incremental encoders typically consist of a rotating disk with a pattern of lines or marks on it, and a sensor that reads the pattern as the disk rotates. The sensor generates a series of pulses for each line or mark that it reads. By counting the number of pulses, it is possible to determine the position and speed of the motor.

The incremental encoder E6B2-CWZ6C OMRON has been selected. [8]

According to its datasheet it manages 2000 pulses per revolution.

To model it we have to consider that the discrete controller sample frequency is $f_s = 1KHz = \frac{1}{dt}$

To get the motor speed counting pulses we would be able to count pulses for $dt = 1ms$ before the system expects to receive a new value. Then speed would be calculated as

$w = \frac{Pulses\ counted}{dt} \quad [\frac{pulses}{s}]$ which in rad/s is $w = \frac{Pulses\ counted}{dt} . \frac{2.\pi\ rad}{P_{/r}} \quad [\frac{rad}{s}]$

That means the speed precision is $dw = \frac{1\ pulse}{dt} . \frac{2.\pi\ rad}{P_{/r}} = \frac{2.\pi\ rad}{P_{/r}.dt} = \frac{2.\pi\ rad}{2000.10^{-3}s} = \pi\frac{rad}{s}$

This can be modelled using a quantizer that outputs motor speed values separated by a minimum step of $\pi\frac{rad}{s}$.

It is not uncommon for an incremental encoder to encounter false pulses or missing pulses due to a variety of factors, including contamination of the encoder disk, exposure to electromagnetic interference, mechanical vibrations, power supply fluctuations, and other causes. To account for these sources of measurement error, white Gaussian noise is added to the encoder measurement model. It will be represented with zero-median and a standard deviation of 5% of speed value.

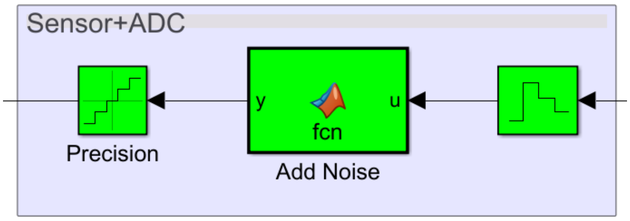

Figure 13. Sensor model in Simulink

Considering the sensor model now, the adequate R_w matrix for the Kalman Filter is calculated. It should take into account the composition of noise variances from the Add Noise block and the quantizer. The Add Noise block has a variance of $(0.05w)^2$ according to its standard deviation. Considering that motor speed values average is $200\frac{rad}{s}$, the Add Noise block variance can be approximated to $(0.05w)^2 = 100\frac{rad^2}{s^2}$

The quantizer variance [9] can be obtained as $\sigma_{ADC}^2 = \frac{q^2}{12} = \frac{\pi^2}{12} = 0{,}82$, which is much lower than the Add Noise block one, and thus can be ignored.

The R_w matrix is defined to be a diagonal 4x4 matrix with all its values equal to 100.

2.6 System Overview

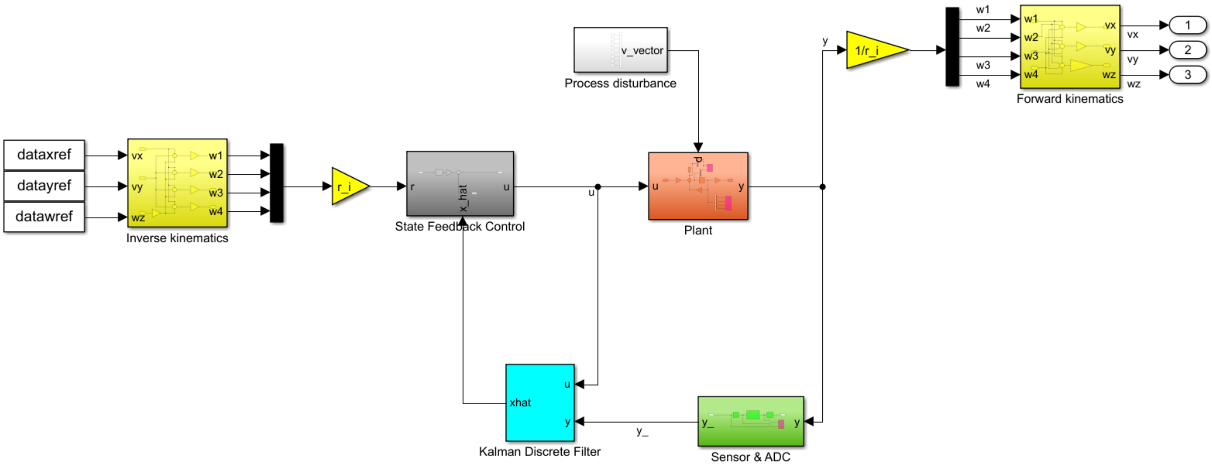

Figure 14. Complete model in Simulink

3. Results

A trajectory is defined for the omnidirectional mobile robot.

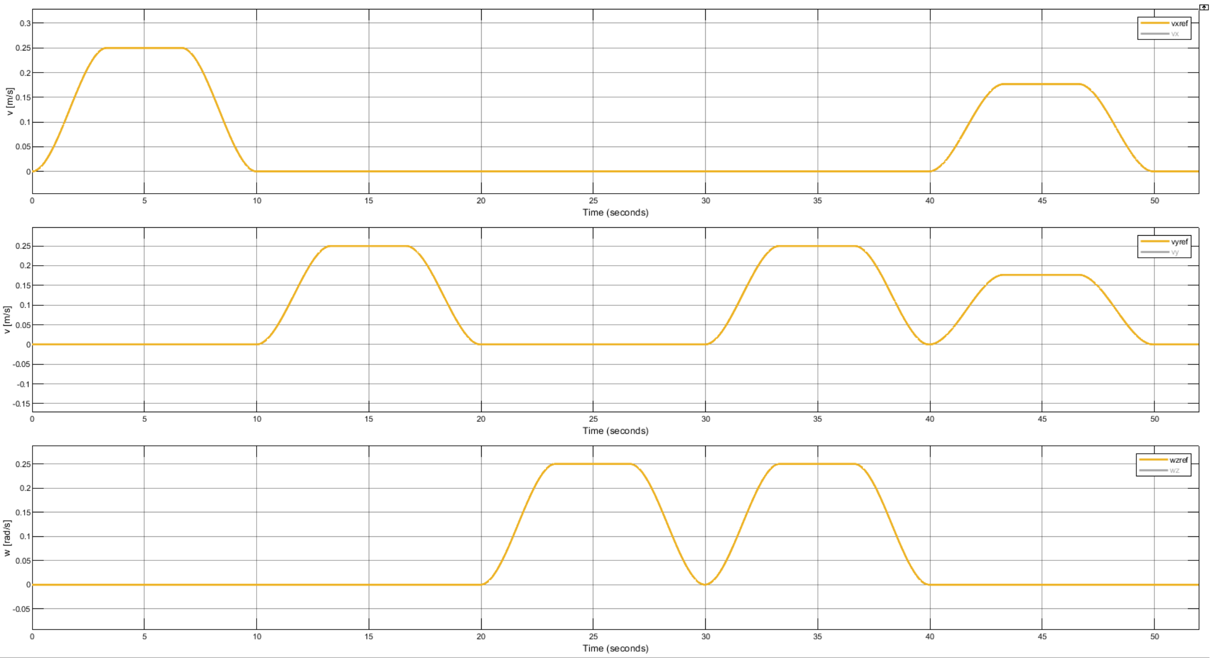

Figure 15. Velocity in x, in y and angular velocity references

The trajectory in Figure 15 involves a sequence of movements, beginning with forward motion, followed by a lateral movement, then a rotational movement, a combination of lateral and rotational movements and finally a diagonal displacement. Each movement lasts 10 seconds and consists of a constant section and two cubic interpolations, one for rise and one for fall.

The linear velocity requirements represent a peak velocity of 1m every 4 seconds in any linear direction. The angular velocity requirement is of 90º every 6 seconds, given that, as a surveillance robot with omnidirectional capabilities, it is no required to rotate fast.

Most of the plots showed in this section will refer exclusively to the first motor velocity and acceleration. This is due to the fact that all state space variables have similar behaviors in terms of noise. When considered important, other variables will be also shown.

3.1 Noise Analysis

In order to design a robust control system, different disturbances are injected.

3.1.1 Measurement Noise

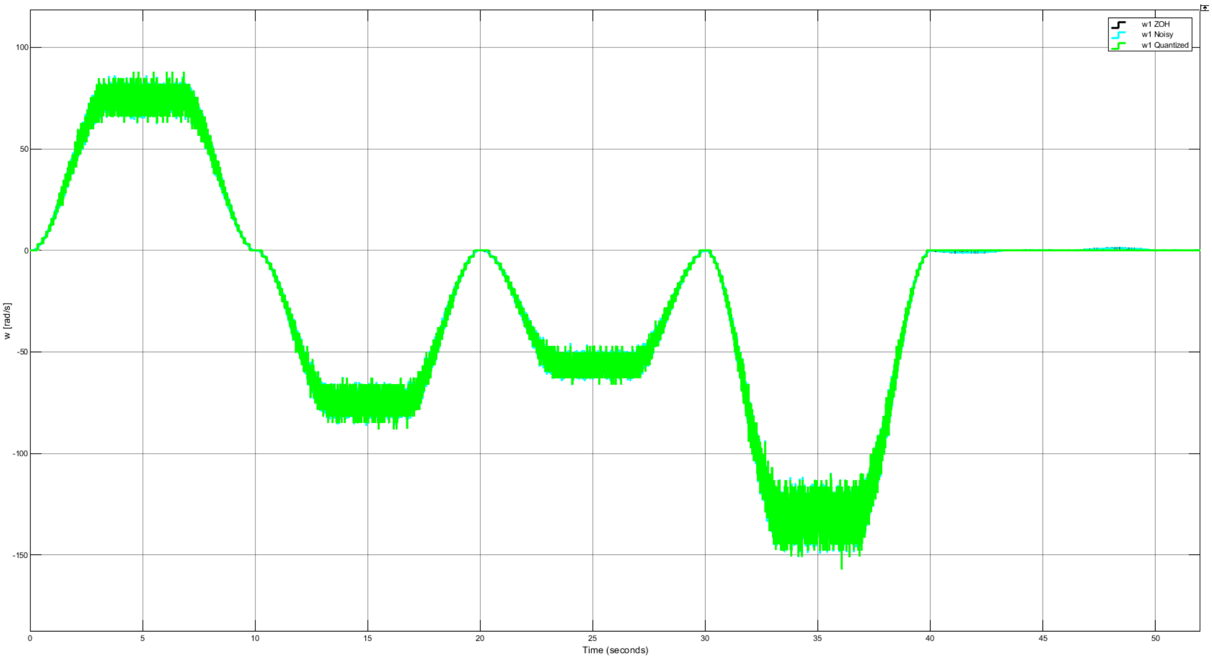

Figure 16. w1 noisy measurement after sensor

As analyzed in 2.5 Sensor Modelling, the pulse loss is proportional to the absolute speed, influencing the most in the higher speeds.

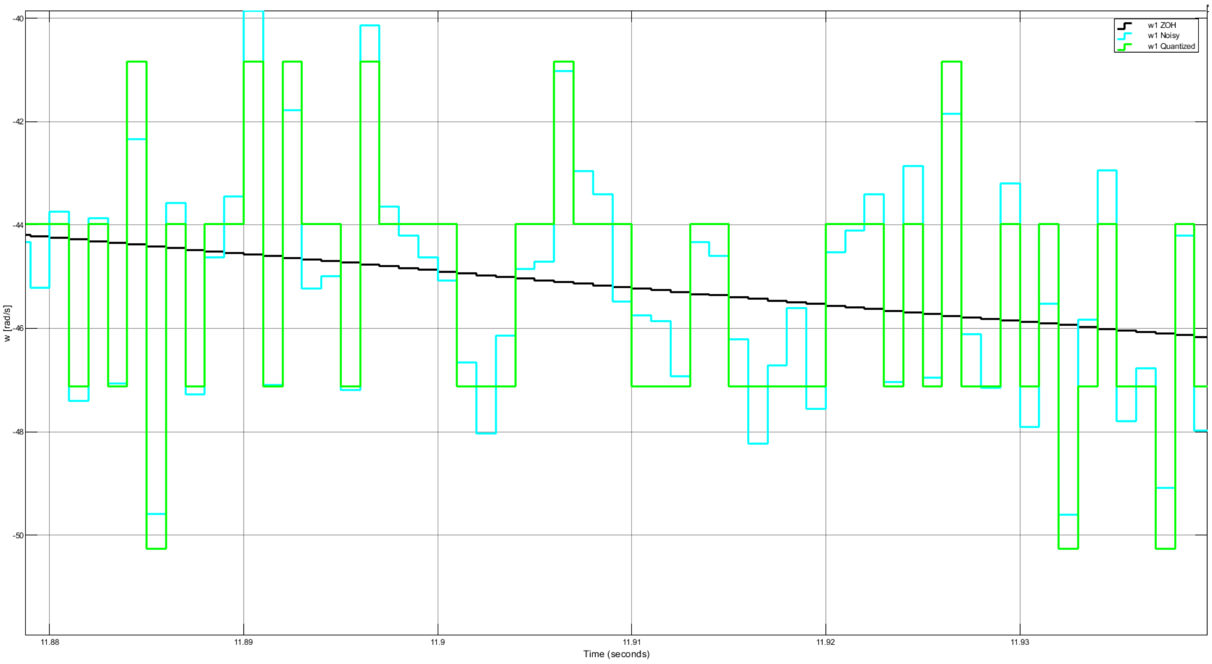

Figure 17. w1 noisy measurement after sensor. Close-up

Shortening the time range it is easy to distinguish the 3 measurement noise sources. The black signal stairs produced by the Zero Order Hold, the blue signal produced by the Add Noise block that represents the artificial pulse generated by external influences, and finally the green signal quantizer representing the encoder working principle.

3.1.2 Process Noise

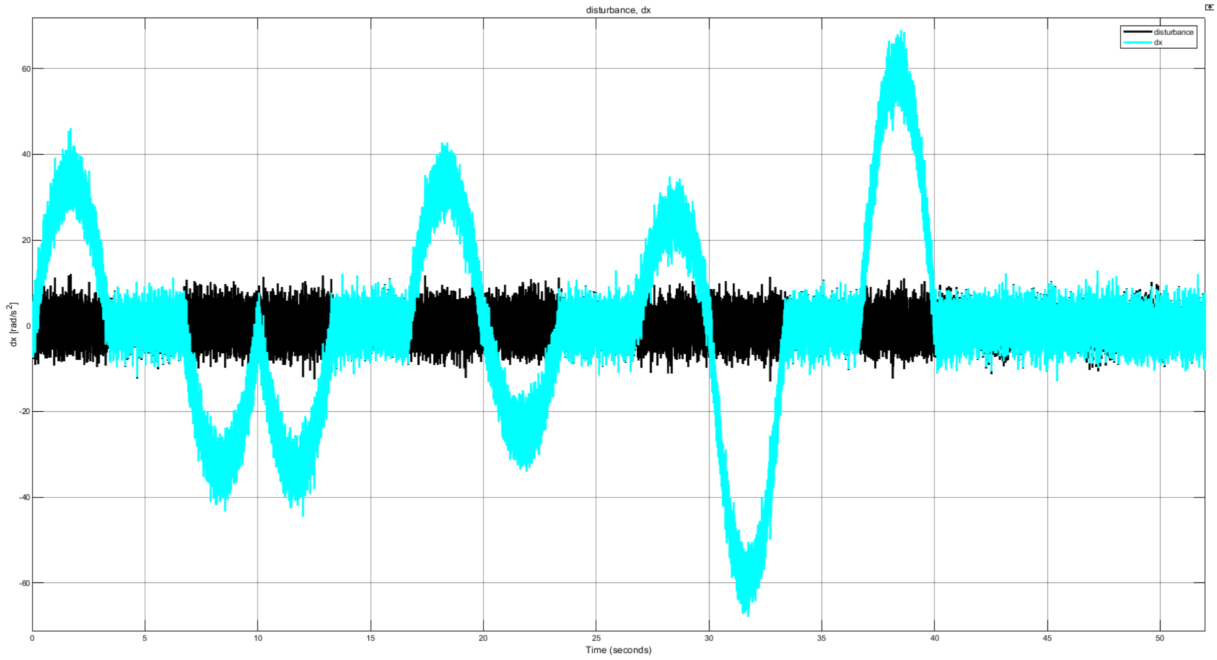

Figure 18. w1 acceleration value compared with process noise

The process noise mentioned in 2.2 State Space Model represents a significant amount of the state derivatives in the process. This particular case analyzes the first motor's acceleration.

3.2 Observer estimations

3.2.1 Discrete Observer

Except from this section, in the rest of the report this observer was not used. It is included in the analysis to show its inferior performance when contrasted with the Kalman Filter alternative.

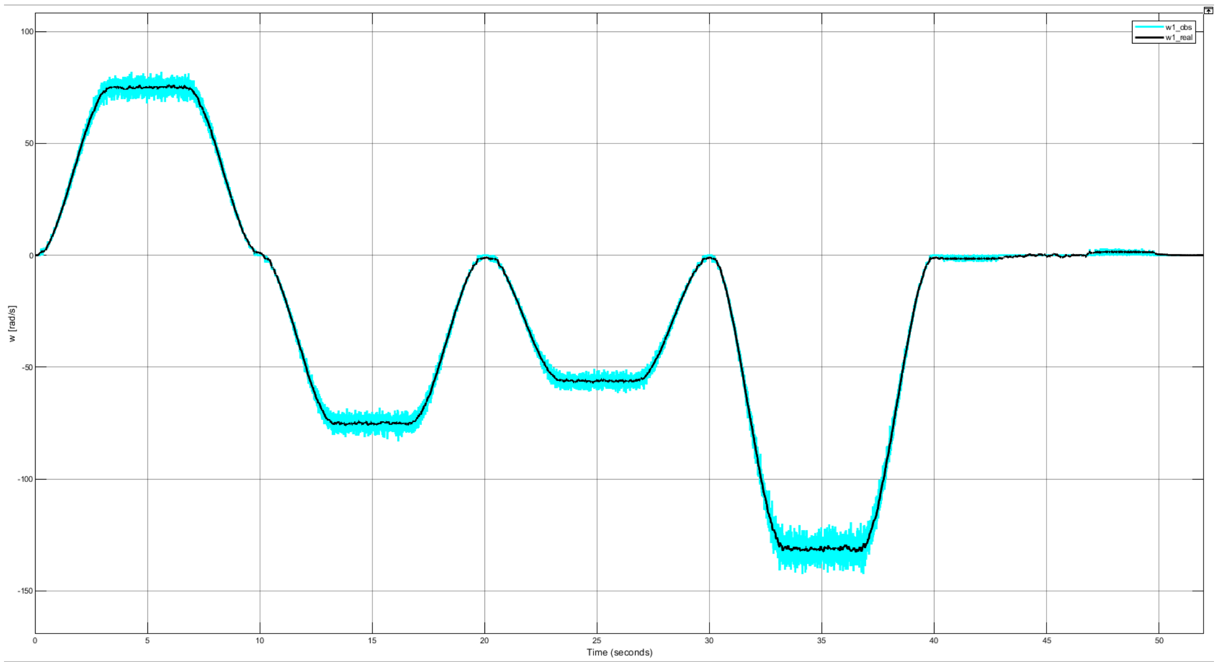

Figure 19. Discrete Observer Estimations (First 4 trajectories)

The discrete observer is affected significantly by the process and measurement noises. A prefiltering instance might have helped, but instead a Kalman Filter was implemented.

3.2.2 Kalman Filter Discrete Observer

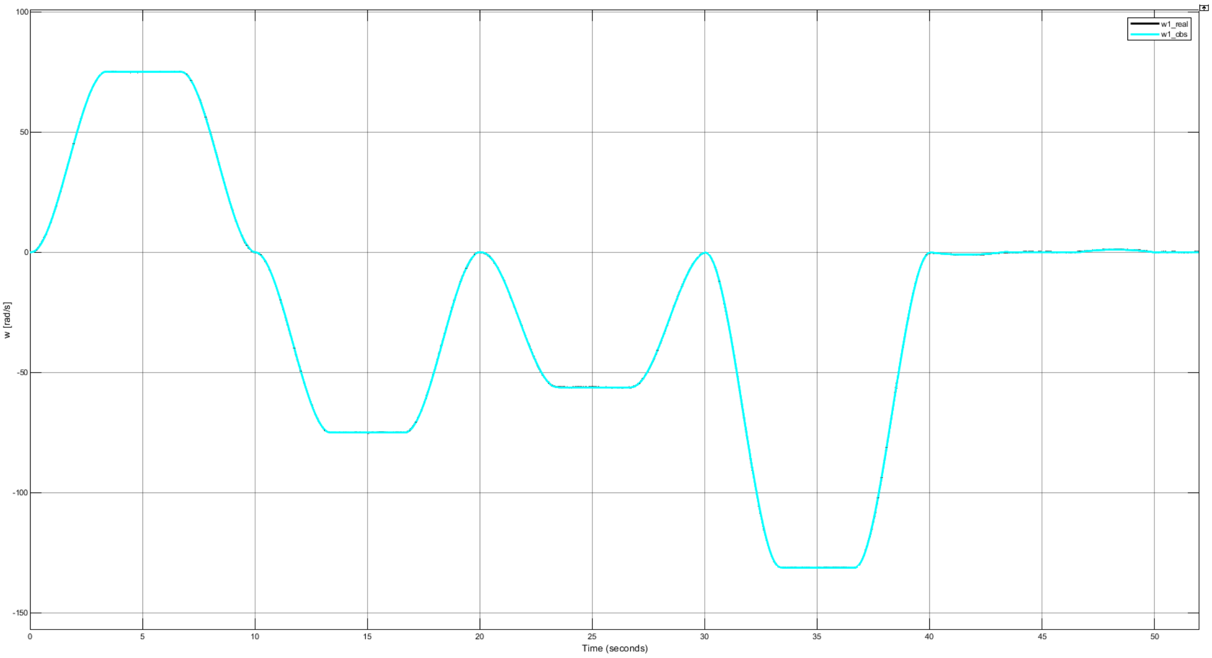

Figure 20. Kalman Filter w1 state estimation

The state variable and its estimation are practically overlapped when analyzed in the previous ranges.

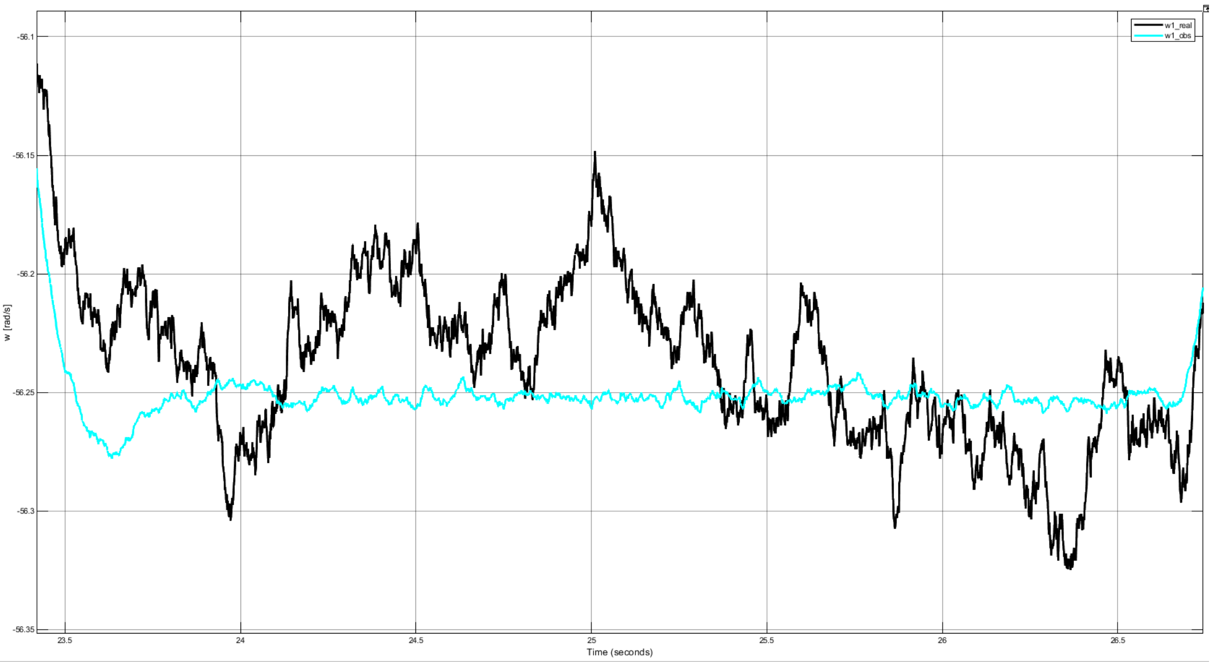

Figure 21. Kalman Filter w1 state estimation. Close-up

Expanding y and x axis the differences are appreciated. The state estimation error is in the order of a 0,2% of its real value.

3.3 Operation Limits

Given the motor specifications, it is extremely important to check that currents, voltages and angular velocities are below the operational limits.

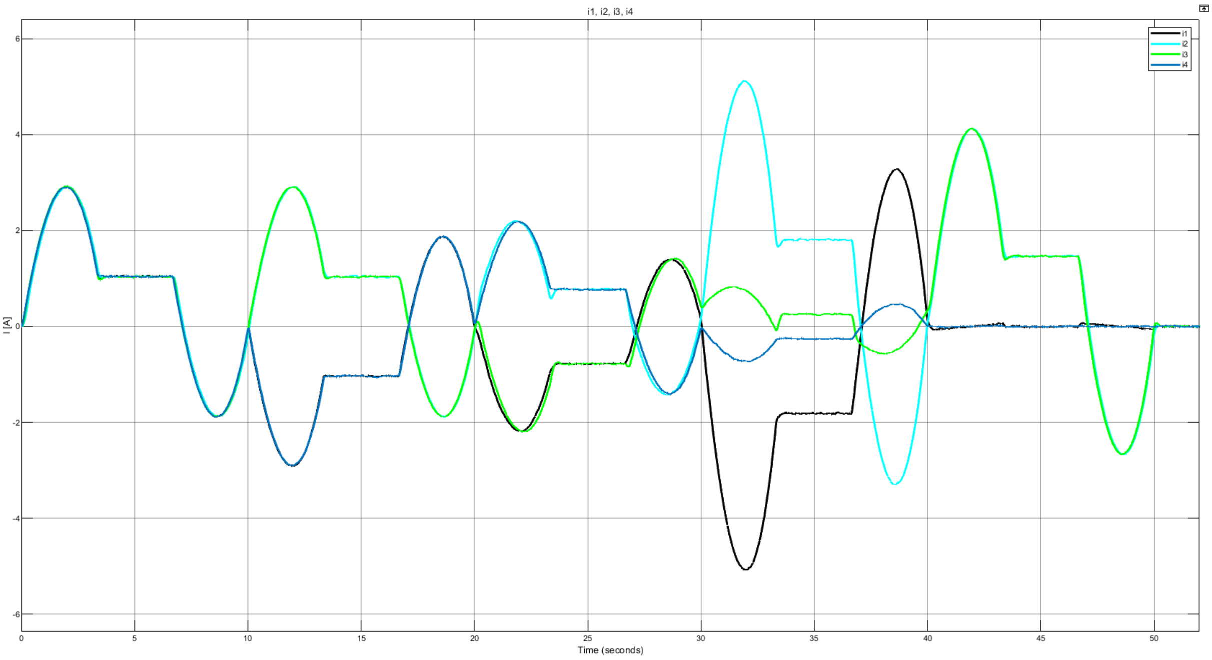

Figure 22. State Variables: Currents

The current limit is 5.5A, and the peak at 32 seconds gets closer to it, but only for a moment.

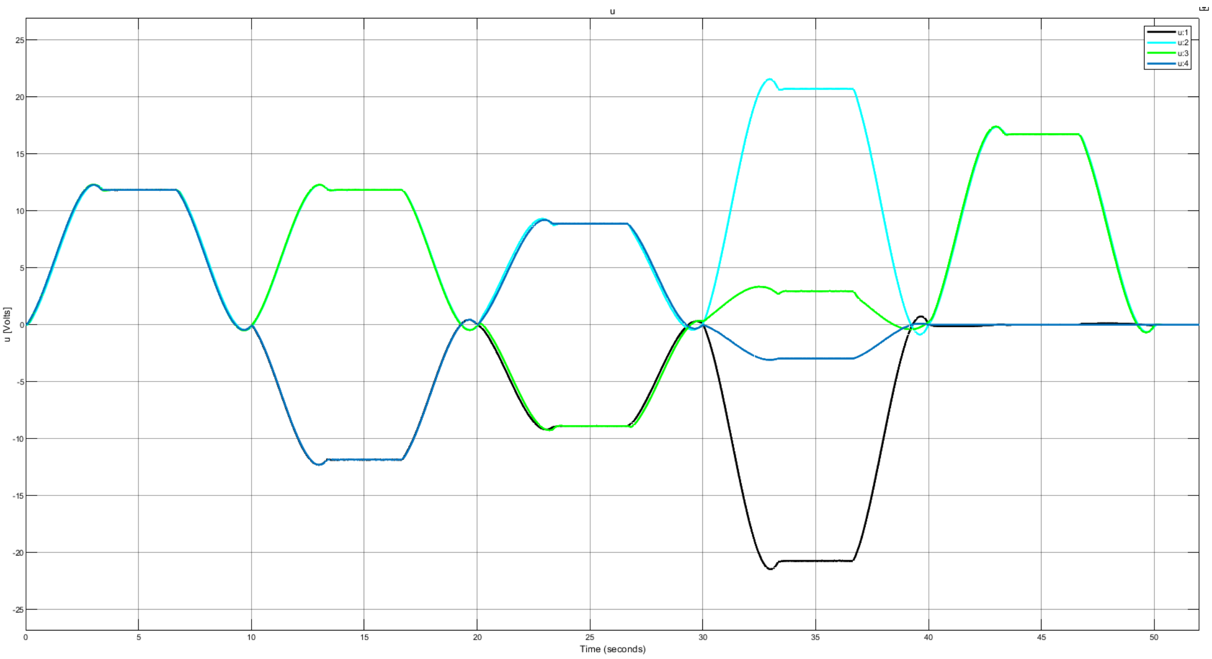

Figure 23.Control Signals: Voltages

The voltage limit of the motor is 48V, so its not exceeded.

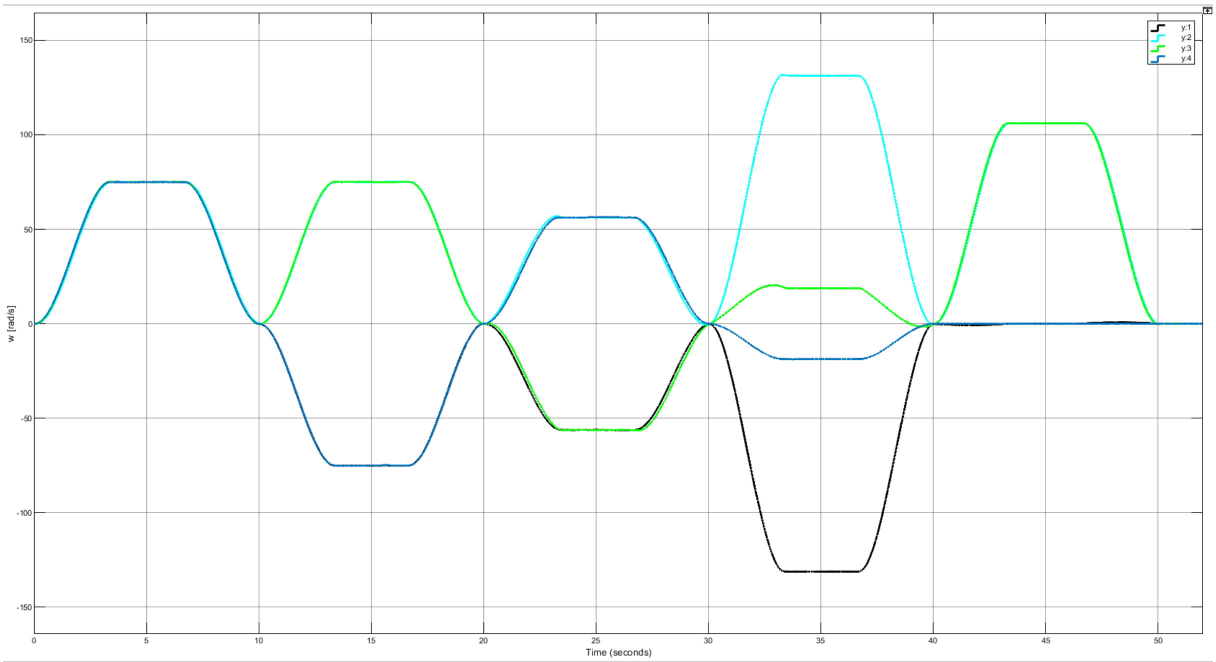

Figure 24. Output Variables: Angular velocity

The angular velocity limit of each motor is $243 \frac{rad}{s}$, so the reached values are far below the limit.

3.4 Outputs vs References

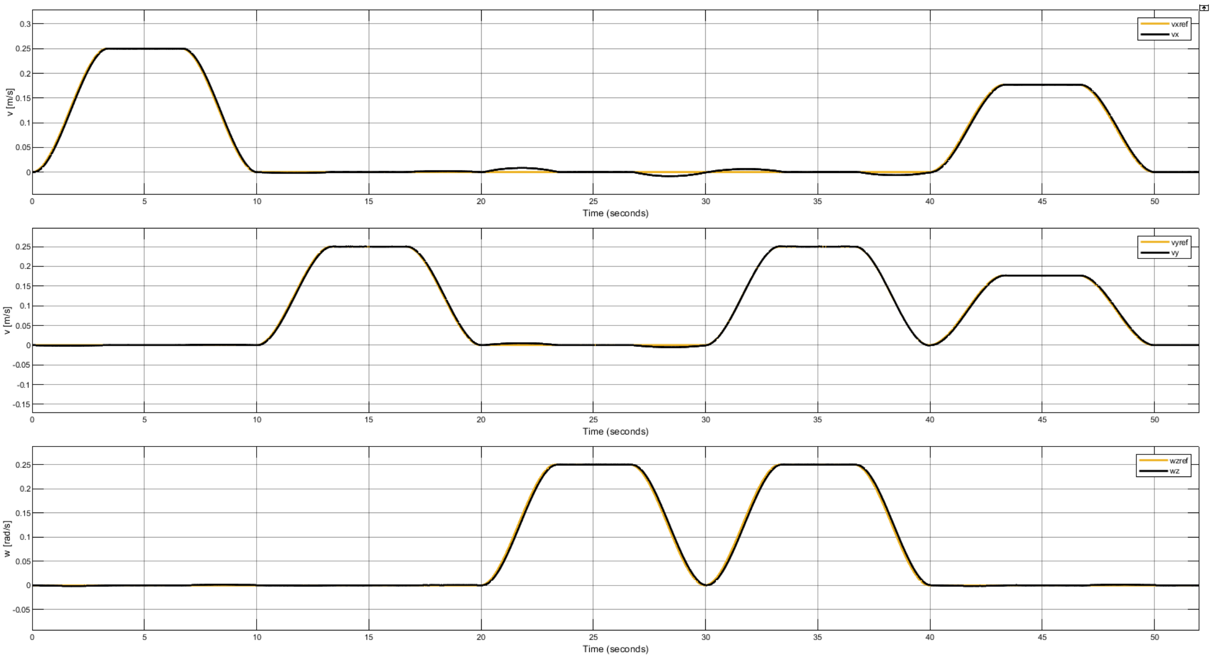

Figure 25. Output variables vs References

It can be seen that the system is following the expected values closely. The highest deviations from the desired values occur in v_x when the rotation sequence starts at 32 seconds. There, v_x has an absolute error of $0.01 \frac{rad}{s}$, which represents a 4% relative error to the maximum forward movement speed. It works successfully under the 5% maximum error expected.

3.4.1 Impulse disturbance response

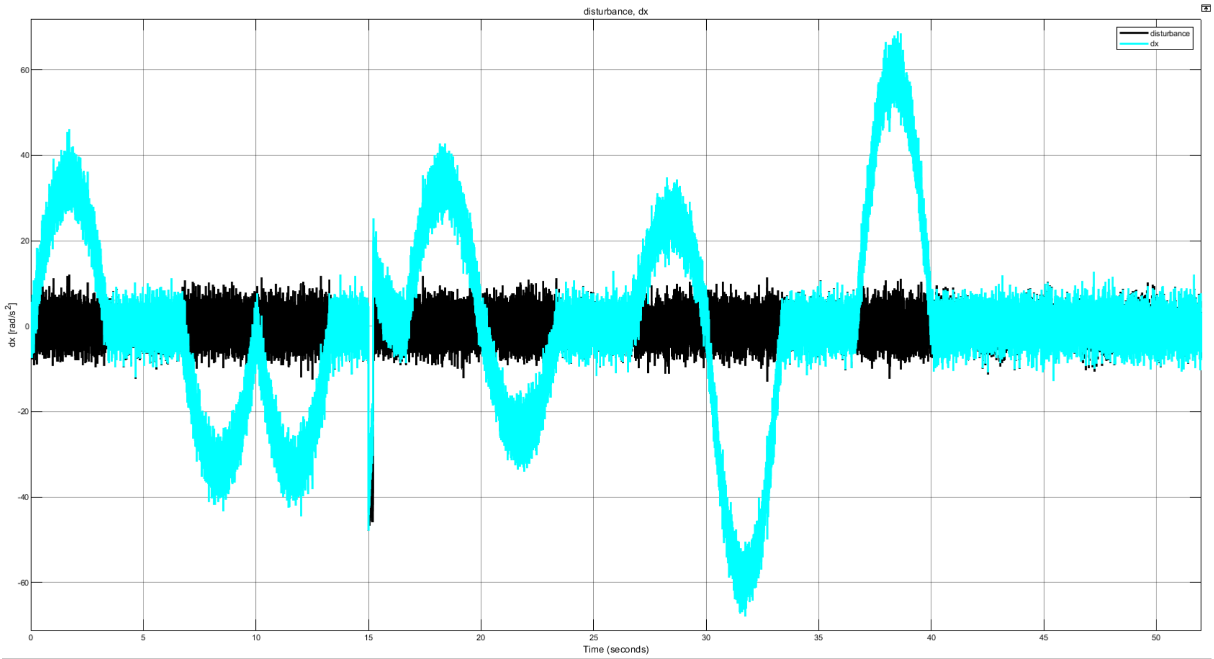

Figure 26. w1 acceleration value compared with process noise and impulse disturbance.

An impulse of $40\,\frac{rad}{s^2}$ is incorporated to the process noise at $15\ s$, representing a crash from the front with a mobile obstacle when displacement is lateral.

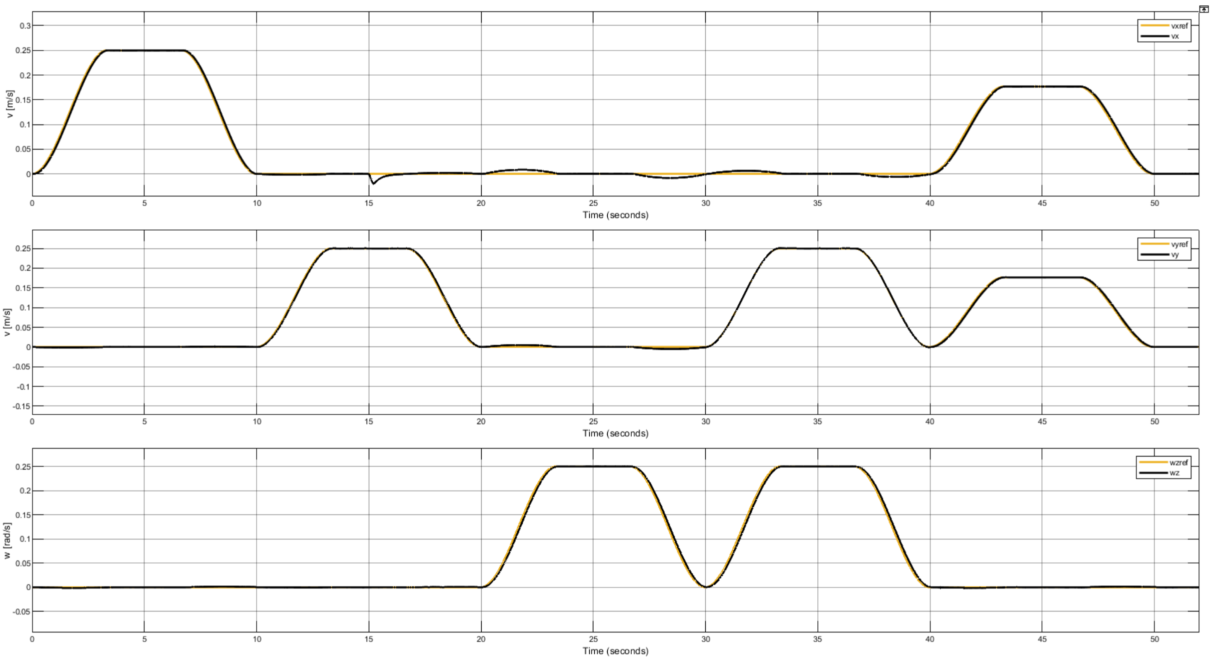

Figure 27. Front crash response when moving laterally. First 4

As seen in the figure above, only the x velocity is slightly affected by the impulse perturbation, and it recovers its expected value fast.

4. Conclusions

The control system developed is stable and capable of accurately tracking references with a 4% final tracking error.

A considerable amount of noise was added to the process and measurements. Most of the noise was efficiently filtered by the Kalman Filter, that estimated the system states with 0,2% error, being incredibly accurate. It was compared with a classic discrete observer, that had a poor behavior in presence of disturbances.

The closed loop system with state feedback control worked appropriately, and its velocity was limited by the control signal limitations. Even though 48 V could not be exceeded, the poles picked showed a great performance, being fast enough to satisfy the initial control objectives.

The encoder sensor modelling as a velocity sensor implied a compromise between the discrete controller time step and the number of pulses measured needed to make an acceptable estimation of velocity.

Simulink represented a great advantage in terms of system visualization, simulation and future potential iterations when implemented in a real system.

Possible improvements for the project could be an LQR controller implementation limiting control signal maximums. This could help push the boundaries of control performance. It would also be a future challenge to implement the designed controller in a raspberry pi, which will be used in the real application that inspired this project.

References

[1] Martínez, S. & Sisto, R. (2009) Control y Comportamiento de robots omnidireccionales. 19-20.34-35

[2] Castaño Giraldo, S. A. (2023, February 16). Modelo de Motor DC. Control Automático Educación. https://controlautomaticoeducacion.com/analisis-de-sistemas/modelo-de-motor-dc/

[3] Moog (2019). BSG23 High Performance Unique Stator Design. Datasheet. https://www.moog.com/content/dam/moog/literature/MCG/BSG23_DtS.pdf

[4] Gonzalez, R. (2020). Observability [PDF document]. Control and Systems. Cuyo National University, Mendoza, Argentina.

[5] Gonzalez, R. (2020). State estimation [PDF document]. Control and Systems. Cuyo National University, Mendoza, Argentina.

[6] Gonzalez, R. (2021). Kalman Filter [PDF document]. Control and Systems. Cuyo National University, Mendoza, Argentina.

[7] Gonzalez, R. (2020). State Feedback Control [PDF document]. Control and Systems. Cuyo National University, Mendoza, Argentina.

[8] Omron Electronics LLC (2017). Incremental 40-mm-dia Rotary Encoder E6B2-C. Datasheet. https://pdf1.alldatasheet.com/datasheet-pdf/view/1243611/OMRON/E6B2-CWZ3E.html

[9] Gonzalez, R. (2020). Typical stages in digital signal processing [PDF document]. Control and Systems. Cuyo National University, Mendoza, Argentina.

[10] Gonzalez, R. (2021). Guía para la presentación del proyecto final de Control y Sistemas [PDF document]. Control and Systems. Cuyo National University, Mendoza, Argentina.