Laboratory Report 4: Longitudinal state–space control of the balancing robot

Matteo Dal Nevo 2087919 Gabriele Boscarini 2063145
Nicola Mancassola 2097141 Batuhan Bilen 2071302

June 20, 2023
Group n.10 Thursday shift

1 Introduction

1.1 Activity Goal

The goal of this laboratory report is to present a solution to the problem of controlling longitudinally a two–wheeled balancing robot. The performance of the designed state-space controller will be discussed and tested both in simulation and on the real system.

1.2 System of interest: two–wheeled balancing robot

A simplified mathematical model for the balancing robot can be derived using the Euler Lagrangian approach, assuming that the motion occurs along a straight line and no lateral motion occurs. The robot can be represented as a multi–body system to each of which a reference frame is attached. It will be formulated a model for the mechanical part, one for the actuation system and one for the inertial measurement system. Referring to Fig.(1), reference frames are:

- **Body frame** {b}
- **Vehicle frame** {v},
- **World frame** {o},
- **Wheel frames(left/right)** {w,l}, {w,r}
- **Rotor frames (left/right)** {rot,l}, {rot,r}

The following set of generalized coordinates is introduced: The *robot tilt angle* θ, *Wheels angles* θ_l and θ_r, the *robot heading angle* ψ, the *rotor angles* $\theta_{rot,l}$ and $\theta_{rot,r}$. Because of the restriction to the longitudinal dynamics only, it can be assumed that $\psi = 0$. Under the pure rolling condition, the two wheel angles θ_l and θ_r are state to be identical and, in the following, γ will denote the common value of the wheels angles.

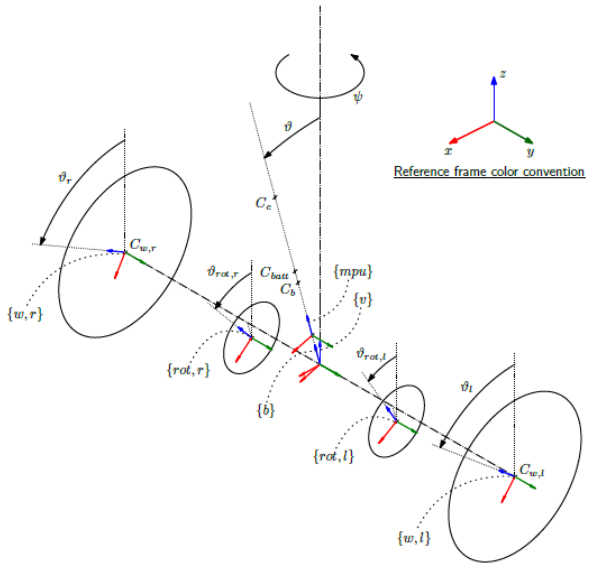


Figure 1: Simplified 3D view

1.2.1 Model of the mechanical part

The equations of motion of the mechanical part are derived using Euler Lagrangian approach. Let $\mathbf{q} = \begin{bmatrix}\gamma & \theta\end{bmatrix}$ be the vector of generalized variables and $\mathbf{x} = \begin{bmatrix}\mathbf{q} & \dot{\mathbf{q}}\end{bmatrix} = \begin{bmatrix}\psi & \theta & \dot{\psi} & \dot{\theta}\end{bmatrix}^T$ be the state variables. The linear model of the system can be written in state–space form

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{1}$$

where $u = 2\tau$, τ is the motor torque generated at the output shaft, and:

$$\mathbf{A} = \begin{bmatrix} \mathbf{0}_{2x2} & \mathbf{I}_{2x2} \\ -\mathbf{M}^{-1}\mathbf{G} & -\mathbf{M}^{-1} - \mathbf{F}_v \end{bmatrix}, \qquad \mathbf{B} = \begin{bmatrix} \mathbf{0}_{2x2} \\ \mathbf{M}^{-1} \end{bmatrix} \begin{bmatrix} 1 \\ -1 \end{bmatrix} \tag{2}$$

and

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} \\ \mathbf{M}_{21} & \mathbf{M}_{22} \end{bmatrix}, \qquad \mathbf{G} = \begin{bmatrix} \mathbf{G}_{11} & \mathbf{G}_{12} \\ \mathbf{G}_{21} & \mathbf{G}_{22} \end{bmatrix} \qquad \mathbf{F}_v = \begin{bmatrix} \mathbf{F}_{v,11} & \mathbf{F}_{v,12} \\ \mathbf{F}_{v,21} & \mathbf{F}_{v,22} \end{bmatrix} \tag{3}$$

with

$$
\begin{aligned}
M_{11} &= 2I_{w,yy} + 2N^2 I_{rot,yy} + (m_b + 2m_w + 2m_{rot})r^2 \\
M_{12} &= M_{21} = 2N(1-N)I_{rot,yy} + (m_b l + 2m_{rot} z^b_{rot})r \\
M_{22} &= I_{b,yy} + 2N(1-N)^2 I_{rot,yy} + m_b l^2 + 2m_{rot}(z^b_{rot})^2
\end{aligned}
$$

$$
\begin{aligned}
G_{11} &= G_{12}\, G_{21} = 0 \\
G_{22} &= -(m_b l + 2m_{rot} z^b_{rot})g
\end{aligned}
\tag{4}
$$

$$
\begin{aligned}
F_{v,11} &= 2(B + B_w) \\
F_{v,12} &= F_{v,21} = -2B \\
F_{v,22} &= 2B
\end{aligned}
$$

where $I_{w,yy}$ is the Principal Moment of inertia of the wheels, N is the gearbox ratio to describe mechanical coupling, $I_{rot,yy}$ is the Principal Moment of inertia of the DC gearmotor rotor, m_b is the mass of the robot body, m_w is the mass of the wheels, m_{rot} is the mass of the DC gearmotor rotor, z^b_{rot} is the Center–of–Mass coord of axis z, w.r.t. body frame b, g is the gravitational acceleration constant, r is the wheel radius, B is the Gearbox viscous friction coefficient (at output shaft), B_w is the Wheel viscous friction coefficient.
All the values of the parameters are reported in the Appendix A.2.

1.2.2 Model of the actuation system

With respect to the previous generalized variables and state vaiables, the equation of motion of the actuation system can be written in compact matrix formulas:

$$
\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{5}
$$

where

$$
\mathbf{A} = \begin{bmatrix} \mathbf{0}_{2x2} & \mathbf{I}_{2x2} \\ -\mathbf{M}^{-1}\mathbf{G} & -\mathbf{M}^{-1} - \mathbf{F}_{\mathbf{v}}^{'} \end{bmatrix}, \qquad \mathbf{B} = \frac{2Nk_t}{R_a} \begin{bmatrix} \mathbf{0}_{2x2} \\ \mathbf{M}^{-1} \end{bmatrix} \begin{bmatrix} 1 \\ -1 \end{bmatrix} \tag{6}
$$

and

$$
u = u_a = L_a \frac{di_a}{dt} + R_a i_a + k_e \frac{d\Delta\theta_{rot}}{dt} \tag{7}
$$

that is the electrical equation, where i_a is the current, R_a, L_a are the resistance and inductance of the armature circuit, k_e is the electric constant $\Delta\theta_{rot} = \theta_{rot} - \theta$ is the angular displacement of the rotor with respect to the stator, and u_a is the supply voltage to the armature circuit. Matrices $\mathbf{M}$ and $\mathbf{G}$ are defined as above, $\mathbf{F}_{\mathbf{v}}^{'}$ is defined as:

$$
\mathbf{F}_{\mathbf{v}}^{'} = \mathbf{F}_v + \frac{2N^2 k_t k_e}{R_a} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{8}
$$

1.2.3 Model of the inertial measurement system

The inertial measurement system provides the acceleration and angular velocity as outputs of the accelerometer and gyroscope. The acceleration is equal to the sum of the actual linear acceleration and the gravity acceleration vector. Defined y_a as the acceleration and y_g the angular velocity, it holds that:

$$\mathbf{y}_a = \begin{bmatrix} r\ddot{\gamma}\cos\theta + z^b_{mpu}\ddot{\theta} + g\sin\theta \\ r\ddot{\gamma}\sin\theta - z^b_{mpu}\dot{\theta}^2 - g\cos\theta \end{bmatrix} \tag{9}$$

$$\mathbf{y}_g = \dot{\theta} \tag{10}$$

where z^b_{mpu} is the z coordinate of the sensor center with respect to the body frame b.

2 Tasks, Methodologies and Results

All the controllers for the two–wheeled balancing robot presented in this report has been designed in order to simultaneously stabilize the robot body to its upward vertical position, and the robot base to a desired longitudinal position set-point.
These tasks are performed following two conventional methodologies:

- both the balance–and–position controllers are designed in the discrete–time domain (direct digital design).
- the state feedback matrix (state feedback controller gain) of the balance/position state–space controller is designed with the linear quadratic (LQ) optimal design techniques.

It is also desired to have the following performance specifications:

- perfect steady-state tracking of step position references.
- perfect steady-state rejection of constant voltage disturbances.

Moreover, at steady-state the maximum deviations for the first state, the second state and for the input value are

$$|\gamma - \gamma^*| < \pi/36 = \bar{\gamma}\ \ (5\text{ deg}), \qquad |\theta| < \pi/360 = \bar{\theta}\ \ (0.5\text{ deg}), \qquad |u| < 1\text{ V} = \bar{u}$$

where γ^* is a constant wheel angle position set–point.

2.1 Balance and position state–space control using LQR methods

According to the state-space description of the model mentioned in (1), the state of the plant is $\mathbf{x} = [\gamma, \theta, \dot{\gamma}, \dot{\theta}]^T$. It is assumed that the plant's state is fully accessible. However, since the measurements provided by the onboard sensors, namely the incremental encoder and the MPU (accelerometer and gyroscope) are not the desired ones, this assumption is not true in practice. Nevertheless, those quantities can be accurately determined implementing a "simple" state observer. For the estimate of the robot tilt angle θ, noting that the accelerometer–based estimate θ_a is more reliable at low frequency, while the gyroscope–based estimate θ_g is more reliable at high

frequency, one can resort to a complementary filtering approach for combining ("fusing") them together, thus yielding a final estimate $\hat{\theta}$ that is sufficiently accurate at every frequency.
Following this approach, a first order filter has been used:

$$H(s) = \frac{1}{T_c s + 1} \tag{11}$$

where T_c is calculated as $1/(2\pi f_c)$ and f_c denotes the cutoff frequency. Since the entire control system runs in the discrete–time domain, the filters must be discretized. $H(z)$ represents the discrete equivalent of H(s) with backward Euler discretization method. It is obtained such as

$$H(z) = \frac{C}{1 - (1 - C)\, z^{-1}} \quad \text{with} \quad C = \frac{T}{T_c + T} \tag{12}$$

where T represents the sampling time. Complementary filter block calculates the estimated tilt angle $\hat{\theta}$ as follows

$$\hat{\theta}[k] = C\, \hat{\theta}_a[k] + (1 - C)(\hat{\theta}[k-1] + T\, y_g[k]) \tag{13}$$

where y_g is the output of gyroscope. Moreover, the wheel angle γ is estimated thanks to state observer such as

$$\hat{\gamma} = \frac{\Delta\theta_{rot}}{N} + \hat{\theta} \quad \text{with} \quad \Delta\theta_{rot} = N(\gamma - \theta) \tag{14}$$

where $\Delta\theta_{rot}$ denotes the rotor angular displacement measured by the encoder.
After calculating the estimated tilt angle $\hat{\theta}$ and estimated wheel angle $\hat{\gamma}$, the angular speeds estimates $\dot{\hat{\gamma}}$ and $\dot{\hat{\theta}}$ are obtained from $\hat{\gamma}$ and $\hat{\theta}$. For such purpose, estimated tilt angle and estimated wheel angle are filtered from real derivative filter as following

$$H_w(z) = \frac{1 - z^{-N}}{NT} \quad \text{with} \quad N = 3 \tag{15}$$

2.2 Nominal Controller

To construct the discrete-time state-space controllers it is necessary to discretize firstly the continuous-time plant model (1) with exact discretization method, and sampling time equal to the controller sampling time $T = 0.01$ s. The resulting plant model $\mathbf{\Sigma}_d = (\mathbf{\Phi}, \mathbf{\Gamma}, \mathbf{H}, 0)$ have been considered for every controller (with suitable extension when considering a robust controller). For the design of the tracking controller the output matrix is chosen equal to $\mathbf{H} = [1\,0\,0\,0]$.
In this section, a nominal state-space controller is considered. The values of the feedforward gains are $\mathbf{N}_x = [1\,0\,0\,0\,0]^T$ for the state vector, $N_u = 0$ for the input and remain the same for all the controllers. The state feedback matrix $\mathbf{K}$ is designed using LQR methods, in order to minimize the discrete-time quadratic cost function

$$J = \sum_{k=0}^{+\infty} \mathbf{x}^T[k]\, \mathbf{Q}\, \mathbf{x}[k] + \rho\, r\, u^2[k] \tag{16}$$

Since the system is reachable and the realisation is minimal, the poles can be freely allocated. Considering the general formulation of the LQ problem, the state-costs matrix $\mathbf{Q}$ is forced to

be diagonal and following the steady-state specifications according to the Bryson's rule the cost weights **Q** and r are selected as follows

$$\mathbf{Q} = \begin{bmatrix} \frac{1}{\bar{\gamma}^2} & 0 & 0 & 0 \\ 0 & \frac{1}{\bar{\theta}^2} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 131 & 0 & 0 & 0 \\ 0 & 13131 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{and} \quad r = \frac{1}{\bar{u}} = 1$$

The extra weight ρ in (16) is used to adjust the relative weighting between the state and input contributions to the total cost function value. Therefore a possible choice could be

$$\rho \in \{500,\ 5000\}$$

In order to validate the nominal controller designed above and also the next controllers, two different tests are done.

1. The first test is with initial state $\mathbf{x}_0 = [0\,0\,0\,0]^T$, a step reference input applied at $t = 10$ s with amplitude $\gamma^* = 0.1/r$ rad where r is the wheel radius (this choice corresponds to a longitudinal position displacement of 10 cm), and no load disturbance.
   The state-feedback gains and the performance indices values are reported in Table (1) and Table (2). The parameter that are used in order to describe the performance of the system are: $|\gamma|$ that stands for the maximum displacement of the position from the reference, $|\theta|$ that is the maximum displacement of the tilt angle θ from the reference and the rise time t_r considered from 10 % to 90 % of the reference.

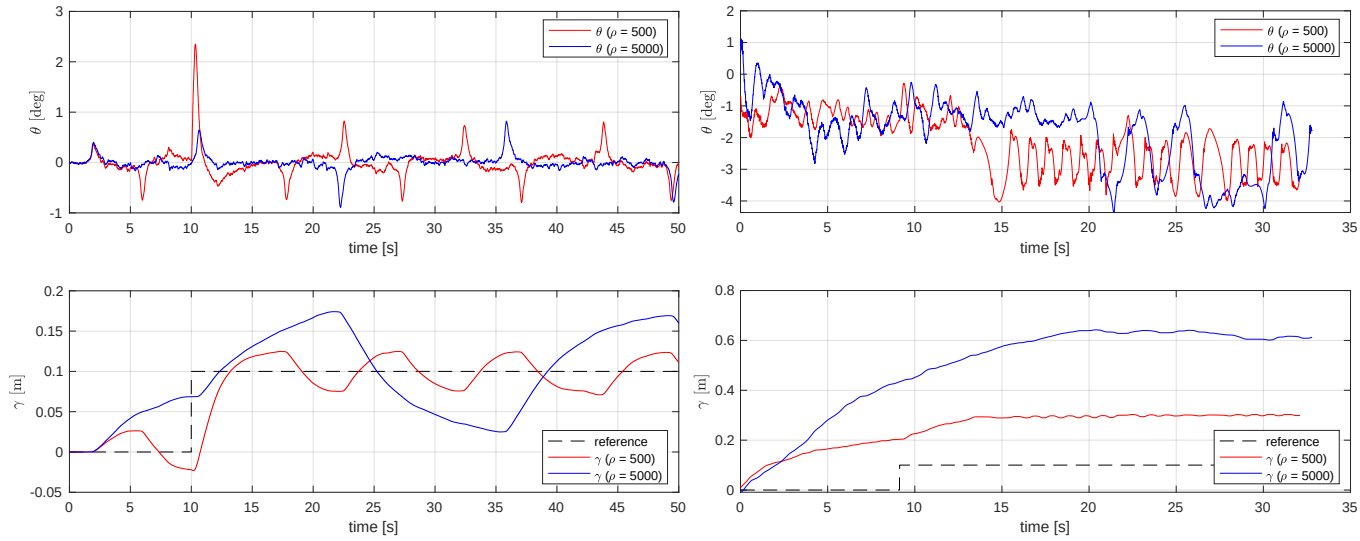
ρ	500	5000
$\boldsymbol{K}$	[-0.4754, -56.3184, -1.2726, -6.3470]	[-0.1507 -53.2141 -1.1488 -6.0237]

Table 1: Gain matrices for nominal controller

	Black-Box model	Real system
ρ	500	500
$t_r[s]$	1.9	-
$\|\gamma\|[m]$	0.025	0.2
$\|\theta\|[deg]$	2.35	4
ρ	5000	5000
$t_r[s]$	-	-
$\|\gamma\|[m]$	0.075	0.5
$\|\theta\|[deg]$	0.65	4.4

Table 2: Performance measures for a step reference input of $\gamma^* = 0.1$ m amplitude for the robust tracking LQR control, for $\rho \in \{500,\ 5000\}$

The step response obtained by applying the controllers to the black-box model and to the real system are shown in Fig.(2). The controller tries to stabilize the system, but since the control action is too slow the robot starts moving forward without taking into account the position reference.
It is evident that the specifications are not satisfied, indeed the controller is not able to guarantees the perfect tracking of the constant position set point in both the responses. This is not to be expected, since the overall closed-loop transfer function presents one pole in the origin and because of the I.M.P. the controller should be able to guarantees perfect tracking. This behavior may be due to the presence of the noise not modeled in the system, and also a difference in the plant parameters leads to a significantly position error. For this reasons, the rise time of the step response in Fig.(2b) is not present. It can be seen from the step response that increasing the weight ρ comports a slower response, because the cost function (16) is set in such a way that the control effort is oriented to keep low the input value rather than having a fast convergence of the state $\mathbf{x}[k]$.
In general, a smaller value of ρ, or an increase of the matrix Q diagonal values could lead to an improvement in performance. This supports the previous argument, for which the error between the response and the reference is also due to the presence of external disturbances. Indeed, in the robust case the tracking error goes to zero.



(a) Black box response

(b) Real system response

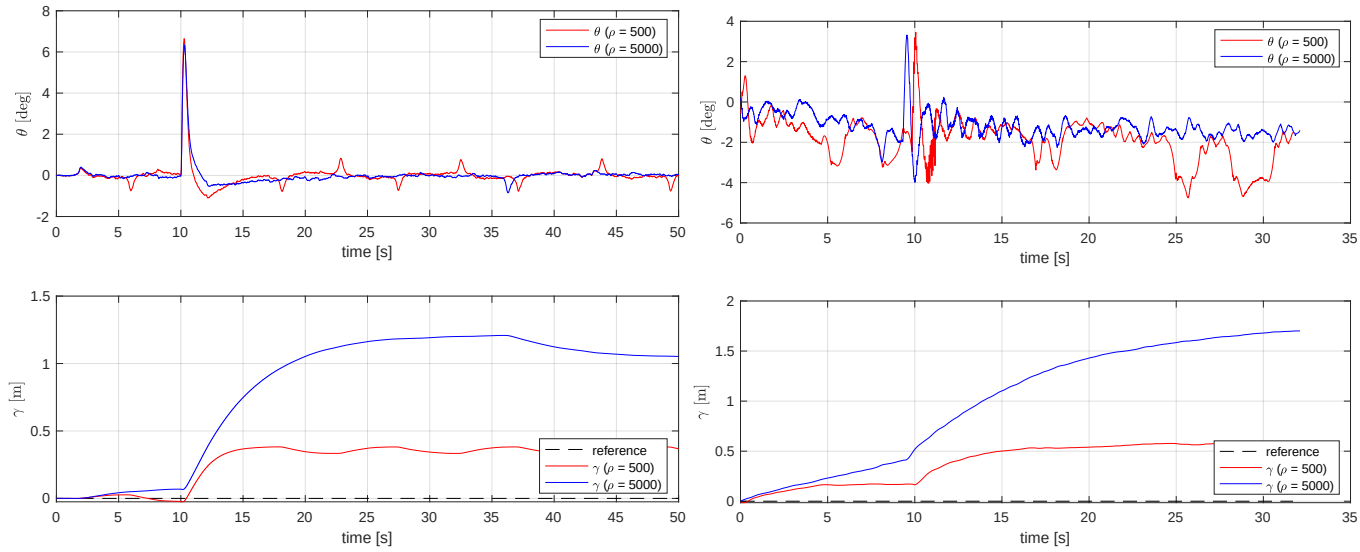
Figure 2: Responses to a step with $\gamma^* = 0.1$ m amplitude of the nominal tracking LQR

2. The second test is with initial state as in the previous point, a reference input $\gamma^* = 0$ m, and a load disturbance of amplitude 5.0 V applied at the voltage driver input at $t = 10$ s. The state-feedback gains and the performance indices values are reported in Table (3).

	Black-Box model	Real system
ρ	500	500
$\|\gamma\|[m]$	0.38	0.55
$\|\theta\|[deg]$	6.65	4
ρ	5000	5000
$\|\gamma\|[m]$	1.2	1.7
$\|\theta\|[deg]$	6.35	4.8

Table 3: Performance measures for a constant voltage disturbance of 5 V applied at the motor input for the nominal tracking LQR control, for $\rho \in \{500, 5000\}$

As before, the robot is not able to stay fixed in the upward position and moves forward. When the disturbance arrives the controller is unable to guarantee the perfect tracking of the constant position set–point (zero in this case), as shown in Fig.(3a-3b). Such disturbance could result from the application of a constant longitudinal force to the robot, similarly to what happens when the robot is forced to climb an inclined plane.
This performance is not considered satisfactory, since none of the specifications have been met. In order to increase performance a robust controller will be implemented.



(a) Black-Box model response

(b) Real system response

Figure 3: Responses to a step disturbance of 5 V applied at the motor input for the nominal tracking LQR

2.3 Robust Controller

A robust controller is then considered. In order to simultaneously track constant references and reject constant disturbances, an additional pole in the origin must be included in the controller transfer function, according to the internal model principle. In order to achieve this result, an additional fictitious state is considered, such that $x_I[k+1] = x_I[k] + (y[k] - r[k])$, where y is the system response and r the reference to be tracked. This leads to an enlarged state-space representation, where the state vector is given by $\mathbf{x}_e = [x_I, \mathbf{x}]^T$. The corresponding matrix description is therefore given by

$$\mathbf{\Phi}_e = \begin{bmatrix} 1 & \mathbf{H} \\ \mathbf{0}_{4\times 1} & \mathbf{\Phi} \end{bmatrix} \qquad \mathbf{\Gamma}_e = \begin{bmatrix} 0 \\ \mathbf{\Gamma} \end{bmatrix} \qquad \mathbf{H}_e = \begin{bmatrix} 0 & \mathbf{H} \end{bmatrix} \tag{17}$$

There are 5 poles to be allocated, namely the poles of the enlarged closed-loop system $\mathbf{\Phi}_e - \mathbf{\Gamma}_e \mathbf{K}$. The augmented state feedback matrix $\mathbf{K}_e = [K_I \, \mathbf{K}]$ can be designed again by resorting to LQR methods. In this case the cost matrix $\mathbf{Q}$ contains an extra weight q_{11} for the integrator state variable x_I. This weight must be chosen different from zero, otherwise the LQR cannot be designed. Indeed, if $q_{11} = 0$ the pair $(\mathbf{\Phi}_e, \sqrt{\mathbf{Q}})$ is not detectable and the Algebraic Riccati Equation has infinite solutions. In addition, the state-feedback matrices do not stabilize the closed-loop system.
However, the weight cannot be chosen with the Bryson's rule, since there is no reasonable and immediate way to identify the maximum deviation of the integrator state from its steady–state value. In practice, the weight q_{11} must be chosen by trial and error.
Two possible choices to consider for such weight are:

$$q_{11} \in \{0.1 \; 1\}$$

The matrix $\mathbf{Q}$ and the weight r are

$$\mathbf{Q} = \begin{bmatrix} q_{11} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{\bar{\gamma}^2} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\bar{\theta}^2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} q_{11} & 0 & 0 & 0 & 0 \\ 0 & 131 & 0 & 0 & 0 \\ 0 & 0 & 13131 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{and} \quad r = \frac{1}{\bar{u}} = 1$$

This architecture (robust controller) is tested for all the values of (ρ, q_{11}). In this section two different tests are done and the goal is to verify that the controller is capable of perfectly tracking a constant position reference at steady state, even in presence of a constant load disturbance. Tests are carried out imposing the same initial conditions of point 1 and 2 of the nominal case.

1. An initial state $\mathbf{x}_0 = [0\,0\,0\,0]^T$ and a step reference input applied at t = 10 s with amplitude $\gamma^* = 0.1$ m and no input disturbance are considered. The table (4) and table (5) show the feedback gain matrices and the performance indices obtained for a 0.1 m step reference, respectively.

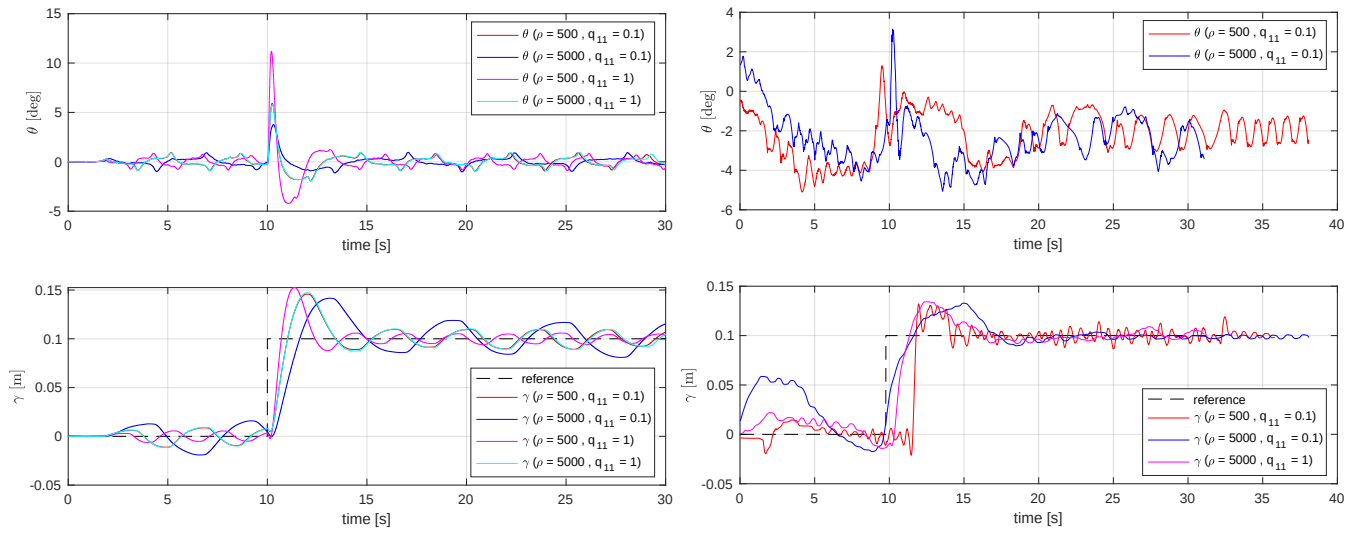
$(\rho,\ q_{11})$	K_I	$\boldsymbol{K}$
(500, 0.1)	-0.0130	[-1.9901 -66.1464 -1.7355 -7.4260]
(500, 1)	-0.0409	[-4.1535 -77.3850 -2.2991 -8.6581]
(5000, 0.1)	-0.0041	[-0.9541 -59.2920 -1.4228 -6.6899]
(5000, 1)	-0.0130	[-1.9182 -65.3451 -1.7093 -7.3529]

Table 4: Gains matrices for robust controller

Only the significant responses are reported for the robust controller parts. It can be seen in Fig.(4b) that the real system response presents the same issue as in the nominal case presented in point 1. Indeed the response does not follow the reference at the start of the simulation, so the robot start moving longitudinally. Then the controller steers to zero the position of the robot, trying to restore the stability. When the step reference changes its value, it can be seen that the controller is capable of perfectly tracking a constant position reference at steady state, even in presence of an unmodeled disturbance in the plant.
As the weight q_{11} associated with the integrator state increases, the integrator gain K_I increases as well and the rise time decreases. Indeed, the system approaches the steady-state value faster, as shown in Figure 4.
Considering the simulated results in the table above, the right trade-off between the maximum displacement of γ and rise time is obtained by the choice of ($\rho = 500$, $q_{11} = 1$), in particular for the real system response. However, the displacement of θ reaches its maximum value, equal to 12 deg, compared to the other results.

	Black-Box model	Real system
$(\rho,\ q_{11})$	(500, 0.1)	(500, 0.1)
$t_r[s]$	0.9	0.84
$\lvert\gamma\rvert[m]$	0.012	0.038
$\lvert\theta\rvert[deg]$	6.3	2.8
$(\rho,\ q_{11})$	(500, 1)	(500, 1)
$t_r[s]$	0.51	0.16
$\lvert\gamma\rvert[m]$	0.01	0.032
$\lvert\theta\rvert[deg]$	10.7	12
$(\rho,\ q_{11})$	(5000, 0.1)	(5000, 0.1)
$t_r[s]$	1.34	1.52
$\lvert\gamma\rvert[m]$	0.015	0.033
$\lvert\theta\rvert[deg]$	2.94	5
$(\rho,\ q_{11})$	(5000, 1)	(5000, 1)
$t_r[s]$	0.9	0.77
$\lvert\gamma\rvert[m]$	0.012	0.034
$\lvert\theta\rvert[deg]$	6.3	3

Table 5: Performance measures for a step reference input of $\gamma^* = 0.1$ m amplitude for the robust tracking LQR control, for $\rho \in \{500,\ 5000\}$ and $q_{11} \in \{0.1,\ 1\}$



(a) Black-Box model response

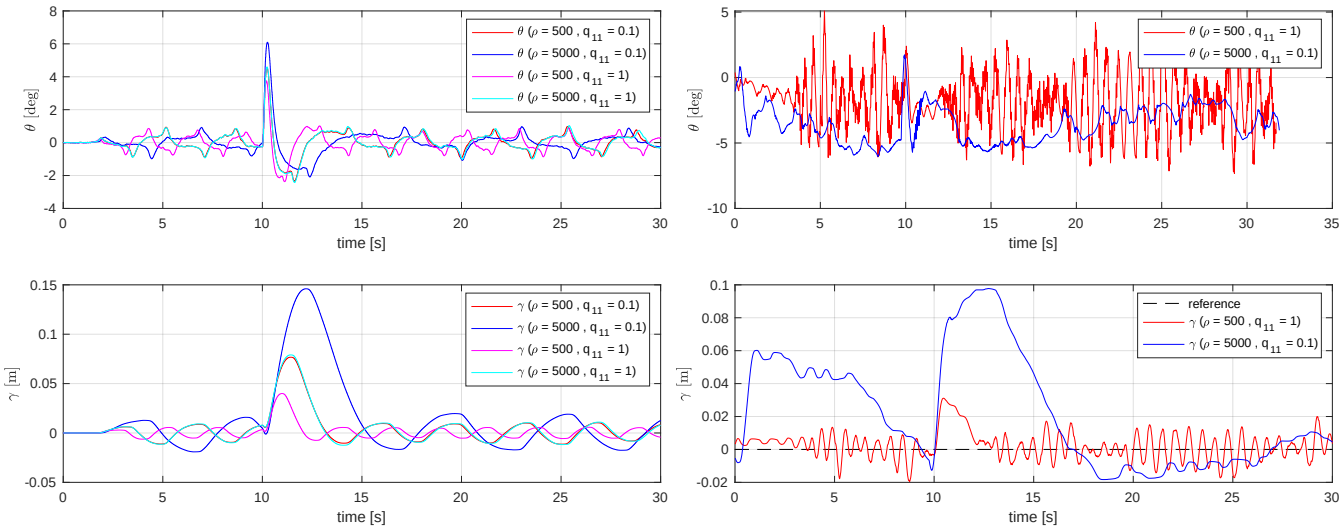
(b) Real system response

Figure 4: Responses to a step with $\gamma^* = 0.1$ m amplitude of the robust tracking LQR

2. In the following simulation, a load disturbance of amplitude 5.0 V is applied at the voltage driver input at $t = 10$ s. The state-feedback gains and the performance indices values are reported in Table (6) and Table (7), respectively. It turned out that, for the real system, the optimal response with respect to $|\gamma|$ is given by the following parameters: $\rho = 500$, $q_{11} = 1$. Indeed, the cost associated to the state variables is higher then the cost associated to the input and therefore the controller stabilizes the robot quickly, at the expense of a large displacement of the tilt angle and a saturation of the control input.

	Black-Box model	Real system
$(\rho,\ q_{11})$	(500, 0.1)	(500, 0.1)
$\lvert\gamma\rvert[m]$	0.075	0.068
$\lvert\theta\rvert[deg]$	4.5	3.54
$(\rho,\ q_{11})$	(500, 1)	(500, 1)
$\lvert\gamma\rvert[m]$	0.04	0.03
$\lvert\theta\rvert[deg]$	3.75	7.3
$(\rho,\ q_{11})$	(5000, 0.1)	(5000, 0.1)
$\lvert\gamma\rvert[m]$	0.146	0.097
$\lvert\theta\rvert[deg]$	6.1	6.04
$(\rho,\ q_{11})$	(5000, 1)	(5000, 1)
$\lvert\gamma\rvert[m]$	0.075	0.057
$\lvert\theta\rvert[deg]$	4.5	6.3

Table 6: Performance measures for a constant voltage disturbance of 5 V applied at the motor input for the nominal tracking LQR control, for $\rho \in \{500, 5000\}$ and $q_{11} \in \{0.1, 1\}$



(a) Black-Box model response

(b) Real system response

Figure 5: Responses to a step disturbance of 5 V applied at the motor input for the robust tracking LQR

2.4 PI implementation with nominal controller

Ideally, if the same motor control signal is sent to the motors control board, both motors should rotate at the same angle. But in the real environment there are many reasons for which the motors rotate at different angles under the same input signals, such as defects of the mechanical, electrical motors parameters, terrain and hindrance on the ground. Therefore, a method to synchronized both the wheels is needed and it is necessary to add an additional controller.
Let's define ψ as the angle between the initial direction $\psi(0) = 0$ of the robot and the possible lateral drift. In our tests, the goal is to remove this behaviour, so this angle should tend to 0.
With such condition, it is possible to calculate ψ according to the following formula:

$$\psi = \frac{r}{w}(\theta_r - \theta_l)$$

Where the parameters $\theta_r - \theta_l$ represent the difference in wheel speeds measured by the encoder.
To control ψ, a PI controller is implemented and the transfer function is the following:

$$C(s) = K_P + \frac{K_I}{s} \quad \text{where} \quad K_P = 3.3 \qquad K_I = 0.7$$

In order to move along a perfect straight line, the motors are controlled with different voltages according to the following relationships:

$$u_r = u_\Sigma + u_\Delta$$

$$u_l = u_\Sigma - u_\Delta$$

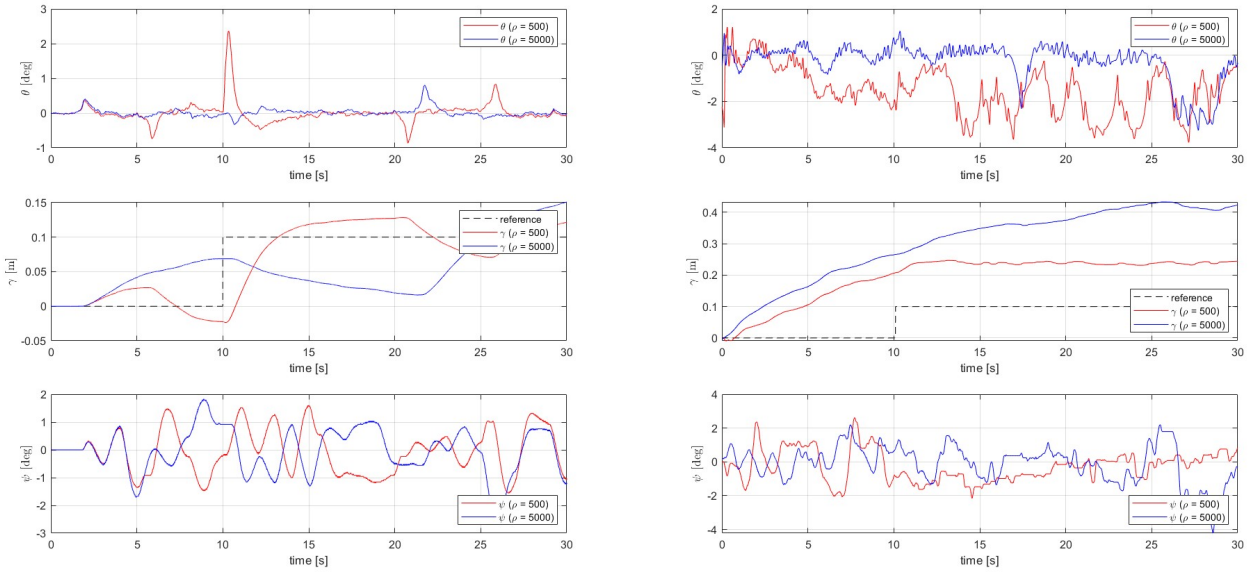
where the parameter u_Σ is given by the longitudinal control, while u_Δ is given by the PI controller.

1. As in the previous points, in order to test the effectiveness of the balance-position controller, an initial state $x_0 = [0\,0\,0\,0]^T$ and a step reference input applied at $t = 10$ s with amplitude $\gamma^* = 0.1$ m and no input disturbance are considered.
   Firstly, the nominal controller with the parameters reported in Table 1 is tested. Let's define $|\psi|$ as the maximum angle reached by our robot during the tracking tests. The step responses obtained by applying the controllers to both the models are shown in Figure (6) and the specifications are reported in Table (7).

	Black box	Real system
ρ	500	500
$t_r[s]$	1.7	-
$\|\gamma\|[m]$	0.028	0.15
$\|\theta\|[deg]$	2.3	3.8
$\|\psi\|[deg]$	1.59	2.1
ρ	5000	5000
$t_r[s]$	-	-
$\|\gamma\|[m]$	0.83	0.32
$\|\theta\|[deg]$	0.78	3.2
$\|\psi\|[deg]$	1.79	2.5

Table 7: Performance measures for a step reference input of $\gamma^* = 0.1$ m amplitude for the nominal tracking PI-LQR control, for $\rho \in \{500, 5000\}$

One can notice that the step response obtained introducing the combined longitudinal and heading–angle state–space controller does not change much compared to that without it, with respect to the state variable γ. Indeed, the LQR controller is not able to guarantee the perfect steady-state tracking as before. But, in this case, the PI controller is indirectly able to maintain the angle θ in a smaller range.
The plots of the new control variable ψ shows that the PI control action is perfectly able to stabilize the yaw angle in the equilibrium position (0 rad), thanks to the integral action.
Also in this case, as for the nominal LQR controller, the robot moves longitudinally in order to maintain its upward position. But since the LQR controller is very slow, the robot is only able to remain in equilibrium, without ever reaching the desired reference tracking and therefore the specifications are not satisfied.



(a) Black box response

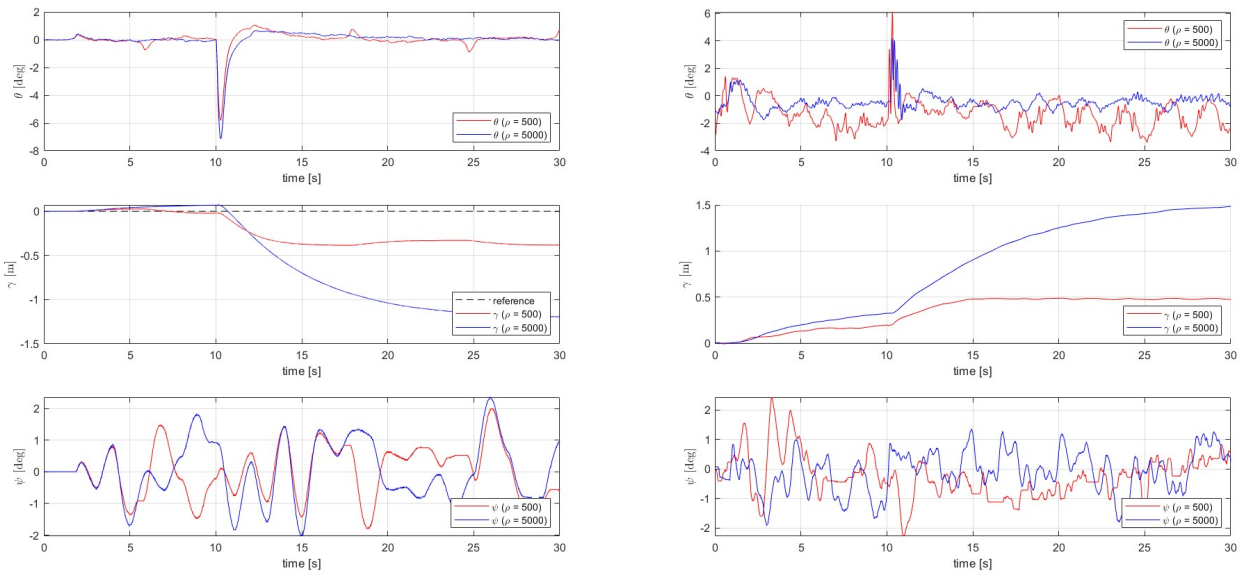
(b) Real system response

Figure 6: Responses to a step with $\gamma^* = 0.1$ m amplitude of the nominal tracking PI-LQR

2. As second test, a load disturbance of amplitude 5.0 V is applied at the voltage driver input at $t = 10$ s in order to evaluate the ability of the system to reject the constant disturbances.

	Black-Box model	Real system
ρ	500	500
$\|\gamma\|[m]$	0.38	0.48
$\|\theta\|[deg]$	5.7	6.06
$\|\psi\|[deg]$	2.0	2.3
ρ	5000	5000
$\|\gamma\|[m]$	1.19	1.48
$\|\theta\|[deg]$	7.1	4.19
$\|\psi\|[deg]$	2.3	1.7

Table 8: Performance measures for a constant voltage disturbance of 5 V applied at the motor input for the nominal tracking PI-LQR control, for $\rho \in \{500, 5000\}$



(a) Black box response

(b) Real system response

Figure 7: Responses to a step disturbance of 5 V applied at the motor input for the nominal tracking PI-LQR

The presented images, Fig.(7), compare the simulation in the black box with the real world model. With a nominal controller, the system is not able to reject the constant disturbance. Therefore, the final position does not meet the specifications. However, with the implementation of the PI controller, we can observe that the robot tends to correct its direction when the ψ angle is different from 0.
Despite the continuous movements made by the robot to maintain a upward position, this control approach is not able to respects the specifications.

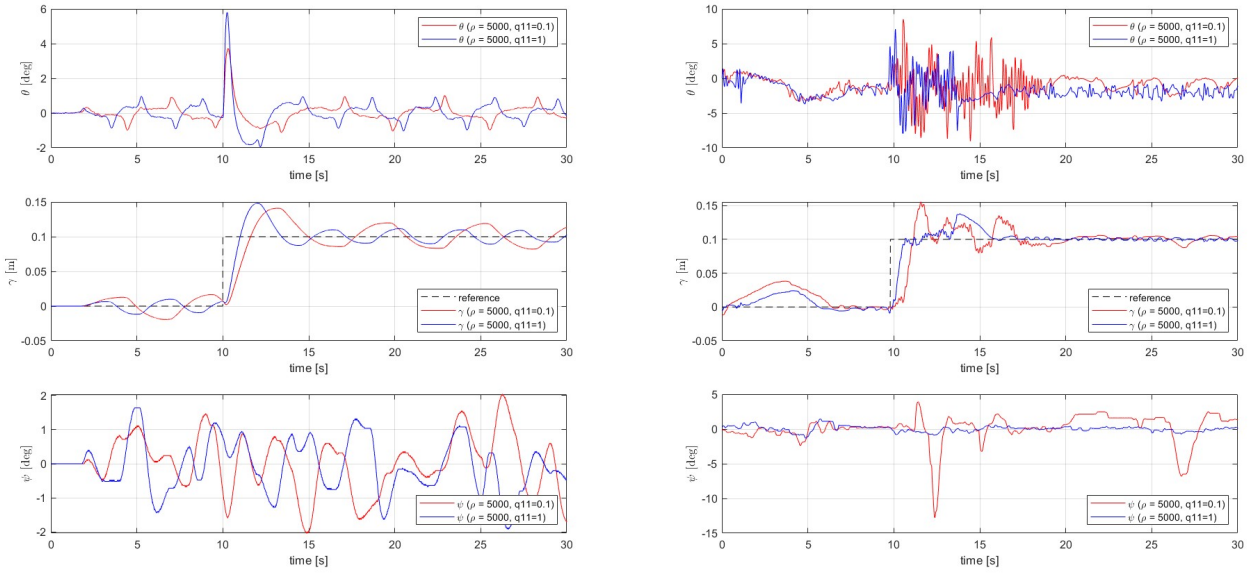
2.5 PI implementation with robust controller

In order to improve the system's response, the controller is modified as described in section 2.3 to ensure asymptotic stability of the output. Several tests are done by varying ρ and q_{11}.
The resulting gain matrices are reported in Table (4).

1. As previously mentioned, the response to a step input for $\gamma^* = 0.1$ m is initially tested. Not all the responses are reported in the Fig.(8) for the robust controller tests, only the most significant ones for the explanation of the results. However, all the specifications are reported in Table (9), except for $\rho = 500$ and $q_{11} = 1$. This particular test is reported in the Appendix A.1.
   The responses, with respect to θ and γ, are pretty similar to those reported for the robust controllers only. While, as you can see in Fig.(8b), when the step reference occurs the response of the angle ψ has diverged of 12 deg. The introduction of the PI controller is able to correct this deviation quickly, and the robot moved back along its original trajectory.

	Black-Box model	Real system
$(\rho,\ q_{11})$	(500, 0.1)	(500, 0.1)
$t_r[s]$	0.63	0.85
$\lvert\gamma\rvert[m]$	0.046	0.072
$\lvert\theta\rvert[deg]$	5.86	4.9
$\lvert\psi\rvert[deg]$	1.5	4.3
$(\rho,\ q_{11})$	(500, 1)	(500, 1)
$t_r[s]$	0.39	-
$\lvert\gamma\rvert[m]$	0.054	-
$\lvert\theta\rvert[deg]$	11,1	-
$\lvert\psi\rvert[deg]$	1.75	-
$(\rho,\ q_{11})$	(5000, 0.1)	(5000, 0.1)
$t_r[s]$	1.44	1.0
$\lvert\gamma\rvert[m]$	0.040	0.054
$\lvert\theta\rvert[deg]$	3.68	8.5
$\lvert\psi\rvert[deg]$	1.95	12
$(\rho,\ q_{11})$	(5000, 1)	(5000, 1)
$t_r[s]$	0.92	0.5
$\lvert\gamma\rvert[m]$	0.048	0.037
$\lvert\theta\rvert[deg]$	5.67	6.8
$\lvert\psi\rvert[deg]$	1.63	1.4

Table 9: Performance measures for a step reference input of $\gamma^* = 0.1$ m amplitude for the robust tracking PI-LQR control, for $\rho \in \{500,\ 5000\}$ and $q_{11} \in \{0.1,\ 1\}$



(a) Black box response

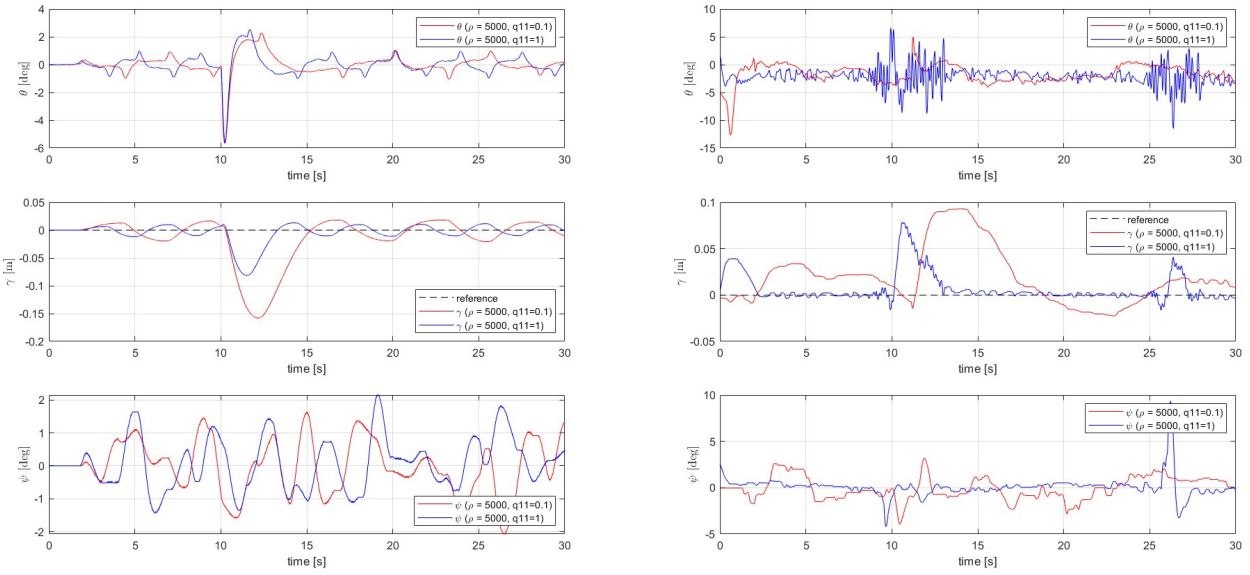
(b) Real system response

Figure 8: Responses to a step with $\gamma^* = 0.1$ m amplitude of the robust tracking PI-LQR

2. In the following test a constant voltage disturbance of 5 V is applied to the motor input to check the response of the system. All the specifications are reported in Table (10) and also the responses are shown in the Fig.(9).

	Black-Box model	Real system
$(\rho,\ q_{11})$	(500, 0.1)	(500, 0.1)
$\lvert\gamma\rvert[m]$	0.079	0.079
$\lvert\theta\rvert[deg]$	5.6	7.8
$\lvert\psi\rvert[deg]$	2.23	3.38
$(\rho,\ q_{11})$	(500, 1)	(500, 1)
$\lvert\gamma\rvert[m]$	0.04	0.042
$\lvert\theta\rvert[deg]$	4.7	9.98
$\lvert\psi\rvert[deg]$	2.03	11.59
$(\rho,\ q_{11})$	(5000, 0.1)	(5000, 0.1)
$\lvert\gamma\rvert[m]$	0.15	0.092
$\lvert\theta\rvert[deg]$	5.6	5.0
$\lvert\psi\rvert[deg]$	1.6	3.9
$(\rho,\ q_{11})$	(5000, 1)	(5000, 1)
$\lvert\gamma\rvert[m]$	0.079	0.078
$\lvert\theta\rvert[deg]$	5.6	6.48
$\lvert\psi\rvert[deg]$	2.1	9.2

Table 10: Performance measures for a constant voltage disturbance of 5 V applied at the motor input for the robust tracking PI-LQR control, for $\rho \in \{500,\ 5000\}$ and $q_{11} \in \{0.1,\ 1\}$



(a) Black box response

(b) Real system response

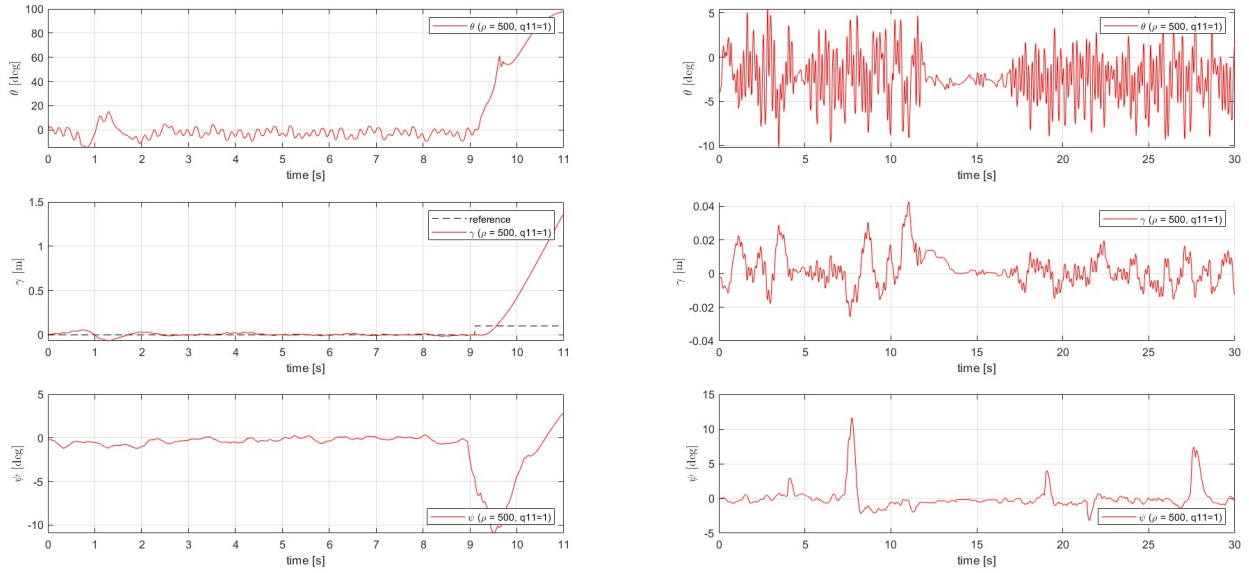
Figure 9: Responses to a step disturbance of 5 V applied at the motor input for the robust tracking PI-LQR

The presented images compare the simulation in the black box with the real-world model. The reported data refers to the cases of higher and lower stability of the real system: the response with parameters $\rho = 5000$ and $q_{11} = 1$ is better in terms of velocity in correcting the tilt angle displacement. All the controllers are able to maintain the upward vertical position without moving longitudinally. It can be observed that for the more stable controller described earlier, there is less oscillation in the angle ψ both in step tracking and disturbance rejection. With more effective vertical control, the wheels will move less, limiting the mechanical and electrical differences of the motor that cause the rotation of the robot itself.

A Appendix

A.1 Robot fall

The controllers presented so far are able to stabilize the robot in a vertical position and maintaining its stability during all the transient response, even when a constant disturbance is introduced. A different behaviour occurs when a robust controller with ($\rho = 500$, $q_{11} = 1$) are considered.
The gain matrix $\mathbf{K}$ reported in Table (4). Additionally, a PI controller is used to correct the yaw angle deviation that tends to cause curvature, as described in section 2.4. The responses regarding straight displacement and disturbance rejection are reported in Fig.(10).



(a) Response to a step reference input of $\gamma^* = 0.1$ m (b) Response to a step disturbance of 5 V

Figure 10: Responses of the real system of robot for the robust tracking PI-LQR

From the experiment, it can be noticed that the controller effectively tracks the assigned reference, thanks to the integral action introduced by the augmented state. But this particular choice of the parameters makes the system highly sensitive to small deviations from our reference and tends to make the control very aggressive and unstable.
This particular behaviour can be observed in the angle θ, which exhibits strong oscillations (more

than 10 deg), in order to reach the desired reference as quickly as possible. Initially, the robot starts from a vertical position with an angle θ close to 0, so it does not have any issue maintaining the upward equilibrium. However, when a step reference of $\gamma^* = 0.1$ m is introduced, the system aggressively tries to reach the asymptotic value, causing the robot to tip over.
A similar situation can be seen in the disturbance rejection test, but in this case, despite the significant oscillations, it manages to maintain the vertical position.

A.2 Matlab code

```
%% General parameters and conversion gains

% controller sampling time
Ts = 1e-2;

% gravity acc [m/s^2]
g = 9.81;

% conversion gains
rpm2rads = 2*pi/60; % [rpm]   > [rad/s]
rads2rpm = 60/2/pi; % [rad/s]   > [rpm]
rpm2degs = 360/60; % [rpm]   > [deg/s]
degs2rpm = 60/360; % [deg/s]   > [rpm]
deg2rad = pi/180; % [deg]   > [rad]
rad2deg = 180/pi; % [rad]   > [deg]
g2ms2 = g; % [acc_g]   > [m/s^2]
ms22g = 1/g; % [m/s^2]   > [acc_g]
ozin2Nm = 0.706e-2; % [oz*inch]   > [N*m]

% robot initial condition
x0 =[ ...
    0, ... % gam(0)
    5*deg2rad, ... % th(0)
    0, ... % dot_gam(0)
    0]; % dot_th(0)

%% DC motor data

% motor id: brushed DC gearmotor Pololu 30:1 37Dx68L mm

% electromechanical params
mot.UN = 12; % nominal voltage
mot.taus = 110/30 * ozin2Nm; % stall torque @ nom voltage
mot.Is = 5; % stall current @ nom voltage
mot.w0 = 350 * 30 * rpm2rads; % no load speed @ nom voltage
mot.I0 = 0.3; % no load current @ nom voltage

mot.R = mot.UN/mot.Is; % armature resistance
```

```
39 mot.L = NaN; % armature inductance
40 mot.Kt = mot.taus/mot.Is; % torque constant
41 mot.Ke = (mot.UN - mot.R*mot.I0)/(mot.w0); % back EMF constant
42 mot.eta = NaN; % motor efficiency
43 mot.PN = NaN; % nominal output power
44 mot.IN = NaN; % nominal current
45 mot.tauN = NaN; % nominal torque
46
47 % dimensions
48 mot.rot.h = 30.7e-3; % rotor height
49 mot.rot.r = 0.9 * 17e-3; % rotor radius
50
51 mot.stat.h = 68.1e-3; % stator height
52 mot.stat.r = 17e-3; % stator radius
53
54 % center of mass (CoM) position
55 mot.rot.xb = 0; % (left) rot CoM x pos in body frame
56 mot.rot.yb = 42.7e-3; % (left) rot CoM y pos in body frame
57 mot.rot.zb = -7e-3; % (left) rot CoM z pos in body frame
58
59 mot.stat.xb = 0; % (left) stat CoM x pos in body frame
60 mot.stat.yb = 52.1e-3; % (left) stat CoM y pos in body frame
61 mot.stat.zb = -7e-3; % (left) stat CoM z pos in body frame
62
63 % mass
64 mot.m = 0.215; % total motor mass
65 mot.rot.m = 0.35 * mot.m; % rotor mass
66 mot.stat.m = mot.m - mot.rot.m; % stator mass
67
68 % moment of inertias (MoI) wrt principal axes
69 mot.rot.Ixx = mot.rot.m/12 * (3*mot.rot.r^2 + mot.rot.h^2); % MoI
      along r dir
70 mot.rot.Iyy = mot.rot.m/2 * mot.rot.r^2; % MoI along h dim
71 mot.rot.Izz = mot.rot.Ixx; % MoI along r dir
72
73 mot.stat.Ixx = mot.stat.m/12 * (3*mot.stat.r^2 + mot.stat.h^2); %
      MoI along r dir
74 mot.stat.Iyy = mot.stat.m/2 * mot.stat.r^2; % MoI along h dir
75 mot.stat.Izz = mot.stat.Ixx; % MoI along r dir
76
77 % viscous friction coeff (motor side)
78 mot.B = mot.Kt*mot.I0/mot.w0;
79
80 %% Gearbox data
81
82 gbox.N = 30; % reduction ratio
83 gbox.B = 0.025; % viscous friction coeff (load side)
```

```
84
85 %% Battery data
86
87 % electrical data
88 batt.UN = 11.1; % nominal voltage
89
90 % dimensions
91 batt.w = 136e-3; % battery pack width
92 batt.h = 26e-3; % battery pack height
93 batt.d = 44e-3; % battery pack depth
94
95 % center of mass (CoM) position
96 batt.xb = 0; % CoM x pos in body frame
97 batt.yb = 0; % CoM y pos in body frame
98 batt.zb = 44e-3; % CoM z pos in body frame
99
100 % mass
101 batt.m = 0.320;
102
103 % moment of inertias (MoI) wrt principal axes
104 batt.Ixx = batt.m/12 * (batt.w^2 + batt.h^2); % MoI along d dim
105 batt.Iyy = batt.m/12 * (batt.d^2 + batt.h^2); % MoI along w dim
106 batt.Izz = batt.m/12 * (batt.w^2 + batt.d^2); % MoI along h dim
107
108 %% H bridge PWM voltage driver data
109
110 drv.Vbus = batt.UN; % H bridge DC bus voltage
111 drv.pwm.bits = 8; % PWM resolution [bits]
112 drv.pwm.levels = 2^drv.pwm.bits; % PWM levels
113 drv.dutymax = drv.pwm.levels-1; % max duty cycle code
114 drv.duty2V = drv.Vbus/drv.dutymax; % duty cycle code (0 255 ) to
      voltage
115 drv.V2duty = drv.dutymax/drv.Vbus; % voltage to duty cycle code (0
       255 )
116
117 %% Wheel data
118
119 % dimensions
120 wheel.h = 26e-3; % wheel height
121 wheel.r = 68e-3/2; % wheel radius
122
123 % center of mass (CoM) position
124 wheel.xb = 0; % (left) wheel CoM x pos in body frame
125 wheel.yb = 100e-3; % (left) wheel CoM y pos in body frame
126 wheel.zb = 0; % (left) wheel CoM z pos in body frame
127
128 % mass
```

```
129 wheel.m = 50e-3;
130
131 % moment of inertias (MoI) wrt principal axes
132 wheel.Ixx = wheel.m/12 * (3*wheel.r^2 + wheel.h^2); % MoI along r
       dim
133 wheel.Iyy = wheel.m/2 * wheel.r^2; % MoI along h dim
134 wheel.Izz = wheel.Ixx; % MoI along r dim
135
136 % viscous friction coeff
137 wheel.B = 0.0015;
138
139 %% Chassis data
140
141 % dimensions
142 chassis.w = 160e-3; % frame width
143 chassis.h = 119e-3; % frame height
144 chassis.d = 80e-3; % frame depth
145
146 % center of mass (CoM) position
147 chassis.xb = 0; % CoM x pos in body frame
148 chassis.yb = 0; % CoM x pos in body frame
149 chassis.zb = 80e-3; % CoM x pos in body frame
150
151 % mass
152 chassis.m = 0.456;
153
154 % moment of inertias (MoI) wrt principal axes
155 chassis.Ixx = chassis.m/12 * (chassis.w^2 + chassis.h^2); % MoI
       along d dim
156 chassis.Iyy = chassis.m/12 * (chassis.d^2 + chassis.h^2); % MoI
       along w dim
157 chassis.Izz = chassis.m/12 * (chassis.w^2 + chassis.d^2); % MoI
       along h dim
158
159 %% Body data
160
161 % mass
162 body.m = chassis.m + batt.m + 2*mot.stat.m;
163
164 % center of mass (CoM) position
165 body.xb = 0; % CoM x pos in body frame
166 body.yb = 0; % CoM y pos in body frame
167 body.zb = (1/body.m) * (chassis.m*chassis.zb + ... % CoM z pos in
       body frame
168     batt.m*batt.zb + 2*mot.stat.m*mot.stat.zb);
169
170 % moment of inertias (MoI) wrt principal axes
```

```
body.Ixx = chassis.Ixx + chassis.m*(body.zb - chassis.zb)^2 + ... % MoI along d dim
    batt.Ixx + batt.m*(body.zb - batt.zb)^2 + ...
    2*mot.stat.Ixx + ...
    2*mot.stat.m*(mot.stat.yb^2 + (body.zb - mot.stat.zb)^2);

body.Iyy = chassis.Iyy + chassis.m*(body.zb - chassis.zb)^2 + ... % MoI along w dim
    batt.Iyy + batt.m*(body.zb - batt.zb)^2 + ...
    2*mot.stat.Iyy + ...
    2*mot.stat.m*(body.zb - mot.stat.zb)^2;

body.Izz = chassis.Izz + batt.Izz + ... % MoI along h dim
2*mot.stat.Izz + 2*mot.stat.m*mot.stat.yb^2;

%% Sensors data    Hall effect encoder

% Hall effect encoder
sens.enc.ppr = 16*4; % pulses per rotation at motor side (w/ quadrature decoding)
sens.enc.pulse2deg = 360/sens.enc.ppr;
sens.enc.pulse2rad = 2*pi/sens.enc.ppr;
sens.enc.deg2pulse = sens.enc.ppr/360;
sens.enc.rad2pulse = sens.enc.ppr/2/pi;

%% Sensors data    MPU6050 (accelerometer + gyro)

% center of mass (CoM) position
sens.mpu.xb = 0;
sens.mpu.yb = 0;
sens.mpu.zb = 13.5e-3;

% MPU6050 embedded accelerometer specs
sens.mpu.acc.bits = 16;
sens.mpu.acc.fs_g = 16; % full scale in "g" units
sens.mpu.acc.fs = sens.mpu.acc.fs_g * g2ms2; % full scale in [m/s^2]
sens.mpu.acc.g2LSB = floor(2^(sens.mpu.acc.bits-1)/sens.mpu.acc.fs_g); % sensitivity [LSB/g]
sens.mpu.acc.ms22LSB = sens.mpu.acc.g2LSB * ms22g; % sensitvity [LSB/(m/s^2)]
sens.mpu.acc.LSB2g = sens.mpu.acc.fs_g/2^(sens.mpu.acc.bits-1); % out quantization [g/LSB]
sens.mpu.acc.LSB2ms2 = sens.mpu.acc.LSB2g * g2ms2; % out quantization [ms2/LSB]
sens.mpu.acc.bw = 94; % out low pass filter BW [Hz]
```

```
209 sens.mpu.acc.noisestd = 400e-6*sqrt(100); % output noise std [
      g rms]
210 sens.mpu.acc.noisevar = sens.mpu.acc.noisestd^2; % output noise var
      [g^2]
211
212 % MPU6050 embdedded gyroscope specs
213 sens.mpu.gyro.bits = 16;
214 sens.mpu.gyro.fs_degs = 250; % full scale in [deg/s (dps)]
215 sens.mpu.gyro.fs = sens.mpu.gyro.fs_degs * deg2rad; % full scale in
      [rad/s]
216 sens.mpu.gyro.degs2LSB = floor(2^(sens.mpu.gyro.bits-1)/sens.mpu.
      gyro.fs_degs); % sensitivity [LSB/degs]
217 sens.mpu.gyro.rads2LSB = sens.mpu.gyro.degs2LSB * rad2deg; %
      sensitivity [LSB/rads]
218 sens.mpu.gyro.LSB2degs = sens.mpu.gyro.fs_degs/2^(sens.mpu.gyro.
      bits-1); % out quantization [degs/LSB]
219 sens.mpu.gyro.LSB2rads = sens.mpu.gyro.LSB2degs * deg2rad; % out
      quantization [rads/LSB]
220 sens.mpu.gyro.bw = 98; % out low pass filter BW [Hz]
221 sens.mpu.gyro.noisestd = 5e-3*sqrt(100); % output noise std [
      degs rms]
222 sens.mpu.gyro.noisevar = sens.mpu.acc.noisestd ^2; % output noise
      var [degs^2]
```