Homework III in EL2450 Hybrid and Embedded Control Systems

Nadine Drollinger
890120-5420
nadined@kth.se

Alexandros Filotheou
871108-5590
alefil@kth.se

Roberto Sánchez-Rey
840616-9139
rosr@kth.se

Emma Thilén
930715-5466
ethilen@kth.se

Task 1

Given the translational and rotational inputs u_ω and u_ω, the individual wheel inputs can be computed by

$$\begin{aligned} u_\omega &= \frac{u_r + u_l}{2} \\ u_\Psi &= u_r - u_l \end{aligned} \quad \Leftrightarrow \quad \begin{aligned} u_l &= u_\omega - \frac{u_\Psi}{2} \\ u_r &= u_\omega + \frac{u_\Psi}{2} \end{aligned}$$

Task 2

On the basis of the continuous state equations for translation.

$$\begin{aligned} \dot{x} &= R\, u_\omega\, cos\theta \\ \dot{y} &= R\, u_\omega\, sin\theta \end{aligned} \tag{1}$$

Since the robot itself uses centimeters as reference distance unit, the given 200 value for the speed turns to be: $u_\omega = 2\,[m/s]$. Furthermore, assuming:

1. $\dot{x} = \frac{\Delta x}{\Delta t}$
2. $\dot{y} = \frac{\Delta y}{\Delta t}$
3. $\Delta x = x_f - x_o;\ \Delta y = y_f - y_o\ \Delta t = t_f - t_o$

and multiplying each term of (1) by itself and adding them toghether, one gets the following equation that can be solved with values of the variations in position coordinates and time, plus the above mentioned value of the speed.

$$\left.\begin{array}{rcl} R & = & \sqrt{\frac{1}{u_\omega^2}\left[\left(\frac{\Delta x}{\Delta t}\right)^2+\left(\frac{\Delta y}{\Delta t}\right)^2\right]} \\ \Delta x & = & 1.3\,[m] \\ \Delta y & = & -0.07\,[m] \\ \Delta t & = & 6.4250\,[s] \\ u_\omega & = & 2\,[m/s] \end{array}\right\} R = 0.1013$$

For the calculation of L, the equation for the robot pose is used.

$$\dot{\theta} = \frac{R}{L}\, u_\Psi$$

As earlier, we have considered $\dot{\theta} = \dfrac{\Delta\theta}{\Delta t}$. From file `Rotation.cvs`, created for $u_\Psi = 400\,[cm/s]$ one gets $\Delta\theta = 330.27^\circ$, $\Delta t = 4.1667\,[s]$. Hence:

$$L = \frac{\Delta t}{\Delta\theta}\, R\, u_\Psi = 0.5113$$

Task 3

In order to reach a conclusion about the stability of the angular displacement, it suffices to find a Lyapunov function $V(x)$ such that $V(0) = 0$, $V(x) > 0$ for all $x \neq 0$ and $\dot{V}(x) \leq 0$ for all x. Considering $x = \theta - \theta^G$ and $V(x) = x^2$:

$$V(0) = 0,\ V(x) > 0,\ \text{for all } x \neq 0, \text{ and}$$

$$\begin{aligned} \dot{V}(x) = 2x\dot{x} &= 2(\theta - \theta^G)\dot{\theta} \\ &= 2(\theta - \theta^G)\frac{R}{L}u_\Psi \end{aligned}$$

- When $\theta - \theta^G \leq 0$, $\dot{V}(x) = 2(\theta - \theta^G)\dfrac{R}{L} \leq 0$
- When $\theta - \theta^G > 0$, $\dot{V}(x) = -2(\theta - \theta^G)\dfrac{R}{L} < 0$

Hence $\dot{V}(x) \leq 0$ for all x, meaning that the system is stable for all $\theta \in (-180^\circ, 180^\circ]$.

Practically, as one can see in equation 2, the system is stable but it exhibits Zeno behaviour, which is defined as an infinite number of switches in finite time.

A solution $\chi = (\tau, q, y)$ exhibits Zeno behaviour if

$$\tau_\infty = \sum_{i=0}^{\infty}(\tau' - \tau) < \infty \tag{2}$$

whereas τ is the the time trajectory. As we can see from the automaton in figure 1, if we start at q_1 we switch to q_2 from where we are immediately switched back and vice versa.

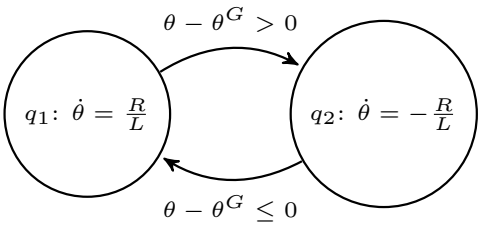


Figure 1

A formal way to look at this is the following: Without loss of generality, let us assume that at $t = 0$ the angular subsystem is at $\theta = \theta_0^0$, that $\theta - \theta^G \leq 0$ and that $R/L = K = ct > 0$. Then:

1. The system is in state q_1 for the first time and $\dot{\theta} = K$. Which means that

$$\theta(t) = Kt + \theta_0^0$$

The transition to state q_2 will happen when $\theta = \theta^G$, at time $t = t_{1\to2}^1$, and at that time:

$$\begin{aligned} \theta^G &= Kt_{1\to2}^1 + \theta_0^0 \Leftrightarrow \\ t_{1\to2}^1 &= \frac{\theta^G - \theta_0^0}{K} \end{aligned} \tag{3}$$

2. The time is $t_{1\to2}^1$ and the system is at state q_2. Here, $\dot{\theta} = -K$ which means that

$$\theta(t) = -Kt + \theta_0^1$$

and at this time instant

$$\begin{aligned} \theta^G &= -Kt_{1\to2}^1 + \theta_0^1 \Leftrightarrow \\ \theta^G &= -K\frac{\theta^G - \theta_0^0}{K} + \theta_0^1 \Leftrightarrow \\ \theta^G &= -\theta^G + \theta_0^0 + \theta_0^1 \Leftrightarrow \\ \theta_0^1 &= 2\theta^G - \theta_0^0 \end{aligned}$$

Hence

$$\theta(t) = -Kt + 2\theta_G - \theta_0^0$$

and at the time of the second switch, at $t_{2\to1}^1$ when $\theta = \theta^G$:

$$\begin{aligned} \theta^G &= -Kt_{2\to1}^1 + 2\theta^G - \theta_0^0 \Leftrightarrow \\ Kt_{2\to1}^1 &= \theta^G - \theta_0^0 \Leftrightarrow \\ t_{2\to1}^1 &= \frac{\theta^G - \theta_0^0}{K} \end{aligned} \tag{4}$$

Comparing equations 3 and 4 we observe that two switches must happen at the exact same time which is $t = t_{1\to2}^1 = t_{2\to1}^1$. This fact implies that the system might exhibit Zeno behaviour, however, we do not know whether the behaviour of θ will be the same in the next round yet. Continuing,

3. The time now is $t^1_{2\to 1}$ and the system is at state q_1. Here, $\dot{\theta} = K$ which means that

$$\theta(t) = Kt + \theta_0^2$$

and at this time instant

$$\begin{aligned}
\theta^G &= Kt^1_{2\to 1} + \theta_0^2 \Leftrightarrow \\
\theta^G &= K\frac{\theta^G - \theta_0^0}{K} + \theta_0^2 \Leftrightarrow \\
\theta^G &= \theta^G - \theta_0^0 + \theta_0^2 \Leftrightarrow \\
\theta_0^2 &= \theta_0^0
\end{aligned}$$

So, when going back to state q_1 again, this process starts from the beginning, and, hence, we are reassured that these switches will be inifinite in number.

Hence, equation 2 converges, and an infinite amount of switches should happen at the same time. Ergo, the angular subsystem exhibits a chattering Zeno behaviour under this control law.

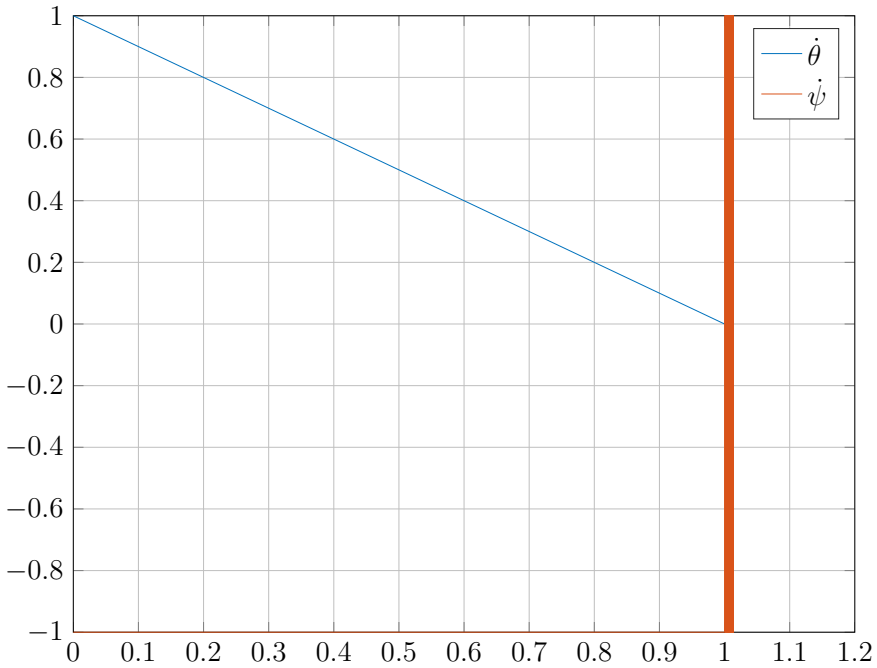


Figure 2

Task 4

Examining file `rot2.slx` one sees that $\dot{\theta} = R/Lu_\Psi$ is stable. We can verify this analytically:

$$
\begin{aligned}
\dot{u}_\Psi &= K_L(u_\Psi^R - u_\Psi) \\
\dot{u}_\Psi &= K_L u_\Psi^R - K_L u_\Psi \\
\dot{u}_\Psi \cdot e^{K_L t} &= K_L u_\Psi^R \cdot e^{K_L t} - K_L u_\Psi \cdot e^{K_L t} \\
\dot{u}_\Psi \cdot e^{K_L t} + K_L u_\Psi \cdot e^{K_L t} &= K_L u_\Psi^R \cdot e^{K_L t} \\
\frac{d}{dt}(u_\Psi e^{K_L t}) &= K_L u_\Psi^R e^{K_L t} \\
u_\Psi e^{K_L t} &= \int K_L u_\Psi^R e^{K_L t} = u_\Psi^R e^{K_L t} + \lambda \\
u_\Psi &= u_\Psi^R + \lambda e^{-K_L t}
\end{aligned}
$$

where, from inspection of the parameters of the integrator, the initial condition is $\lambda = 1$. Hence

$$
u_\Psi(t) = \begin{cases} 1 + e^{-K_L t} & \theta - \theta^G \leq 0 \\ -1 + e^{-K_L t} & \theta - \theta^G > 0 \end{cases} \tag{5}
$$

In order to reach a conclusion about the stability of the angular displacement, it suffices to find a Lyapunov function $V(x)$ such that $V(0) = 0$, $V(x) > 0$ for all $x \neq 0$ and $\dot{V}(x) \leq 0$ for all x. Considering $x = \theta - \theta^G$ and $V(x) = x^2$:

$$
V(0) = 0,\ V(x) > 0,\ \text{for all } x \neq 0, \text{ and}
$$

$$
\begin{aligned}
\dot{V}(x) = 2x\dot{x} &= 2(\theta - \theta^G)\dot{\theta} \\
&= 2(\theta - \theta^G)\frac{R}{L}u_\Psi \\
&= 2(\theta - \theta^G)\frac{R}{L}(u_\Psi^R + e^{-K_L t})
\end{aligned}
$$

- When $\theta - \theta^G \leq 0$, $\dot{V}(x) = 2(\theta - \theta^G)\frac{R}{L}(1 + e^{-K_L t}) \leq 0$
  With

$$
\begin{aligned}
0 < e^{-K_L t} &\leq 1 \\
0 < 1 < 1 + e^{-K_L t} &\leq 2
\end{aligned} \tag{6}
$$

- When $\theta - \theta^G > 0$, $\dot{V}(x) = 2(\theta - \theta^G)\frac{R}{L}(-1 + e^{-K_L t}) < 0$
  With

$$
\begin{aligned}
0 < e^{-K_L t} &\leq 1 \\
-1 < -1 + e^{-K_L t} &\leq 0
\end{aligned} \tag{7}
$$

Hence, from inequalities 6 and 7 we conclude that $\dot{V}(x) \leq 0$ for all x, meaning that the system is stable for all $\theta \in (-180°, 180°]$. In practice, the plot of the continuous state trajectory in figure 3 suggests that this is the case.

From what one can see in the discrete state trajectory plot in figure 3, the time interval between two consecutive switches is constantly being reduced in size every time a switch

happens. This means that the sum in equation 2 converges, and, hence, the angular system is verified to exhibit genuine Zeno behaviour.

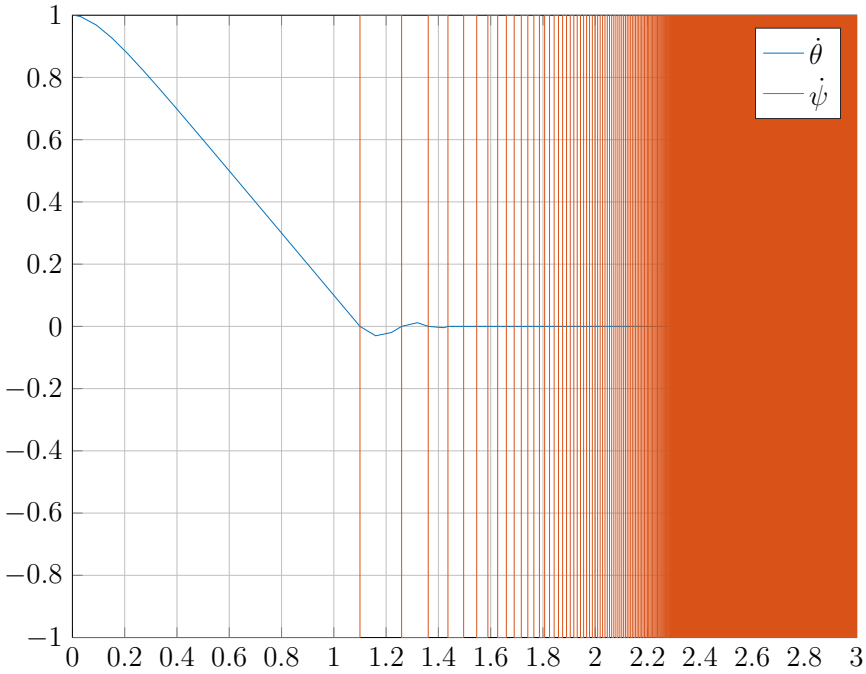


Figure 3

Task 5

From what we can see in the continuous and discrete state trajectory plot in figure 4, there is no zeno behaviour. There may be an infinite amount of state switches, but they do not happen in an finite amount of time. The zero-order-hold implemented in this system holds the current value until the next sampling time instant. Therefore, the system's reaction time is limited by the ZOH and therefore it can not exhibit zeno behaviour.

$\dot{\theta} = R/Lu_{\Psi}$ is marginally stable, as it does not go to zero but as well not to instability.

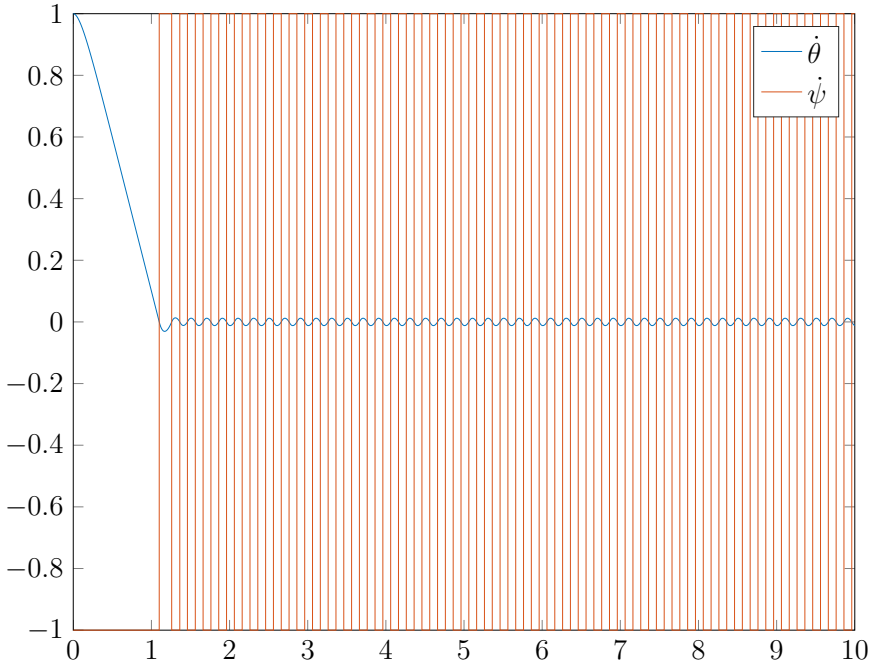


Figure 4

Task 6

The system dynamics can be discretized using the Euler forward method. With sampling time T_s, the derivatives can then be approximated by

$$\frac{d}{dt}\chi(t) = \frac{q-1}{T_s}\chi(t) \tag{8}$$

Applying equation 8 to the continuous state-space equations gives

$$\begin{aligned}
\frac{q-1}{T_s}x(t) &= Ru_\omega(t)cos(\theta(t)) \\
\frac{q-1}{T_s}y(t) &= Ru_\omega(t)cos(\theta(t)) \\
\frac{q-1}{T_s}\theta(t) &= \frac{R}{L}u_\psi(t)
\end{aligned}$$

Multiplying both sides by T_s together with some rearranging after letting the shift operator act upon the variables yields

$$\begin{aligned}
x(t+T_s) &= x(t) + T_sRu_\omega(t)cos(\theta(t)) \\
y(t+T_s) &= y(t) + T_sRu_\omega(t)cos(\theta(t)) \\
\theta(t+T_s) &= \theta(t) + \frac{T_sR}{L}u_\psi(t)
\end{aligned}$$

Letting $x[k], y[k], \theta[k]$ and $u_\omega[k], u_\psi[k]$ denote the robot's states and the control input respectively at time $t = kT_s, k = 0, 1, 2...$, the discretized system can be written as

$$
\begin{aligned}
x[k+1] &= x[k] + T_s R u_\omega[k] cos(\theta[k]) \\
y[k+1] &= y[k] + T_s R u_\omega[k] cos(\theta[k]) \\
\theta[k+1] &= \theta[k] + \frac{T_s R}{L} u_\psi(t)
\end{aligned}
$$

Task 7

From the discrete state equation for θ and the definition of u_Ψ^R for this part of the controller:

$$
\begin{aligned}
\theta[k+1] &= \theta[k] + \frac{T_s R}{L} u_\Psi^R[k] \\
&= \theta[k] + \frac{T_s R}{L} K_\Psi^R(\theta^R - \theta[k]) \\
&= \theta[k](1 - \frac{T_s R}{L} K_\Psi^R) + \frac{T_s R}{L} K_\Psi^R \theta^R
\end{aligned}
$$

By subtracting θ^R from both sides one gets

$$
\begin{aligned}
\theta[k+1] - \theta^R &= \theta[k](1 - \frac{T_s R}{L} K_\Psi^R) + (\frac{T_s R}{L} K_\Psi^R - 1)\theta^R \\
&= (\theta[k] - \theta^R)(1 - \frac{T_s R}{L} K_\Psi^R) \qquad (9)
\end{aligned}
$$

We now define state θ' as

$$
\theta'[k] = \theta[k] - \theta^R
$$

Then, equation 9 becomes

$$
\theta'[k+1] = \theta'[k](1 - \frac{T_s R}{L} K_\Psi^R) \qquad (10)
$$

In order for this system to be stable, that is, $|\theta - \theta^R| \to 0$ as $k \to \infty$, the solution of the equation inside the parentheses should lie inside the unit circle:

$$
\begin{aligned}
\left|1 - \frac{T_s R}{L} K_\Psi^R\right| &< 1 \\
-1 < 1 - \frac{T_s R}{L} K_\Psi^R &< 1 \\
-2 < -\frac{T_s R}{L} K_\Psi^R &< 0 \\
0 < \frac{T_s R}{L} K_\Psi^R &< 2 \\
0 < K_\Psi^R &< \frac{2L}{T_s R} \qquad (11)
\end{aligned}
$$

Hence, the maximum value K_Ψ^R can take for the system to be marginally stable is $K_{\Psi,max}^R = \dfrac{2L}{T_s R}$

Theoretically, the value of K_Ψ^R can be chosen to be any value inside the interval defined in inequality 11. However, first, it would be wise to choose a value that is far enough from the maximum value so as to avoid overshoot, but close enough to it, so that convergence happens in reasonable time. Hence, in practice, it is reasonable that one would need to experiment with different values and choose one that results in balancing a small angular error, a minimal overshoot, if any, and a quick enough settling time.

On another note, for a first order system, the dead-beat controller brings the system to steady state in one single step, provided that the system can be brought to zero by some input. Hence the dead-beat controller is a good choice regarding speed and convergence. As one can see from equation 10, the error $\theta[k+1] - \theta^R$ will go to zero in one single step if K_Ψ^R is set to

$$K_\Psi^R = \frac{L}{T_s R} = \frac{1}{2} K_{\Psi,max}^R$$

i.e. we can design a dead-beat controller using this value.

Task 8

For the purpose of simulating this part of the controller, the initial point of the robot was taken to be $I(x_0, y_0) \equiv (0, 0)$. The goal was set to $G(x_g, y_g) \equiv (-0.37, 1.68)$, which is node 1 in the simulation environment. The angle between the line connecting I and G and the x-axis is hence $\theta_R = tan^{-1}(1.68/-0.37) = 102.42°$.

Figures 6, 8, 10, 12, 14 show the angular response of the robot for different values of K_Ψ^R inside the interval set by inequality 11. Figure 16 verifies that the upper limit for K_Ψ^R is indeed $K_{\Psi,max}^R = \frac{2L}{T_s R}$ by showing that the angular response of the robot cannot converge for $K_\Psi^R > K_{\Psi,max}^R$.

Here, one can see that the smaller the value of K_Ψ^R is, the larger the settling time, the lower the rise time and the smoother the response is. However, as the value of K_Ψ^R increases, the steady-state response begins to oscillate, with the amplitude of this oscillation proportional to the value of K_Ψ^R.

Figures 7, 9, 11, 13 and 7 focus on the steady-state value of the aforementioned responses. As it is evident, none of the responses converge to the value $\theta_R = 102.42$. This is reasonable since with only a purely proportional control signal, as the angular error, i.e. $e(\theta) = \theta^R - \theta$, tends to zero, the product of K_Ψ^R and $e(\theta)$ isn't large enough to force the robot to rotate exactly θ^R degrees.

Another way to look at this is by looking at the steady-state response of the system, which is linear, for a step input of magnitude θ^R. Figure 5 shows the structure of the system. The z-transform of the input is then $R(z) = \dfrac{\theta^R}{1 - z^{-1}}$ and the equation of the closed-loop system is

$$Y(z) = \frac{K_\Psi^R G(z)}{1 + K_\Psi^R G(z)} R(z)$$

The steady-state response is

$$lim_{t\to\infty} y(t) = lim_{z\to 1}(1 - z^{-1}) \frac{\theta^R}{1 - z^{-1}} \frac{K_\Psi^R G(z)}{1 + K_\Psi^R G(z)} = \theta^R \cdot lim_{z\to 1} \frac{K_\Psi^R G(z)}{1 + K_\Psi^R G(z)}$$

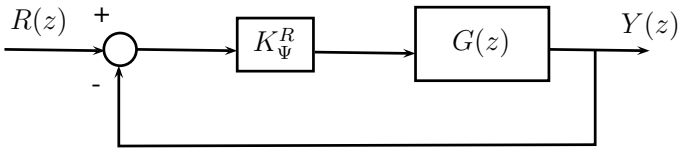


Figure 5: The structure of the system under rotational control. $G(z)$ is the discretized transfer function of the linear system whose state-space equation is $\dot{\theta} = \frac{R}{L} u_\Psi$

The steady-state response cannot reach exactly θ^R as the above limit cannot converge to 1 under our limitations for K_Ψ^R and the linear dynamics of $G(z)$.

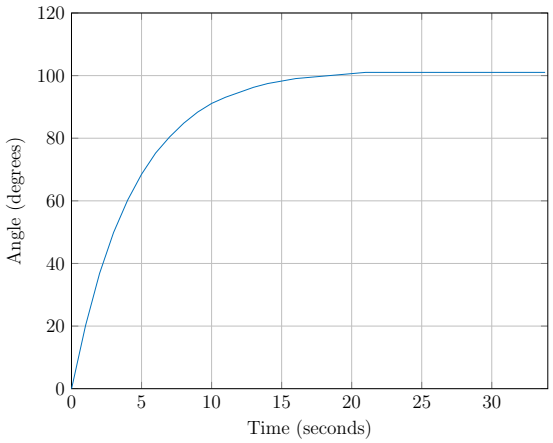


Figure 6: The orientation of the robot over time for $K_\Psi^R = 0.1 K_{\Psi,max}^R$

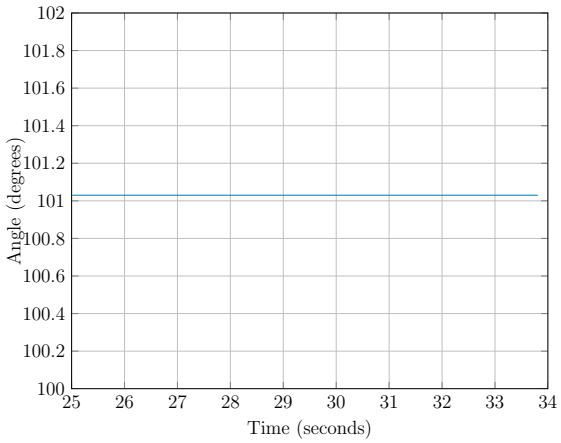


Figure 7: The steady state orientation of the robot for $K_\Psi^R = 0.1 K_{\Psi,max}^R$

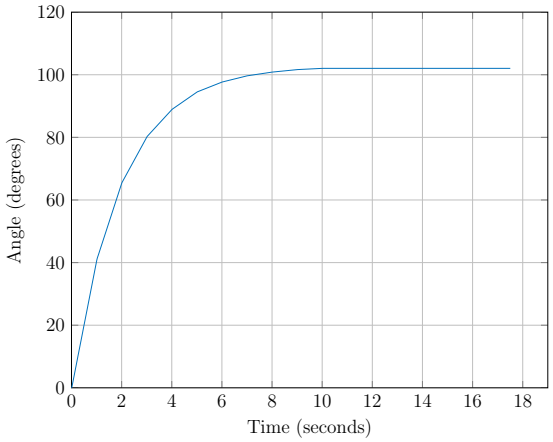


Figure 8: The orientation of the robot over time for $K_\Psi^R = 0.2 K_{\Psi,max}^R$

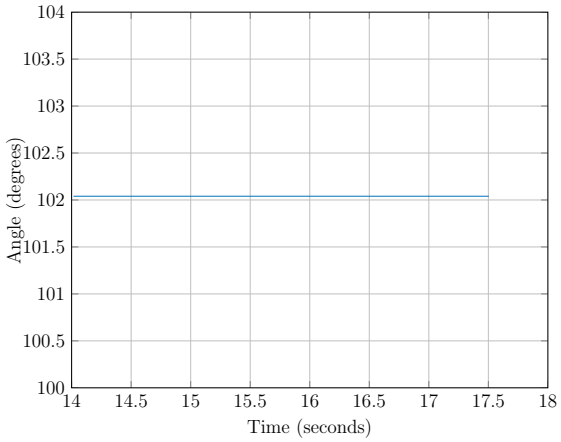


Figure 9: The steady state orientation of the robot for $K_\Psi^R = 0.2 K_{\Psi,max}^R$

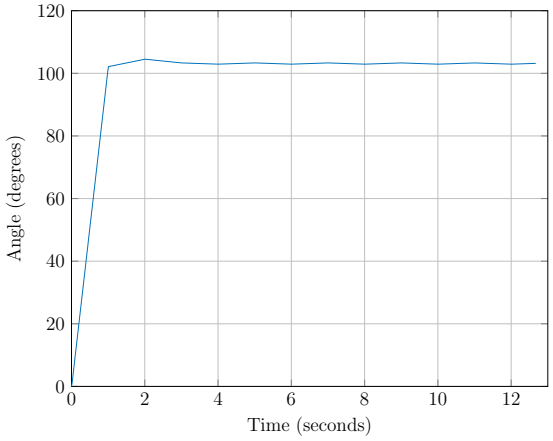


Figure 10: The orientation of the robot over time for $K_{\Psi}^{R} = 0.5K_{\Psi,max}^{R}$

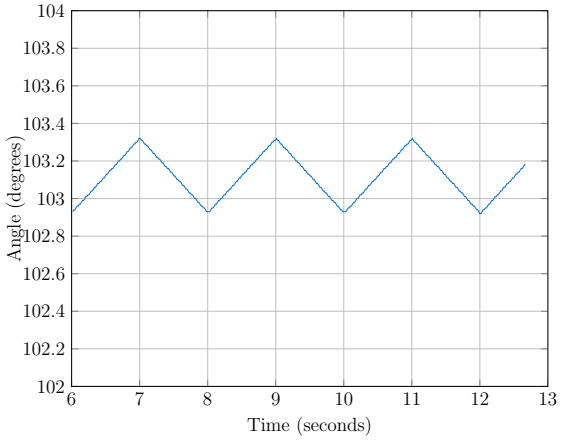


Figure 11: The steady state orientation of the robot for $K_{\Psi}^{R} = 0.5K_{\Psi,max}^{R}$

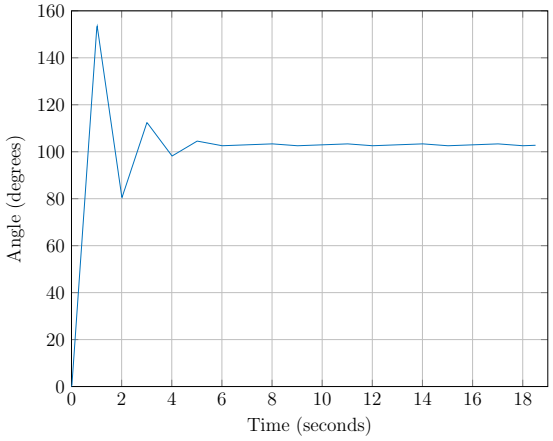


Figure 12: The orientation of the robot over time for $K_{\Psi}^{R} = 0.75K_{\Psi,max}^{R}$

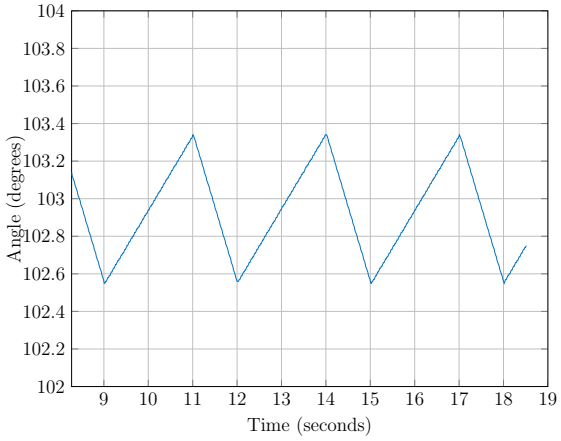


Figure 13: The steady state orientation of the robot for $K_{\Psi}^{R} = 0.75K_{\Psi,max}^{R}$

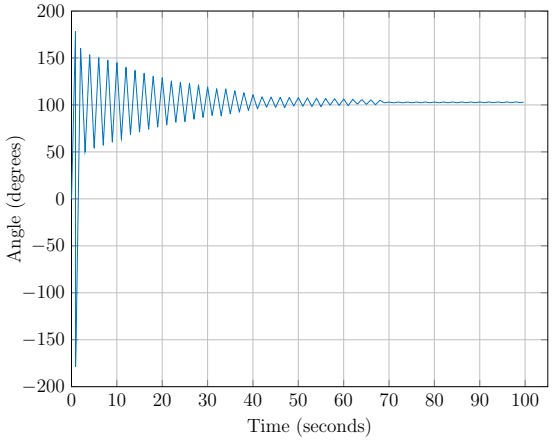


Figure 14: The orientation of the robot over time for $K_{\Psi}^{R} = K_{\Psi,max}^{R}$. This is the upper limit value of K_{Ψ}^{R} before the system becomes unstable

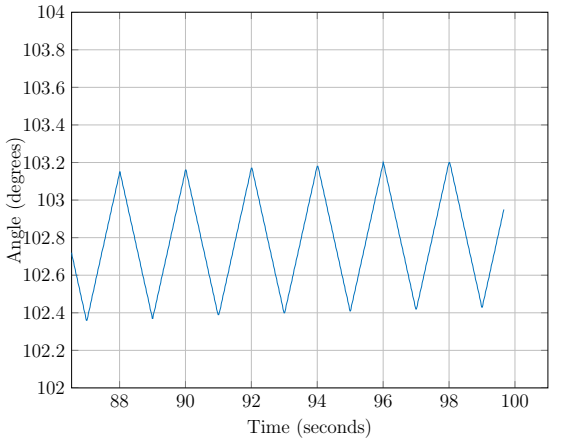


Figure 15: The steady state orientation of the robot for $K_{\Psi}^{R} = K_{\Psi,max}^{R}$

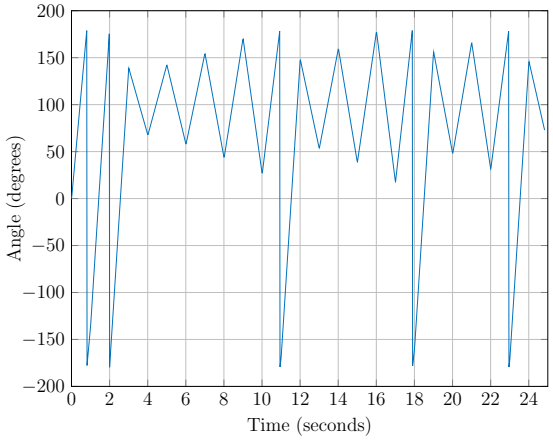


Figure 16: The orientation of the robot over time for $K_\Psi^R = K_{\Psi,max}^R + 1$. The system is indeed unstable

Task 9

From the definition of $d_0[k]$, and the fact that θ does not change over time:

$$\begin{aligned}
d_0[k] &= cos(\theta[k])(x_0 - x[k]) + sin(\theta[k])(y_0 - y[k]) \\
&= cos(\theta[k])(x_0 - x[k-1] - T_s R u_\omega^R[k-1] cos(\theta[k-1])) \\
&+ sin(\theta[k])(y_0 - y[k-1] - T_s R u_\omega^R[k-1] sin(\theta[k-1])) \\
&= cos(\theta^R)(x_0 - x[k-1] - T_s R u_\omega^R[k-1] cos(\theta^R)) \\
&+ sin(\theta^R)(y_0 - y[k-1] - T_s R u_\omega^R[k-1] sin(\theta^R)) \\
&= cos(\theta^R)(x_0 - x[k-1]) + sin(\theta^R)(y_0 - y[k-1]) - T_s R K_\omega^T d_0[k-1] \\
&= d_0[k-1] - T_s R K_\omega^T d_0[k-1] \\
&= (1 - T_s R K_\omega^R) d_0[k-1]
\end{aligned}$$

Hence

$$d_0[k+1] = (1 - T_s R K_\omega^R) d_0[k]$$

In order for this system to be stable, that is, $d_0 \to 0$ as $k \to \infty$, the solution of the equation inside the parentheses should lie inside the unit circle:

$$\begin{aligned}
\left|1 - T_s R K_\omega^R\right| &< 1 \\
-1 < 1 - T_s R K_\omega^R &< 1 \\
-2 < -T_s R K_\omega^R &< 0 \\
0 < T_s R K_\omega^R &< 2 \\
0 < K_\omega^R &< \frac{2}{T_s R}
\end{aligned} \tag{12}$$

Hence, the maximum value K_ω^R can take for the system to be marginally stable is $K_{\omega,max}^R = \dfrac{2}{T_s R}$.

Theoretically, the value of K_ω^R can be chosen to be any value inside the interval defined in inequality 12. However, first, it would be wise to choose a value that is far enough from the maximum value so as to avoid overshoot, but close enough to it, so that convergence happens in reasonable time. Hence, in practice, it is reasonable that one would need to experiment with different values and choose one that results in balancing a small angular error, a minimal overshoot, if any, and a quick enough settling time.

Just like discussed before in the response of task 7, the dead-beat controller is a good choice in order to get fast convergence. The dead-beat control is in this case obtained when $K_\omega^R = \frac{1}{T_s R} = \frac{1}{2} K_{\omega,max}^R$.

Task 10

This part of the controller is responsible for compensating translational errors during rotation. Since the rotational speed u_Ψ is zero, it is expected that the robot will not move away from its origin. Figure 17 plots the robot's distance from its origin over time for $K_\omega^R = 0.5 K_{\omega,max}^R$.

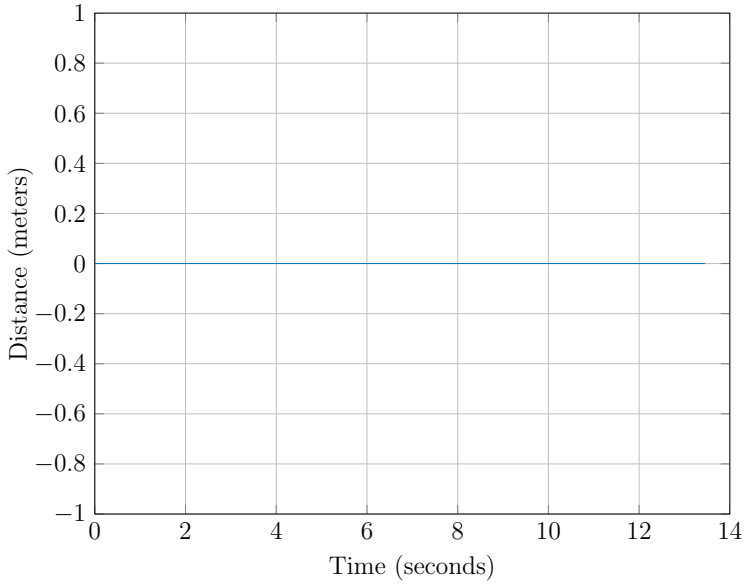


Figure 17: The distance of the robot from its origin position over time for $K_\omega^R = 0.5 K_{\omega,max}^R$

Task 11

Figures 18, 20, 22, 24, 26 show the bearing error of the robot for different values of K_Ψ^R inside the interval set by inequality 11. Figures 19, 21, 23, 25 and 19 focus on the steady-state bearing error. Figure 28 illustrates the $d_0[k]$ error, which is at all times zero.

The evolution of the bearing and distance error is the same when both of the rotational controllers are enabled compared to when only one of them is enabled. This happens because the behaviour of each controller does not affect the behaviour of the other, since this is an ideal system. In reality, we expect that the distance error will be non-zero.

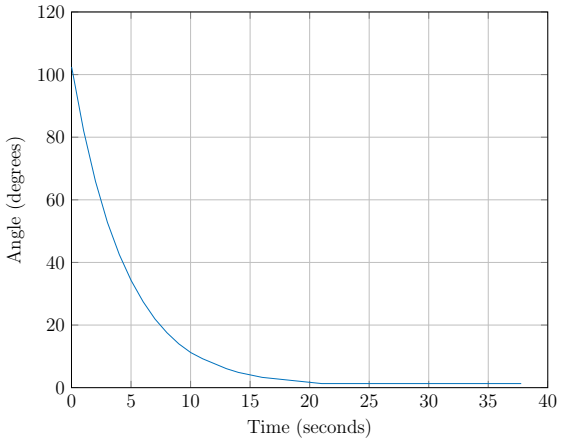


Figure 18: The error in orientation of the robot over time for $K_{\Psi}^{R} = 0.1K_{\Psi,max}^{R}$

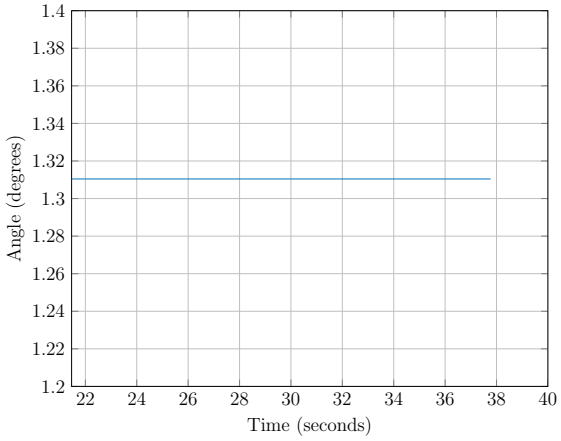


Figure 19: The steady state error in orientation of the robot for $K_{\Psi}^{R} = 0.1K_{\Psi,max}^{R}$

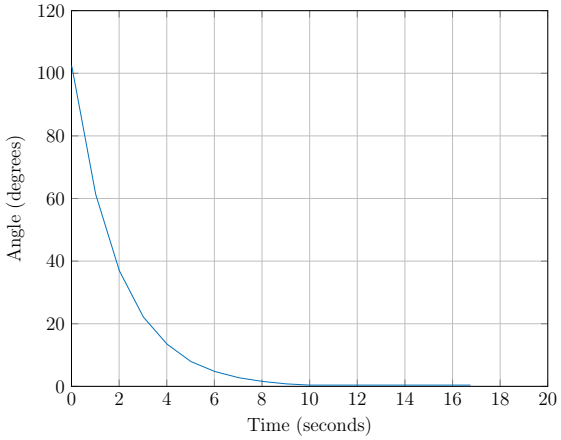


Figure 20: The error in orientation of the robot over time for $K_{\Psi}^{R} = 0.2K_{\Psi,max}^{R}$

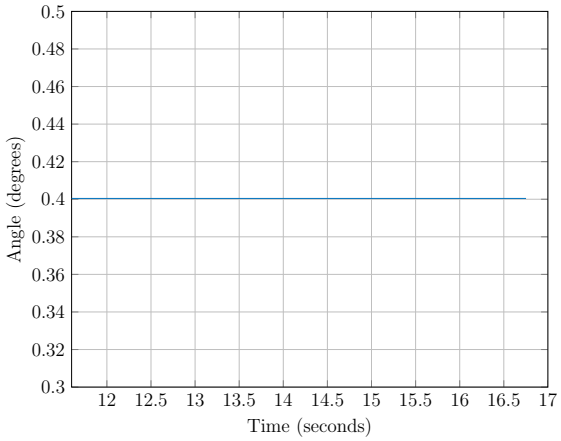


Figure 21: The steady state error in orientation of the robot for $K_{\Psi}^{R} = 0.2K_{\Psi,max}^{R}$

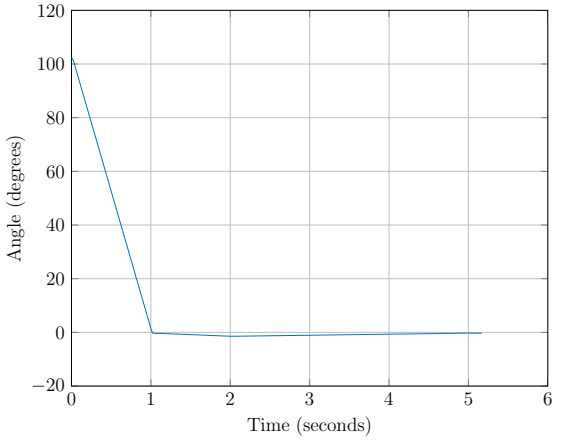


Figure 22: The error in orientation of the robot over time for $K_{\Psi}^{R} = 0.5K_{\Psi,max}^{R}$

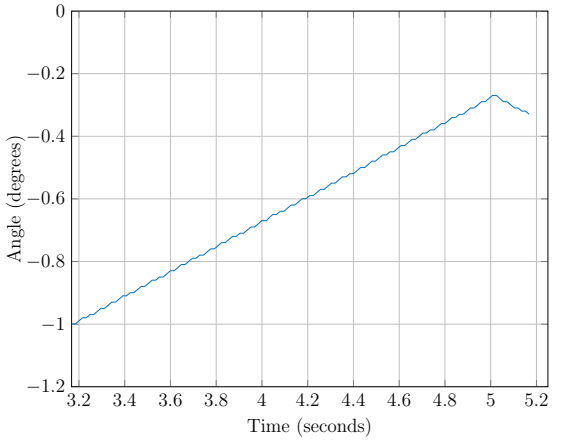


Figure 23: The steady state error in orientation of the robot for $K_{\Psi}^{R} = 0.5K_{\Psi,max}^{R}$

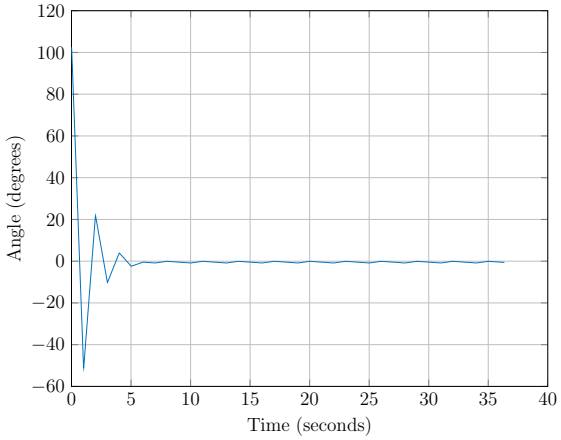


Figure 24: The error in orientation of the robot over time for $K_{\Psi}^{R} = 0.75K_{\Psi,max}^{R}$

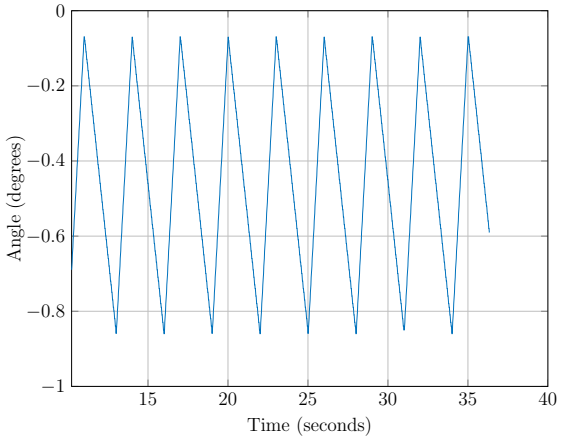


Figure 25: The steady state error in orientation of the robot for $K_{\Psi}^{R} = 0.75K_{\Psi,max}^{R}$

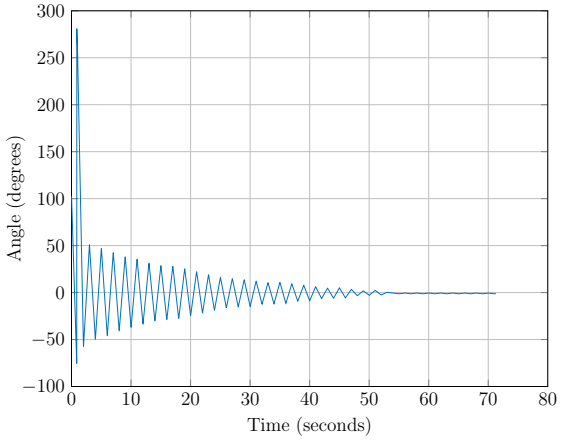


Figure 26: The error in orientation of the robot over time for $K_{\Psi}^{R} = K_{\Psi,max}^{R}$.

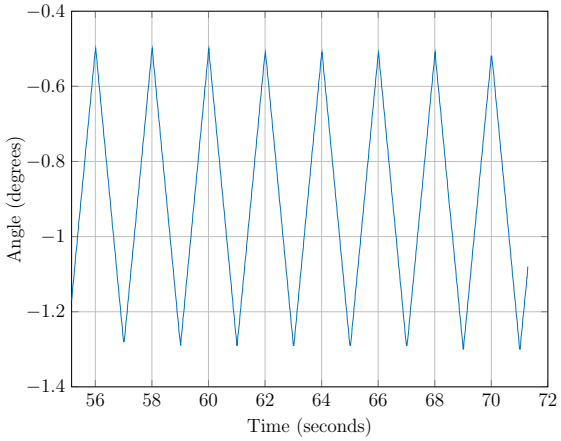


Figure 27: The steady state error in orientation of the robot for $K_{\Psi}^{R} = K_{\Psi,max}^{R}$

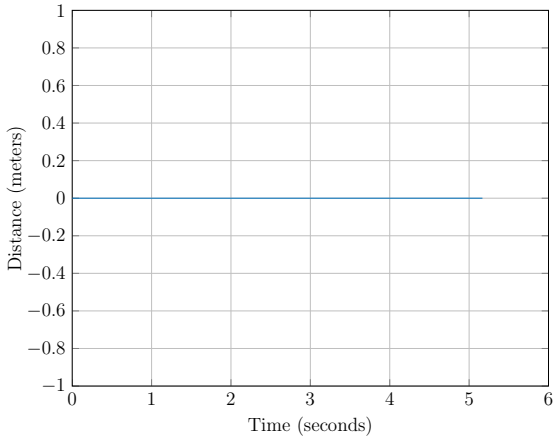


Figure 28: The distance of the robot from its origin position over time for all legitimate values or K_{ω}^{R}

Task 12

From the definition of $d_g[k]$:

$$\begin{aligned} d_g[k] &= cos(\theta_g)(x_g - x[k]) + sin(\theta_g)(y_g - y[k]) \\ &= cos(\theta_g)(x_g - x[k-1] - T_s R u_\omega^T[k-1] cos(\theta_g)) + \\ & sin(\theta_g)(y_g - y[k-1] - T_s R u_\omega^T[k-1] sin(\theta_g)) \\ &= cos(\theta_g)(x_g - x[k-1]) + sin(\theta_g)(y_g - y[k-1]) - T_s R K_\omega^T d_g[k-1] \\ &= d_g[k-1] - T_s R K_\omega^T d_g[k-1] \\ &= (1 - T_s R K_\omega^T) d_g[k-1] \end{aligned}$$

Hence

$$d_g[k+1] = (1 - T_s R K_\omega^T) d_g[k]$$

In order for this system to be stable, that is, $d_g \to 0$ as $k \to \infty$, the solution of the equation inside the parentheses should lie inside the unit circle:

$$\begin{aligned} \left|1 - T_s R K_\omega^T\right| &< 1 \\ -1 < 1 - T_s R K_\omega^T &< 1 \\ -2 < -T_s R K_\omega^T &< 0 \\ 0 < T_s R K_\omega^T &< 2 \\ 0 < K_\omega^T &< \frac{2}{T_s R} \end{aligned} \tag{13}$$

Hence, the maximum value K_ω^T can take for the system to be marginally stable is $K_{\omega,max}^T = \dfrac{2}{T_s R}$

Theoretically, the value of K_ω^T can be chosen to be any value inside the interval defined in inequality 13. However, first, it would be wise to choose a value that is far enough from the maximum value so as to avoid overshoot, but close enough to it, so that convergence happens in reasonable time. Hence, in practice, it is reasonable that one would need to experiment with different values and choose one that results in balancing a small angular error, a minimal overshoot, if any, and a quick enough settling time.

Just like discussed before, the dead-beat controller is a good choice in order to get fast convergence. Also in this case the dead-beat control is obtained for $K_\omega^T = \frac{1}{T_s R} = \frac{1}{2} K_{\omega,max}^T$.

Task 13

For the purpose of simulating this part of the controller, the initial point of the robot was taken to be $I(x_0, y_0) \equiv (0, 0)$. The goal was set to $G(x_g, y_g) \equiv (1.0, 0.0)$.

Figures 29, 31, 33, 35 and 37 show the displacemental response of the robot for various values of K_ω^T inside the interval set by inequality 13. Figure 39 verifies that the upper

limit for K_ω^T is indeed $\frac{2}{T_s R}$ by showing that the displacemental response of the robot cannot converge for $K_\omega^T > K_{\omega,max}^T$.

Here, one can see that the smaller the value of K_ω^T is, the larger the settling time, the lower the rise time and the smoother the response is. However, as the value of K_ω^T increases, the steady-state response begins to oscillate, with the amplitude of this oscillation proportional to the value of K_ω^T.

Figures 30, 32, 34, 36 and 38 focus on the steady-state value of the aforementioned responses. In contrast to the proportional rotational controller, the robot *can* arrive to its reference signal. The equations that govern the robot's translational movement are non-linear, as opposed to the one that governs its rotational movement, which in turn means that the translational system's behaviour is not bounded within the laws that govern control with proportional controllers on linear systems.

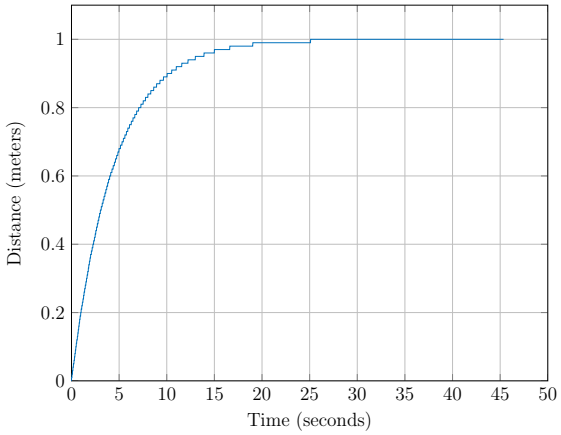


Figure 29: The orientation of the robot over time for $K_\omega^T = 0.1K_{\omega,max}^T$

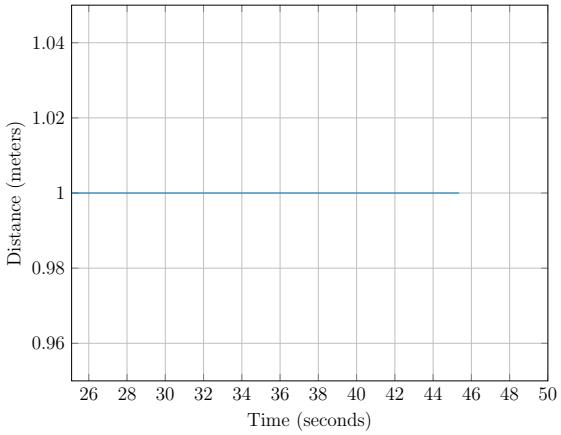


Figure 30: The steady state orientation of the robot for $K_\omega^T = 0.1K_{\omega,max}^T$

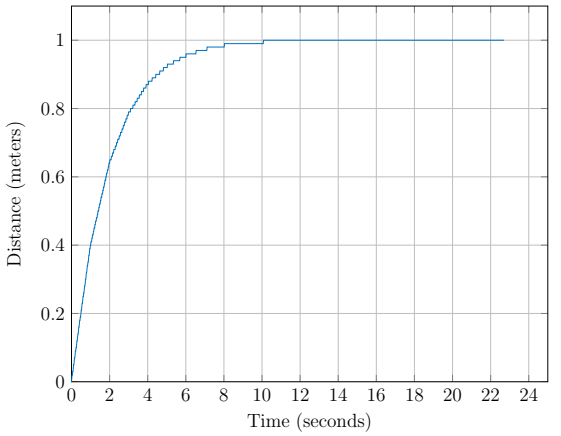


Figure 31: The orientation of the robot over time for $K_\omega^T = 0.2K_{\omega,max}^T$

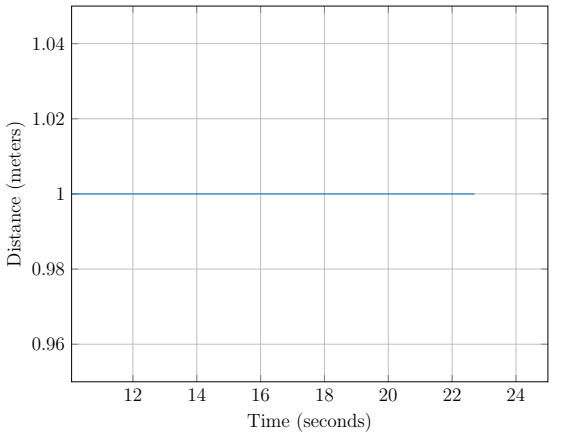


Figure 32: The steady state orientation of the robot for $K_\omega^T = 0.2K_{\omega,max}^T$

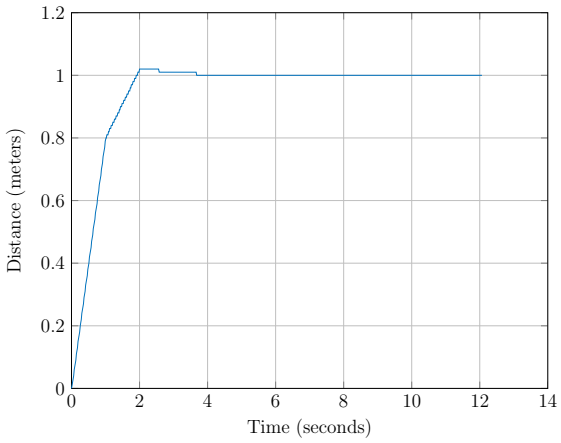


Figure 33: The orientation of the robot over time for $K_\omega^T = 0.5K_{\omega,max}^T$

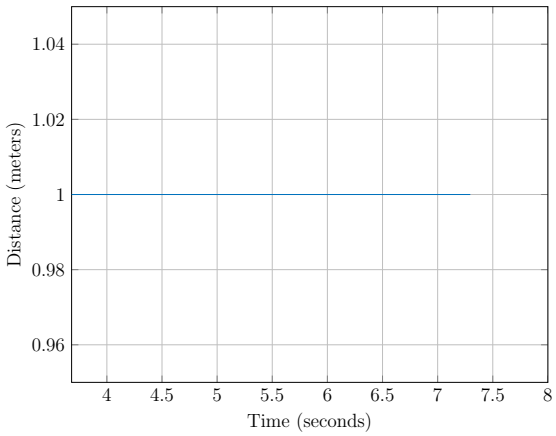


Figure 34: The steady state orientation of the robot for $K_\omega^T = 0.5K_{\omega,max}^T$

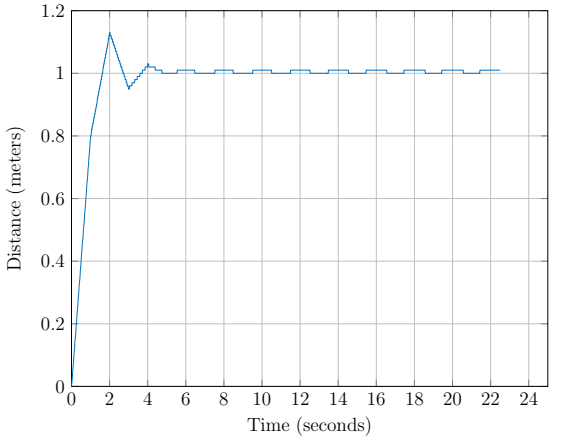


Figure 35: The orientation of the robot over time for $K_\omega^T = 0.75K_{\omega,max}^T$

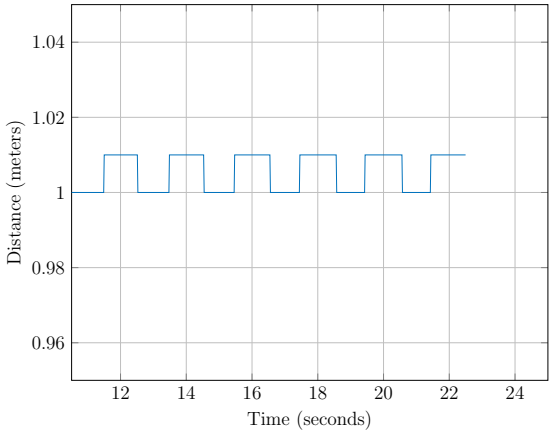


Figure 36: The steady state orientation of the robot for $K_\omega^T = 0.75K_{\omega,max}^T$

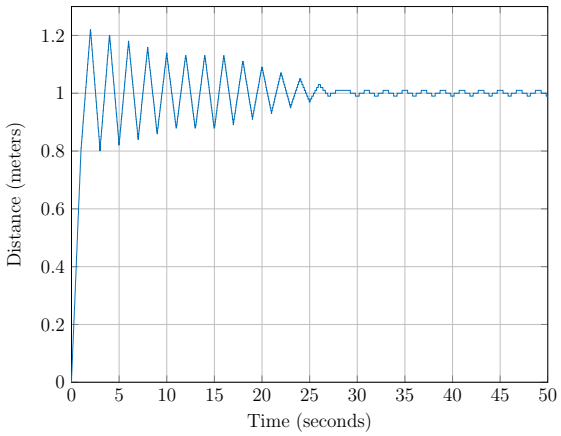


Figure 37: The orientation of the robot over time for $K_\omega^T = K_{\omega,max}^T$.

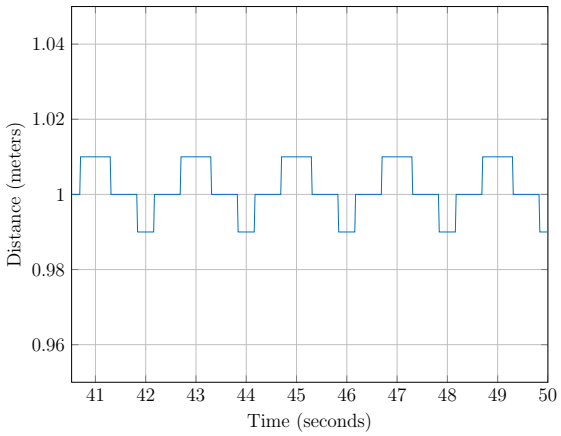


Figure 38: The steady state orientation of the robot for $K_\omega^T = K_{\omega,max}^T$

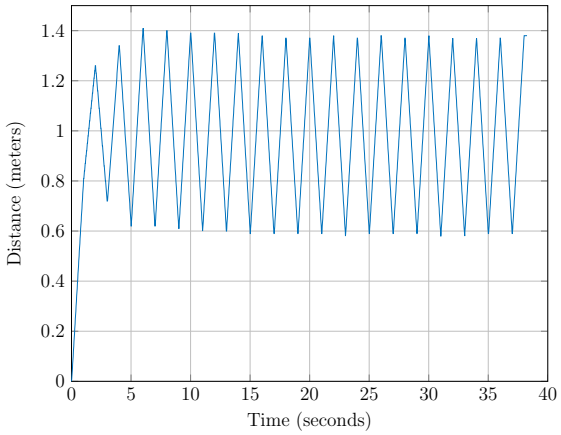


Figure 39: The orientation of the robot over time for $K_\omega^T = 1.1K_{\omega,max}^T$. The system is marginally stable

Task 14

From the definition of $d_p[k]$:

$$d_p[k+1] = p(\theta[k+1] - \theta_g) \tag{14}$$

But

$$\theta[k+1] = \theta[k] + T_s \frac{R}{L} u_\Psi^T[k]$$

Hence equation 14 becomes

$$\begin{aligned} d_p[k+1] &= p(\theta[k] + T_s \frac{R}{L} u_\Psi^T[k] - \theta_g) \\ &= p(\theta[k] + T_s \frac{R}{L} K_\Psi^T d_p[k] - \theta_g) \\ &= pT_s \frac{R}{L} K_\Psi^T d_p[k] + p(\theta[k] - \theta_g) \\ &= pT_s \frac{R}{L} K_\Psi^T d_p[k] + d_p[k] \\ &= d_p[k](1 + pT_s \frac{R}{L} K_\Psi^T d_p[k]) \end{aligned}$$

In order for this system to be stable, that is, $d_p \to 0$ as $k \to \infty$, the solution of the equation inside the parentheses should lie inside the unit circle:

$$
\begin{aligned}
\left|1+pT_s\frac{R}{L}K_\omega^T\right| &< 1 \\
-1 < 1+pT_s\frac{R}{L}K_\omega^T &< 1 \\
-2 < pT_s\frac{R}{L}K_\omega^T &< 0 \\
-\frac{2L}{pT_sR} < K_\omega^T &< 0
\end{aligned} \tag{15}
$$

Hence, the maximum value K_ω^T can take for the system to be marginally stable is $K_{\omega,max}^T = 0$, and the minimum is $K_{\omega,min}^T = -\frac{L}{pT_sR}$.

Theoretically, the value of K_ω^T can be chosen to be any value inside the interval defined in inequality 12. However, first, it would be wise to choose a value that is far enough from the maximum value so as to avoid overshoot, but close enough to it, so that convergence happens in reasonable time. Hence, in practice, it is reasonable that one would need to experiment with different values and choose one that results in balancing a small angular error, a minimal overshoot, if any, and a quick enough settling time.

Just like discussed before, the dead-beat controller is a good choice in order to get fast convergence. The dead-beat control is in this case obtained when $K_\Psi^T = \frac{L}{pT_sR} = \frac{1}{2}K_{\Psi,min}^T$.

Task 15

Inequality 15 tells us that the higher the value of p, the broader the region of values for K_Ψ^T is so that the systems is stable. Hence, the lower the value of p is, the worse the robot's ability to follow a line is.

Task 16

This part of the controller is responsible for compensating for rotational errors during translation. Since the translational velocity u_ω is zero, it is expected that the robot will not rotate away from its original bearing. Figure 40 plots the robot's bearing with regard to its original bearing of 0 degrees over time for $K_\Psi^T = 0.5K_{\Psi,min}^T$.

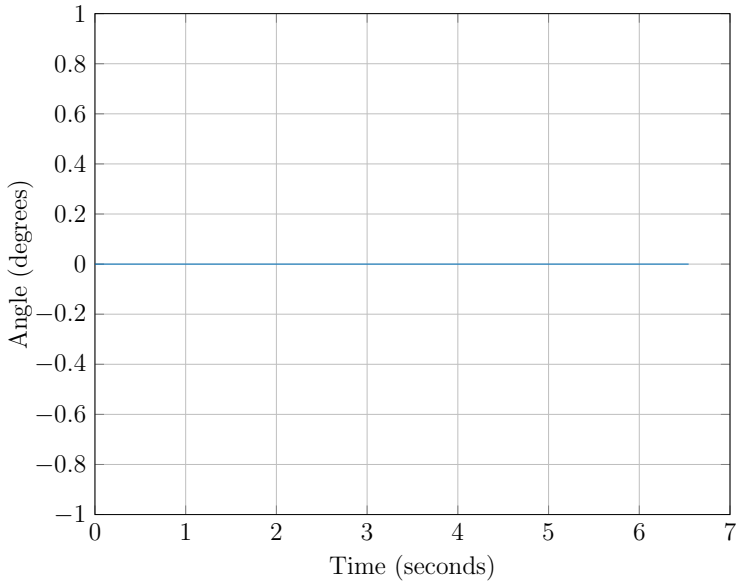


Figure 40: The angular displacement of the robot from its original bearing over time for $K_{\Psi}^{T} = 0.5K_{\Psi,min}^{T}$

Task 17

Figures 41, 43, 45, 47 and show the displacemental error of the robot for different values of K_{ω}^{T} inside the interval set by inequality 15. Figures 42, 44, 46 and 48 focus on the steady-state displacemental error. Figure 49 illustrates the $d_0[k]$ error, which is at all times zero.

The evolution of the bearing and displacement error is the same when both of the translational controllers are enabled compared to when only one of them is enabled. This happens because the behaviour of each controller does not affect the behaviour of the other, since this is an ideal system. In reality, we expect that the angular error will be non-zero.

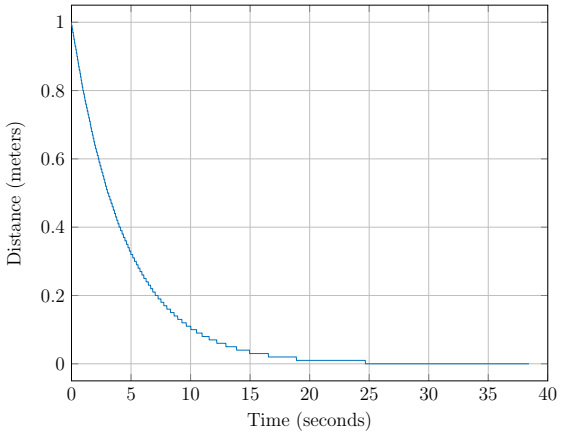


Figure 41: The error in displacement of the robot over time for $K_{\omega}^{T} = 0.1K_{\omega,max}^{T}$

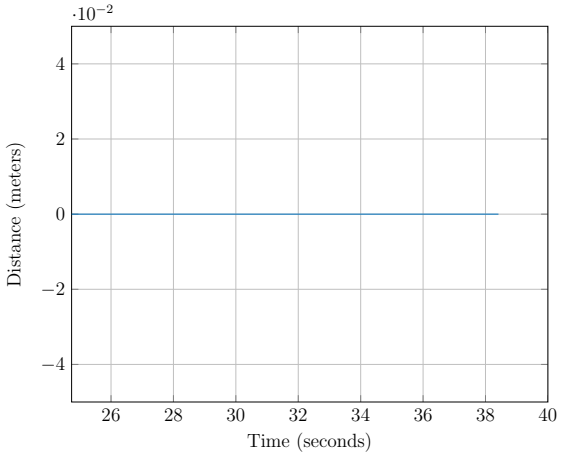


Figure 42: The steady state error in displacement of the robot for $K_{\omega}^{T} = 0.1K_{\omega,max}^{T}$

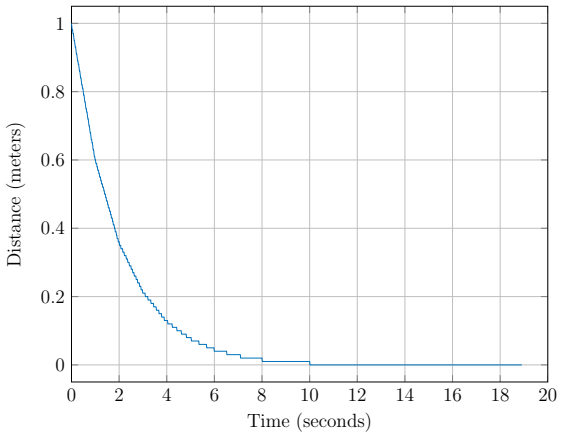


Figure 43: The error in displacement of the robot over time for $K_\omega^T = 0.2K_{\omega,max}^T$

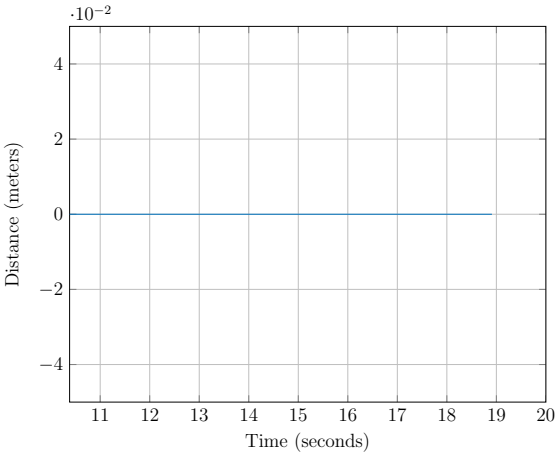


Figure 44: The steady state error in displacement of the robot for $K_\omega^T = 0.2K_{\omega,max}^T$

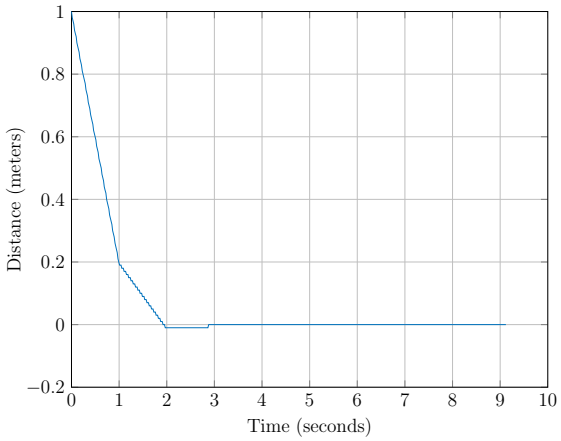


Figure 45: The error in displacement of the robot over time for $K_\omega^T = 0.5K_{\omega,max}^T$

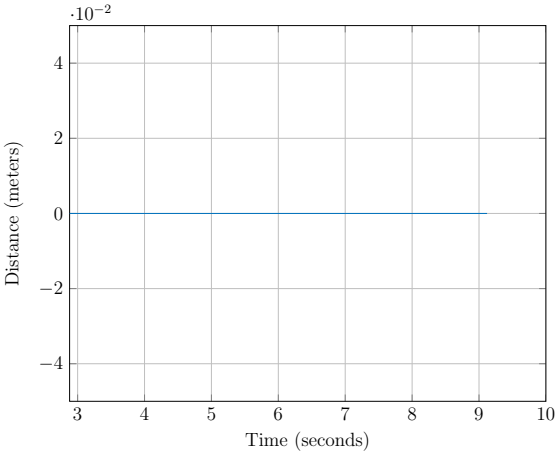


Figure 46: The steady state error in displacement of the robot for $K_\omega^T = 0.5K_{\omega,max}^T$

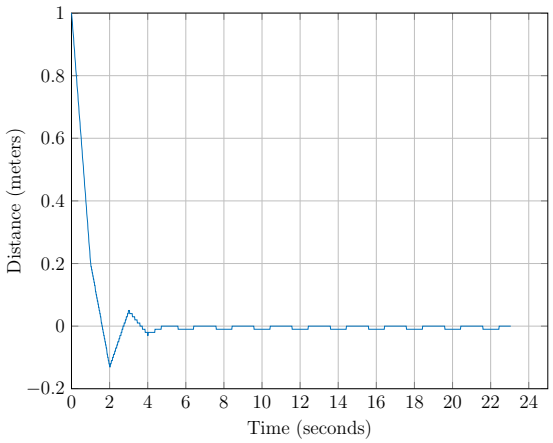


Figure 47: The error in displacement of the robot over time for $K_\omega^T = 0.75K_{\omega,max}^T$

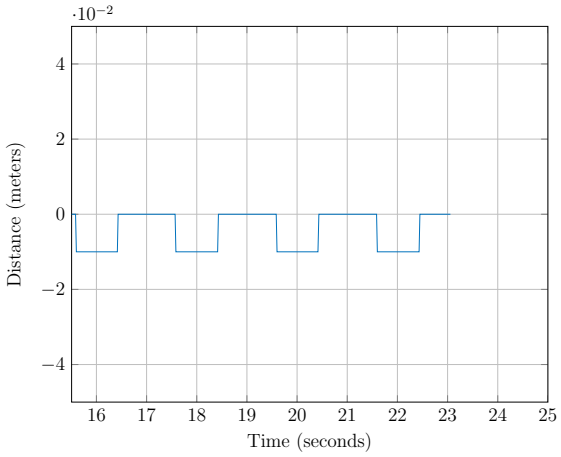


Figure 48: The steady state error in displacement of the robot for $K_\omega^T = 0.75K_{\omega,max}^T$

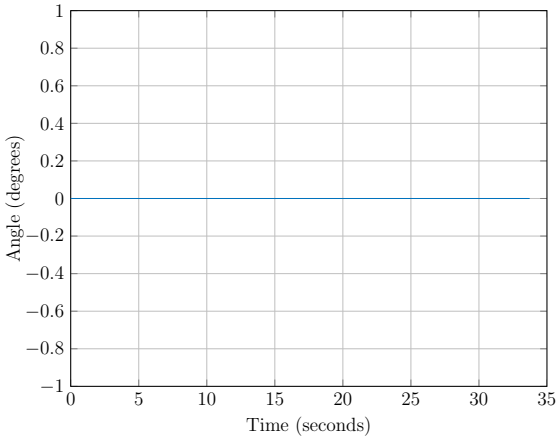


Figure 49: The steady state error in orientation of the robot for all legitimate values of K_ω^T

Task 18

The hybrid controller H is modelled formally by an 8-tuple

$$H = (Q, X, Init, f, D, E, G, R)$$

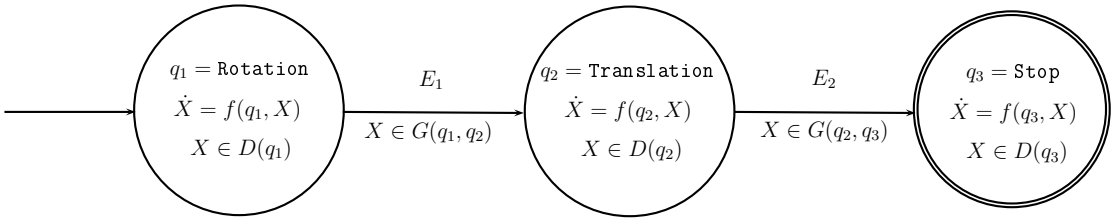


Figure 50: The hybrid automaton that controls the navigation of the robot from an initial state (x_0, y_0, θ_0) to a goal location (x_g, y_g).

- $Q \equiv \{q_1, q_2, q_3\}$ denotes the set of discrete states. The robot is in state q_1 when executing a rotation, in state q_2 when executing line-following, and in state q_3 when it is indefinitely stopped.
- $Init \equiv \{q_1\}$. The initial state is taken to be q_1.
- $X \equiv \{(x, y, \theta) : x, y \in \mathbb{R}^2, \theta \in (-180°, 180°]\}$ denotes the continuous states
- Vector fields f

$$f(q_1, X) = \begin{bmatrix} \dot{x} = Ru_\omega cos\theta \\ \dot{y} = Ru_\omega sin\theta \\ \dot{\theta} = \frac{R}{L} u_\Psi \end{bmatrix}$$

$$f(q_2, X) = \begin{bmatrix} \dot{x} = Ru_\omega cos\theta \\ \dot{y} = Ru_\omega sin\theta \\ \dot{\theta} = \frac{R}{L} u_\Psi \end{bmatrix}$$

$$f(q_2, X) = \begin{bmatrix} \dot{x} = 0 \\ \dot{y} = 0 \\ \dot{\theta} = 0 \end{bmatrix}$$

- D shows what conditions need to be satisfied in order for the automaton to stay in a state.

  $D(q_1) = \{(x, y, \theta) : x, y \in \mathbb{R}^2, |\theta_g - \theta| > \delta\}$

  $D(q_2) = \{(x, y, \theta) : x, y \in \mathbb{R}^2, (x_g - x)^2 + (y_g - y)^2 > \xi, \theta \in (-180°, 180°]\}$

  $D(q_3) = \{x, y \in \mathbb{R}^2, \theta \in (-180°, 180°]\}$
- E: The edges show which transitions are possible.

  $E_1 = \{q_1, q_2\}$

  $E_2 = \{q_2, q_3\}$

  $E = E_1 \cup E_2$
- G: The guards show under what conditions the system can transition from one to another state.

  $G(\{q_1, q_2\}) = \{\theta_g - \theta \leq \delta\}$

  $G(\{q_2, q_3\}) = \{(x_g - x)^2 + (y_g - y)^2 \leq \xi\}$
- R: Resets illustrate the values that the state takes when transitioning between states.

  $R = \{x, y, \theta\}$

Task 19

Evaluating the performance of the hybrid automaton primarily involves the evaluation of the steady-state errors regarding the position and angle of the robot with regard to the selected goal position. In order to obtain a broader understanging of how the four K_* gains influence the trajectory of the robot, twelve combinations were considered for the 4-tuple

$$(K_{\Psi}^{R}, K_{\omega}^{T}, K_{\omega}^{R}, K_{\Psi}^{T})$$

where

$$K_{\Psi}^{R} \in \{0.1, 0.2, 0.5\} \cdot K_{\Psi,max}^{R}$$

$$K_{\omega}^{T} \in \{0.1, 0.2, 0.5, 0.75\} \cdot K_{\omega,max}^{T}$$

$$K_{\omega}^{R} = 0.5K_{\omega,max}^{R}, \text{ and } K_{\Psi}^{T} = -0.5K_{\Psi,min}^{T}$$

Notice the minus sign for K_{Ψ}^{T}. Although theoretically (and using the approximative form for $d_p[k]$) this gain should be negative, simulations showed that the robot achieved lower levels of both displacemental and angular errors when considered positive[1]. For purposes of homogeneity we shall denote $K_{\Psi,max}^{T} = -K_{\Psi,min}^{T}$, so $K_{\Psi}^{T} = 0.5K_{\Psi,max}^{T} > 0$.

Figures 51(a) - 56(f) illustrate the evolution of the continuous and discrete state trajectories, over time, for the aforementioned combinations of values of the gains K_*. The goal was set to be node 1 $N1(-0.37, 1.68)$, and the robot's initial pose was $(x_0, y_0, \theta_0) \equiv (0, 0, 0)$. Hence, the distance to the goal was $d_g = 1.7203$ meters and the angle to the goal $\theta^R = 102.42°$. The distance and angle tolerance thresholds where taken to be $\delta = 2$ cm and $\xi = 2°$ respectively.

Since the actual final pose is not defined deterministically, five simulations of each possible combination of settings for the aforementioned 4-tuple were conducted. Hence, all the following figures express the mean continuous and discrete state trajectories.

Table 1 illustrates the steady-state errors e_d and e_θ regarding the distance and the angle that the robot had to travel, respectively.

$K_{\Psi}^{R}/K_{\Psi,max}^{R}$	$K_{\omega}^{T}/K_{\omega,max}^{T}$	e_d (cm)	e_θ (deg)
0.1	0.1	1.73	0.329
0.1	0.2	0.60	0.144
0.1	0.5	0.90	0.330
0.1	0.75	1.9	0.054
0.2	0.1	1.47	0.419
0.2	0.2	1.21	0.308
0.2	0.5	0.67	0.116
0.2	0.75	2.12	0.026
0.5	0.1	1.26	0.303
0.5	0.2	0.75	0.585
0.5	0.5	1.05	0.695
0.5	0.75	0.86	0.347

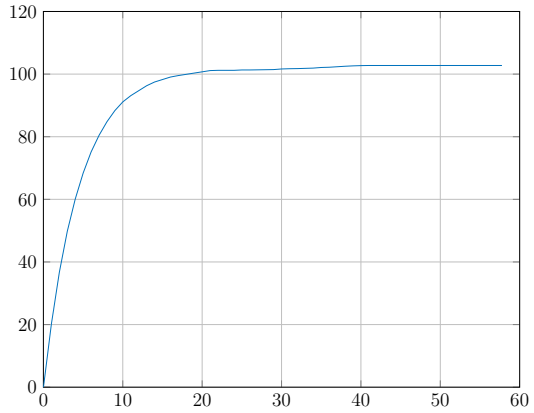
Table 1: Mean steady-state errors regarding the distance and bearing to Node 1 for the 12 different combinations considered for the 4-tuple $(K_{\Psi}^{R}, K_{\omega}^{T}, K_{\omega}^{R}, K_{\Psi}^{T})$

[1]Analysis on Task 14 showed that $K_{\Psi,min}^{T}$ is considered negative

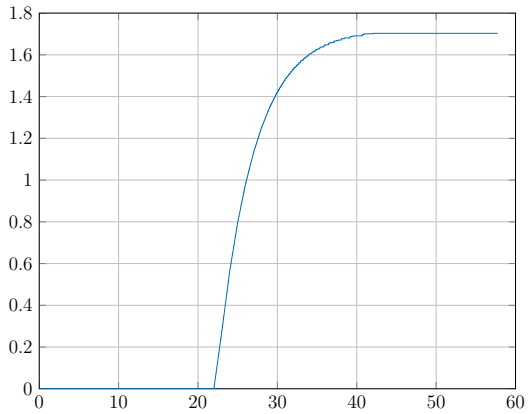
The combination of the two control components made it possible for the robot to move to any location inside the simulation environment. In particular, for the single goal considered here, the controller made it possible for the robot to approach node 1 within a radius of length δ (with one exception, when $(K_\Psi^R/K_{\Psi,max}^R, K_\omega^T/K_{\omega,max}^T) \equiv (0.2, 0.75)$) and with a bearing well within the 2° upper threshold. The second component of the line-following controller was partly responsible for reducing the bearing errors compared to the case when only the rotational controller was enabled. However, in order for the second component's input to be able to kick in in the first place, it was necessary to limit the translational velocity of the robot to a value of $u_\omega = 400$, so as to permit the correctional input of this component to become active, since there is an upper limit to the velocities the robot can achieve, either in simulation or in real-life.

The discrete state trajectory figures express what was expected: as the value of a gain K_* increases, the time that the robot stays in the corresponding state decreases. What was not expected, though, was the temporary spikes into state `Stop` for all cases when $K_\omega^T/K_{\omega,max}^T = 0.75$. This makes sense, since that high a value for the gain of the translational part of the line-following controller makes the robot overshoot the goal. Its momentum is such that, while it approaches the goal within a distance of δ and its state transitions from `Translation` to `Stop`, it cannot physically stop and steps out of the circle with radius δ, and thus its state goes back to `Translation`. The duration the robot stays at state `Stop` is the duration of time it spends inside this virual circle with radius δ. This time interval is observed to be unequal among all three cases, due to the fact that the robot travels forwards (part I of the line-following controller) and rotates (part II of the line-following controller) simultaneously.

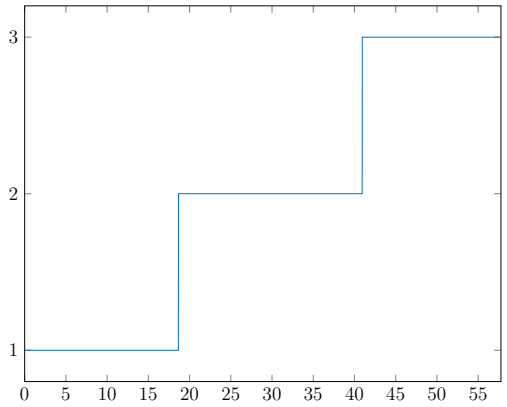
Furthermore, we note that the robot does not start to translate immediately after the state switch to q_2.



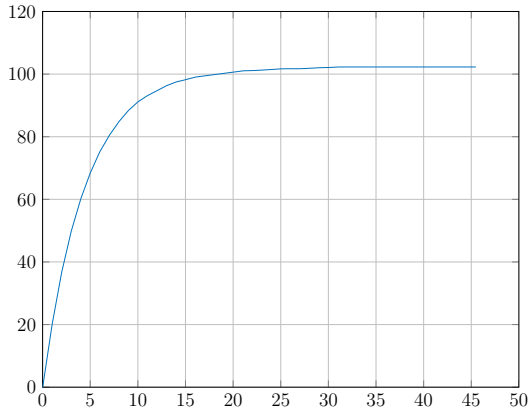
(a) The bearing of the robot over time for $(K^R_\Psi, K^T_\omega) \equiv (0.1K^R_{\Psi,max}, 0.1K^T_{\omega,max})$



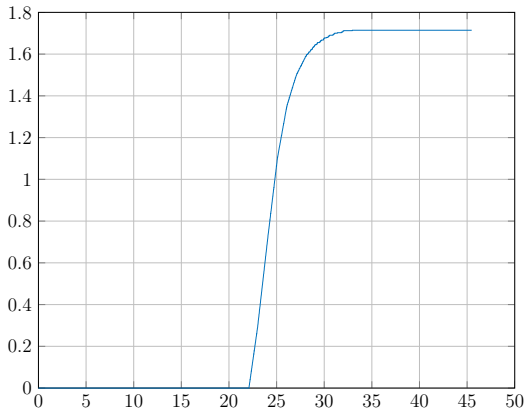
(b) The distance of the robot to the origin over time for $(K^R_\Psi, K^T_\omega) \equiv (0.1K^R_{\Psi,max}, 0.1K^T_{\omega,max})$



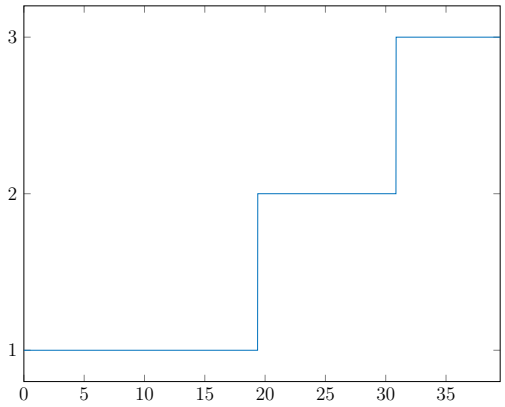
(c) The discrete state trajectory for $(K^R_\Psi, K^T_\omega) \equiv (0.1K^R_{\Psi,max}, 0.1K^T_{\omega,max})$. 1 denotes `Rotation`, 2 denotes `Translation` and 3 denotes `Stop`



(d) The bearing of the robot in over time for $(K^R_\Psi, K^T_\omega) \equiv (0.1K^R_{\Psi,max}, 0.2K^T_{\omega,max})$

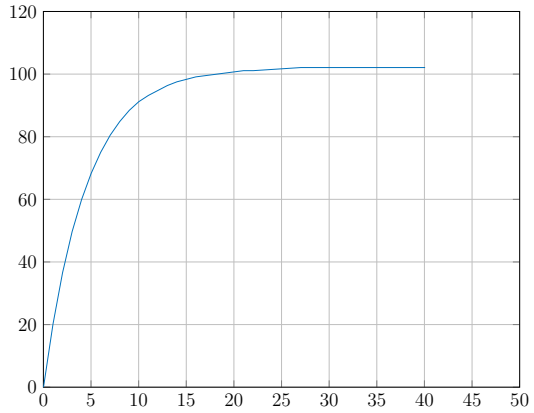


(e) The distance of the robot to the origin over time for $(K^R_\Psi, K^T_\omega) \equiv (0.1K^R_{\Psi,max}, 0.2K^T_{\omega,max})$

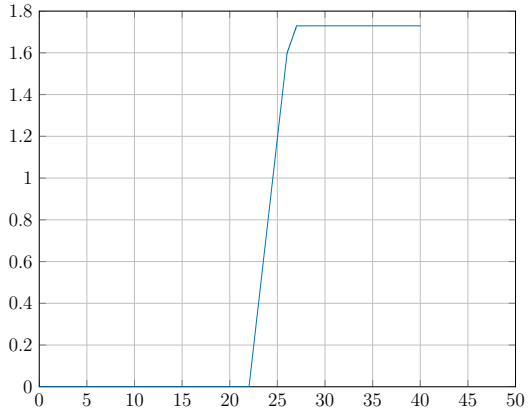


(f) The discrete state trajectory for $(K^R_\Psi, K^T_\omega) \equiv (0.1K^R_{\Psi,max}, 0.2K^T_{\omega,max})$. 1 denotes `Rotation`, 2 denotes `Translation` and 3 denotes `Stop`

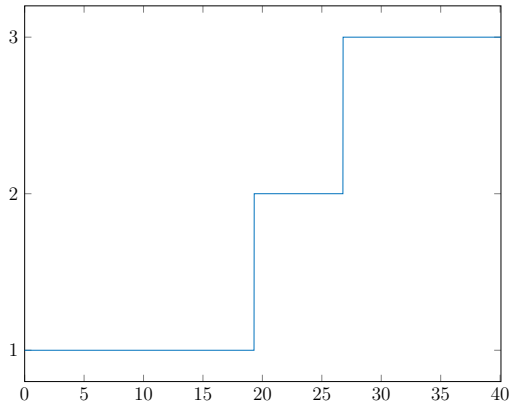
Figure 51



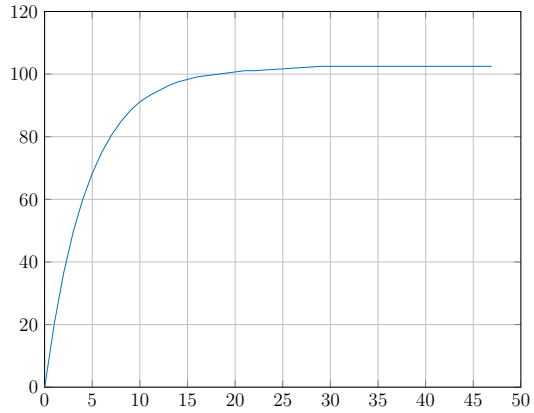
(a) The bearing of the robot over time for $(K^R_\Psi, K^T_\omega) \equiv (0.1K^R_{\Psi,max}, 0.5K^T_{\omega,max})$



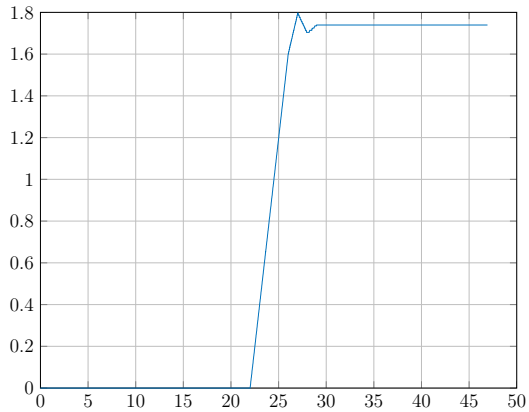
(b) The distance of the robot to the origin over time for $(K^R_\Psi, K^T_\omega) \equiv (0.1K^R_{\Psi,max}, 0.5K^T_{\omega,max})$



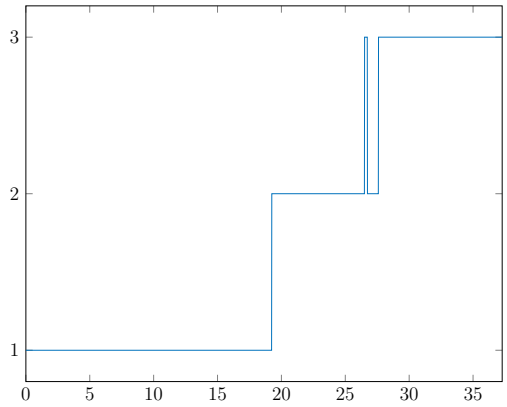
(c) The discrete state trajectory for $(K^R_\Psi, K^T_\omega) \equiv (0.1K^R_{\Psi,max}, 0.5K^T_{\omega,max})$. 1 denotes `Rotation`, 2 denotes `Translation` and 3 denotes `Stop`



(d) The bearing of the robot in over time for $(K^R_\Psi, K^T_\omega) \equiv (0.1K^R_{\Psi,max}, 0.75K^T_{\omega,max})$

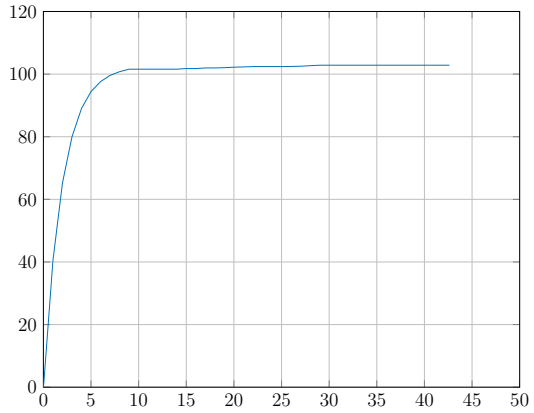


(e) The distance of the robot to the origin over time for $(K^R_\Psi, K^T_\omega) \equiv (0.1K^R_{\Psi,max}, 0.75K^T_{\omega,max})$

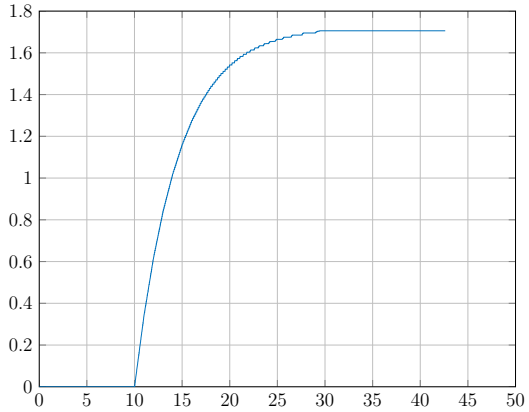


(f) The discrete state trajectory for $(K^R_\Psi, K^T_\omega) \equiv (0.1K^R_{\Psi,max}, 0.75K^T_{\omega,max})$. 1 denotes `Rotation`, 2 denotes `Translation` and 3 denotes `Stop`

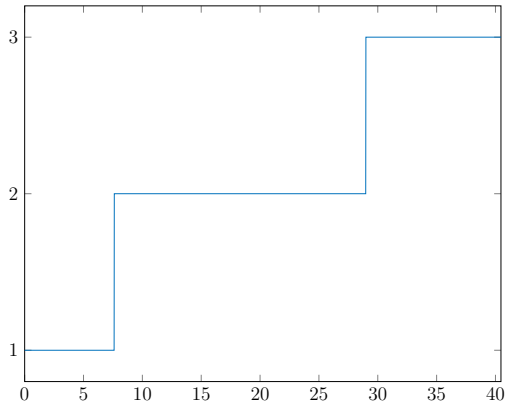
Figure 52



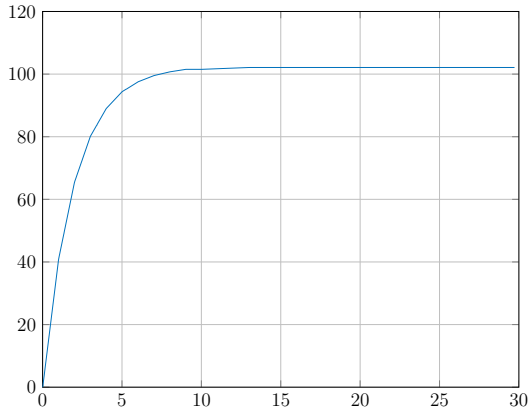
(a) The bearing of the robot over time for $(K_\Psi^R, K_\omega^T) \equiv (0.2K_{\Psi,max}^R, 0.1K_{\omega,max}^T)$



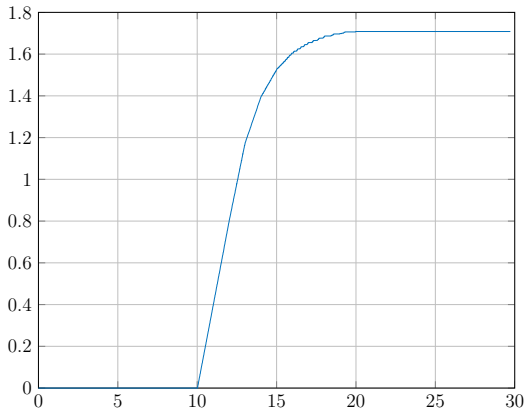
(b) The distance of the robot to the origin over time for $(K_\Psi^R, K_\omega^T) \equiv (0.2K_{\Psi,max}^R, 0.1K_{\omega,max}^T)$



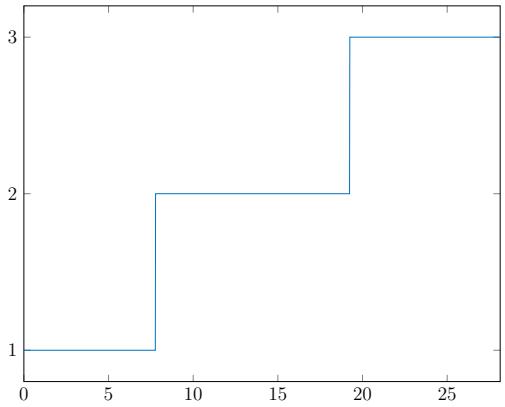
(c) The discrete state trajectory for $(K_\Psi^R, K_\omega^T) \equiv (0.2K_{\Psi,max}^R, 0.1K_{\omega,max}^T)$. 1 denotes `Rotation`, 2 denotes `Translation` and 3 denotes `Stop`



(d) The bearing of the robot in over time for $(K_\Psi^R, K_\omega^T) \equiv (0.2K_{\Psi,max}^R, 0.2K_{\omega,max}^T)$

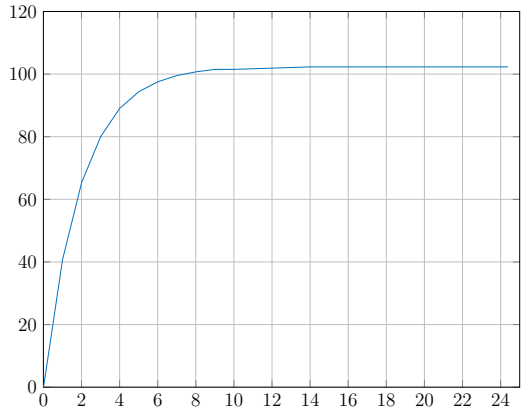


(e) The distance of the robot to the origin over time for $(K_\Psi^R, K_\omega^T) \equiv (0.2K_{\Psi,max}^R, 0.2K_{\omega,max}^T)$

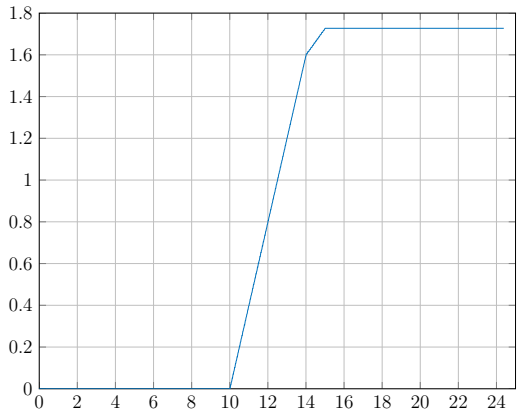


(f) The discrete state trajectory for $(K_\Psi^R, K_\omega^T) \equiv (0.2K_{\Psi,max}^R, 0.2K_{\omega,max}^T)$. 1 denotes `Rotation`, 2 denotes `Translation` and 3 denotes `Stop`

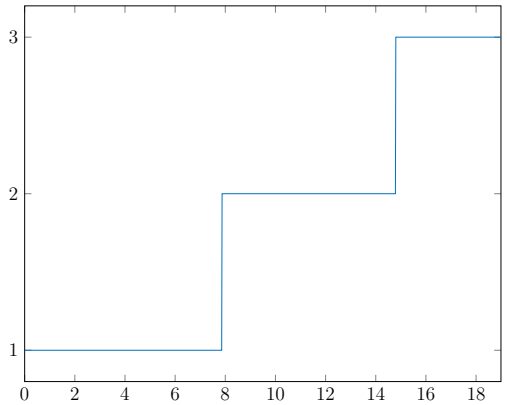
Figure 53



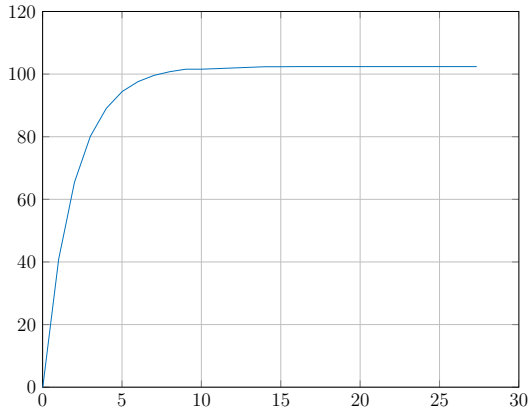
(a) The bearing of the robot over time for $(K_\Psi^R, K_\omega^T) \equiv (0.2K_{\Psi,max}^R, 0.5K_{\omega,max}^T)$



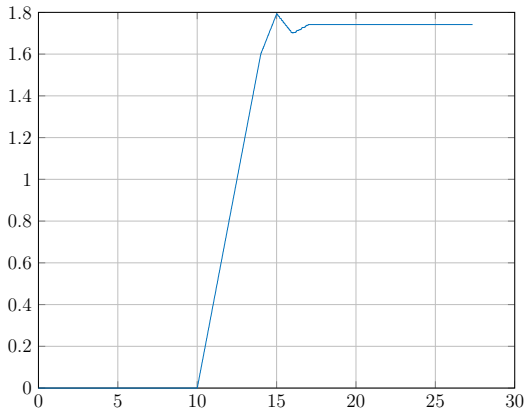
(b) The distance of the robot to the origin over time for $(K_\Psi^R, K_\omega^T) \equiv (0.2K_{\Psi,max}^R, 0.5K_{\omega,max}^T)$



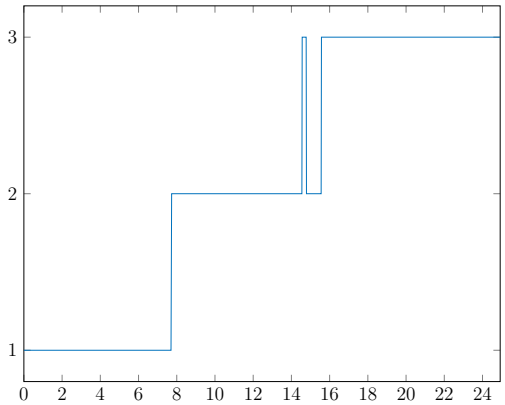
(c) The discrete state trajectory for $(K_\Psi^R, K_\omega^T) \equiv (0.2K_{\Psi,max}^R, 0.5K_{\omega,max}^T)$. 1 denotes `Rotation`, 2 denotes `Translation` and 3 denotes `Stop`



(d) The bearing of the robot in over time for $(K_\Psi^R, K_\omega^T) \equiv (0.2K_{\Psi,max}^R, 0.75K_{\omega,max}^T)$

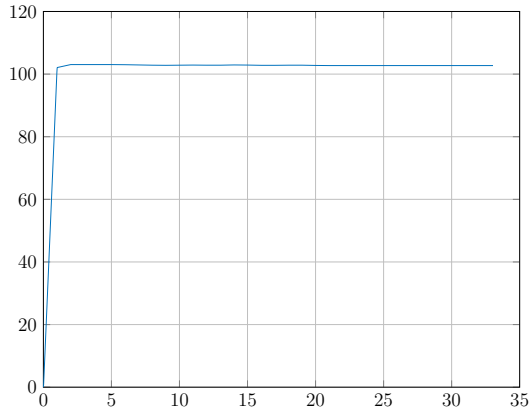


(e) The distance of the robot to the origin over time for $(K_\Psi^R, K_\omega^T) \equiv (0.2K_{\Psi,max}^R, 0.75K_{\omega,max}^T)$

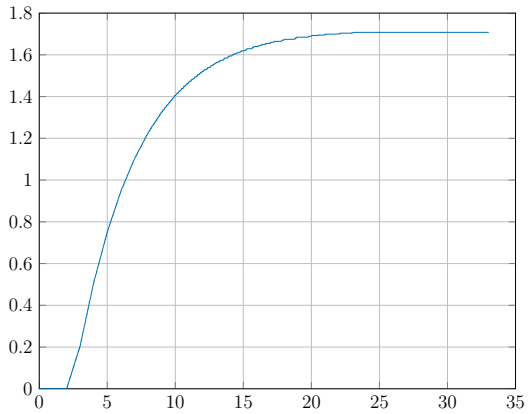


(f) The discrete state trajectory for $(K_\Psi^R, K_\omega^T) \equiv (0.2K_{\Psi,max}^R, 0.75K_{\omega,max}^T)$. 1 denotes `Rotation`, 2 denotes `Translation` and 3 denotes `Stop`

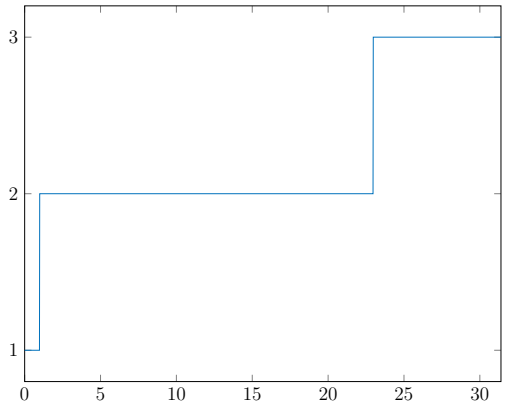
Figure 54



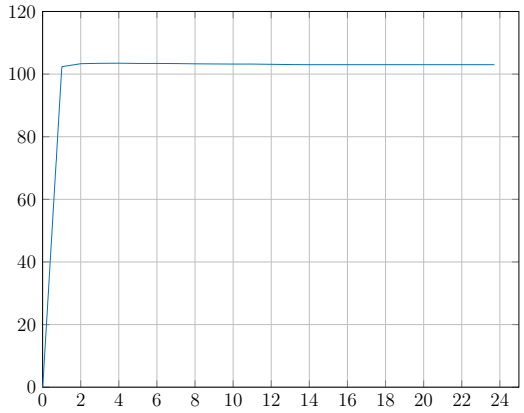
(a) The bearing of the robot over time for $(K^R_\Psi, K^T_\omega) \equiv (0.5K^R_{\Psi,max}, 0.1K^T_{\omega,max})$



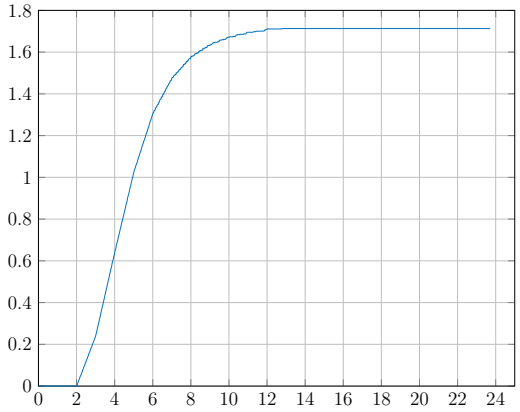
(b) The distance of the robot to the origin over time for $(K^R_\Psi, K^T_\omega) \equiv (0.5K^R_{\Psi,max}, 0.1K^T_{\omega,max})$



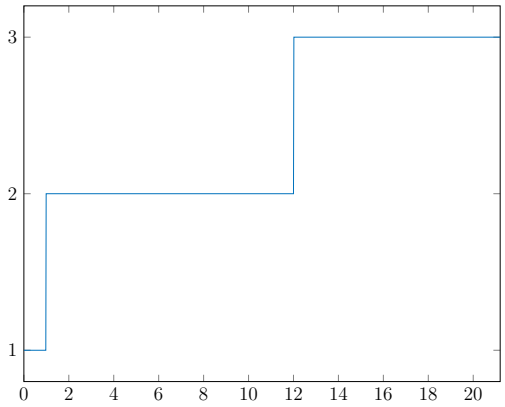
(c) The discrete state trajectory for $(K^R_\Psi, K^T_\omega) \equiv (0.5K^R_{\Psi,max}, 0.1K^T_{\omega,max})$. 1 denotes `Rotation`, 2 denotes `Translation` and 3 denotes `Stop`



(d) The bearing of the robot in over time for $(K^R_\Psi, K^T_\omega) \equiv (0.5K^R_{\Psi,max}, 0.2K^T_{\omega,max})$

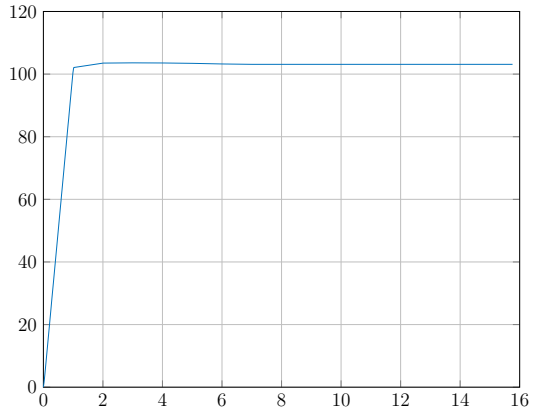


(e) The distance of the robot to the origin over time for $(K^R_\Psi, K^T_\omega) \equiv (0.5K^R_{\Psi,max}, 0.2K^T_{\omega,max})$

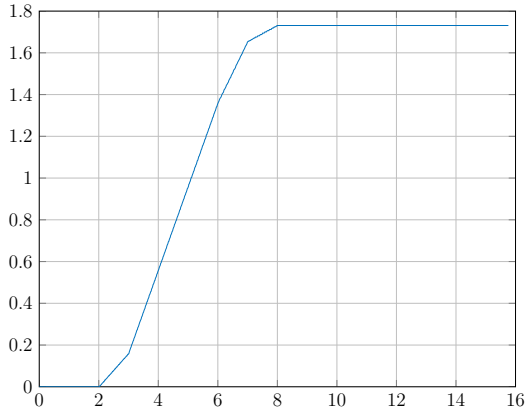


(f) The discrete state trajectory for $(K^R_\Psi, K^T_\omega) \equiv (0.5K^R_{\Psi,max}, 0.2K^T_{\omega,max})$. 1 denotes `Rotation`, 2 denotes `Translation` and 3 denotes `Stop`

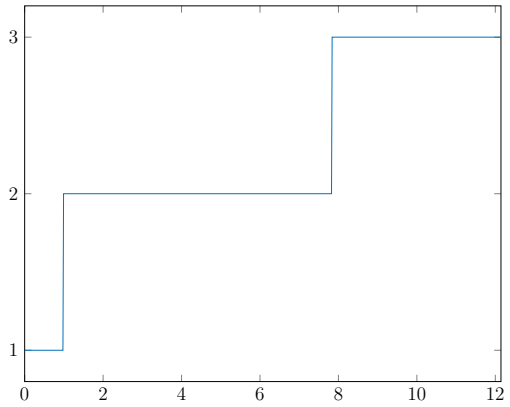
Figure 55



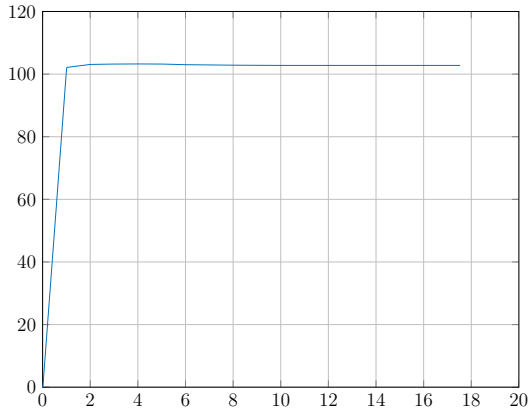
(a) The bearing of the robot over time for $(K^R_\Psi, K^T_\omega) \equiv (0.5K^R_{\Psi,max}, 0.5K^T_{\omega,max})$



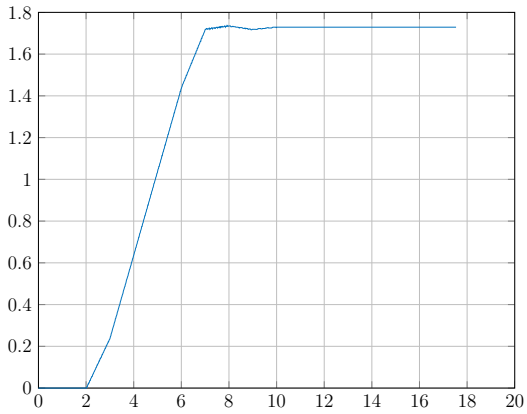
(b) The distance of the robot to the origin over time for $(K^R_\Psi, K^T_\omega) \equiv (0.5K^R_{\Psi,max}, 0.5K^T_{\omega,max})$



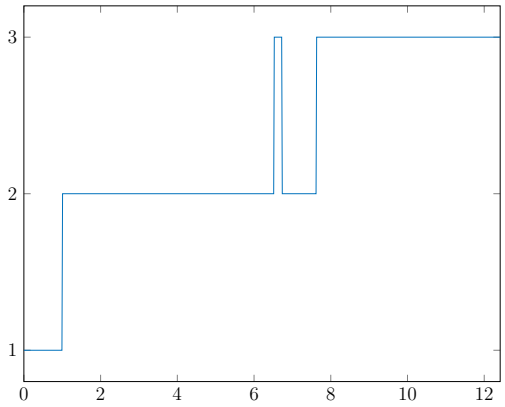
(c) The discrete state trajectory for $(K^R_\Psi, K^T_\omega) \equiv (0.5K^R_{\Psi,max}, 0.5K^T_{\omega,max})$. 1 denotes `Rotation`, 2 denotes `Translation` and 3 denotes `Stop`



(d) The bearing of the robot in over time for $(K^R_\Psi, K^T_\omega) \equiv (0.5K^R_{\Psi,max}, 0.75K^T_{\omega,max})$



(e) The distance of the robot to the origin over time for $(K^R_\Psi, K^T_\omega) \equiv (0.5K^R_{\Psi,max}, 0.75K^T_{\omega,max})$



(f) The discrete state trajectory for $(K^R_\Psi, K^T_\omega) \equiv (0.5K^R_{\Psi,max}, 0.75K^T_{\omega,max})$. 1 denotes `Rotation`, 2 denotes `Translation` and 3 denotes `Stop`

Figure 56

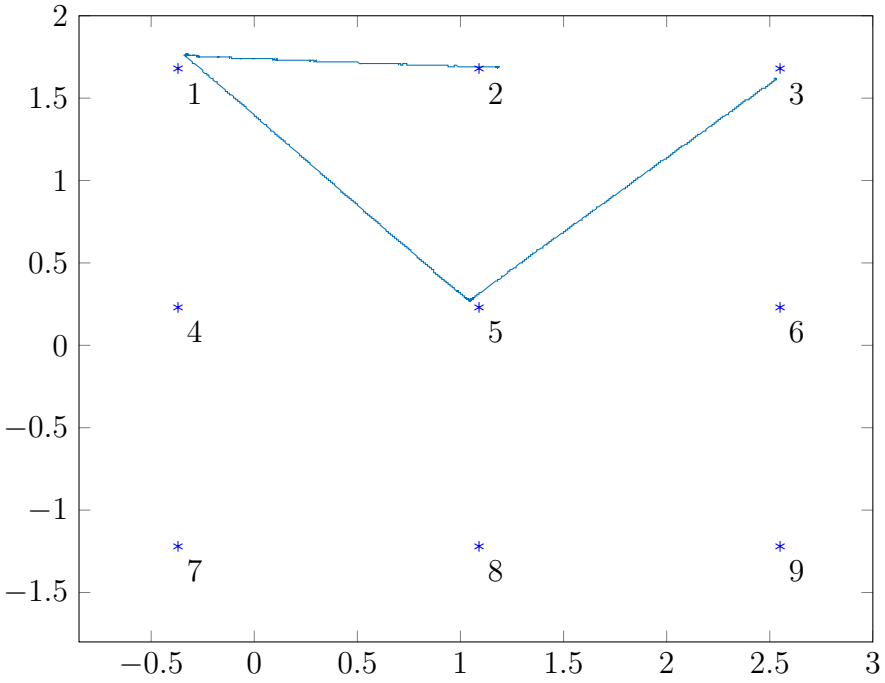


Figure 57: The trajectory of the robot in real-life conditions. The anglular tolerance was set to $\xi = 4°$ and the distance tolerance to $\delta = 6$ cm. Direction of travel is counter-clockwise, starting from node 2.

Task 21

Reaching a certain accuracy is difficult. First one has to orient and then to translate the robot. As there is no automatic line-following, one has to correct many times to reach a certain point, while reaching a certain area should be easier. The accuracy in both bearing and distance is related to the sensitivity of the motors driving the wheels to their input signals and their output rotation precision.

Task 22

Figure 57 plots the trajectory of the robot for an experiment performed at KTH's Smart Mobility Lab of the Automatic Control Department. The robot was initially placed in the vicinity of node 2. Its first goal was dictated to be node 1, and from there, nodes 5 and 3.

The robot's bearing and distance tolarance from a target was set to $\xi = 4°$ and the distance tolerance to $\delta = 6$ cm. The controller's gains were set to

$$(\frac{K_\Psi^R}{K_{\Psi,max}^R}, \frac{K_\omega^T}{K_{\omega,max}^T}, \frac{K_\omega^R}{K_{\omega,max}^R}, \frac{K_\Psi^T}{K_{\Psi,max}^T}) \equiv (0.2, 0.2, 0.5, 0.5)$$

where $K_{*,max} > 0$.

The robot's distance errors from nodes 1, 5 and 3 were $e_d^1 = 8.94$ cm, $e_d^5 = 6.40$ cm and $e_d^3 = 6.32$ cm and respectively. Its respective bearing errors were $e_\theta^1 = 3.43°$ $e_\theta^5 = 2.14°$ and $e_\theta^3 = 0.017°$.

Figure 58 shows the bearing of the robot over time. In this context, poses are sampled

once every 30 clock ticks. Hence, the denser the pose map, the slower it took the robot to move from place A to place B.

It wasn't before this experiment that we considered a positive K_ω^T, instead of a negative one, as the theoretical analysis pointed to. The second component of the line-following controller showed significantly worse performance when $K_\omega^T < 0$ than when $K_\omega^T > 0$.

When the robot was asked to go from node 2 to node 1, the angle between it and node 1 was $1.68° < \xi = 4°$. Hence, the robot first executed line-following, and not rotation. The augmented error in distance, relative to the other two, is hence the fact that $K_\omega^T < 0$ during the experimental phase.

Figure 59 illustrates the evolution of the position of the robot over time.

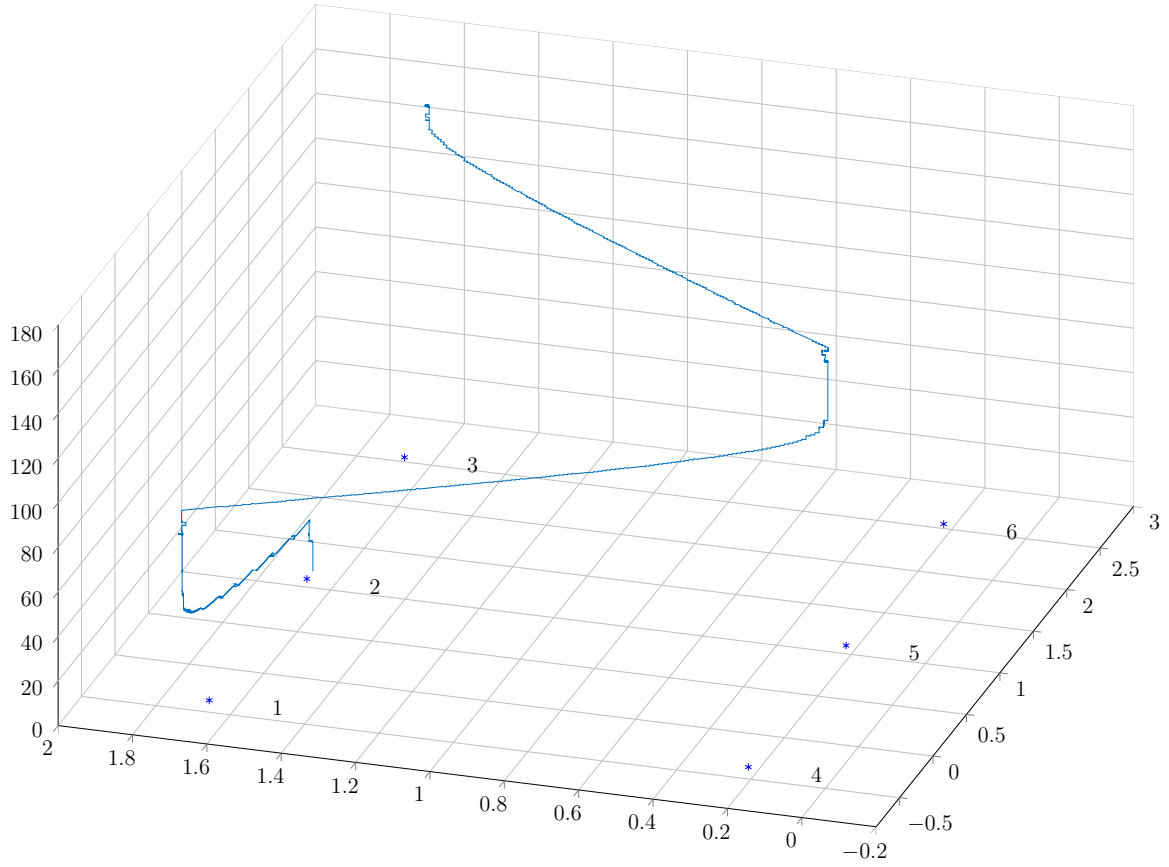


Figure 59: The evolution of the robot's position over time. The vertical axis denotes time units in seconds.

Figure 60 plots the the distance of the robot from the origin $O(0, 0)$ over time for the duration of this experiment. Figures 61, 62, and 63 illustrate the evolution of the error in the robot's distance to each of its three goals.

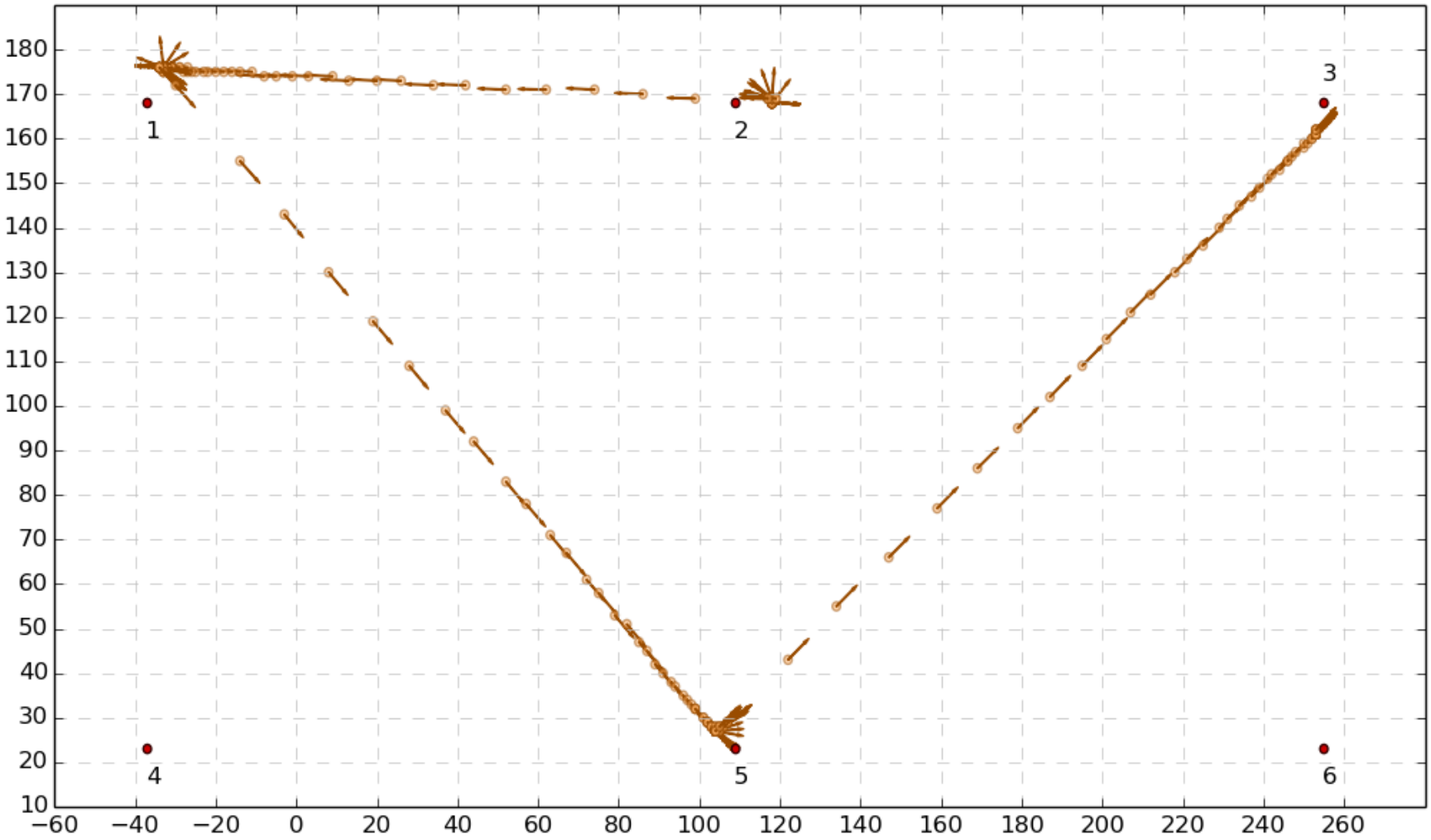


Figure 58: The robot's pose over time. Starting point is node 2.

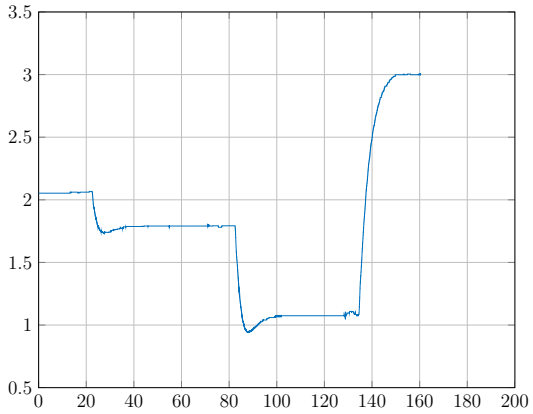


Figure 60: The robot's distance to the origin over time.

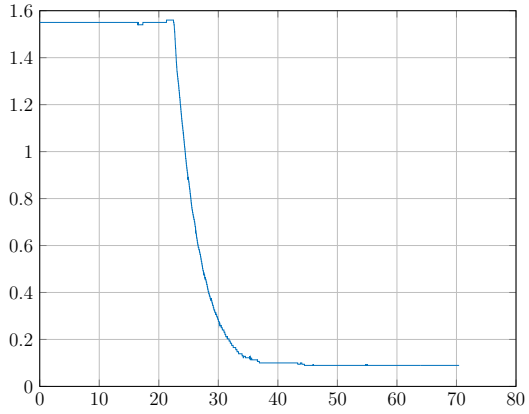


Figure 61: The robot travels from node 2 to node 1. Its steady-state distance error is 8.94 cm.

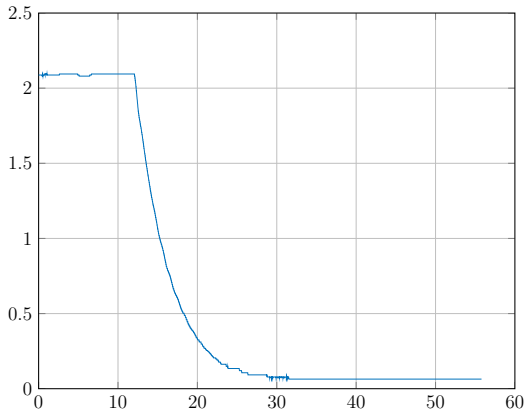


Figure 62: The robot travels from node 1 to node 5. Its steady-state distance error is 6.40 cm.

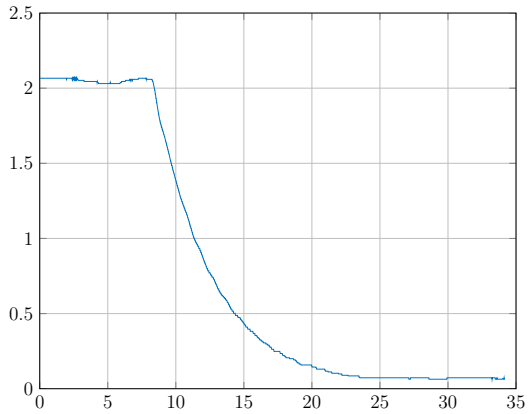


Figure 63: The robot travels from node 5 to node 3. Its steady-state distance error is 6.32 cm.

Figure 64 plots the bearing of the robot over time for the duration of this experiment. Figures 65, 66, and 67 illustrate the evolution of the error in the robot's orientation towards a goal.

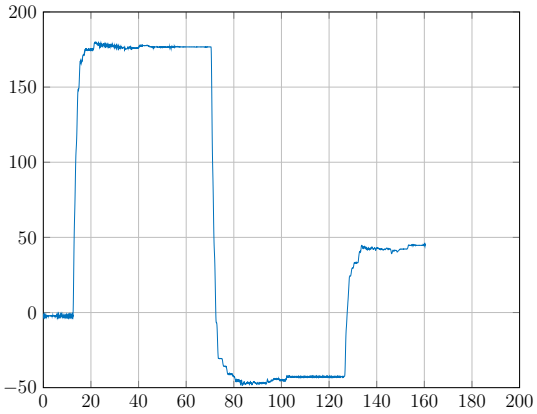


Figure 64: The robot's bearing over the duration of the experiment.

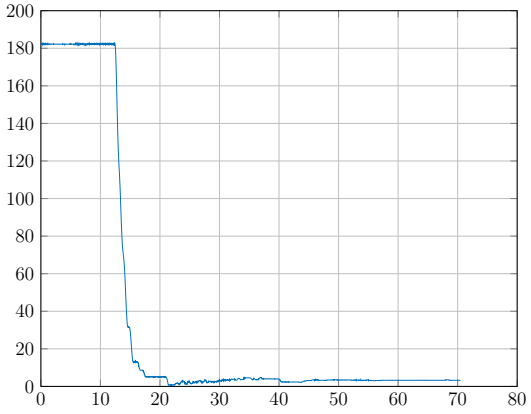


Figure 65: The robot travels from node 2 to node 1. Its steady-state bearing error towards its goal orientation is 3.43°.

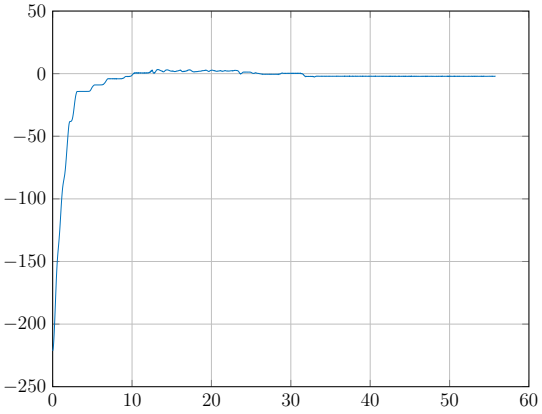


Figure 66: The robot travels from node 1 to node 5. Its steady-state bearing error towards its goal orientation is 2.14°.

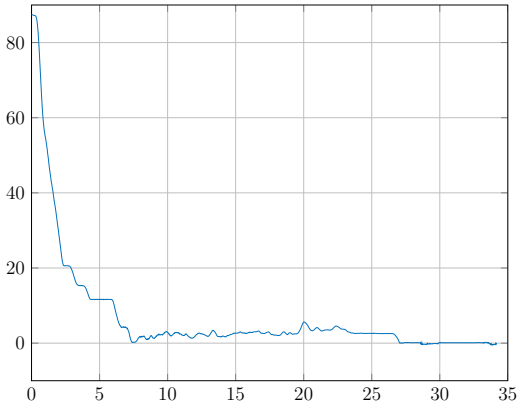


Figure 67: The robot travels from node 5 to node 3. Its steady-state bearing error towards its goal orientation is 0.017°.

Qualitatively, these experimental results are consistent with the simulations performed that preceded it, given the fact that theoretical and practical results will never coincide exactly quantitatively. All bearing errors are within the angular tolerance, while, the distance errors are close to the distance tolerance set, despite the fact that this experiment was conducted with a negative gain for the bearing correction component of the line-following controller.