Homework 3 (Tasks 1-19) in EL2450 Hybrid and Embedded Control Systems

Devendra Sharma
950815-5497
devendra@kth.se

Kartik Chari
960807-0174
kartikc@kth.se

Merav Modi
19950120-0191
merav@kth.se

Khushdeep Singh
19940126-5955
ksmann@kth.se

February 26, 2019

Task 1

Given the translational velocity (v) and rotational velocity (ω), we can calculate u_r and u_l as explained below. Given equations:

$$v = \frac{u_r + u_l}{2} \tag{1}$$

$$\omega = u_r - u_l \tag{2}$$

Solving equation (1) and (2) by adding them, will give us u_r and u_l:

$$u_r = \frac{2v + \omega}{2}$$
$$u_l = \frac{2v - \omega}{2}$$

Task 2

The transition system $\mathcal{T}$ is defined as:

$$\mathcal{T} = (S, S^0, \sum, \rightarrow, AP, L)$$

where

- S is the set of the discretized states of the robot i.e. $S = \{R_1, R_2, ..., R_{\mathrm{K}}\}$
  Here, K is considered as 18. If we consider a bounding rectangle around our robot with length = 38 cm = breadth, the area occupied by it is 0.1444 m^2. So, the minimum area of each region in workspace should be 0.1444 m^2. For K = 18, the area of each region turned out to be 0.5 m^2 with length = 0.707m = breadth.(See fig.1)

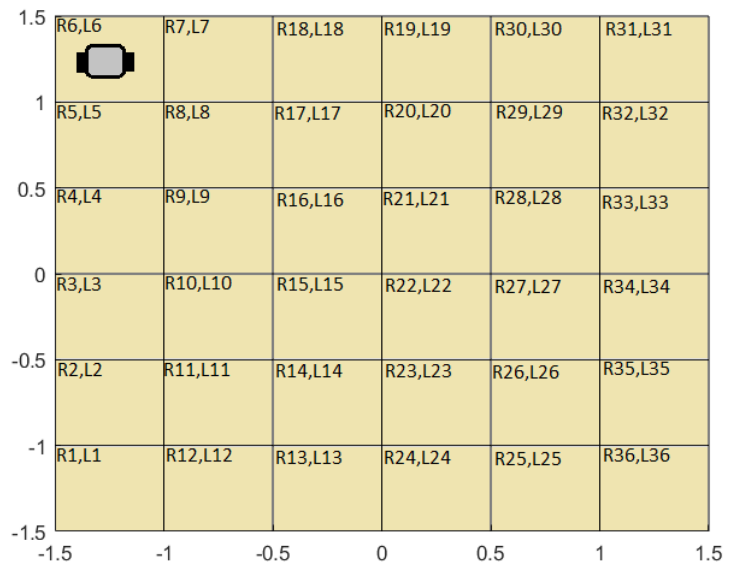


Figure 1: Workspace discretization in Rectangular regions

- $S^0 = \{R_6\}$
- $\sum = \{up, down, right, left, stop\}$
- $\rightarrow = \{(R_1, stop, R_1), (R_1, up, R_2), (R_1, right, R_{12}),$
  $(R_2, stop, R_2), (R_2, up, R_3), (R_2, down, R_1), (R_2, right, R_{11}),$
  $(R_3, stop, R_3), (R_3, up, R_4), (R_3, down, R_2), (R_3, right, R_{10}),$
  $(R_4, stop, R_4), (R_4, up, R_5), (R_4, down, R_3), (R_4, right, R_9),$
  $(R_5, stop, R_5), (R_5, up, R_6), (R_5, down, R_4), (R_5, right, R_8),$
  $(R_6, stop, R_6), (R_6, down, R_5), (R_6, right, R_7),$
  $(R_7, stop, R_7), (R_7, down, R_8), (R_7, left, R_6), (R_7, right, R_{18}),$
  $(R_8, stop, R_8), (R_8, up, R_7), (R_8, down, R_9), (R_8, left, R_5), (R_8, right, R_{17}),$
  $(R_9, stop, R_9), (R_9, up, R_8), (R_9, down, R_{10}), (R_9, left, R_4), (R_9, right, R_{16}),$
  $(R_{10}, stop, R_{10}), (R_{10}, up, R_9), (R_{10}, down, R_{11}), (R_{10}, left, R_3), (R_{10}, right, R_{15}),$
  $(R_{11}, stop, R_{11}), (R_{11}, up, R_{10}), (R_{11}, down, R_{12}), (R_{11}, left, R_2), (R_{11}, right, R_{14}),$
  $(R_{12}, stop, R_{12}), (R_{12}, up, R_{11}), (R_{12}, left, R_1), (R_{12}, right, R_{13}),$
  $(R_{13}, stop, R_{13}), (R_{13}, up, R_{14}), (R_{13}, left, R_{12}), (R_{13}, right, R_{24}),$
  $(R_{14}, stop, R_{14}), (R_{14}, up, R_{15}), (R_{14}, down, R_{13}), (R_{14}, left, R_{11}), (R_{14}, right, R_{23}),$
  $(R_{15}, stop, R_{15}), (R_{15}, up, R_{16}), (R_{15}, down, R_{14}), (R_{15}, left, R_{10}), (R_{15}, right, R_{22}),$
  $(R_{16}, stop, R_{16}), (R_{16}, up, R_{17}), (R_{16}, down, R_{15}), (R_{16}, left, R_9), (R_{16}, right, R_{21}),$
  $(R_{17}, stop, R_{17}), (R_{17}, up, R_{18}), (R_{17}, down, R_{16}), (R_{17}, left, R_8), (R_{17}, right, R_{20}),$
  $(R_{18}, stop, R_{18}), (R_{18}, down, R_{17}), (R_{18}, left, R_7), (R_{18}, right, R_{19}),$
  $(R_{19}, stop, R_{19}), (R_{19}, down, R_{20}), (R_{19}, left, R_{18}), (R_{19}, right, R_{30}),$
  $(R_{20}, stop, R_{20}), (R_{20}, up, R_{19}), (R_{20}, down, R_{21}), (R_{20}, left, R_{17}), (R_{20}, right, R_{29}),$
  $(R_{21}, stop, R_{21}), (R_{21}, up, R_{20}), (R_{21}, down, R_{22}), (R_{21}, left, R_{16}), (R_{21}, right, R_{28}),$
  $(R_{22}, stop, R_{22}), (R_{22}, up, R_{21}), (R_{22}, down, R_{23}), (R_{22}, left, R_{15}), (R_{22}, right, R_{27}),$
  $(R_{23}, stop, R_{23}), (R_{23}, up, R_{22}), (R_{23}, down, R_{24}), (R_{23}, left, R_{14}), (R_{23}, right, R_{26}),$
  $(R_{24}, stop, R_{24}), (R_{24}, up, R_{23}), (R_{24}, left, R_{13}), (R_{24}, right, R_{25}),$
  $(R_{25}, stop, R_{25}), (R_{25}, up, R_{26}), (R_{25}, left, R_{24}), (R_{25}, right, R_{36}),$

$(R_{26}, stop, R_{26}), (R_{26}, up, R_{27}), (R_{26}, down, R_{25}), (R_{26}, left, R_{23}), (R_{26}, right, R_{35}),$
$(R_{27}, stop, R_{27}), (R_{27}, up, R_{28}), (R_{27}, down, R_{26}), (R_{27}, left, R_{22}), (R_{27}, right, R_{34}),$
$(R_{28}, stop, R_{28}), (R_{28}, up, R_{29}), (R_{28}, down, R_{27}), (R_{28}, left, R_{21}), (R_{28}, right, R_{33}),$
$(R_{29}, stop, R_{29}), (R_{29}, up, R_{30}), (R_{29}, down, R_{28}), (R_{29}, left, R_{20}), (R_{29}, right, R_{32}),$
$(R_{30}, stop, R_{30}), (R_{30}, down, R_{29}), (R_{30}, left, R_{19}), (R_{30}, right, R_{31}),$
$(R_{31}, stop, R_{31}), (R_{31}, down, R_{32}), (R_{31}, left, R_{30}),$
$(R_{32}, stop, R_{32}), (R_{32}, up, R_{31}), (R_{32}, down, R_{33}), (R_{32}, left, R_{29}),$
$(R_{33}, stop, R_{33}), (R_{33}, up, R_{32}), (R_{33}, down, R_{34}), (R_{33}, left, R_{28}),$
$(R_{34}, stop, R_{34}), (R_{34}, up, R_{33}), (R_{34}, down, R_{35}), (R_{34}, left, R_{27}),$
$(R_{35}, stop, R_{35}), (R_{35}, up, R_{34}), (R_{35}, down, R_{36}), (R_{35}, left, R_{26}),$
$(R_{36}, stop, R_{36}), (R_{36}, up, R_{35}), (R_{36}, left, R_{25}),$
$\}$

- $AP = \{r_1, r_2, r_3, r_4, r_5, r_6, r_7, r_8, r_9, r_{10}, r_{11}, r_{12}, r_{13}, r_{14}, r_{15}, r_{16}, r_{17}, r_{18}\}$ This means each criterion checks whether the robot is in the region R or not.

- $L(r_1) = L_1;\ L(r_2) = L_2;\ L(r_3) = L_3; L(r_4) = L_4;\ L(r_5) = L_5;\ L(r_6) = L_6$
$L(r_7) = L_7;\ L(r_8) = L_8;\ L(r_9) = L_9; L(r_{10}) = L_{10};\ L(r_{11}) = L_{11};\ L(r_{12}) = L_{12}$
$L(r_{13}) = L_{13};\ L(r_{14}) = L_{14};\ L(r_{15}) = L_{15}; L(r_{16}) = L_{16};\ L(r_{17}) = L_{17};\ L(r_{18}) = L_{18}$
Here, L_i ∀ i = {1,2,...18 } denotes the label of the state of robot. In fig. 1, L_i written inside the region means that the state R_i is True in that particular region

Task 3

The desired path is as follows:
$Path = \{(R_6, right, R_7), (R_7, right, R_{18}), (R_{18}, down, R_{17}), (R_{17}, right, R_{20}), (R_{20}, down, R_{21}),$
$inf[(R_{20}, down, R_{21}), (R_{21}, up, R_{20})]^*\}$
This is shown in the fig.(fig:path) as follows:

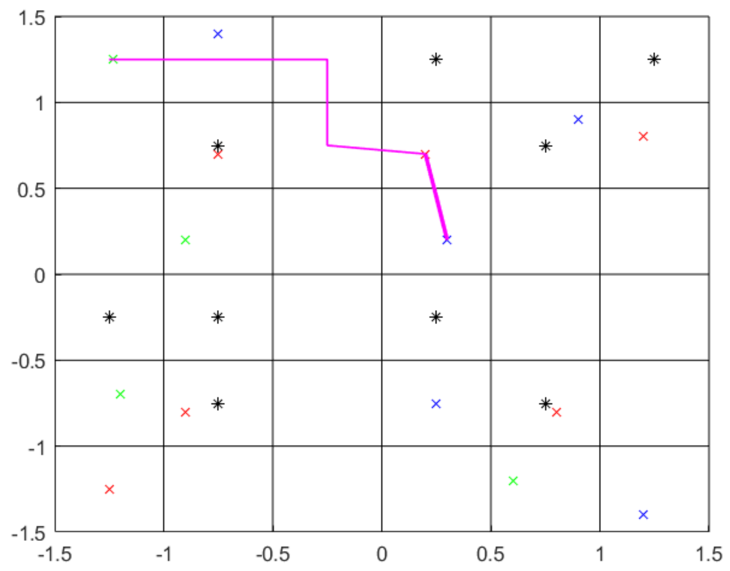


Figure 2: The 2D workspace with the path and obstacles

Task 4

During one transition, the robot must always remain in the two regions of the transition because if we simultaneously control both the relative orientation and distance of the robot to the goal region then instead of being on a straight line the robot will go move from initial points to goal points choosing a random zig-zag path. Also, since we have to move between two points as fast as possible, on simultaneous control it will take more time as compared to the normal situation. And, most importantly it is ***'safe'*** property to not give simultaneous control is because in this kind of control, the robot might collide with some obstacle in its path; causing failure.

Task 5

Since we have to design a proportional controller:

$$\omega[k] = K_{\Psi}(\theta^{R} - \theta[k]) \tag{3}$$

Given:

$$\dot{\theta} = \frac{R\omega}{L} \tag{4}$$

On using Euler Forward Algorithm, we can solve Equation (4), giving us:

$$\theta[k+h] = \dot{\theta}[k]\tau_s + \theta[k] \tag{5}$$

Substituting value of $\dot{\theta}$ from Equation(4) in Equation(6) gives us:

$$\theta[k+h] = \theta[k] + \frac{R\omega\tau_s}{L} \tag{6}$$

where τ_s is sampling time. And, we have already given controller that we have to design. Therefore,

$$\theta[k+h] = \theta[k] + \frac{R\tau_{\mathrm{s}}K_{\Psi}}{L}(\theta^{\mathrm{R}} - \theta[k]) \tag{7}$$

Simplifying Equation(8) will give us the below solution:

$$\theta[k+h] - \theta^{\mathrm{R}} = [1 - \frac{R\tau_{\mathrm{s}}K_{\Psi}}{L}](\theta^{\mathrm{R}} - \theta[k]) \tag{8}$$

Now, we know for the system to be stable, the solution should exist inside the unit circle,

$$\left|(1 - \frac{RK_{\Psi}\tau_s}{L})\right| < 1 \tag{9}$$

$$0 < \frac{RK_{\Psi}\tau_s}{L} < 2 \tag{10}$$

Solving the above condition will give us the value range for K_{Ψ}:

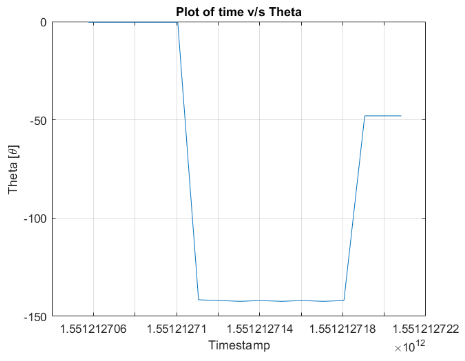
$$\boxed{0 < K_{\Psi} < \frac{2L}{R\tau_s}}$$

Thus, the maximum value of K_{Ψ} to achieve the goal is: $\frac{2L}{R\tau_s}$

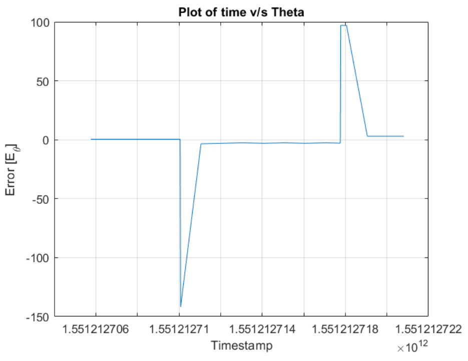
Since we are dealing with proportional controller, we choose K_{Ψ} in order to get faster response i.e. minimum rise time, but also we need to keep in mind that we should get minimum oscillation as well. Therefore, we can choose K_{Ψ} by experimenting different values and hence keep on checking value for least error.

Task 6

On running the online simulator with the above controller, we see that the robot stays the x0 and y0 position and rotates and reaches the reference angle of the goal. When we change the reference goal the robot can be seen rotating. It can be seen that the robot does not move transitionally in any direction. In the simulation of the robot, the robot was in node 2 it starts to rotate from node 3 and then is rotated to node 4 and finally to node 6, the corresponding value of θ_g for node 3 is $0°$, node 4 is $-145°$ and node 6 is $-45°$.



(a) Theta v/s timestamp



(b) Error in theta v/s timestamp

Figure 3: **Plots for evaluation of Rotation controller performance**

The two graphs show the error and the actual theta variation wrt the time stamp. In figure 3.b we can see that the error at node 4 is present and similarly at node 6 also. A small error is noticed in the angle it is supposed to be pointing towards is notice wrt the reference. This error can be removed by including the integral controller in the system.

Task 7

For this task, we are considering a given assumption that $\theta[k+1] = \theta[k]$ i.e. θ does not change with time. We have to design the proportional controller:

$$v[k] = K_\omega d_0[k] \tag{11}$$

The equation of $d_0[k]$ is given by:

$$d_0[k] = cos(\theta[k])(x_0 - x[k]) + sin(\theta[k])(y_0 - y[k]) \tag{12}$$

Solving the Equation(12) by substituting the values of $x[k]$ and $y[k]$, with unit time shift, it will give us the below equation:

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

Now, we know for the system to be stable, the solution should exist inside the unit circle,

$$\begin{aligned} |(1 - \tau_s R K_\omega)| &< 1 \\ -1 < (1 - \tau_s R K_\omega) &< 1 \\ 0 < \tau_s R K_\omega &< 2 \end{aligned}$$

$$\boxed{0 < K_\omega < \frac{2}{R\tau_s}}$$

Thus, the maximum value of K_ω for $d_0[k]$ to converge to 0 is $= \dfrac{2}{R\tau_s}$
Since we are dealing with proportional controller, we choose K_ω in order to get faster response i.e. minimum rise time, but also we need to keep in mind that we should get minimum oscillation as well. Therefore, we can choose K_ω by experimenting different values and hence keep on checking value for least error.

Task 8

When executing the online simulator for this system, we first placed the robot at node 1 and gave the new start location as node 2 and ran the controller, the robot moves to node 2 and similar to node 3. While testing we gave the set point as node 2 and while the robot was moving from node 1, we immediately changed the set point back to node 1 and the robot came back to node 1. Hence the controller was working as expected. The error was not observable from the simulator hence we needed to plot it in MATLAB, but on doing so we were getting a very erratic plot for the same. (See fig 4)

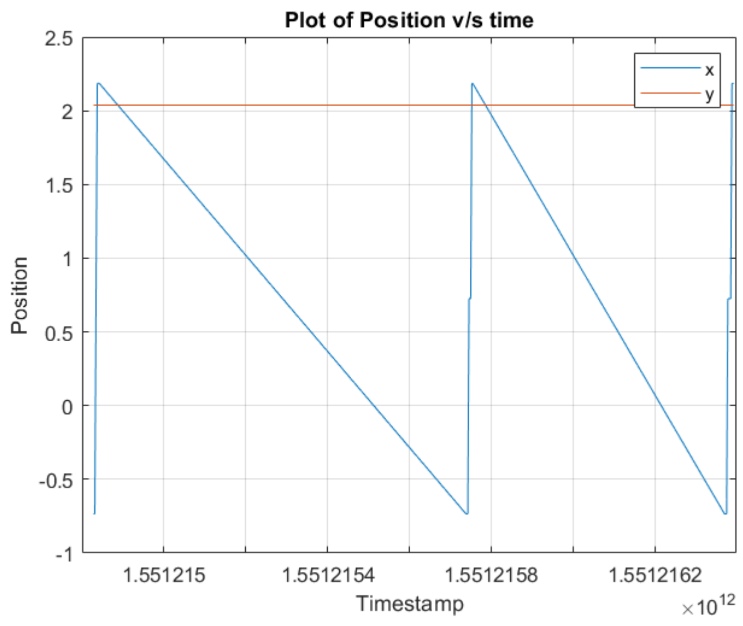
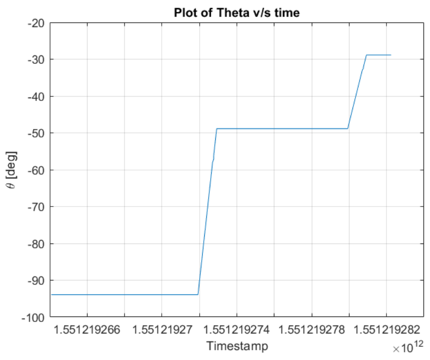


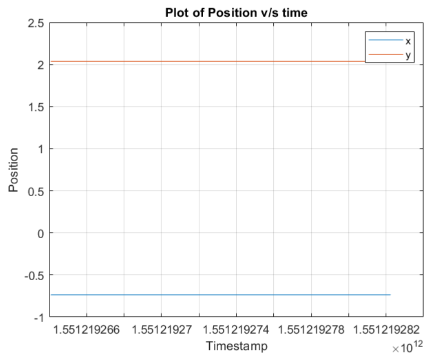
Figure 4: Evaluation of robustness of the controller to hold its position

Task 9

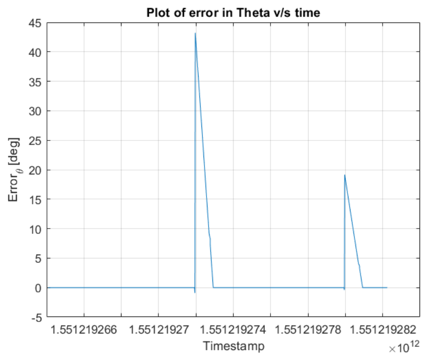
On running both the controller together, we can see that the robot rotates and aligns itself w.r.t the reference line. The second controller reduces the error between the start set point and the robot location. The error plot for the rotational controller and the transnational controller can be seen in the graphs (figure 5). The variation in these curves can be noticed as it becomes a better when both controllers are used together.



(a) Theta v/s timestamp



(b) Position v/s timestamp



(c) Error in Theta v/s timestamp

Figure 5: **Plots for evaluation of controller performance**

Task 10

For this task, we are considering a given assumption that $\theta[k] = \theta_g$ i.e. the robot is headed correctly. We have to design the proportional controller:

$$v[k] = K_\omega d_g[k] \tag{13}$$

The equation of $d_g[k]$ is given by:

$$d_g[k] = cos(\theta[k])(x_g - x[k]) + sin(\theta[k])(y_g - y[k]) \tag{14}$$

Using Euler forward equations that we have solved before i.e.

$$x[k+h] = x[k] + \tau_s Rvcos\theta$$
$$y[k+h] = y[k] + \tau_s Rvsin\theta$$

Solving the Equation(12) by substituting the above values, with unit time shift, it will give us the below equation:

$$d_g[k] = (1 - \tau_s R K_\omega) d_g[k-1]$$
$$d_g[k+1] = (1 - \tau_s R K_\omega) d_g[k]$$

Now, we know for the system to be stable, the solution should exist inside the unit circle,

$$|(1 - \tau_s R K_\omega)| < 1$$
$$-1 < (1 - \tau_s R K_\omega) < 1$$
$$0 < \tau_s R K_\omega < 2$$

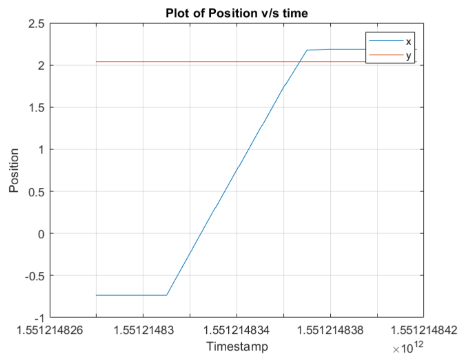
$$\boxed{0 < K_\omega < \frac{2}{R\tau_s}}$$

Thus, the maximum value of K_ω for $d_g[k]$ to converge to 0 is $= \frac{2}{R\tau_s}$

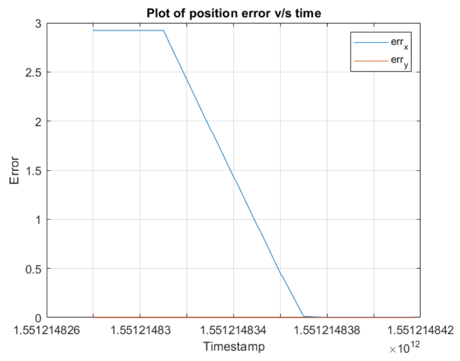
Since we are dealing with proportional controller, we choose K_ω in order to get faster response i.e. minimum rise time, but also we need to keep in mind that we should get minimum oscillation as well. Therefore, we can choose K_ω by experimenting different values and hence keep on checking value for least error.

Task 11

On running the online simulator we can see that the robot starts from node 1 to move towards node 3. It starts slowly and maintains the speed and as seen in the parameter d$_g$, the terms (x$_g$-x) and (y$_g$-y) will tend to become smaller as the robot approaches the goal and simultaneously reduce the speed of the robot. Eventually coming to a full stop at the goal coordinates of node 3. The change in the coordinate position of the robot and the error plot can be seen the fig 6 :



(a) Position v/s timestamp



(b) Error in Position v/s timestamp

Figure 6: **Plots for evaluation of Transnational controller performance**

The error in position can be seen as max when the robot is at node 1 and the goal is at node 3. As it comes closer the error decreases to almost zero but does not become zero.

Task 12

By taking into consideration that $\theta[k]$ is close to θ_g, we can write:

$$d_p[k] \approx p(\theta_g - \theta[k]) \tag{15}$$

Thus, we can also write $d_p[k+1]$ i.e.

$$d_p[k+1] \approx p(\theta_g - \theta[k+1]) \tag{16}$$

We have already solved for $\theta[k+1]$ and controller is given as:

$$\omega[k] = K_\Psi d_p[k] \tag{17}$$

Substituting the values in Equation(16) will give us,

$$d_p[k+1] = p(\theta_g - \theta[k] - \frac{\tau_s R K_\Psi d_p[k]}{L}) \tag{18}$$

Solving the Equation(18) will give us the following equation:

$$d_p[k+1] = [1 - \frac{p\tau_s R K_\Psi}{L}]d_p[k] \tag{19}$$

To be stable, the poles must lie within the unit circle:

$$\left|1 - \frac{p\tau_s R K_\Psi}{L}\right| < 1 \implies -1 < 1 - \frac{p\tau_s R K_\Psi}{L} < 1$$

$$0 < \frac{p\tau_s R K_\Psi}{L} < 2 \implies 0 < K_\Psi < \frac{2L}{pR\tau_s}$$

The maximum value of K_Ψ for $d_p[k]$ to converge to 0 is =

$$\boxed{K_\Psi < \frac{2L}{pR\tau_s}} \tag{20}$$

Task 13

We got the below condition on solving for K_Ψ :

$$K_\Psi < \frac{2L}{pR\tau_s}$$

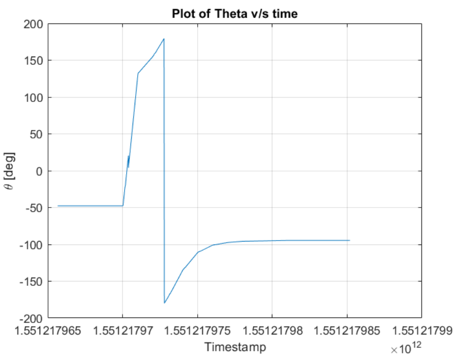
Since we have designed a proportional controller, the value of p decides the robot's ability to follow the line.There are two possible conditions for p :

- When we take larger value of p that means robot will have ample time to look for goal orientation, thus resulting in slower convergence of the system.
- On the other hand, on taking smaller value of p, convergence will be much faster. But we should keep in mind that we are dealing with proportional controller, and if we take much smaller value of p it may cause oscillations in the system.

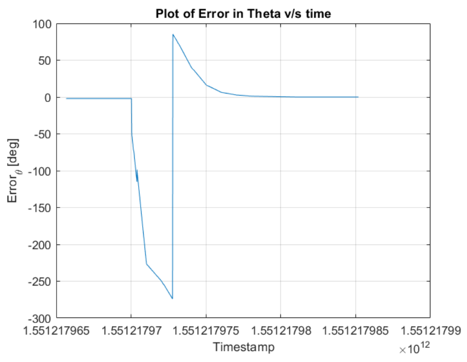
Therefore, in order to consider all the cases in mind, we should tune p for the best performance while working with the controller.

Task 14

On running the simulator we see that the robot rotates continuously at its location and when a reference location is given, the robot aligns itself with respect to the angle of the goal. The first node is the start point and the 5 as the goal. The robot aligns itself to the reference line. Now the start node is changed to node 2. On changing the reference line, we note that the robot starts rotating continuously and hence the variation can be seen in the curve of error. Now when the position of the robot is set to node 2. It stabilizes itself as the reference line is found.



(a) Theta v/s timestamp



(b) Error in theta v/s timestamp

Figure 7: **Plots for evaluation of Rotation controller performance**

Task 15

Task 16

The Hybrid Controller is modelled using Hybrid Automaton of 8 tuple:

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

here:

- $Q \equiv \{q1, q2, q3\}$ which denotes the set of discrete states. The state q1 represents rotational controller, q2 corresponds translational controller, and q3 represents stop condition.
- $Init \equiv \{q3\}$ denotes the initial state, in this case we have taken the stop condition.
- $X \equiv \{(x, y, \theta) \in \mathbb{R}^3 : \theta \in (-180°, 180°]\}$ denotes the continuous state space.
- Vector fields f given by:

$$f(q1, X) = \begin{bmatrix} \dot{x} = Rvcos\theta \\ \dot{y} = Rvsin\theta \\ \dot{\theta} = \dfrac{R\omega}{L} \end{bmatrix}$$

$$f(q2,X) = \begin{bmatrix} \dot{x} = Rvcos\theta \\ \dot{y} = Rvsin\theta \\ \dot{\theta} = \frac{R\omega}{L} \end{bmatrix}$$

$$f(q3,X) = \begin{bmatrix} \dot{x} = 0 \\ \dot{y} = 0 \\ \dot{\theta} = 0 \end{bmatrix}$$

- D shows conditions for automaton to stay in a state.
  $D(q1) = \{(x,y,\theta) : x,y \in \mathbb{R}^2, |\theta_{\mathrm{g}} - \theta| \ > \delta\}$
  $D(q2) = \{(x,y,\theta) : x,y \in \mathbb{R}^2, \sqrt{(x_g - x)^2 + (y_g - y)^2} \ > \xi\}$
  D$(q3)$ = {x,y $\in \mathbb{R}^2, \sqrt{(x_g - x)^2 + (y_g - y)^2} \sim 0$ }
- E: Edges show possible transitions.
  $E1 = \{q1, q2\}$
  $E2 = \{q2, q3\}$
  $E3 = \{q3, q1\}$
  $E = E1 \cup E2 \cup E3$
- G : G is the guard conditions.
  $G(\{q1,q2\}) = \{|\theta_g - \theta| \ \leq \delta\}$
  $G(\{q2,q3\}) = \{\sqrt{(x_g - x)^2 + (y_g - y)^2} \ \leq \xi\}$
  G({q3,q1}) = $\{\sqrt{(x_g - x)^2 + (y_g - y)^2} \ > \xi\}$

- R: denotes the reset map.
  $R = \{x, y, \theta\}$

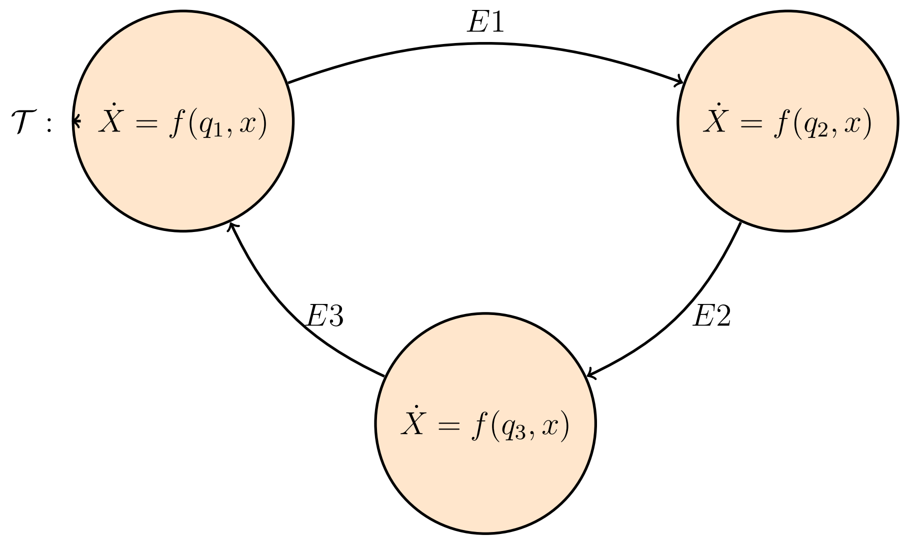


Figure 8: Product Interleaving Transition Systems $\mathcal{T}_{\mathrm{int}}$.

Task 17

Solution to the task

Task 18

Solution to the task

Task 19

Solution to the task

References

[1] Hassan K Khalil. *Nonlinear systems.* Prentice Hall, Upper Saddle river, 3. edition, 2002. ISBN 0-13-067389-7.

[2] Tobias Oetiker, Hubert Partl, Irene Hyna, and Elisabeth Schlegl. *The Not So Short Introduction to LaTeX 2ε.* Oetiker, OETIKER+PARTNER AG, Aarweg 15, 4600 Olten, Switzerland, 2008. http://www.ctan.org/info/lshort/.

[3] Shankar Sastry. *Nonlinear systems: analysis, stability, and control*, volume 10. Springer, New York, N.Y., 1999. ISBN 0-387-98513-1.