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

Gengwu Du	Jiayi Feng	Yanfeng Yu	Ziqiao Wang
19980213-1459	19980818-0393	19980204-9339	19980630-9291
gengwu@kth.se	jiayif@kth.se	yfyu@kth.se	ziqiao@kth.se

March 5, 2021

Task 1

The velocities of the robot are modelled as the following:

$$\begin{aligned} v &= \frac{u_r + u_l}{2} \\ \omega &= u_r - u_l \end{aligned} \tag{1}$$

where v is the translation velocity, ω is the rotational velocity, and u_r, u_l are the angular velocities (1°/s) of the respective right and left wheel. Given the velocities v and ω, the above equation system 1 can be rewritten as:

$$\begin{aligned} u_r &= \frac{2v + \omega}{2} \\ u_l &= \frac{2v - \omega}{2} \end{aligned} \tag{2}$$

Task 2

We are given that the robot operates in a workspace with dimensions $3 \times 3m^2$. The properties of interest, red, green, blue and obstacle are plotted on the workspace as in figure 1 with obstacle being black dots.

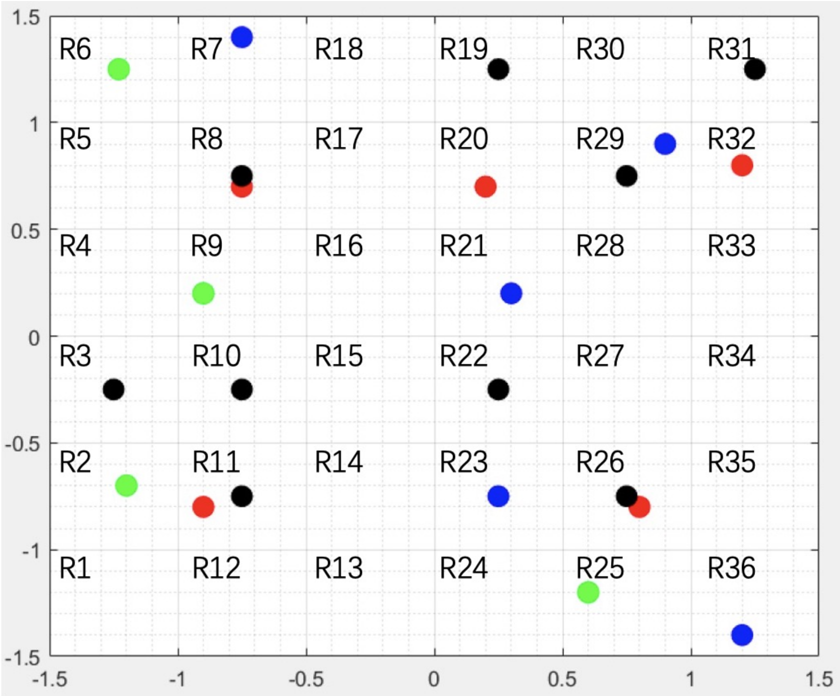

Figure 1: Discretized workspace where the robot operates in

The workspace is discretized into 36 regions with each region having an area of $(0.5 \times 0.5)m^2$, which is able to contain a robot with the largest area being $0.38 \times 0.38 = 0.1444m^2$.

The proposed transition system will be defined as follows:

$$\mathcal{T} = (S, S_0, \Sigma, \longrightarrow, AP, \mathcal{L}) \tag{3}$$

where:

- The set of states $S = \{R_1, R_2, \ldots, R_{36}\}$
- The set of initial state $S_0 = \{R_6\}$, where the starting point (-1.25,1.25) is in R_6
- The set of actions $\Sigma = \{up, down, left, right\}$
- Some transition relation examples
  - $(R_{27}, up, R_{28}) \in\rightarrow$
  - $(R_{27}, down, R_{26}) \in\rightarrow$
  - $(R_{27}, left, R_{22}) \in\rightarrow$
  - $(R_{27}, right, R_{34}) \in\rightarrow$

  However, some cases are not available for map boundary condition, where

  $-$action *up* is not available in the state$\{R_6, R_7, R_{18}, R_{19}, R_{30}, R_{31}\}$

  $-$action *down* is not available in the state$\{R_1, R_{12}, R_{13}, R_{24}, R_{25}, R_{26}\}$

  $-$action *left* is not available in the state$\{R_1, R_2, R_3, R_4, R_5, R_6\}$

  $-$action *right* is not available in the state$\{R_{31}, R_{32}, R_{33}, R_{34}, R_{35}, R_{36}\}$
- The set of atomic proposition $AP = \{r, g, b, obs\}$, where *obs* stands for obstacle, and r, g, b stands for red, green, yellow, respectively.

- The labeling functions are shown in the table below:

$\mathcal{L}(R_6) = \{g\}$	$\mathcal{L}(R_7) = \{b\}$	$\mathcal{L}(R_{18}) = \{\emptyset\}$	$\mathcal{L}(R_{19}) = \{obs\}$	$\mathcal{L}(R_{30}) = \{\emptyset\}$	$\mathcal{L}(R_{31}) = \{obs\}$
$\mathcal{L}(R_5) = \{\emptyset\}$	$\mathcal{L}(R_8) = \{obs, r\}$	$\mathcal{L}(R_{17}) = \{\emptyset\}$	$\mathcal{L}(R_{20}) = \{r\}$	$\mathcal{L}(R_{29}) = \{obs, b\}$	$\mathcal{L}(R_{32}) = \{r\}$
$\mathcal{L}(R_4) = \{\emptyset\}$	$\mathcal{L}(R_9) = \{g\}$	$\mathcal{L}(R_{16}) = \{\emptyset\}$	$\mathcal{L}(R_{21}) = \{b\}$	$\mathcal{L}(R_{28}) = \{\emptyset\}$	$\mathcal{L}(R_{33}) = \{\emptyset\}$
$\mathcal{L}(R_3) = \{obs\}$	$\mathcal{L}(R_{10}) = \{obs\}$	$\mathcal{L}(R_{15}) = \{\emptyset\}$	$\mathcal{L}(R_{22}) = \{obs\}$	$\mathcal{L}(R_{27}) = \{\emptyset\}$	$\mathcal{L}(R_{34}) = \{\emptyset\}$
$\mathcal{L}(R_2) = \{g\}$	$\mathcal{L}(R_{11}) = \{obs, r\}$	$\mathcal{L}(R_{14}) = \{\emptyset\}$	$\mathcal{L}(R_{23}) = \{b\}$	$\mathcal{L}(R_{26}) = \{obs, r\}$	$\mathcal{L}(R_{35}) = \{\emptyset\}$
$\mathcal{L}(R_1) = \{\emptyset\}$	$\mathcal{L}(R_{12}) = \{\emptyset\}$	$\mathcal{L}(R_{13}) = \{\emptyset\}$	$\mathcal{L}(R_{24}) = \{\emptyset\}$	$\mathcal{L}(R_{25}) = \{g\}$	$\mathcal{L}(R_{36}) = \{b\}$

Table 1: Labeling functions

Task 3

In order to meet the requirements, we can follow the sequence 4 to plan the path. The robot will eventually moves repeatedly between the regions R_{20} and R_{21}:

$$R_6, R_7, R_{18}, R_{17}, (R_{20}, R_{21})^* \tag{4}$$

The desired path is shown in the figure 2:

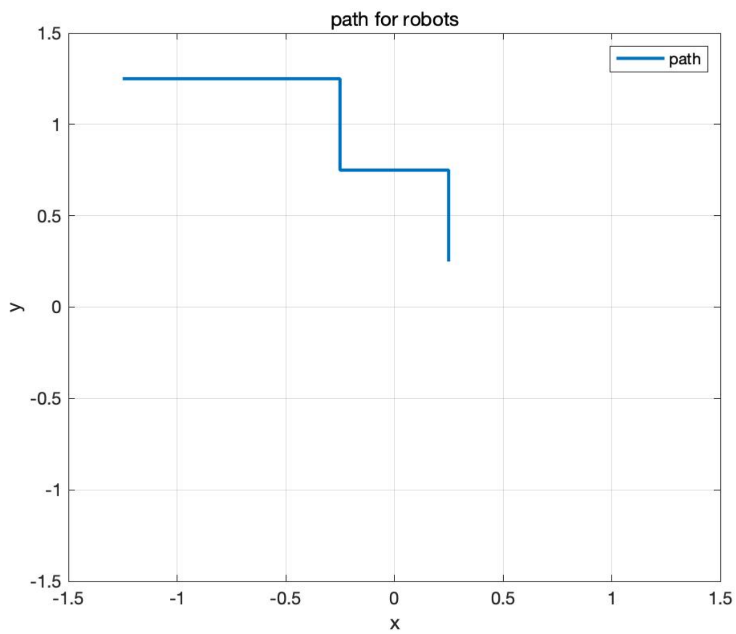

Figure 2: The desired path for the robot

Task 4

It wouldn't be better to simultaneously control them both since in this case it wouldn't guarantee a straight line for the robot to move on. Instead, by separating the 'rotation' and 'go to goal' control, the robot's route will always be straight. In this case, we can effectively prevent the robot from entering unplanned areas that could possibly contain i.e. obstacles.

Task 5

According to the robot dynamics we know that:

$$\dot{\theta} = \frac{R}{L}\omega \tag{5}$$

Then, discretize the equation 5 by forward Euler method gives:

$$\theta[k+1] = \theta[k] + \frac{Rh}{L}\omega[k] \tag{6}$$

where h is the sampling time.

Insert $\omega[k] = K_\Psi(\theta^R - \theta[k])$ into the equation 6 we get:

$$\theta[k+1] = \theta[k] + \frac{Rh}{L}K_\Psi(\theta^R - \theta[k]) \tag{7}$$

In order to make $\theta[k]$ approaches θ^R asymptotically, we can set a new state as $e[k] = \theta[k] - \theta^R$ and rewrite equation 7 by also subtracting θ^R on both sides:

$$\begin{aligned} \theta[k+1] - \theta^R &= (1 - \frac{Rh}{L}K_\Psi)(\theta[k] - \theta^R) \\ e[k+1] &= (1 - \frac{Rh}{L}K_\Psi)e[k] \end{aligned} \tag{8}$$

Only when $|1 - \frac{Rh}{L}K_\Psi| < 1$, e can approach to 0 asymptotically, then $\theta[k]$ can follow the θ^R, thus we have:

$$\begin{aligned} 1 - \frac{Rh}{L}K_\Psi &< 1 \\ 1 - \frac{Rh}{L}K_\Psi &> -1 \end{aligned} \tag{9}$$

Finally, we get:

$$0 < K_\Psi < 2\frac{L}{Rh} \tag{10}$$

Task 6

By implementing the controller we designed in task 5 into the simulation, we get the result shown in the figure 3. In this figure we have the desired angle $\theta^R = 90^\circ$ and the initial angel was set as 0° with $K_\Psi = 202$.

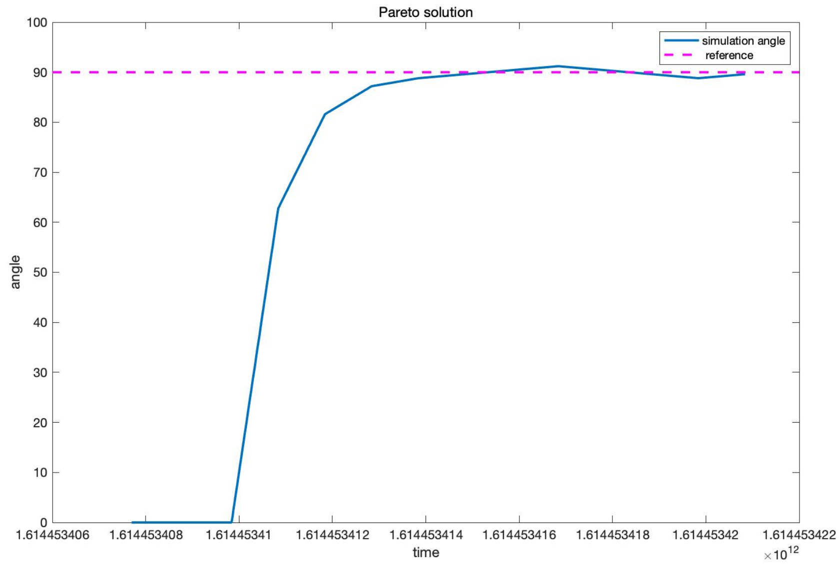

Figure 3: Performance of angular control

From the result we can see that the angle almost follows the reference, and we know that the ω controller is actually a proportion controller, where the performance is really dependent on the controller's coefficient. There may be a static state error because we only use a p controller in this case.

Task 7

From the instruction, we know that:

$$d_0 = \begin{bmatrix} cos\theta, sin\theta \end{bmatrix} \begin{bmatrix} x_0 - x \\ y_0 - y \end{bmatrix} = (x_0 - x)cos\theta + (y_0 - y)sin\theta \tag{11}$$

By assuming θ is a constant value at this time and given the values of $\dot{x}$ and $\dot{y}$ by robot's dynamics, we will get:

$$\begin{aligned} \dot{d}_0 &= -\dot{x}\, cos\theta - \dot{y}\, sin\theta \\ &= -Rv\, cos\theta^2 - Rv\, sin\theta^2 \\ &= -Rv \end{aligned} \tag{12}$$

Discretize the equation 12 by Euler forward method, where h is the sampling time we get:

$$d_0[k+1] = d_0[k] - hRv[k] \tag{13}$$

Insert the v controller to the equation 13:

$$d_0[k+1] = d_0[k] - hRK_\omega d_0[k] = (1 - hRK_\omega)d_0[k] \tag{14}$$

In order to make d_0 approaches to 0 asymptotically, we need fulfill the condition $|1 - hRK_\omega| < 1$, then we can get:

$$0 < K_\omega < \frac{2}{Rh} \tag{15}$$

Task 8

For this task we set the initial angle as $\theta_0 = 0$, initial point at $(1, 0)$, $K_\omega = 0.1$, and we add a disturbance along the x-axis with a value of -0.7 in order to verify if the robot can return to the initial point. The result is shown in the figure 4.

However, it is not always possible to keep the robot at the initial point if we set the $\omega = 0$. Because if we set $\omega = 0$, that means the robot can only move along the initial angle. If the disturbance is not along this direction, the robot can not move back to the initial point.

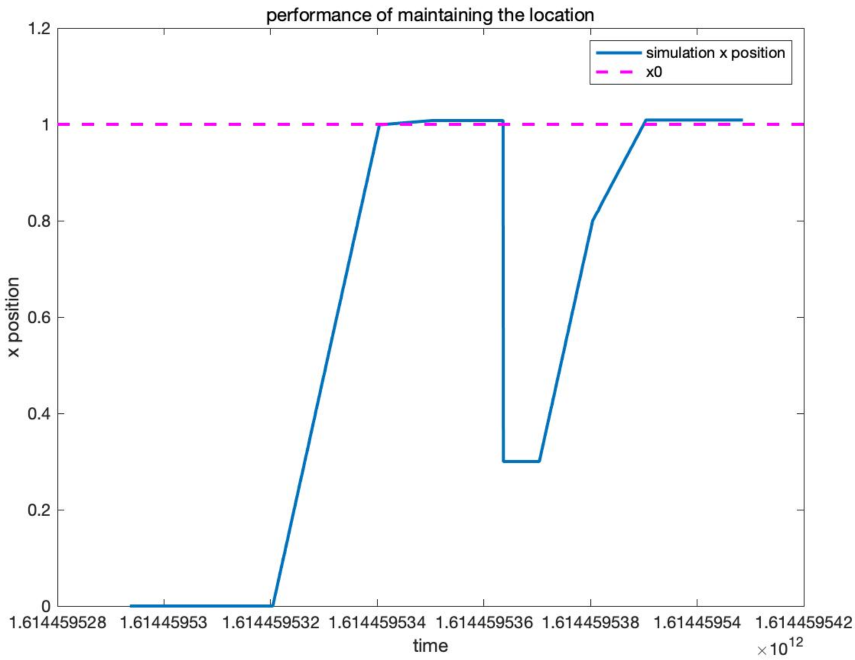

Figure 4: Performance of maintaining the location

Task 9

The angle error and d_0 for both controller are enabled and only one controller is enabled are shown in the figures 5, 6, 7 and 8, separately.

From the figure we can see that the performance of error angle when both controllers are enabled and when only ω controller is enabled are almost the same. That is because the v controller doesn't have much impact on how the angle error changes. However, the performance of d_0 when only v controller is enabled is better than when both controllers are enabled. That is because when both controllers are working, the oscillation of the angle caused by the ω controller as seen in task 6 also causes d_0 to oscillate.

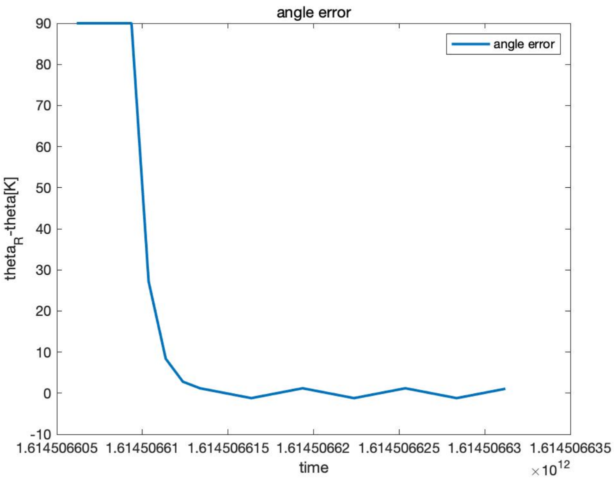

Figure 5: Angle error when both controller worked

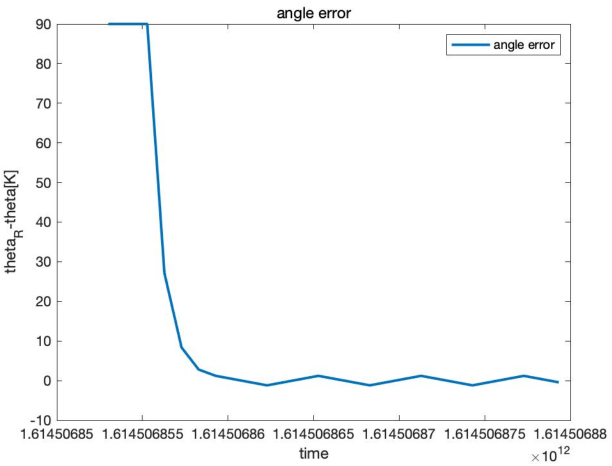

Figure 6: Angle error when ω controller worked

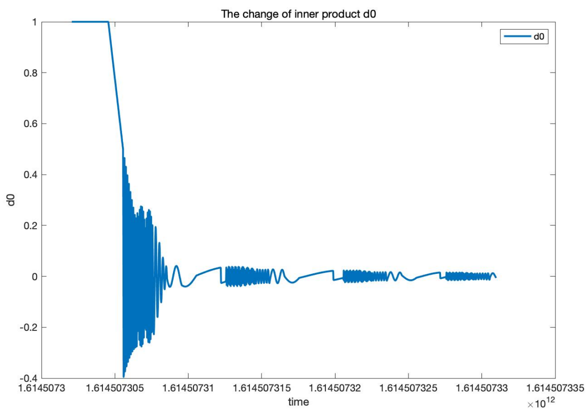

Figure 7: d_0 when both controller worked

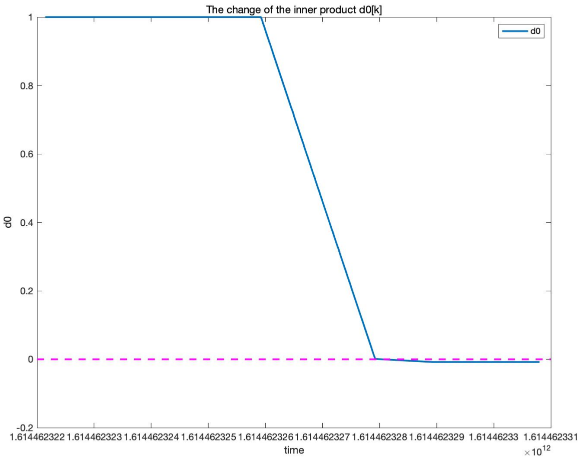

Figure 8: d_0 when v controller worked

Task 10

From the instruction, we know that:

$$d_g = \begin{bmatrix} cos\theta_g, sin\theta_g \end{bmatrix} \begin{bmatrix} x_g - x \\ y_g - y \end{bmatrix} = (x_g - x)cos\theta + (y_g - y)sin\theta \tag{16}$$

By assuming θ is a constant value at this time and given the values of $\dot{x}$ and $\dot{y}$ by robot's dynamics, we will get:

$$\begin{aligned} \dot{d_g} &= -\dot{x}cos\theta_g - \dot{y}sin\theta_g \\ &= -Rv\,cos\theta_g^2 - Rv\,sin\theta_g^2 \\ &= -Rv \end{aligned} \tag{17}$$

Discretize the equation 17 by Euler forward method, where h is the sampling time we get:

$$d_g[k+1] = d_g[k] - hRv[k] \tag{18}$$

Insert the v controller to the equation 18:

$$d_g[k+1] = d_g[k] - hRK_\omega d_g[k] = (1 - hRK_\omega)d_g[k] \tag{19}$$

In order to make d_g approaches to 0 asymptotically, we need fulfill the condition $|1 - hRK_\omega| < 1$, then we can get:

$$0 < K_\omega < \frac{2}{Rh} \tag{20}$$

Task 11

It is possible for the robot to arrive at (x_g, y_g) exactly. we set start point at $(0, 0)$, goal point at $(1, 1)$ and $K_\omega = 2.5$. The result of the simulation is shown in the figure 9. In the figure, the position of x approaches $x_g = 1$ at the end. Same behaviour goes for the position y. The reason behind it is that when $\omega = 0$, there won't be much disturbance impact on the v controller to set the position.

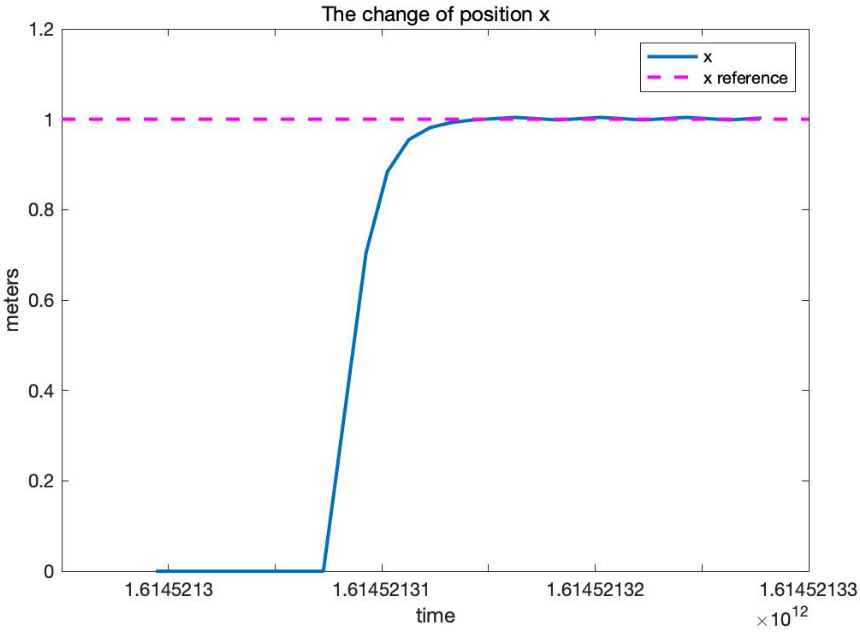

Figure 9: The performance of position x

Task 12

From the instruction, we know that:

$$d_p = \begin{bmatrix} sin\theta_g, -cos\theta_g \end{bmatrix} \begin{bmatrix} x + p\cos\theta - x_0 \\ y + p\sin\theta - y_0 \end{bmatrix} = (x - x_0)sin\theta_g - (y - y_0)cos\theta_g + p\sin(\theta_g - \theta) \quad (21)$$

And we all know that

$$\begin{cases} x - x_0 = \sqrt{(x - x_0)^2 + (y - y_0)^2} * \cos\theta_g, \\ y - y_0 = \sqrt{(x - x_0)^2 + (y - y_0)^2} * \sin\theta_g \end{cases} \quad (22)$$

By assuming $\theta[k]$ is close to θ_g, thus we can get :

$$d_p = p\sin(\theta_g - \theta) \approx p(\theta_g - \theta) \quad (23)$$

After differentiation, we get that

$$\dot{d}_g = -p\dot{\theta} = \frac{pR}{L}\omega \quad (24)$$

Discrete the equation 24 by Euler forward method, where h is the sampling time we get:

$$d_p[k+1] = d_p[k] - \frac{hpR}{L}\omega[k] \quad (25)$$

According to the instruction, we know that:

$$\omega[k] = K_\Psi d_p[k] \quad (26)$$

Therefore,

$$d_p[k+1] = d_p[k] - \frac{hpR}{L}K_\Psi d_p[k] = (1 - \frac{hpR}{L}K_\Psi)d_p[k] \quad (27)$$

In order to make d_p approaches 0 asymptotically, we need to fulfill the condition $|1 - \frac{hpR}{L}K_\Psi| < 1$, then we can get:

$$0 < K_\Phi < \frac{2L}{hpR} \tag{28}$$

Task 13

From the instruction, we are given with:

$$\begin{aligned} w[k] &= K_\Psi d_p[k] \\ &= K_\Psi p(\theta_g - \theta[k]) \end{aligned} \tag{29}$$

We can see that if we keep K_Ψ as a constant and change the p value, then p can be seen as the gain of a proportional controller.

However, In task 12 we get that $0 < K_\Psi < \frac{2L}{pRh}$. In order to keep K_Ψ always in a stable range, we let $K_\Psi \times p$ to be a constant that is smaller than $\frac{2L}{Rh}$. Thus, when we change the value of p that affect the range, the value of $K_\Psi p$ does not change so there is virtually no effect on the system. The figure 10, 11 and 12 shows the results for the different values of p. We can easily see that they are basically the same.

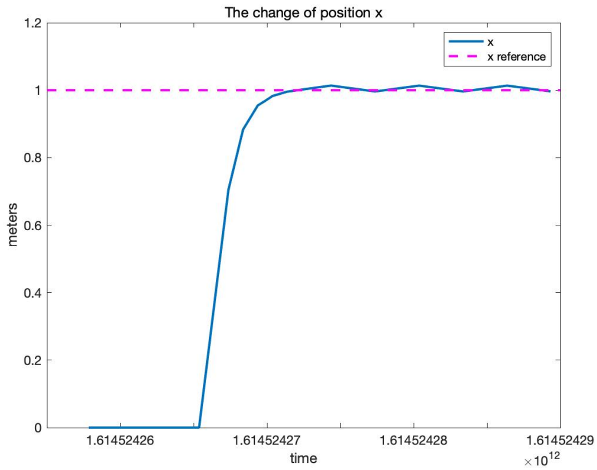

Figure 10: The performance of x when $p = 1$

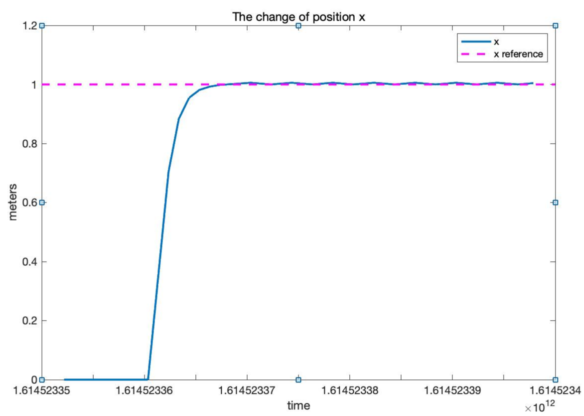

Figure 11: The performance of x when $p = 50$

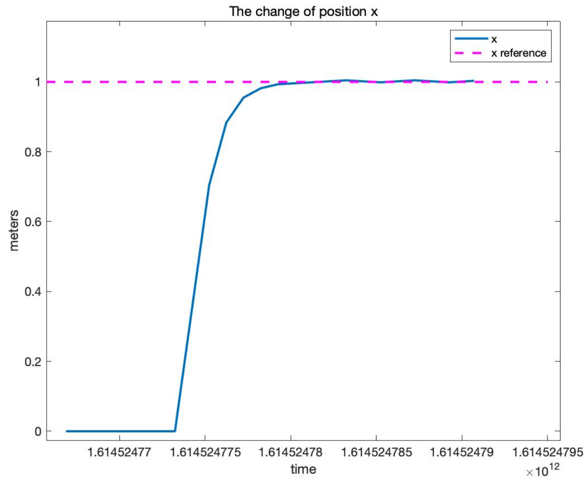

Figure 12: The performance of x when $p = 10000$

Task 14

It is possible to maintain $d_p[k]$ at 0 exactly when we set v = 0 and $K_\Psi = 0.1$. According to the figure 13, we set the initial angle $\theta_0 = 0$ and goal angle $\theta_g = \frac{\pi}{2}$. And we can find that the $d_p = 0$ at the end. This means that the control design is well made, which allows d_p to converge to 0 asymptotically.

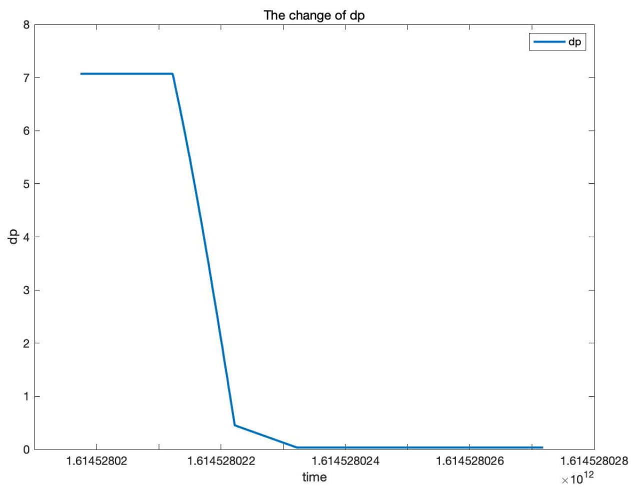

Figure 13: The change of dp

Task 15

When both controllers are executed, the simulation results of d_g and d_p are shown in the figure 14 and 15. In the results we can see that d_p and d_g approaches to the 0 at the end. Here we used the values $K_\Psi = 0.1$, $p = 22$ and $K_\omega = 2.5$. We can compare it to the previous cases in figure 9 and 13 with only one controller and find that they are almost identical.

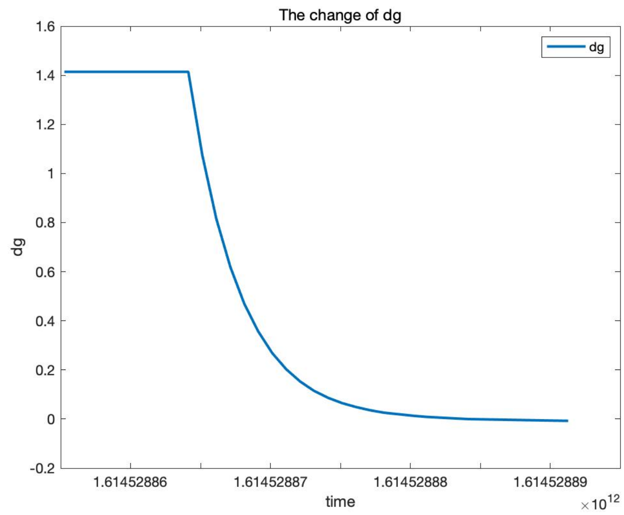

Figure 14: The change of d_g

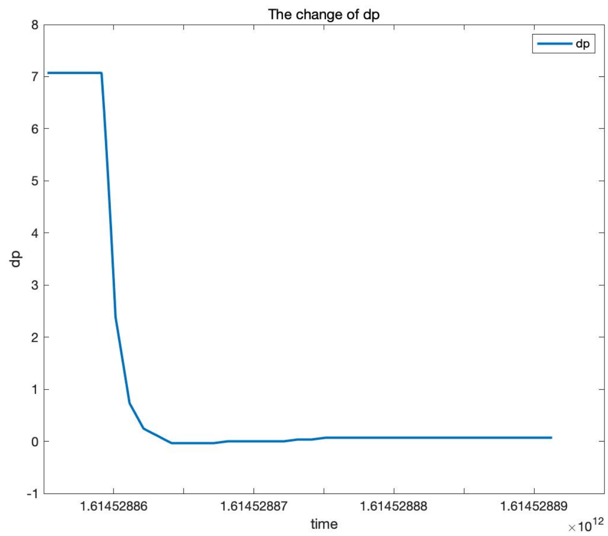

Figure 15: The change of d_p

Task 16

The controlled robot can be modeled formally as following hybrid automaton:

$$H = (Q, X, Init, f, D, E, G, R) \tag{30}$$

- $Q = \{q_1, q_2, q_3\} = \{rotation, translation, goal\}$, where Q are the discrete states of the controller. In state q_1, rotational control is executed, in state q_2 the robot is driving to the goal and lastly in state q_3, the robot stops at the goal.
- $X = \{(x, y) \in \mathrm{R}^2, \theta \in (-\pi, \pi]\}$, where X are the continuous states of the controller.
- $Init = \{rotation, x_0, y_0, \theta_0\}$, where Init is the initial state of the controller.

- f are the vector fields.

  $f(q_1,(x,y,\theta)) = [Rvcos\theta, Rvsin\theta, \frac{R}{L}\omega]$
  where according to Task 5 and Task 6 we can get
  $\omega = K_\Psi(\theta_g - \theta)$, $v = K_\omega d_0 = K_\omega[cos\theta(x_0 - x) + sin\theta(y_0 - y)]$

  $f(q_2,(x,y,\theta)) = [Rvcos\theta, Rvsin\theta, \frac{R}{L}\omega]$
  where according to Task 10 and Task 12 we can get:
  $\omega = K_\Psi d_p = K_\Psi[sin\theta_g(x + pcos\theta - x_0) - cos\theta_g(y + pcos\theta - y_0)]$
  $v = K_\omega d_g = K_\omega[cos\theta_g(x_g - x) + sin\theta_g(y_g - y)]$

  $f(q_3,(x,y,\theta)) = [Rvcos\theta, Rvsin\theta, \frac{R}{L}\omega] = [0,0,0]$

- D is the domain of the discrete states.
  $D(q_1) = \{(x,y) \in \mathrm{R}^2, \theta \in (-\pi,\pi] : |\theta \neq \theta_g\}$
  $D(q_2) = \{(x,y) \in \mathrm{R}^2, \theta \in (-\pi,\pi] : |x \neq x_g, y \neq y_g\}$
  $D(q_3) = \{(x,y) \in \mathrm{R}^2, \theta \in (-\pi,\pi] : |x = x_g, y = y_g\}$

- E are the edges. $E = \{(q_1,q_2),(q_2,q_3),(q_3,q_1)\}$

- G are the conditions that allow to change from a state to next state:
  $G(q_1,q_2) = \{(x,y) \in \mathrm{R}^2, \theta \in (-\pi,\pi] : \theta = \theta_g\}$
  $G(q_2,q_3) = \{(x,y) \in \mathrm{R}^2, \theta \in (-\pi,\pi] : x = x_g, y = y_g\}$
  $G(q_3,q_1) = \{(x,y) \in \mathrm{R}^2, \theta \in (-\pi,\pi] : x \neq x_g, y \neq y_g\}$

- R means reset.
  $R(q_1,q_2,(x,y,\theta)) = R(q_2,q_3,(x,y,\theta)) = R(q_3,q_1,(x,y,\theta))[x,y,\theta]$

Task 17

When the hybrid controller has been implemented, the results of the simulation are shown below. In the simulation of the initial position (x_0, y_0, θ), it is set to be $(0,0,0)$ while the goal (x_g, y_g) is set to be $(1,1)$. And then figure 16, 17, 18, 19 shows the change of robot including position x, y and angle θ. According to these figures, we can find that the discrete state evolves as expected. Firstly, the rotation controller is activated. When the desired angle is reached, it will enter a new discrete state q_2 and the translation controller will activate. Then the robot starts translating through the goal position. Finally, when the goal position is reached, it will enter a new discrete state q_3, where the goal state is activated. Hence, the robot maintains in its position and does not move.

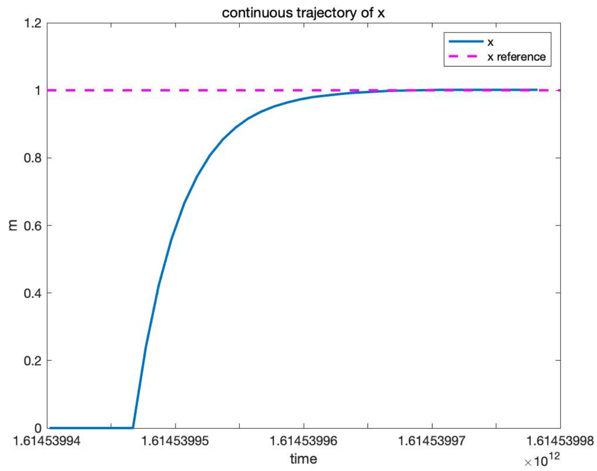

Figure 16: continuous trajectory of x

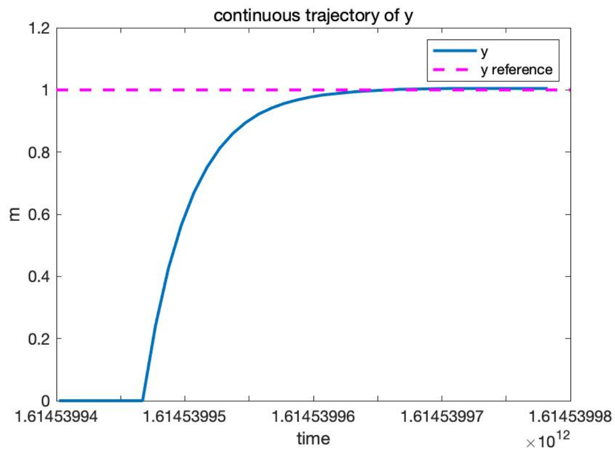

Figure 17: continuous trajectory of y

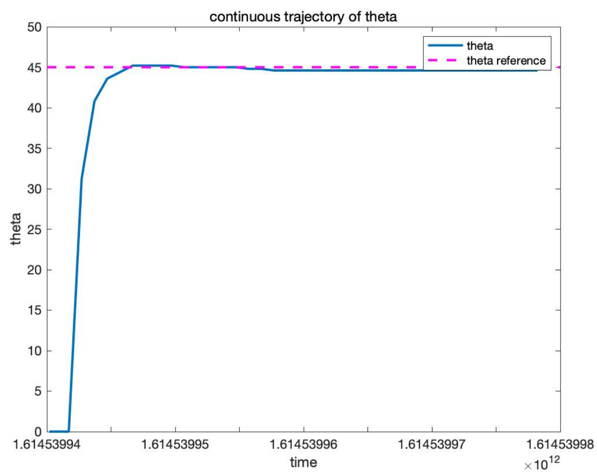

Figure 18: continuous trajectory of theta

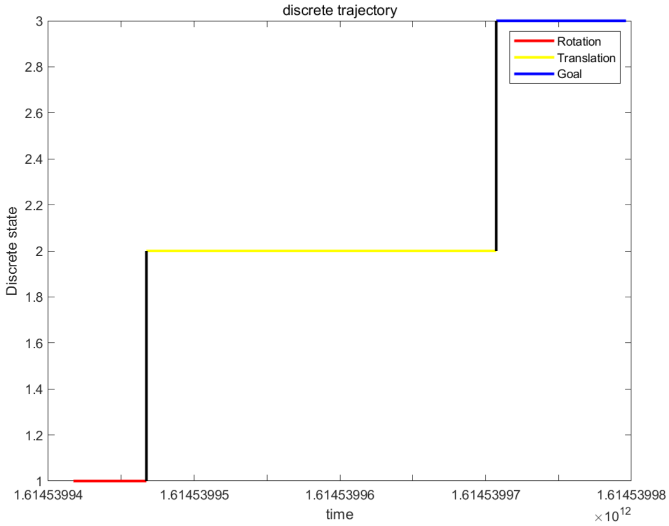

Figure 19: discrete trajectory

Task 18

According to the plan that have been proposed in Task 3, the way-points becomes:

$$(-1.25, 1.25) \rightarrow (-0.25, 1.25) \rightarrow (-0.25, 0.75) \rightarrow (0.25, 0.75) \rightarrow (0.25, 0.25)$$

As is shown in figure 20, the robot can pass each point with high accuracy using effective controllers. Only small errors appeared.

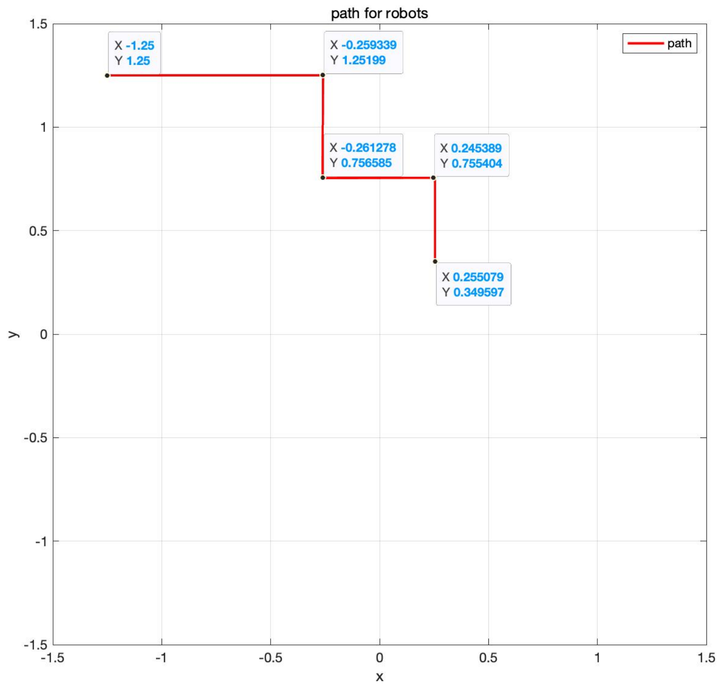

Figure 20: Simulation result for the robot tracking a specified route

Task 19

Safety, in the sense of hybrid controller means that the robot stays within the two regions during transitioning, rather than going into any undetermined states. As a result, the rotational controller calibrates the robot's orientation before the straight line movement. In the following process, the line following controller prevents the robot from deviating the aforementioned straight route. When it reaches the destination, the repeated process will be executed. In addition, the size of the robot is about $38cm$, which is smaller than each region which are $50cm \times 50cm$. The repeated controller performance and small robot area results in higher successful rate for the transition since it makes it harder to reach more than two regions at the same time.