AUT.360 Distributed Control and Optimization of Cyber-Physical Systems

Homework 9

Valtteri Nikkanen 282688

Riku Pekkarinen 267084

Table of contents

We have an omnidirectional robot with the following dynamics:

$$\dot{x} = u \tag{1}$$

Where $x = [p_x \quad p_y]^T \in \mathbb{R}^2$ which represents the position of the robot and $u = [u_x \quad u_y]^T \in \mathbb{R}^2$ is the control input that is to be designed. The robots initial location is at (0,0) and its goal position is at (3.0, 3.0). There is also a circular obstacle at (1.5, 1.8) so $x_o = [1.5 \quad 1.8]^T$ with the radius of 0.7.

The quadratic programming-based controller with control barrier function is as follows

$$u = \arg\min_u \left\| u_{gtg} - u \right\|^2 \tag{2}$$

$$s.t. \ \left(\frac{\partial h}{\partial x}\right)^T u \geq -\gamma(h(x))$$

Where $u_{gtg} = (x_g - x)$ and $h(x) = \|x - x_0\|^2 - 0.8^2$

Quadratic programming has a general from that we need to solve to insert into MATLAB.

$$\min_u \frac{1}{2} u^T Q u + c^T u \tag{3}$$

$$s.t. \ Hu \leq b$$

We now need to read the missing parameters Q, c, H and b in equation 3 from the equation 2.

$$\arg\min_u \left\| u_{gtg} - u \right\|^2$$

$$\arg\min_u \left(u_{gtg} - u\right)^T \left(u_{gtg} - u\right)$$

$$\arg\min_u \ {u_{gtg}}^T u_{gtg} - {u_{gtg}}^T u - u^T u_{gtg} + u^T u$$

$$\arg\min_u \ {u_{gtg}}^T u_{gtg} - {u_{gtg}}^T u - {u_{gtg}}^T u + u^T I u$$

$$\arg\min_u \ {u_{gtg}}^T u_{gtg} - 2{u_{gtg}}^T u + \frac{1}{2} u^T 2 I u$$

From this representation we can read that

$$c = -2u_{gtg} \text{ and } Q = 2I$$

We know that $u_{gtg} = (x_g - x)$ which means that c can be represented as

$$c = \begin{bmatrix} -2(p_{g,x} - p_x) \\ -2(p_{g,y} - p_y) \end{bmatrix}$$

And Q is a 2x2 matrix

$$Q = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

Then H and b can be read from the conditions almost straight away

$$-\left(\frac{\partial h}{\partial x}\right)^T u \leq \gamma(h(x))$$

Where $b = \gamma(h(x))$ and $H = -\left(\frac{\partial h}{\partial x}\right)^T$. We know that we will get the γ(h(x)) later so we don't need to worry about it. H on the other hand can be represented in a nicer manner as we know what h(x) is and can differentiate it.

$$H = -\left(\frac{\partial(\|x - x_0\|^2 - 0.8^2)}{\partial x}\right)^T$$

$$H = -\begin{bmatrix} -2(p_x - p_{o,x}) \\ -2(p_y - p_{o,y}) \end{bmatrix}^T$$

$$H = [-2(p_x - p_{o,x}) \quad -2(p_y - p_{o,y})]$$

We now have all the unknow variables solved for the general form of the general form of quadratic programming presented in equation 3.

Problem 1

1. We are given that our γ(h(x)) = h(x) which means that

$$b = \gamma(h(x)) = \|x - x_0\|^2 - 0.8^2$$

Designing the simulation based on the previously calculated parameters and the new one we get the results presented in figure 1. The robot seems to go fairly wide around the obstacle and stays firmly in the safe zone. In figure 2-4 there is presented how the nominal go-to-goal and the safety filter input signals differ, how the h function evolves as the robot moves and the norm difference between the nominal and the safety filter input signals.

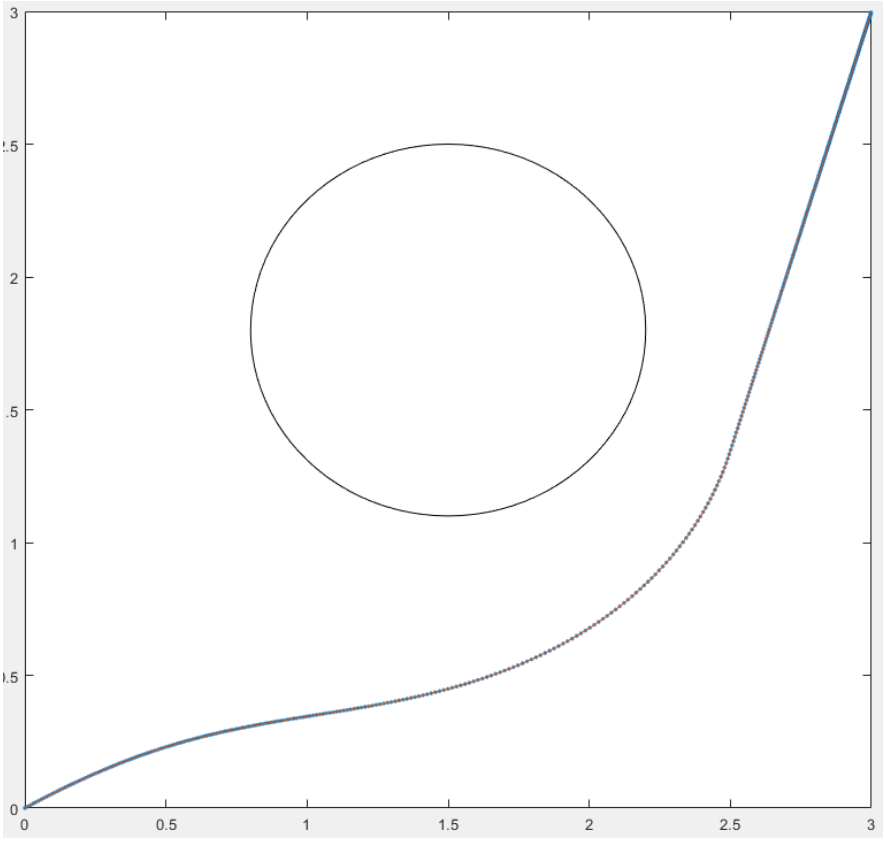

Figure 1: The robot path with γ(h(x)) = h(x)

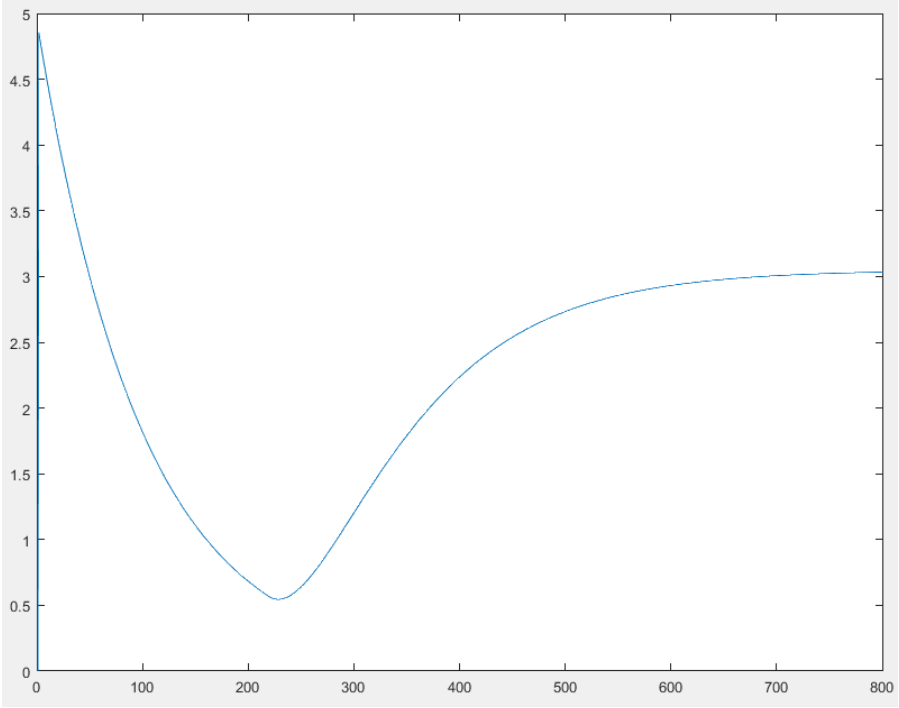

Figure 2: Evolution of the h as the robot moves in figure 1

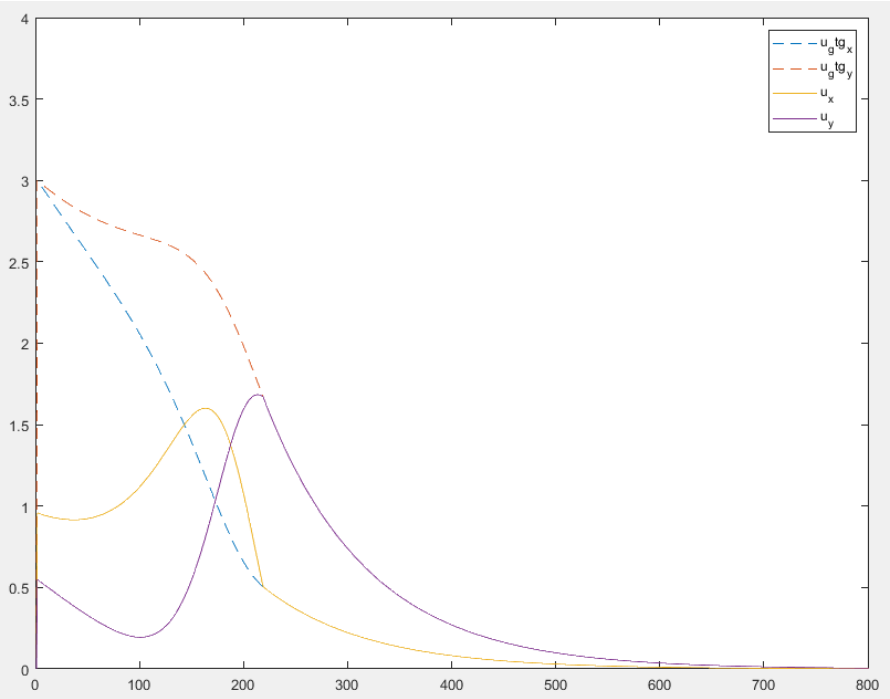

Figure 3: The nominal control input signal vs the input signal after the safety filter as the robot moves in figure 1

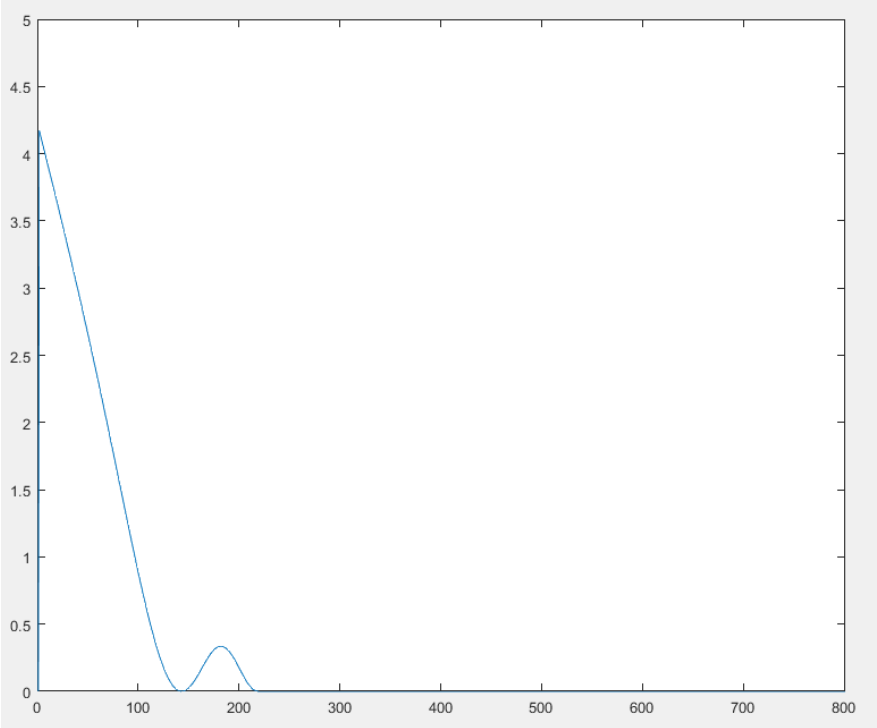

Figure 4: $\|u_{gtg} - u\|^2$ as the robot moves in figure 1. The difference between the nominal control input and the one through the safety filter

2. Now we need to test with different γ(h(x)) functions, γ(h(x)) = 10h(x) and γ(h(x)) = $10h^3(x)$. These simulations are plotted in figures 5 and 9 respectively.
   The option in figure 1 stays further away from the obstacle than the options in figure 5 or 9. The option in figure 5 seems to take the shortest path from the options presented. The third option seems to take a bit wider route than the option two, staying a bit further away and turning away sooner. There is no huge difference between these two.

   We feel that the best option depends on the application of the robot. If it is necessary to stay further away from the obstacle the best solution, from the ones tested, would be the one presented in figure 1. If, however the goal is to reach the goal by the shortest way while avoiding the obstacle we feel like the option in figure 5 is the best one.

   After the figure 5 in figures 6-8 there is presented how the nominal go-to-goal and the safety filter input signals differ, how the h function evolves as the robot moves and the norm difference between the nominal and the safety filter input signals. And the same goes for figure 9 after which the same diagrams are presented in figure 10-12.

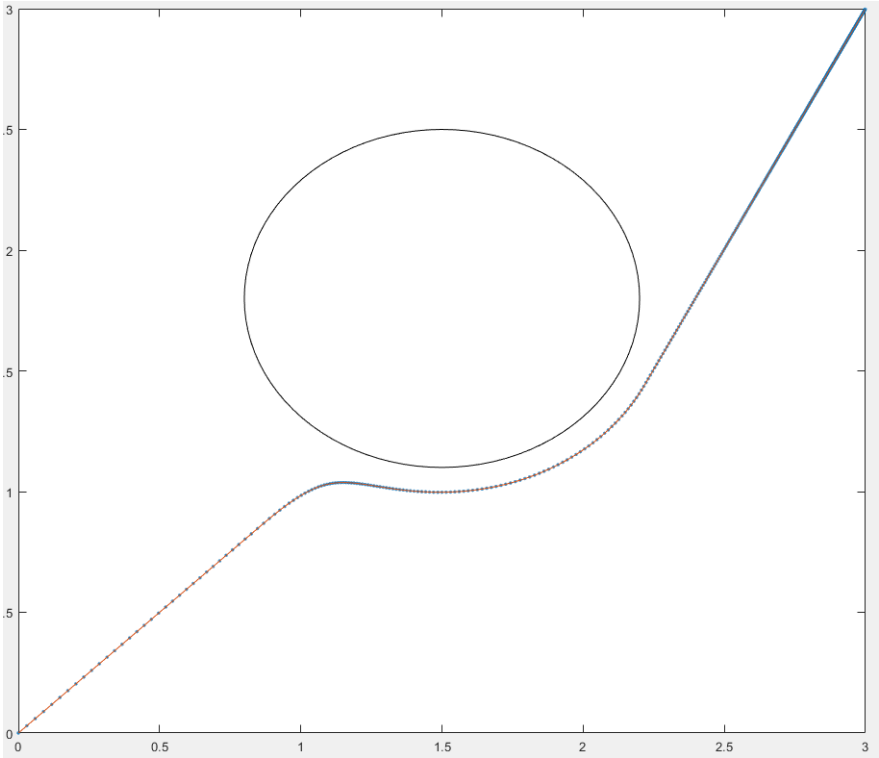

Figure 5: The robot path with γ(h(x)) = 10h(x)

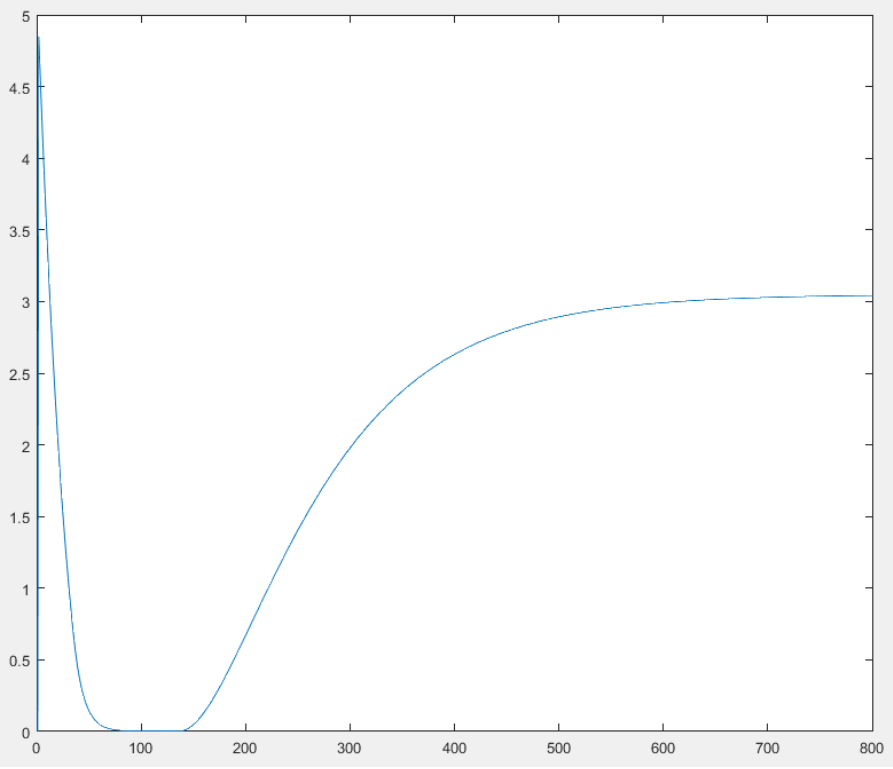

Figure 6: Evolution of the h as the robot moves in figure 5

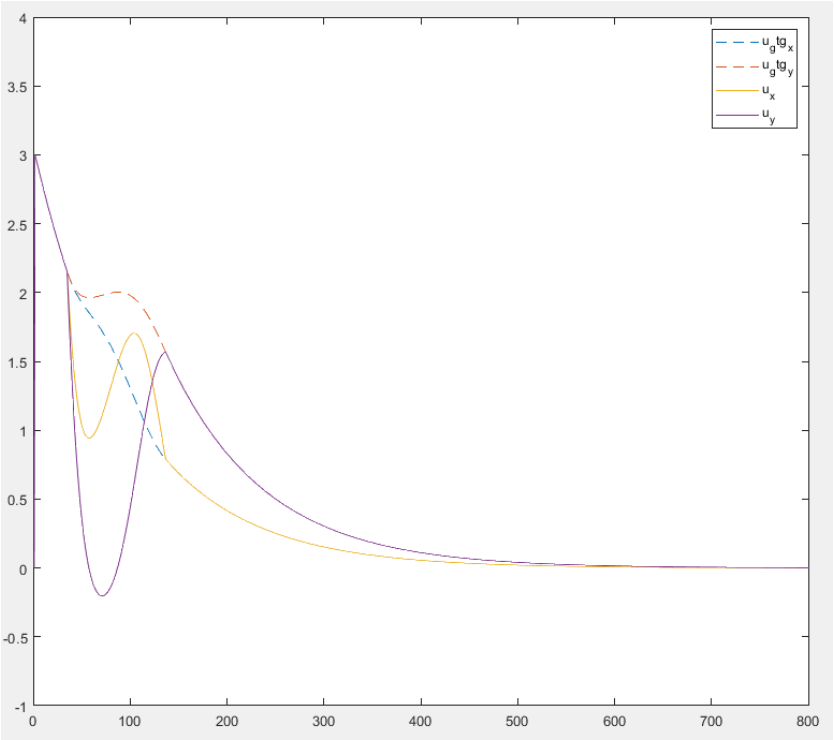

Figure 7: The nominal control input signal vs the input signal after the safety filter as the robot moves in figure 5

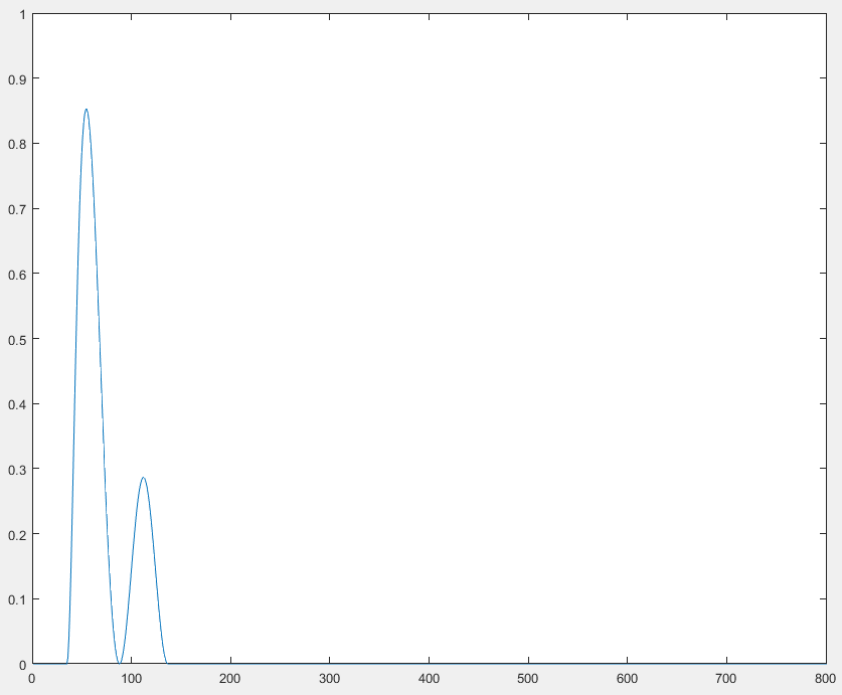

Figure 8: $\|u_{gtg} - u\|^2$ as the robot moves in figure 5. The difference between the nominal control input and the one through the safety filter

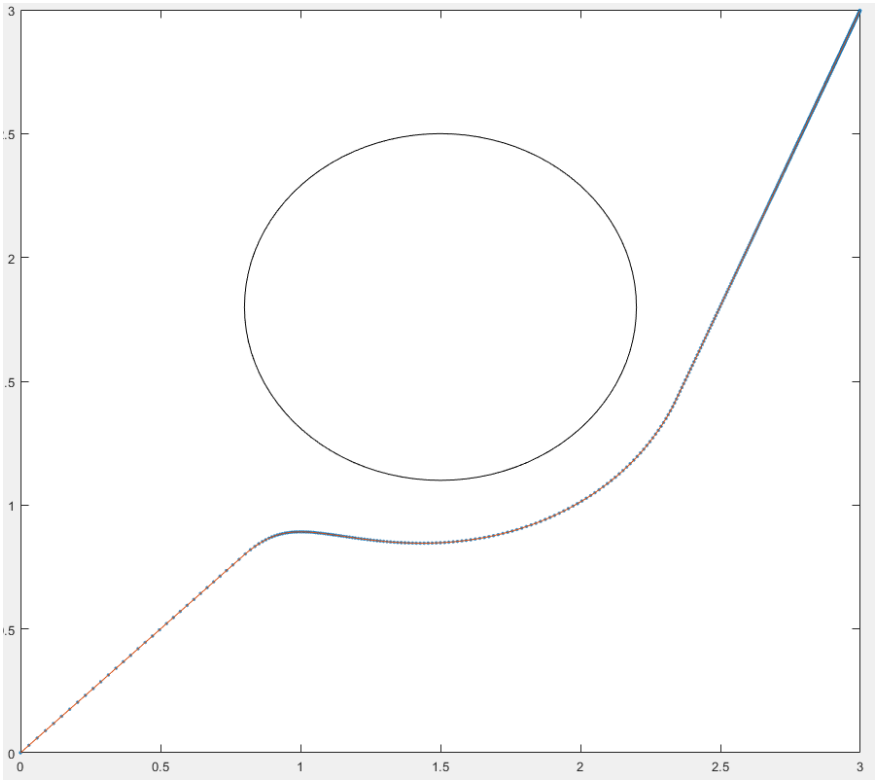

Figure 9: The robot path with $\gamma(h(x)) = 10h^3(x)$

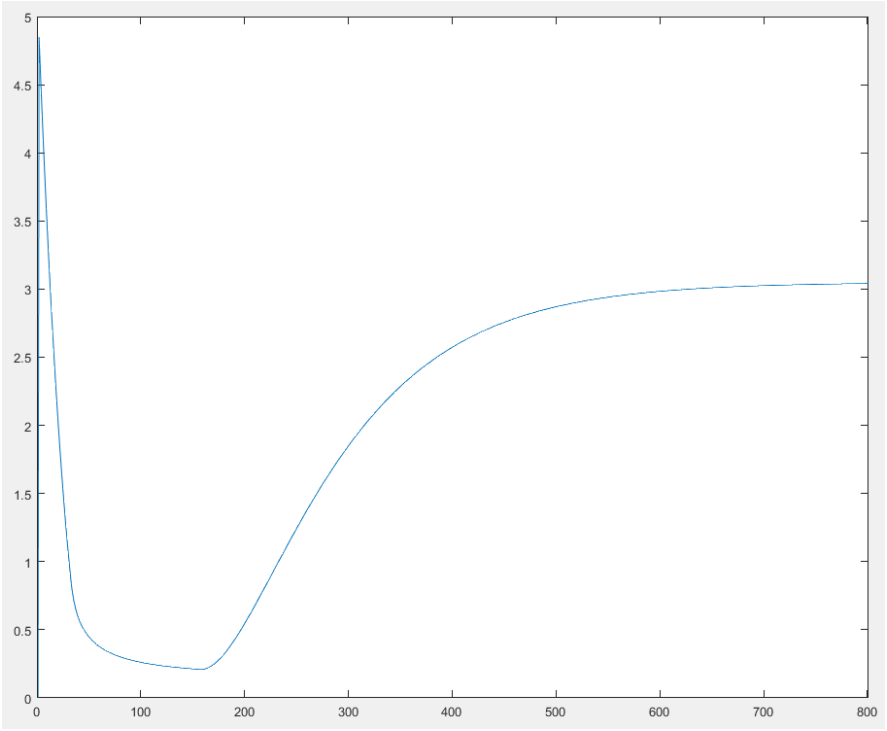

Figure 10: Evolution of the h as the robot moves in figure 9

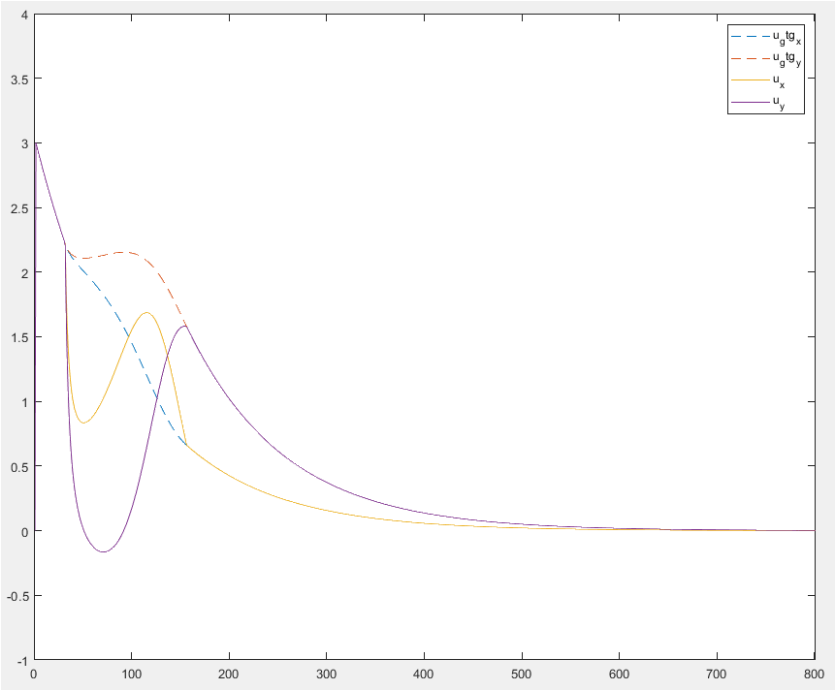

Figure 11: The nominal control input signal vs the input signal after the safety filter as the robot moves in figure 9

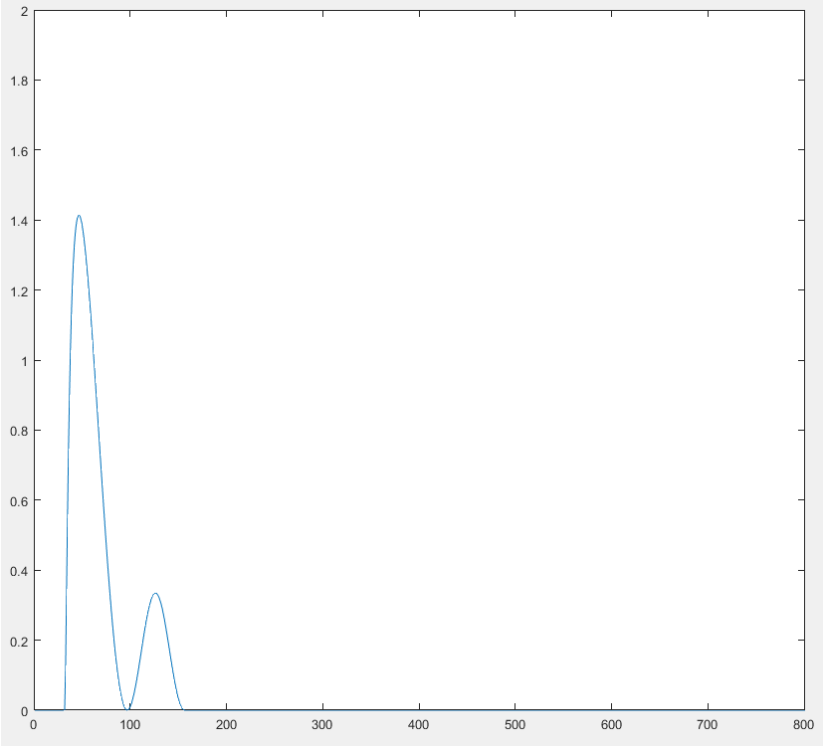

Figure 12: $\|u_{gtg} - u\|^2$ as the robot moves in figure 9. The difference between the nominal control input and the one through the safety filter

Problem 2

Using the same controller as in part 2 of the first problem γ(h(x)) = 10h(x).

1. Moving the obstacle to [1.5 1.5] directly on to the robot's path is plotted and presented in figure 13. It seems like the robot stops and doesn't know how to reach the goal. In figures 14-16 is then shown how the h function evolves as the robot approaches the obstacle which helps to explain why the robot stops moving and then how the nominal and safety filtered input signals differ from each other and behave as the robot approaches the obstacle directly in its path.

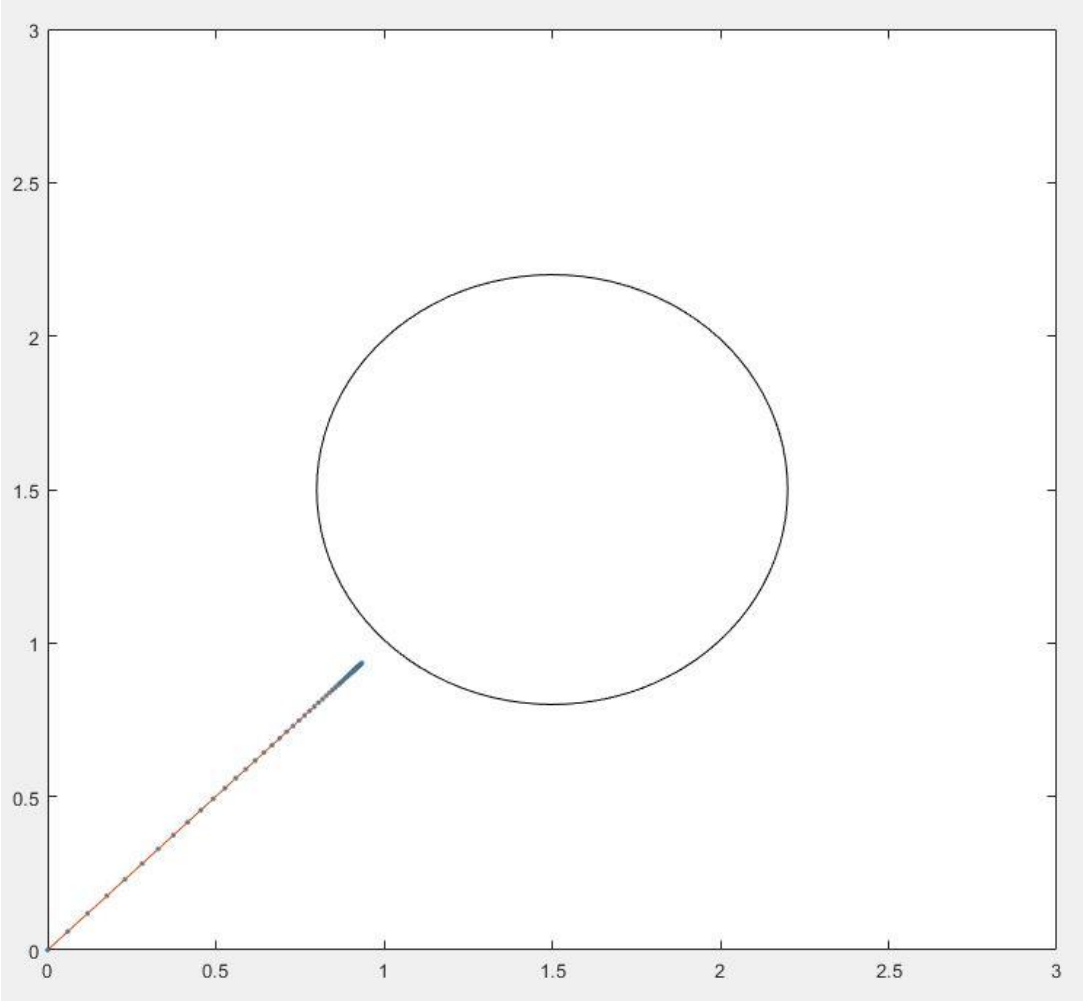

Figure 13: The robot path when obstacle is directly in the path.

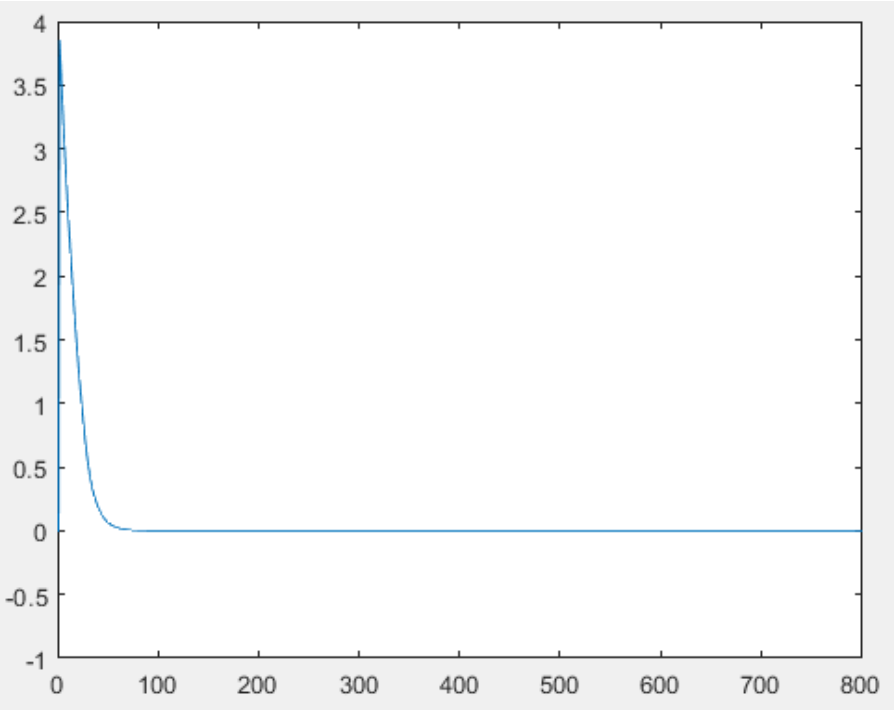

Figure 14: Evolution of the h as the robot moves in figure 13

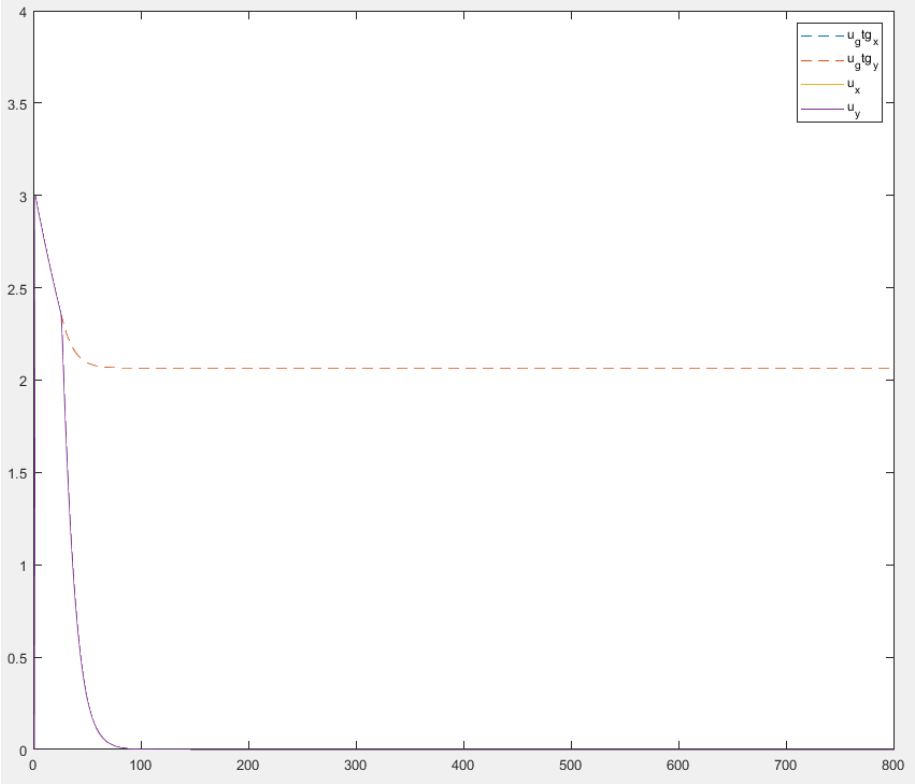

Figure 15: The nominal control input signal vs the input signal after the safety filter as the robot moves in figure 13

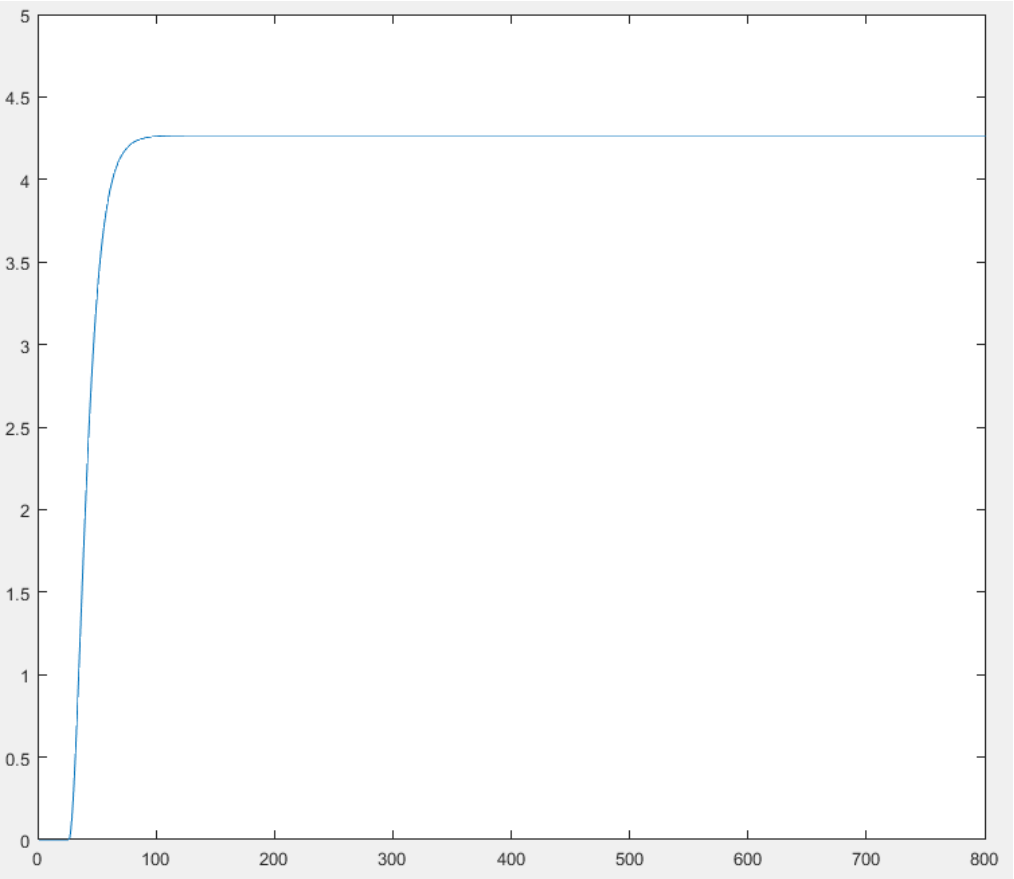

Figure 16: $\|u_{gtg} - u\|^2$ as the robot moves in figure 13. The difference between the nominal control input and the one through the safety filter

In a *deadlock* the robot stops as the control input goes to zero. This is because the path ahead is “as good” in either direction and the robot cannot decide on its own where it should go. So, the control input goes to zero and the robot stops.

2. Setting the sampling time to 250ms and plotting the resulting robot path is presented in figure 17.

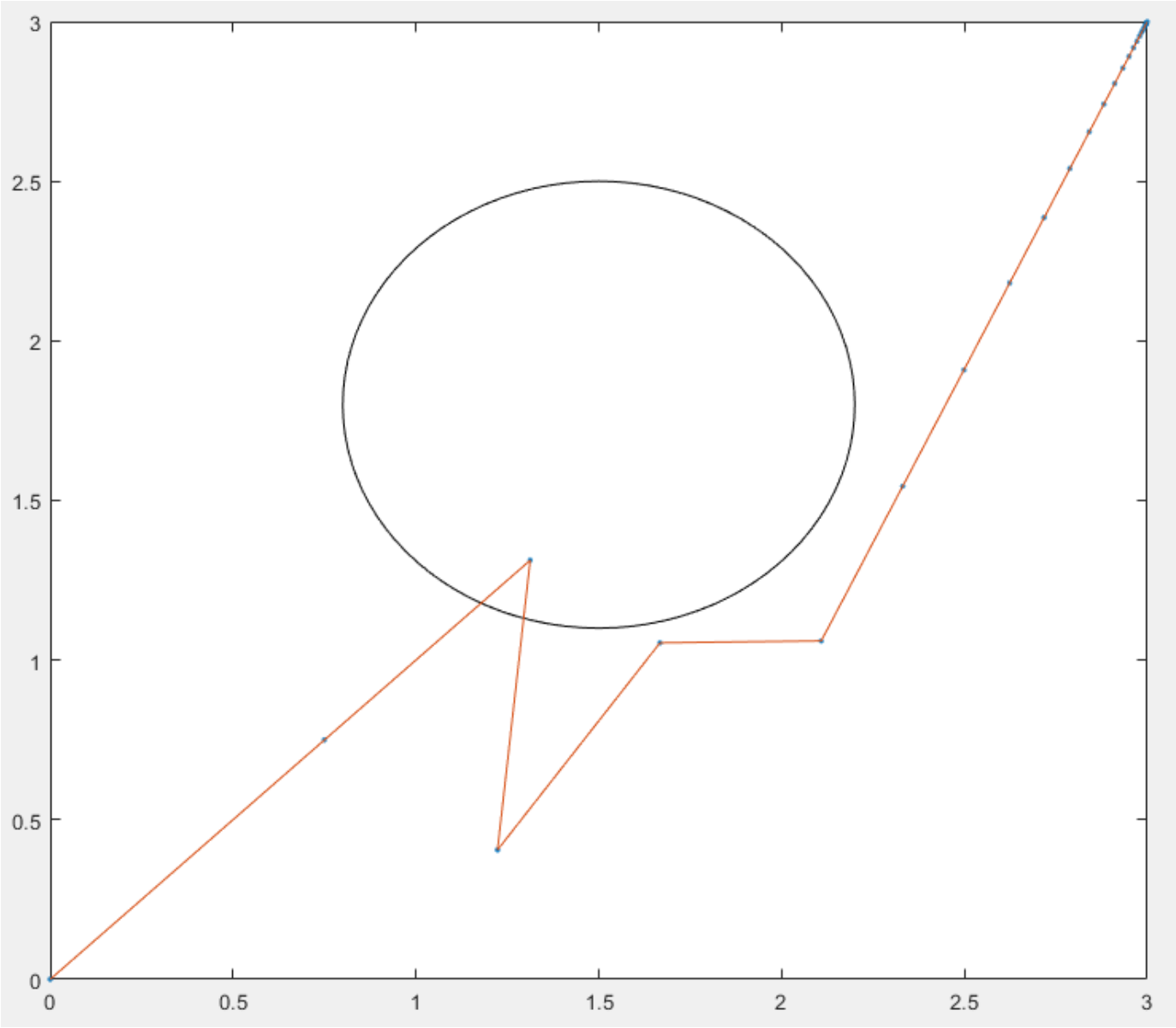

Figure 17: The robot path with γ(h(x)) = 10h(x) and sampling time 250ms

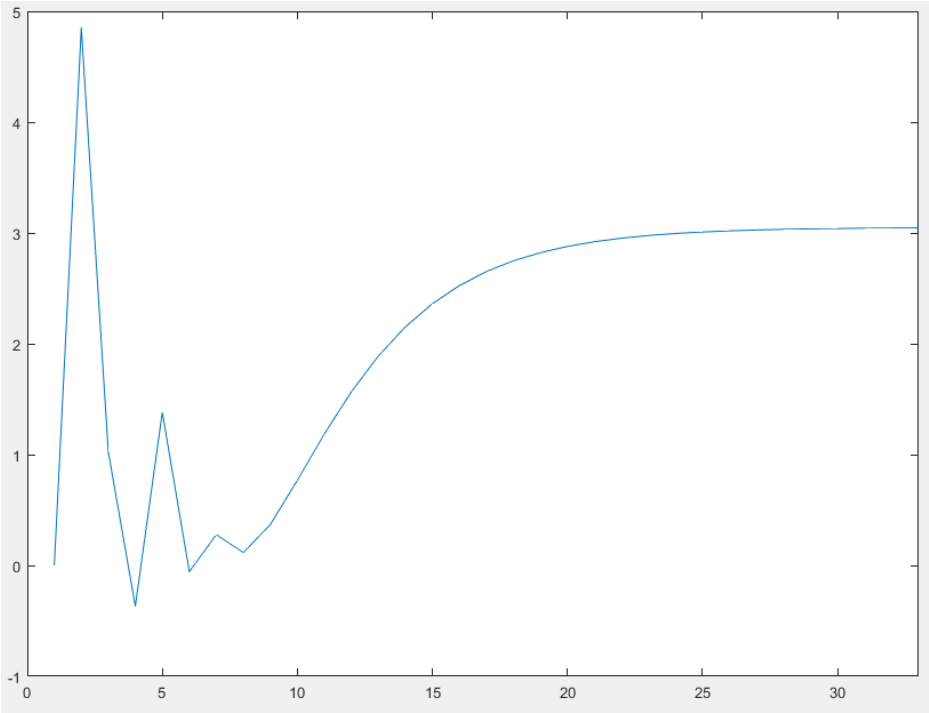

Figure 18: Evolution of the h as the robot moves in figure 17.

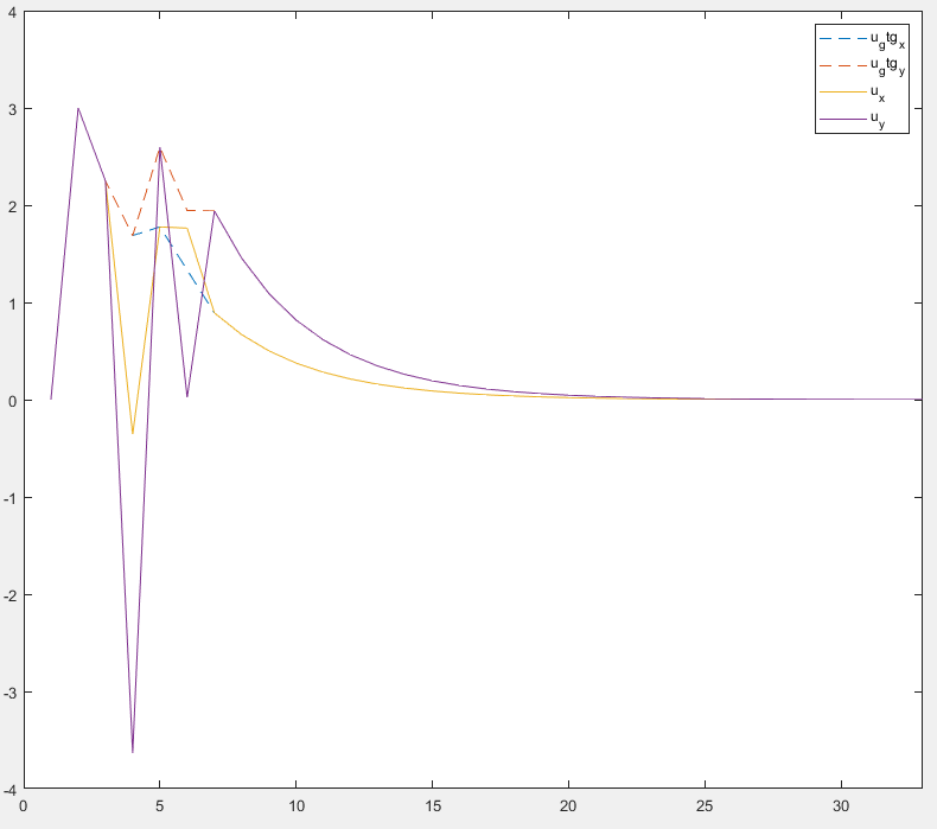

Figure 19: The nominal control input signal vs the input signal after the safety filter as the robot moves in figure 17.

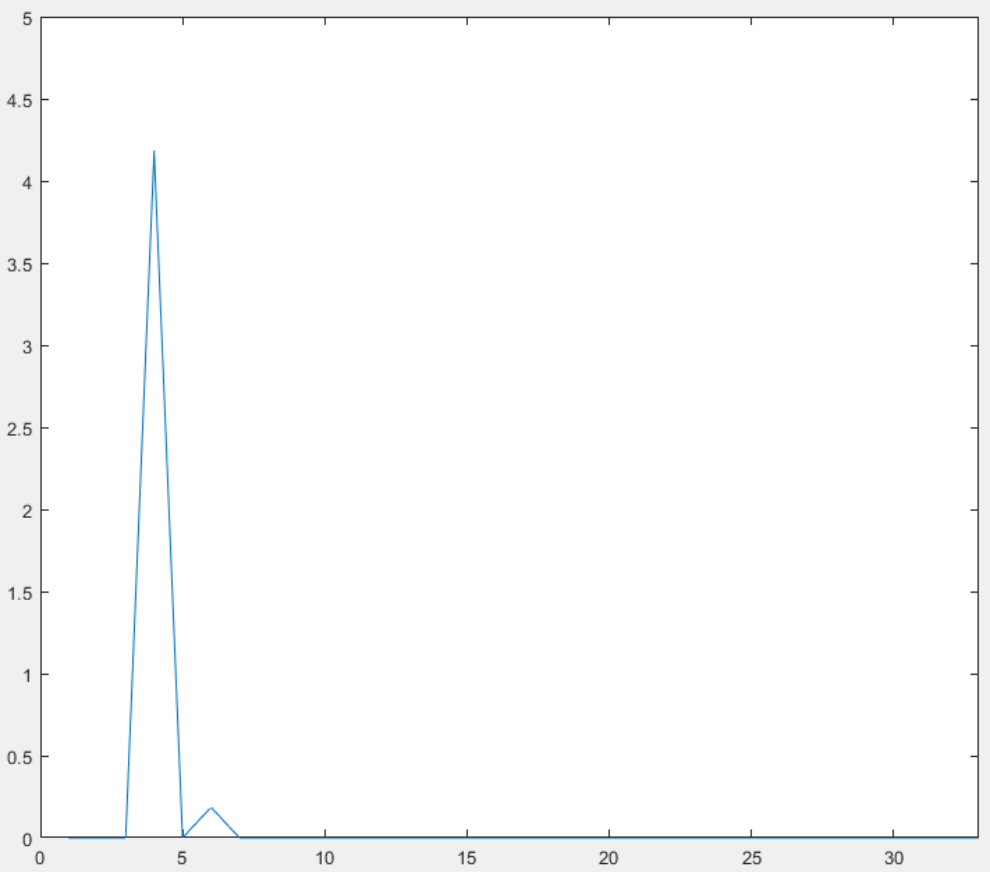

Figure 20: $\|u_{gtg} - u\|^2$ as the robot moves in figure 17. The difference between the nominal control input and the one through the safety filter.

We can see that the robot has a collision with the obstacle. This is because the controller doesn't get to give the next input signal before the robot has already collided with the obstacle. The sampling time makes it so the last time the controller got to give instructions the situation was such that it made the robot move straight into the obstacle. Then it overcorrects and makes this weird zigzag trying to find its way.

By trial and error, we can deduce that the maximum sampling time that still narrowly avoids a collision seems to be 218ms this is presented in figure 21. In figures 22-24 are then the same diagrams as with all the previous tests.

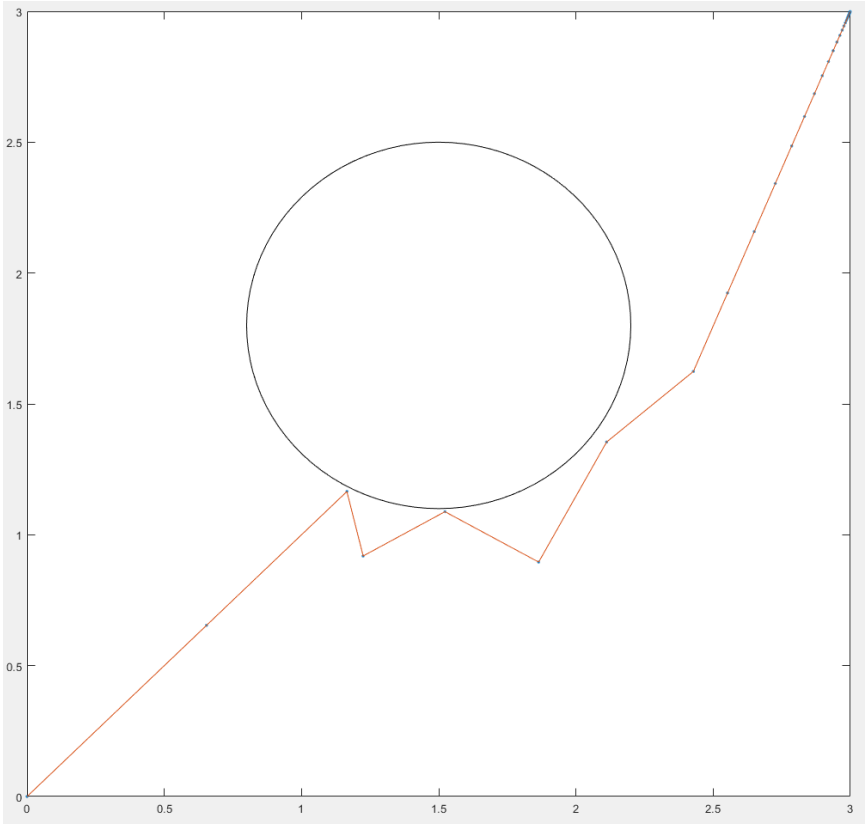

Figure 21: The robot path with γ(h(x)) = 10h(x) and sampling time 218ms.

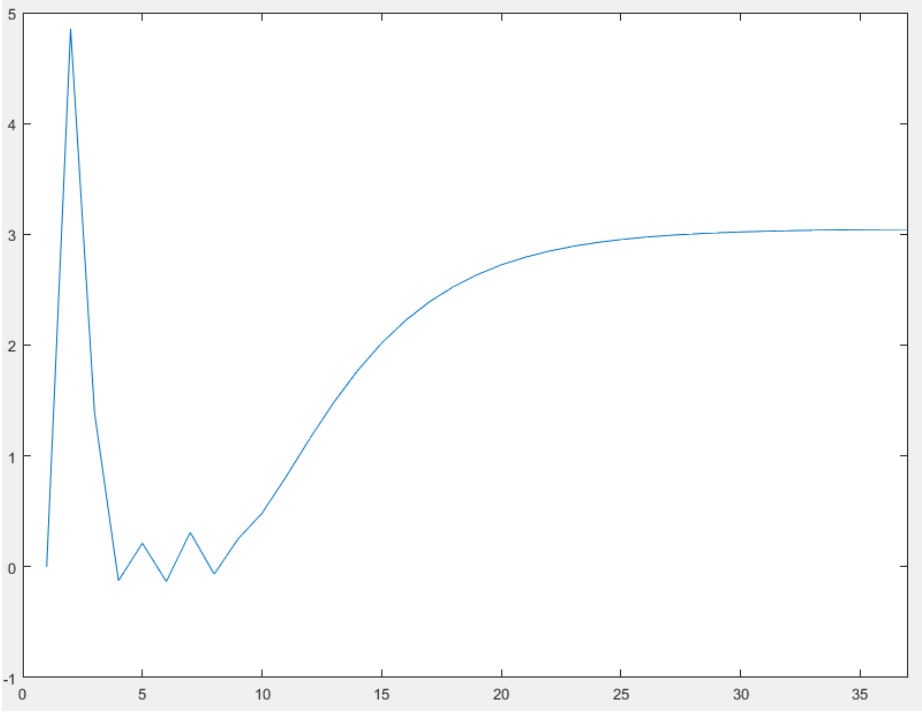

Figure 22: Evolution of the h as the robot moves in figure 21.

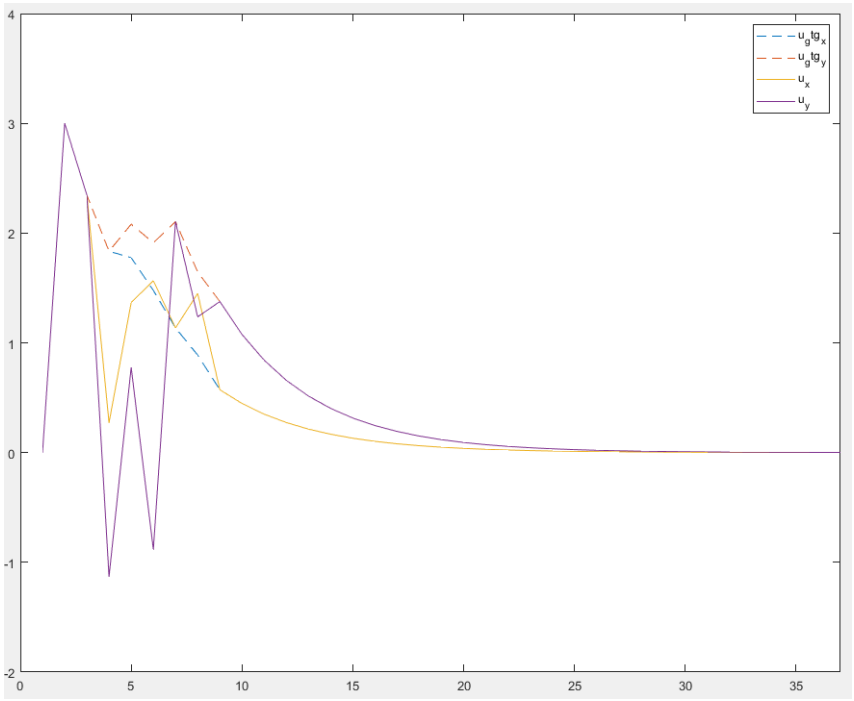

Figure 23: The nominal control input signal vs the input signal after the safety filter as the robot moves in figure 21.

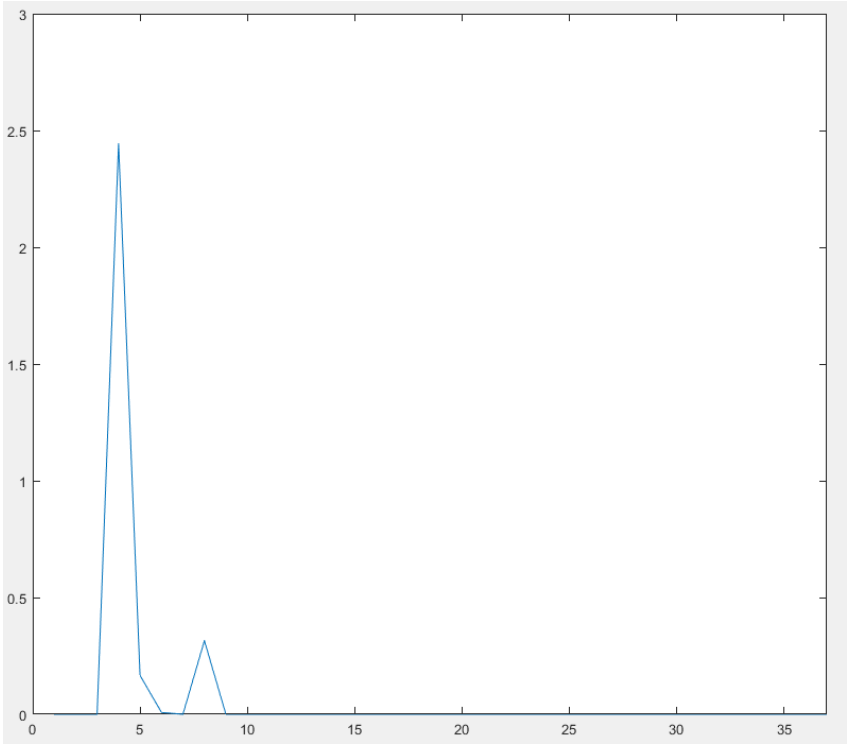

Figure 24: $\|u_{gtg} - u\|^2$ as the robot moves in figure 21. The difference between the nominal control input and the one through the safety filter.

Problem 3

Adding speed constraints to the robot as

$$-1 \leq u_x \leq 1, \qquad -1 \leq u_y \leq 1$$

And still using the controller γ(h(x)) = 10h(x) with sampling time of 10ms.

1. Rerunning the simulation with limiting the controller input without them being used as constraints for the controller. Results are plotted in figure 25. The h, and input signal comparisons are presented in figures 26-28.

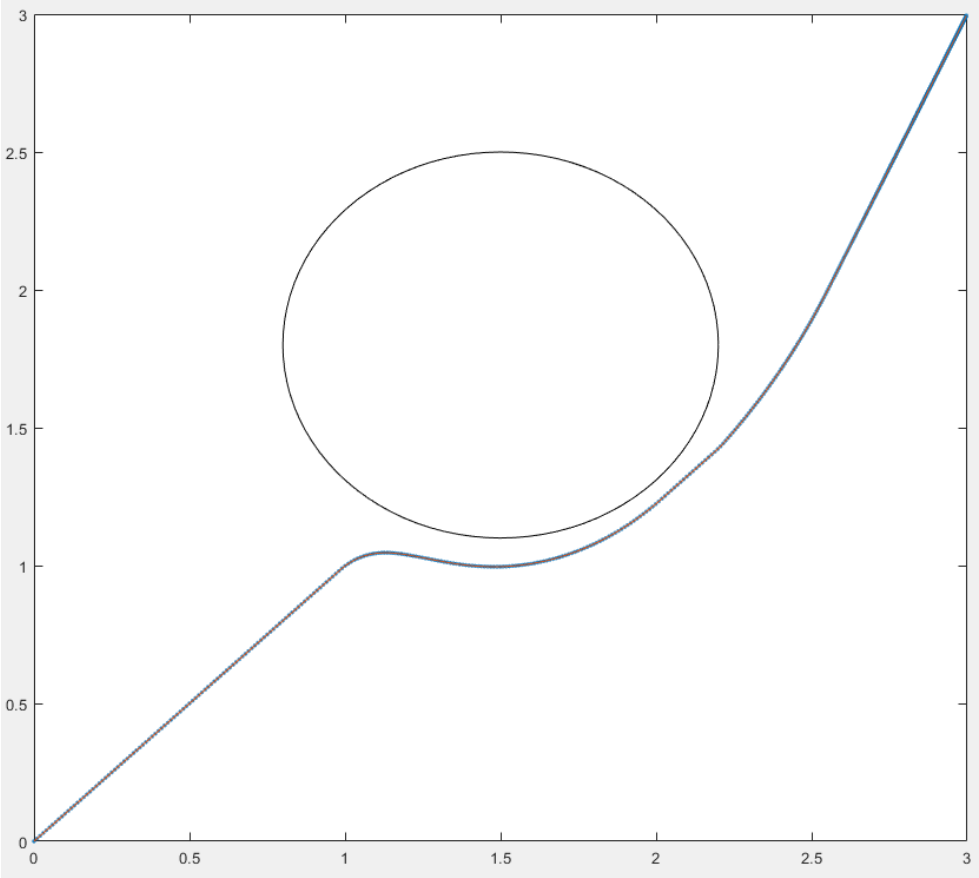

Figure 25: The robot path with γ(h(x)) = 10h(x) and input limited to -1 ≤ u ≤ 1

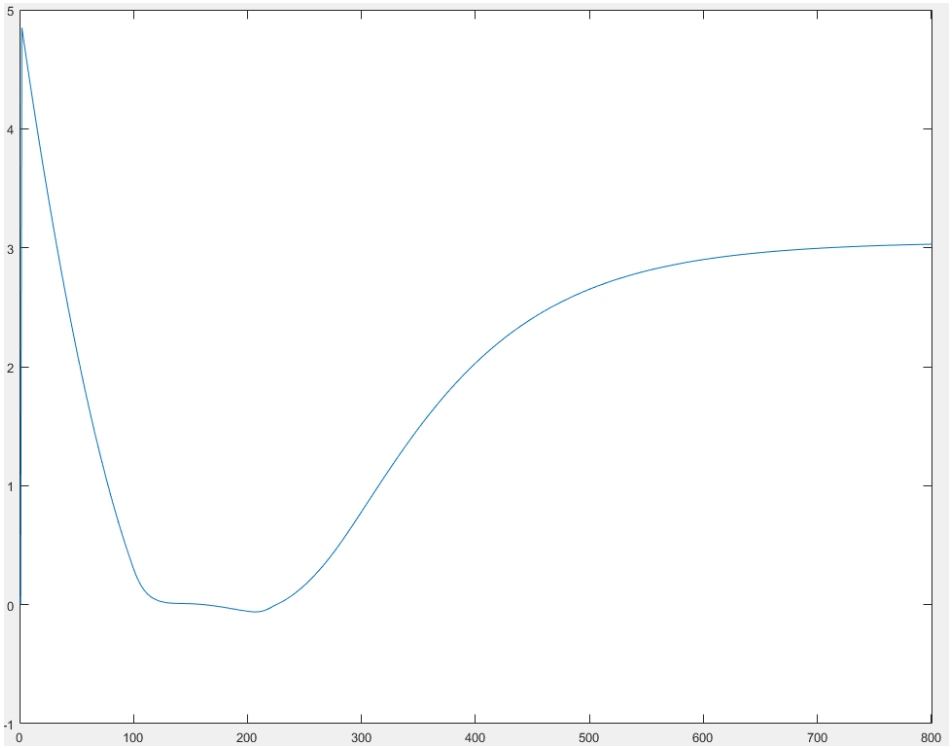

Figure 26: Evolution of the h as the robot moves in figure 25.

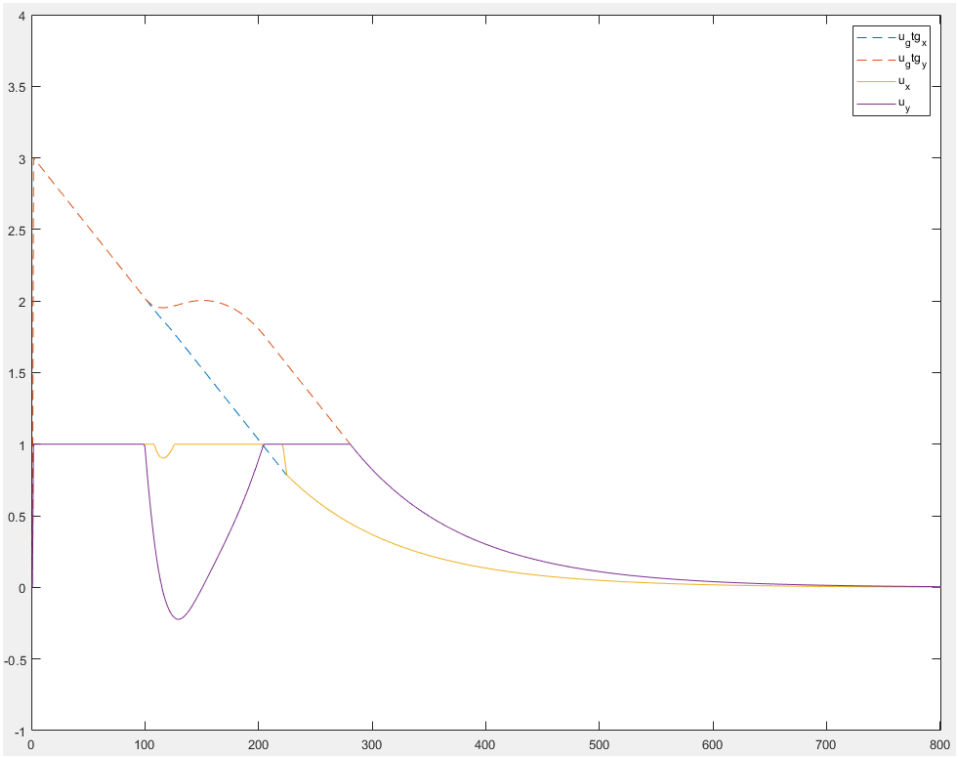

Figure 27: The nominal control input signal vs the input signal after the safety filter as the robot moves in figure 25.

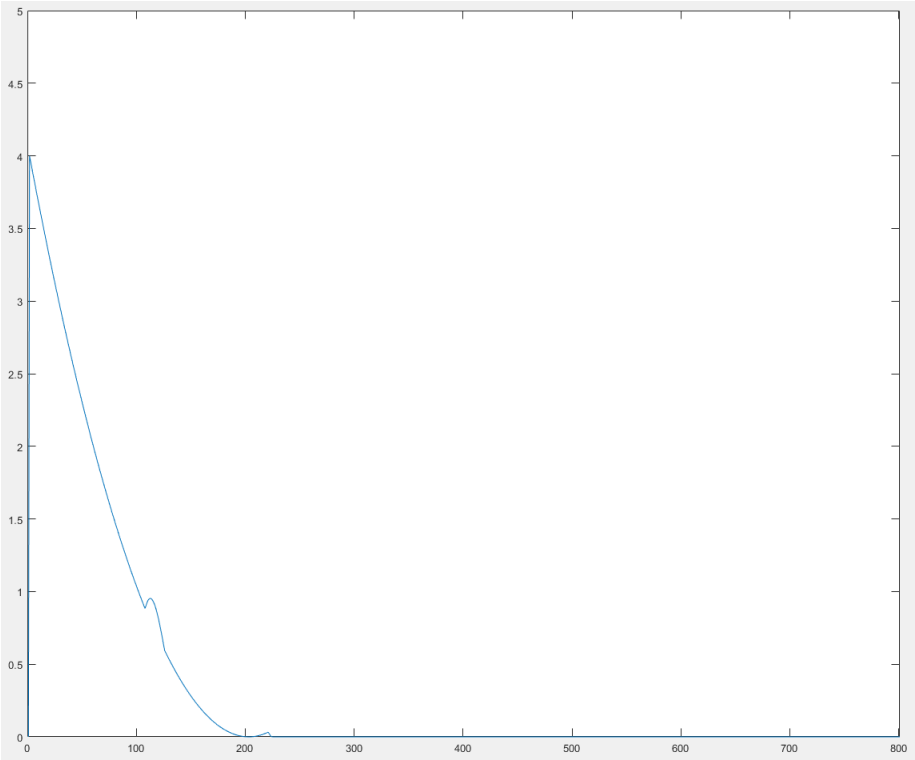

Figure 28: $\|u_{gtg} - u\|^2$ as the robot moves in figure 25. The difference between the nominal control input and the one through the safety filter

The path that the robot takes is stiff as the controller thinks it could go faster than it actually can resulting in those tiny wiggles of frustration as it passes the obstacle. This is because the controller doesn't know that the output it gives is limited by something and outputs larger or smaller input signals which it then needs to take into account on the next control cycle which is different than it anticipated.

2. Rerunning the simulation again but now by giving the controller the speed limitations as constraints results in a smoother looking path for the robot as is shown in figure 29. The h and input signal comparisons are provided in figures 30-32.

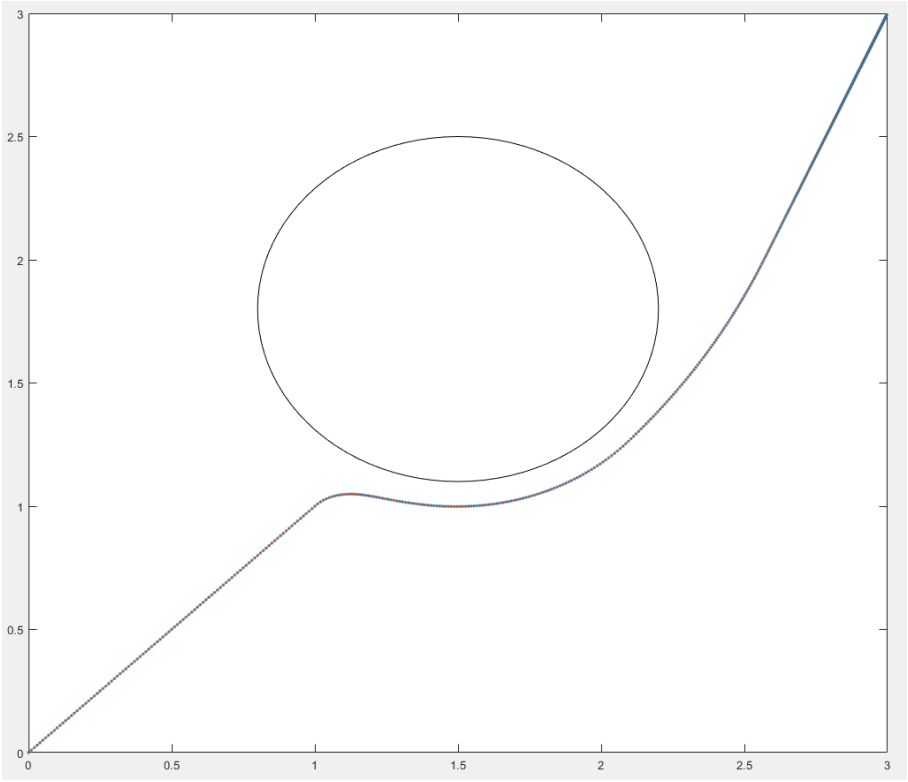

Figure 29: The robot path with γ(h(x)) = 10h(x) and input limited to -1 ≤ u ≤ 1 as control constraints

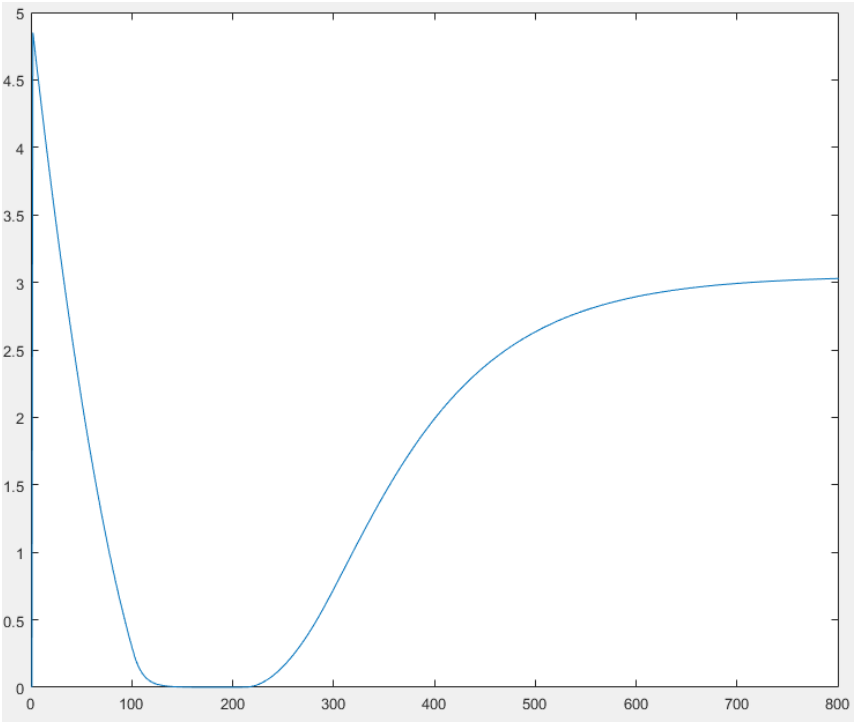

Figure 30: Evolution of the h as the robot moves in figure 29.

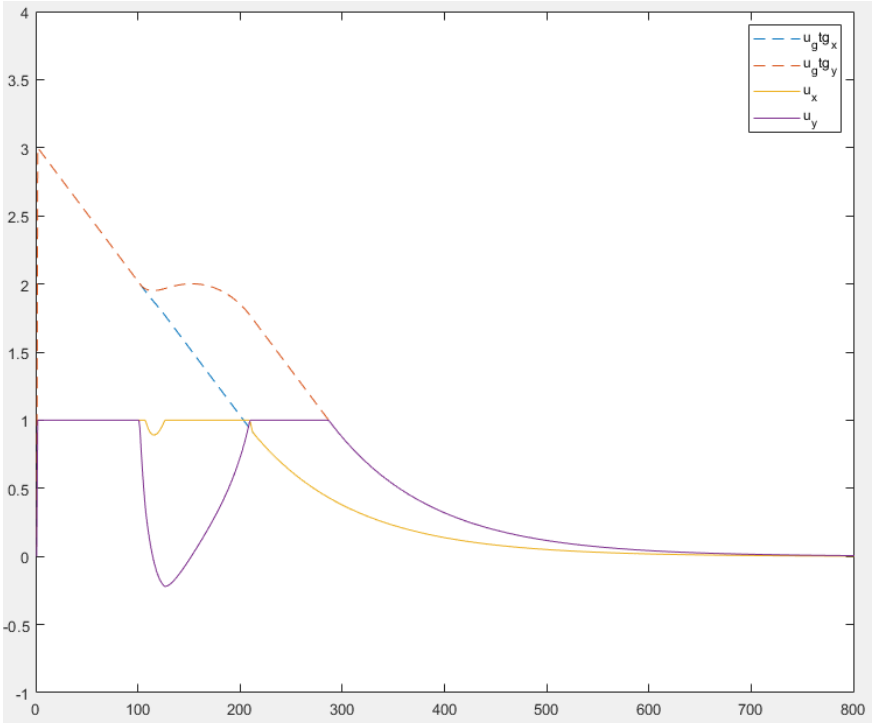

Figure 31: The nominal control input signal vs the input signal after the safety filter as the robot moves in figure 29.

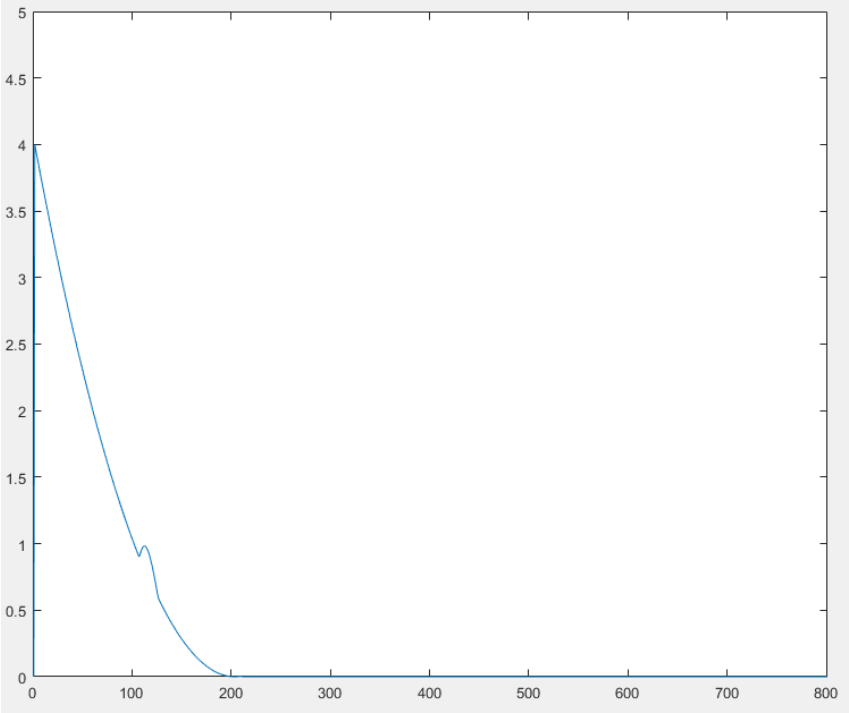

Figure 32: $\|u_{gtg} - u\|^2$ as the robot moves in figure 25. The difference between the nominal control input and the one through the safety filter

The resulting path seems to be a little bit smoother than the one in figure 25. There is not a huge difference but still a noticeable one. Which makes sense as the robot moves as the controller would expect without external limitations.

Sources

(1) Homework 9, M.W.S. Atman, M. Iqbal, A. Gusrialdi, March 31, 2023 https://moodle.tuni.fi/pluginfile.php/3206990/mod_resource/content/1/2023_DistControl_Problem_set9.pdf
(2) Lecture 10 slides, A. Gusrialdi, March 31, 2023. https://moodle.tuni.fi/pluginfile.php/3206003/mod_resource/content/1/AUT360_lecture10_handout2.pdf