Navigation and Control of a small quadrotor

Davide Carminati and Iris David Du Mutel

May 2022

1 Abstract

Unmanned aerial vehicles are gaining popularity due to the many applications in which they can be used. To be able to perform difficult tasks, control strategies must be designed to follow the assigned trajectory. For non-aggressive manoeuvres, basic control structures such as PID controllers can be a perfect fit. However, challenging trajectories can be required in demanding fields of application, which require more elaborate control approaches. In this work, aggressive and non-aggressive trajectories have been handled with two different control approaches: a PID controller for a straight line trajectory, and a cascaded Sliding Mode Controller (SMC) architecture for the ascending helix trajectory.

2 Introduction

Quadrotors are of growing interest to the scientific community, as they are challenging to operate due to their non-linear dynamics. The study of control and navigation strategies is not an old topic for other types of aircraft such as common airplanes and fixed-wing UAV. Multi-copters started to become popular in the early 2000s [1], when scientific institutions started developing their own prototypes. Nowadays, small multi-copters are studied extensively in the control and navigation fields.

A common approach to the control of these aircraft revolves around simple and traditional control strategies such as PID controllers as in [2], [3]. Also, from optimal-control theory, controllers such as Linear Quadratic Regulators (LQR) are also used in [4], [5]. More complex techniques are also used, like Model Predictive Control (MPC) are also a possibility [6] for complex trajectories. SMC [7] and Backstepping [8] approaches are commonly used for challenging trajectories as the one intended in this project.

The objective of this work is that of designing a control strategy to perform two different trajectories: a non-aggressive trajectory and an more complex aggressive trajectory. A quadrotor of small dimensions must be able to perform a simple linear trajectory and also an ascending helix type of trajectory. We chose a simple PID based strategy for the linear trajectory. For the helix path, a more elaborated controller is used. In this case, a Cascaded Sliding Mode Controller (SMC) has been chosen.

The intention behind this project is that of simulating a non-linear model of an quadcopter and successfully fly the proposed trajectories. A linear controller such as the PID is a good choice when the trajectories are linear and simple to keep track of. When more aggressive trajectories are necessary, a more robust option must be available. Using non linear control algorithms becomes helpful when dealing with disturbances and more complex trajectories. A cascade SMC approach is considered in order to follow the aggressive trajectory.

To carry on these tests, MATLAB&Simulink is used as simulation environment. The drone model has been written in MATLAB functions used inside a Simulink model to run the whole setup. A manual tuning is performed for both control approaches.

The first trajectory to be tested is a linear trajectory consisting on three steps: an initial hover to a certain height, an advancement in the horizontal plane, parallel to the ground plane, and a vertical descent back to ground. On the other hand, the ascending helix is a continuous trajectory that is composed by only one longer section that varies position and altitude. In the following sections, the procedures and methods used for this project are described.

3 Methods

In this work, a simulative approach based on the Newton-Euler mathematical formulation is employed considering a state vector as in 1:

$$X = \begin{bmatrix} x \\ y \\ z \\ \psi \\ \theta \\ \psi \\ u \\ v \\ w \\ p \\ q \\ r \end{bmatrix} \tag{1}$$

The model equations correspond to the ones in 2:

$$\begin{aligned}
\dot{x} &= u \\
\dot{y} &= v \\
\dot{z} &= w \\
\dot{\phi} &= p + (\sin\phi\tan\theta)q - (cos\phi\tan\theta)r \\
\dot{\theta} &= (cos\phi)q + (sin\phi)r \\
\dot{\psi} &= -(sin\phi\sec\theta)q + (cos\phi\sec\theta)r \\
\ddot{x} &= -\frac{1}{m}(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi)T \\
\ddot{y} &= -\frac{1}{m}(\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi)T \\
\ddot{z} &= g - \frac{1}{m}(\cos\phi\sin\cos)T \\
\dot{p} &= \frac{I_y - I_z}{I_x}qr - \frac{I_{rz}}{I_x}q(\omega_1 - \omega_2 + \omega_3 - \omega_4) + \frac{1}{I_x}\tau_1 \\
\dot{q} &= \frac{I_z - I_x}{I_y}pr + \frac{I_{rz}}{I_y}q(\omega_1 - \omega_2 + \omega_3 - \omega_4) + \frac{1}{I_y}\tau_2 \\
\dot{r} &= \frac{I_x - I_y}{I_z}pq + \frac{1}{I_z}\tau_3
\end{aligned} \tag{2}$$

A trajectory planner has been implemented to produce the references that are fed to the controllers. A simple linear interpolation is computed between the waypoints of a desired trajectory. Similarly, the velocity profiles are computed and trapezoidal velocity profiles are obtained. This way, the velocity references are not as sharp as a step signal.

Also, a motor mixer has been implemented to obtain the rotation speed of every motor as a function of the outputs of the controllers. These speeds are used for the computation of the gyroscopic effect. The model however receives the control output directly.

3.1 The model parameters

A real platform has been chosen from [9], since experimental tests were made to determine the inertias and mass of a real platform that can be seen in Figure 1.

The parameters chosen for the quadcopter to be simulated are the following:

Feature	Value
I_{xx}	$0.033113kgm^2$
I_{yy}	$0.033889kgm^2$
I_{zz}	$0.050445kgm^2$
Mass	$1.5kg$

Table 1: Main characteristics of the real platform

Figure 1: Real platform

The drone has been considered symmetrical and therefore, the products of inertia are assumed null.

3.2 PID controller

The PID controller is one of the most popular control architectures due to its ease to be used and the good results it generates. The difference between a reference signal and the real variable goes through a linear feedback action to achieve the desired command value. The PID controller has three parts; a proportional term or P, proportional to the error; an integral term or I, proportional to the integral of the error and a derivative term or D, proportional to the derivative of the error. A general expression for a PID controller can be seen in Equation 3:

$$PID(t) = K_p e(t) + K_D \frac{de(t)}{dt} + K_I \int e(t)dt \tag{3}$$

Where K_P, K_I, K_D are the PID parameters and $e(t) = r(t) - y(t)$ is the error between a reference signal $r(t)$ and the real variable $y(t)$.

In this application, six PID controllers are used. The architecture is composed by an inner loop, in charge of the attitude and altitude control; and an outer loop, that deals with the position in the x-y plane through the velocities. The overall structure can be seen in Figure 2:

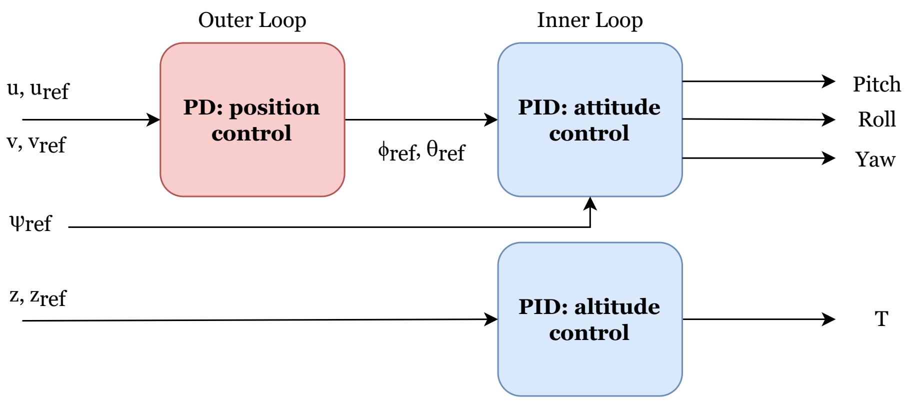

Figure 2: PID based control architecture

In the following sections, the two control loops are discussed in more detail.

3.2.1 Outer loop

The outer loop consists on two PD blocks that deal with the velocity references in the x-y plane. Such references come from the trajectory planner and are u_{ref} that refers to the desired velocity in the x axis and v_{ref} that is the corresponding velocity in the y axis. This fraction of the control scheme can be seen in Figure 3:

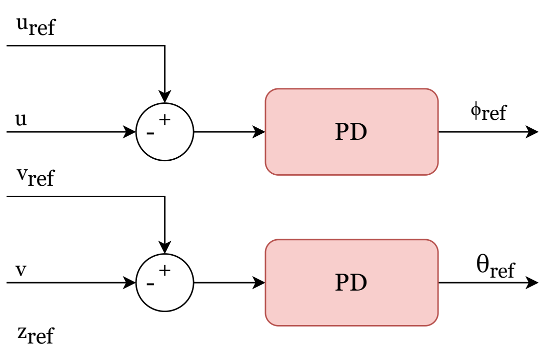

Figure 3: Outer loop: position control

3.2.2 Inner loop

The inner loop is in charge of attitude and altitude. In this case, full PID blocks are used as depicted in Figure 4:

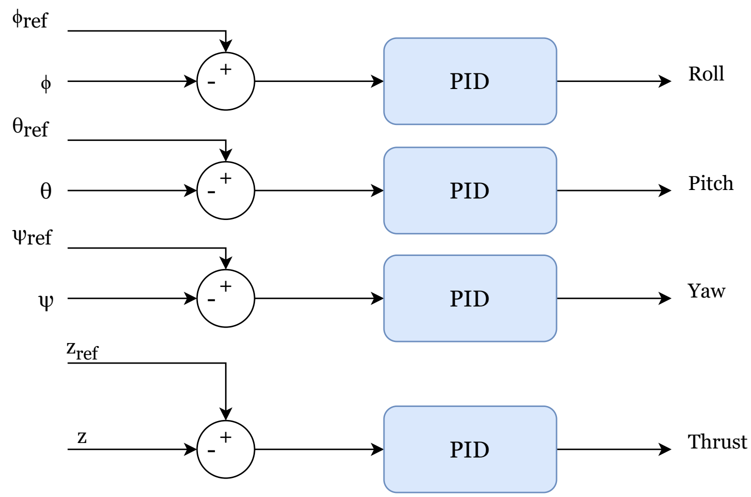

Figure 4: Inner loop: attitude and altitude control

The input for the pitch and roll controllers come from the position control loop. The remaining references are taken directly from the trajectory planner.

Numerical data regarding the controllers can be found in subsection 4.1.

3.3 SMC controller

The Sliding Mode Controller (SMC) is a Variables Structure Controller that employs a *sliding surface* to enforce the system to move towards and stay in a stability condition. In this application, a cascade architecture is used, taking [10] as an example, where two SMC are used for every signal to be controlled. In the case at hand, a pair of sliding mode controllers in a cascade fashion is used for ϕ, θ, ψ and z. The command law for both inner and outer SMC is:

$$u = K_* \tanh(\sigma_*) \tag{4}$$

where K_* and σ_* are respectively the gain and the *sliding surface* for inner or outer SMC. The *sliding surfaces* for the outer SMC are:

$$\sigma_\phi = \phi_{ref} - \phi + \lambda_o^{roll} \int_\tau (\phi_{ref} - \phi) d\tau \tag{5}$$

$$\sigma_\theta = \theta_{ref} - \theta + \lambda_o^{pitch} \int_\tau (\theta_{ref} - \theta) d\tau \tag{6}$$

$$\sigma_\psi = \psi_{ref} - \psi + \lambda_o^{yaw} \int_\tau (\psi_{ref} - \psi) d\tau \tag{7}$$

while for the inner SMC:

$$\sigma_\phi = \dot{\phi}_{ref} - \dot{\phi} + \lambda_i^{roll} \int_\tau (\dot{\phi}_{ref} - \dot{\phi}) d\tau \tag{8}$$

$$\sigma_\theta = \dot{\theta}_{ref} - \dot{\theta} + \lambda_i^{pitch} \int_\tau (\dot{\theta}_{ref} - \dot{\theta}) d\tau \tag{9}$$

$$\sigma_\psi = \dot{\psi}_{ref} - \dot{\psi} + \lambda_i^{yaw} \int_\tau (\dot{\psi}_{ref} - \dot{\psi}) d\tau \tag{10}$$

Please note that in this case, supposing that ϕ, θ and ψ are small, we consider: $\dot{\phi} = p$, $\dot{\theta} = q$ and $\dot{\psi} = r$. The reference roll and pitch angles ϕ_{ref} and θ_{ref} are computed from the error between the desired position provided by the trajectory planner and actual position along X and Y using a PD controller. The scheme for roll and pitch channels can be seen in Figure 5.

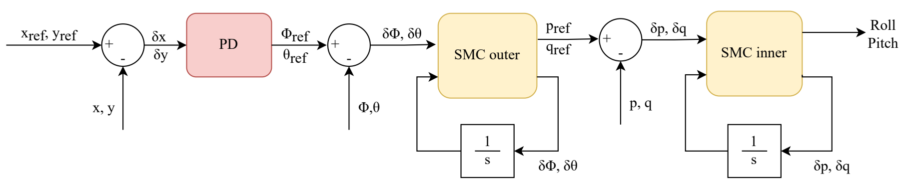

Figure 5: Cascaded SMC + PD control architecture

As for yaw, only the cascade SMC is employed, which is provided with the reference directly by the trajectory planner. The scheme is as follows (Figure 6:

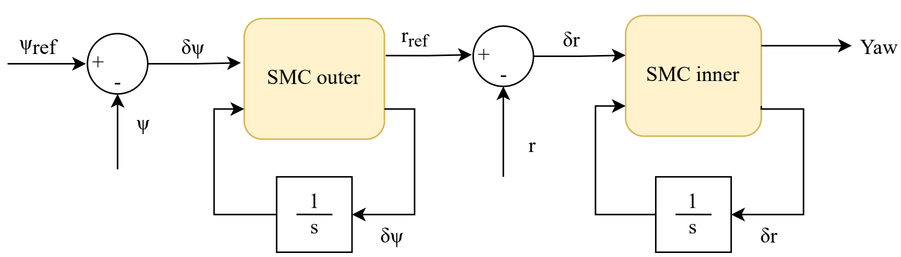

Figure 6: Cascaded SMC for yaw control

Finally, the altitude is driven by a simple SMC with same command law as Equation 4 and *sliding surface*:

$$\sigma_z = w_{ref} - w + \lambda_z (z_{ref} - z) \tag{11}$$

where z_{ref} and w_{ref} are generated by the trajectory planner. The numerical characteristics of these controllers are discussed in subsection 4.2.

4 Results and discussion

In this section, a deeper discussion about the characteristics of the controllers is presented. Also, the main results for each of the trajectories are obtained and commented.

4.1 PID controller

The PID controllers descirbed in subsection 3.2 have the following characteristics:

Attitude control		
Axis	Parameter	Value
Roll	K_P	0.08
	K_I	0.02
	K_D	0.08
Pitch	K_P	0.08
	K_I	0.02
	K_D	0.08
Yaw	K_P	0.15
	K_I	0.0001
	K_D	0.2
Altitude	K_P	60
	K_I	70
	K_D	40

Table 2: Inner loop PID parameters

Velocity control		
Axis	Parameter	Value
Velocity (X)	K_P	0.15
	K_D	0.009
Velocity (Y)	K_P	0.15
	K_D	0.009

Table 3: Outer loop PD parameters

The linear trajectory to be tested consists on reaching 2m of altitude and then going forward 2m. Once done that, the quadcopter must hover back to ground.

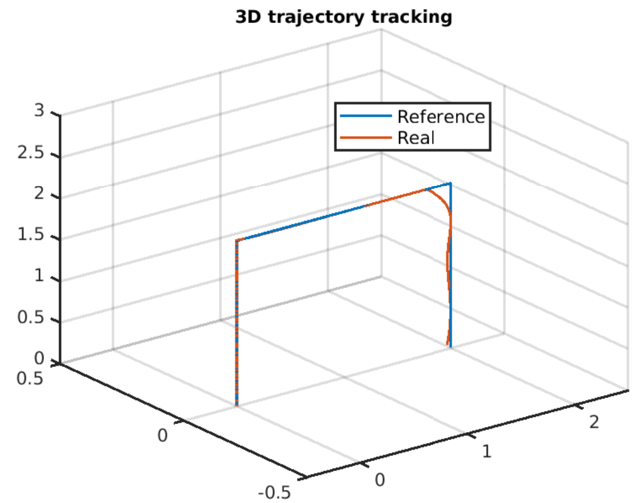

Figure 7: 3D tracking of a straight line trajectory

As seen in Figure 7, the trajectory is followed without major issues, validating the PID approach for this kind of trajectories. Refer to the annex for the rest of the figures regarding velocity, attitude and position signals.

4.2 SMC controller

The parameters of the controllers described in subsection 3.3 are reported in Table 4.2 and Table 5.

The trajectory to be tested consists on an ascending helix. This trajectory entails a growing altitude reference while the x-y position must describe a circumference. The equations of the desired trajectory are the following:

$$
\begin{gathered}
[x, y, z] = \left[2\cos\left(\frac{2\pi}{15}t - \frac{\pi}{2}\right), 2\sin\left(\frac{2\pi}{15}t - \frac{\pi}{2}\right), -0.1t\right] \\
\psi_{ref} = \frac{2\pi}{15}t
\end{gathered} \tag{12}
$$

Figure 8 shows the complete manoeuvre using the cascaded SMC. For additional figures, see the annex.

Attitude control		
Axis	Parameter	Value
Roll	λ_o	1e-3
	K_o	100
	λ_i	0.01
	K_i	10
Pitch	λ_o	1e-3
	K_o	100
	λ_i	0.01
	K_i	10
Yaw	λ_o	0.5
	K_o	2
	λ_i	0.1
	K_i	2
Altitude	λ_z	10
	K_z	40

Table 4: Inner and Outer SMC loop parameters

Position control		
Axis	Parameter	Value
Position (X)	K_P	0.5
	K_D	2
Position (Y)	K_P	0.5
	K_D	2

Table 5: Position PD parameters

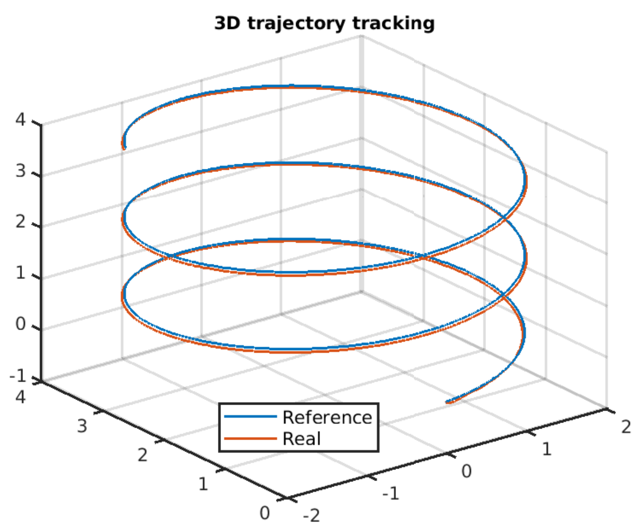

Figure 8: 3D tracking of a straight line trajectory

Even if the SMC approach has been used in [10] for an ascending helix capable of tracking a linearly growing ψ reference, in this work, such results could not be met. After many attempt at tuning the controllers, the system was able to track the references up to a certain point where the attitude became unstable. This issue is thought to be due to unsuitable tuning parameters.

5 Conclusions

As expected, the PID controller is a simple solution for equally simple trajectories. It performs satisfactorily for straight lines and analogous trajectories, and it is a controller that requires little to no knowledge about control to be used. It is easily tuned too.
On the other hand, the cascaded SMC is a more advanced approach for more complex manoeuvres. The tuning parameters of the SMC controllers might not be the fittest as no tracking of the heading reference was achieved when performing the aggressive manoeuvre. In literature, the tuning parameters are not found trough a trial-and-error technique. More advance methods involving Evolutionary Algorithms are used to find an optimal solution.

As a future goal, a finer tuning of the cascaded SMC controllers is due, in order to achieve an accurate tracking of the heading angle in aggressive trajectories.

6 Division of tasks in this project

This work is the product of the collaboration of Iris David Du Mutel and Davide Carminati. Both parts have participated in the writing of this document and have shared their ideas regarding the techniques and results displayed. However, the main tasks carried on by each collaborator are the following:

- Iris: Model construction, Motor mixer, PID architecture, report writing
- Davide: Trajectory planner, Cascaded SMC architecture, report writing

Both parts have participated to a greater or lesser extent in all the tasks listed above. A distinction is made based on the time employed.

References

[1] S Norouzi Ghazbi, Y Aghli, M Alimohammadi, and A A Akbari. Quadrotors unmanned aerial vehicles: A review. *INTERNATIONAL JOURNAL ON SMART SENSING AND INTELLIGENT SYSTEMS*, 9, 2016.

[2] Minh Nguyen Duc, Thang Nguyen Trong, and Yang Sheng Xuan. The quadrotor mav system using pid control. pages 506–510. Institute of Electrical and Electronics Engineers Inc., 9 2015.

[3] Jun Li and Yuntang Li. Dynamic analysis and pid control for a quadrotor. pages 573–578, 2011.

[4] Elias Reyes-Valeria, Rogerio Enriquez-Caldera, Sergio Camacho-Lara, and Jose Guichard. Lqr control for a quadrotor using unit quaternions: Modeling and simulation. pages 172–178, 2013.

[5] Zhao Shulong, An Honglei, Zhang Daibing, and Shen Lincheng. A new feedback linearization lqr control for attitude of quadrotor. pages 1593–1597. Institute of Electrical and Electronics Engineers Inc., 2014.

[6] M. Abdolhosseini, Y. M. Zhang, and C. A. Rabbath. An efficient model predictive control scheme for an unmanned quadrotor helicopter. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 70:27–38, 4 2013.

[7] Model-free-based terminal smc of quadrotor attitude and position. *IEEE Transactions on Aerospace and Electronic Systems*, 52:2519–2528, 10 2016.

[8] A novel nonlinear resilient control for a quadrotor uav via backstepping control and nonlinear disturbance observer. *Nonlinear Dynamics*, 85:1281–1295, 7 2016.

[9] Iris David Du Mutel de Pierrepont Franzetti. Guidance and control algorithms for uav indoor applications, 2020.

[10] Abdel-Razzak Mehreb, Hassan Noura, and François Bateman. Passive fault tolerant control of quadrotor uav using regular and cascaded sliding mode control. 2013.

Annex

6.1 PID graphs

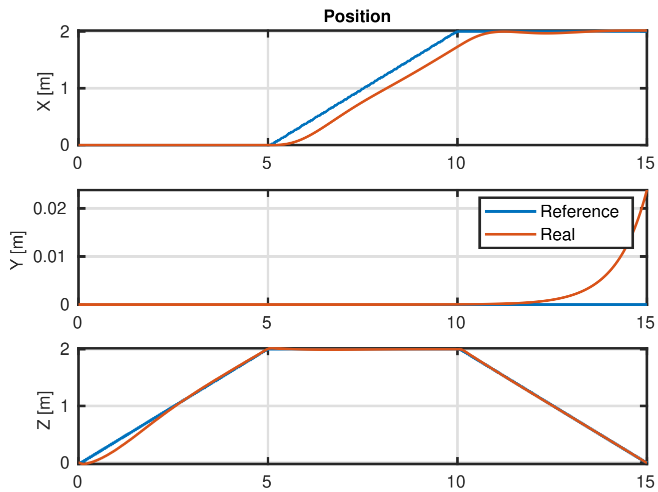

Figure 9: PID: Position references with state variables

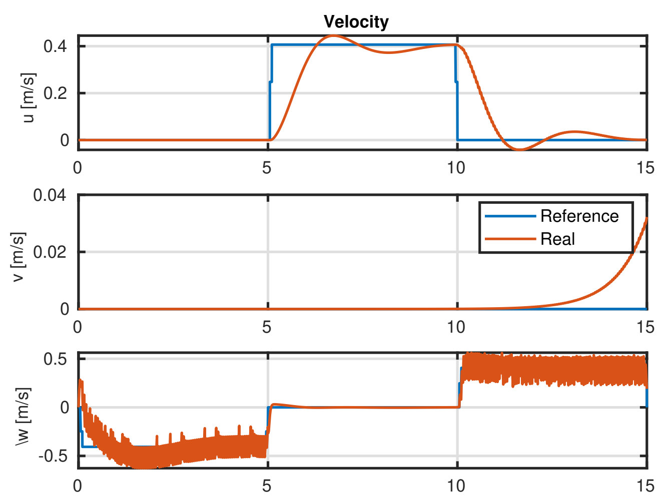

Figure 10: PID: Velocity references with state variables

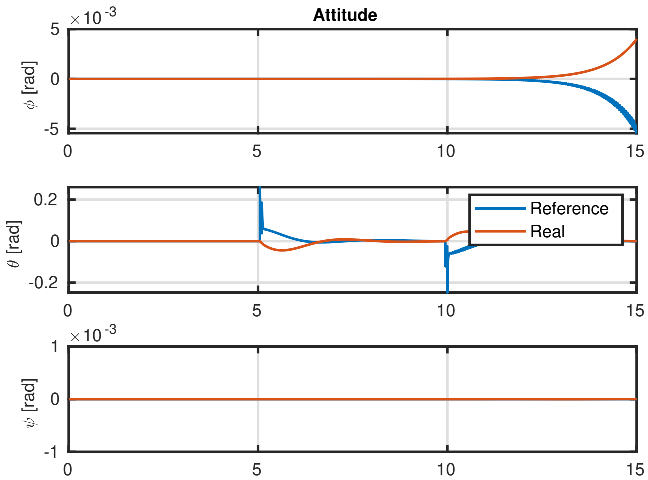

Figure 11: PID: Attitude references with state variables

6.2 Cascaded SMC graphs

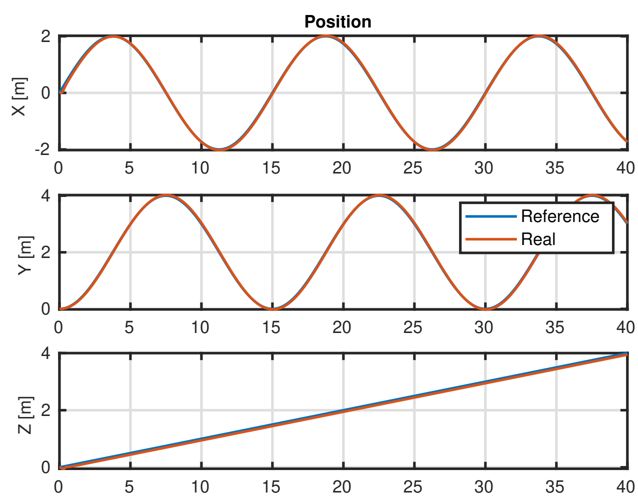

Figure 12: Cascaded SMC: Position references with state variables

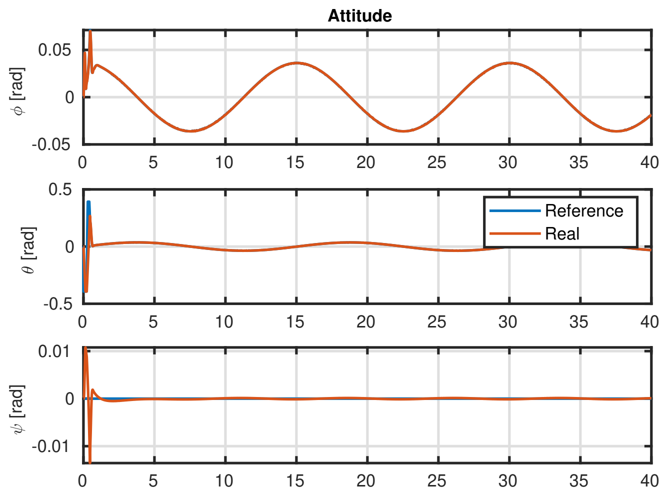

Figure 13: Cascaded SMC: Attitude references with state variables