Project Report
Model Predictive Control

EL2700 – Model Predictive Control

October 1, 2020

Gianni Hotz
Gobi Kumarasamy
Akash Singh
Sanjay Varadharajan

Postal address	**Visiting address**	**Internet**
Royal Institute of Technology	Malvinas Vag 10	www.kth.se/dcs
Division of Decision and Control Systems	Stockholm	
Sweden		

Quadrotor Dynamics

1. The linear system matrices dependent on the equilibrium point x_{eq} derived with the first order Taylor expansion are implemented in the function `Quadcopter_linear_dynamics`. The linearization is chosen to be around the zero also for the yaw angle because it is not expected that a deviation of the yaw angle from zero is necessary. The system matrices are given in the following:

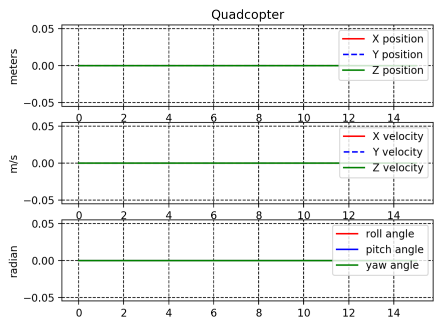
$$A = \begin{pmatrix}
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \frac{f_z(\cos\theta\sin\psi - \cos\psi\sin\phi\sin\theta)}{m} & \frac{f_z\cos\phi\cos\psi\cos\theta}{m} & \frac{f_z(\cos\psi\sin\theta - \cos\theta\sin\phi\sin\psi)}{m} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -\frac{f_z(\cos\psi\cos\theta + \sin\phi\sin\psi\sin\theta)}{m} & \frac{f_z\cos\phi\cos\theta\sin\psi}{m} & \frac{f_z(\sin\psi\sin\theta + \cos\psi\cos\theta\sin\phi)}{m} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -\frac{f_z\cos\phi\sin\theta}{m} & -\frac{f_z\cos\theta\sin\phi}{m} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & w_z\cos\phi(\tan^2\theta + 1) + w_y\sin\phi(\tan^2\theta + 1) & w_y\cos\phi\tan\theta - w_z\sin\phi\tan\theta & 0 & 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -w_z\cos\phi - w_y\sin\phi & 0 & 0 & \cos\phi & -\sin\phi \\
0 & 0 & 0 & 0 & 0 & 0 & \frac{w_z\cos\phi\sin\theta}{\cos\theta^2} + \frac{w_y\sin\phi\sin\theta}{\cos\theta^2} & \frac{w_y\cos\phi}{\cos\theta} - \frac{w_z\sin\phi}{\cos\theta} & 0 & 0 & \frac{\sin\phi}{\cos\theta} & \frac{\cos\phi}{\cos\theta} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{M_y w_z - M_z w_z}{M_x} & \frac{M_y w_y - M_z w_y}{M_x} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{M_x w_z - M_z w_z}{M_y} & 0 & -\frac{M_x w_x - M_z w_x}{M_y} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{M_x w_y - M_y w_y}{M_z} & \frac{M_x w_x - M_y w_x}{M_z} & 0
\end{pmatrix}$$

$$B = \begin{pmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
\frac{\sin\psi\sin\theta + \cos\psi\cos\theta\sin\phi}{m} & 0 & 0 & 0 \\
-\frac{\cos\psi\sin\theta - \cos\theta\sin\phi\sin\psi}{m} & 0 & 0 & 0 \\
\frac{\cos\phi\cos\theta}{m} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & \frac{1}{M_x} & 0 & 0 \\
0 & 0 & \frac{1}{M_y} & 0 \\
0 & 0 & 0 & \frac{1}{M_z}
\end{pmatrix}$$

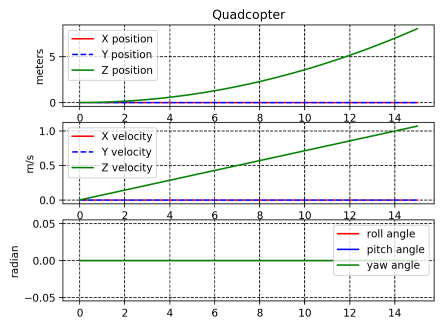
2. The linearization is calculated in CasADi using a sampling time of $h = 0.1\,\text{s}$. With that the following matrices are obtained:

$$A = \begin{pmatrix}
1 & 0 & 0 & 0.1 & 0 & 0 & 0 & 0.0490585 & 0 & 0 & 0.00169298 & 0 \\
0 & 1 & 0 & 0 & 0.1 & 0 & -0.0490585 & 0 & 0 & -0.00169298 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0.1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0.981 & 0 & 0 & 0.0490585 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & -0.981 & 0 & 0 & -0.0490585 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0.1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0.1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0.1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}
\quad
B = \begin{pmatrix}
0 & 0 & 0.0409157 & 0 \\
0 & -0.0409157 & 0 & 0 \\
0.00357143 & 0 & 0 & 0 \\
0 & 0 & 1.63502 & 0 \\
0 & -1.63502 & 0 & 0 \\
0.0714286 & 0 & 0 & 0 \\
0 & 5.00001 & 0 & 0 \\
0 & 0 & 5.00001 & 0 \\
0 & 0 & 0 & 1 \\
0 & 100 & 0 & 0 \\
0 & 0 & 100 & 0 \\
0 & 0 & 0 & 20
\end{pmatrix}$$

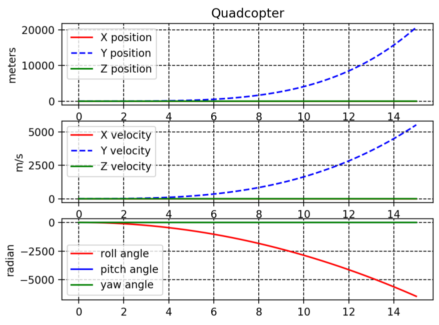
The results of the tests with $u = \begin{bmatrix}0 & 0 & 0 & 0\end{bmatrix}^T$, $\begin{bmatrix}0.1 & 0 & 0 & 0\end{bmatrix}^T$, $\begin{bmatrix}0.1 & -0.001 & 0 & 0\end{bmatrix}^T$ and $\begin{bmatrix}0.1 & 0 & 0.001 & 0\end{bmatrix}^T$ are shown in figure 1. The behavior of the Quadrotor is as expected.



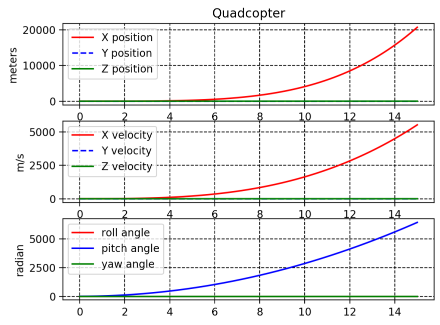
(a) $u = \begin{bmatrix}0 & 0 & 0 & 0\end{bmatrix}^T$



(b) $u = \begin{bmatrix}0.1 & 0 & 0 & 0\end{bmatrix}^T$



(c) $u = \begin{bmatrix}0.1 & -0.001 & 0 & 0\end{bmatrix}^T$



(d) $u = \begin{bmatrix}0.1 & 0 & 0.001 & 0\end{bmatrix}^T$

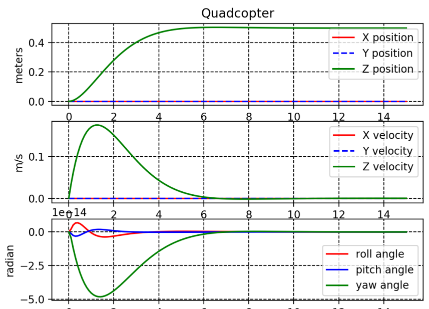
Figure 1: Test runs of the Quadrotor with $u = \begin{bmatrix}0 & 0 & 0 & 0\end{bmatrix}^T$, $\begin{bmatrix}0.1 & 0 & 0 & 0\end{bmatrix}^T$, $\begin{bmatrix}0.1 & -0.001 & 0 & 0\end{bmatrix}^T$ *and* $\begin{bmatrix}0.1 & 0 & 0.001 & 0\end{bmatrix}^T$.

MPC for a Quadrotor

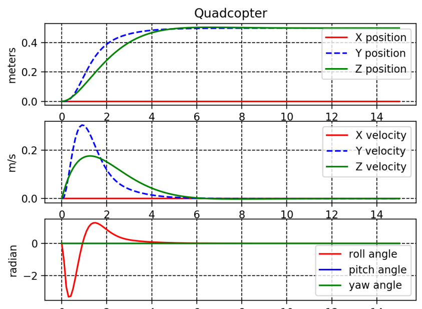
3. The weight matrices Q, R, and bounds on the inputs and states were chosen as below :

```python
# Solve the ARE for our system to extract the terminal weight matrix P
Q = np.diag([1,1,1,1,1,1,1,1,1,1,1,1])
R = np.diag([1,1,1,1])
P = np.matrix(scipy.linalg.solve_discrete_are(A,B,Q,R))

# Instantiate controller
max_fz  = 4 * m_vehicle  * Quadcopter_disturb.g
max_tau = (max_fz/2)*(17.5/100)
tau_z = 0.01

horizon=0.5
ulb=[-max_fz,-max_tau,-max_tau,-tau_z] #lower bounds on control inputs
uub=[max_fz,max_tau,max_tau,tau_z]     #upper bounds on control inputs
xlb=[-np.inf,-np.inf,-np.inf,-np.inf,-np.inf,-np.inf,-np.pi/2,-np.pi/2,-np.
    pi/2,-np.inf,-np.inf,-np.inf]       #lower bounds on states
xub=[np.inf,np.inf,np.inf,np.inf,np.inf,np.inf,np.pi/2,np.pi/2,np.pi/2,np.
    inf,np.inf,np.inf]                 #upper bounds on states
```

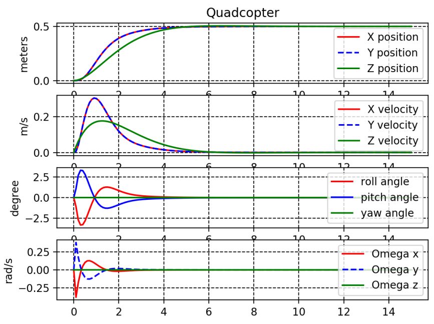
The set points $z = 0.5\,\text{m}$, $z = y = 0.5\,\text{m}$ and $z = y = x = 0.5\,\text{m}$ (with all other states equal to zero) are tested and the Quadrotor behavior is displayed in figure 2. The Quadrotor shows the expected behavior.



(a) $z = 0.5\,\mathrm{m}$



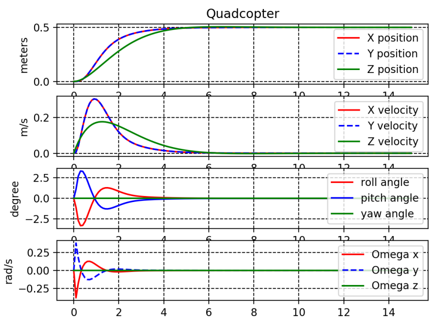
(b) $z = y = 0.5\,\mathrm{m}$



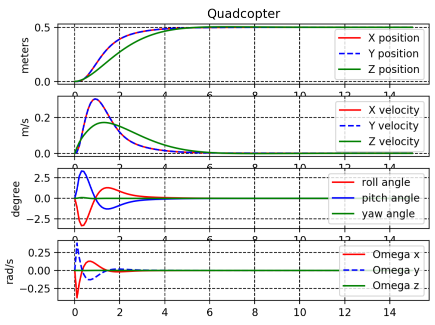
(c) $z = y = x = 0.5\,\mathrm{m}$

Figure 2: Test runs of the Quadrotor with the set points $z = 0.5\,\mathrm{m}$*,* $z = y = 0.5\,\mathrm{m}$ *and* $z = y = x = 0.5\,\mathrm{m}$ *(and all other states equal to zero).*

The behavior of the Quadrotor simulated with the non-linear environment instead of the linear one results in a very similar behavior, as can be seen in figure 3. Because the controller designed on the linear environment has a similar nice performance on the non-linear environment no changes were necessary.



(a) Linear environment.



(b) Nonlinear environment.

Figure 3: Simulation results of the Quadrotor with the set point $z = y = x = 0.5\,\mathrm{m}$ *(and all other states equal to zero) in a linear and non-linear environment.*

Integrator Dynamics

4. If the Quadrotor mass is changed in the simulation with respect to the controller Quadrotor mass, a steady state error is introduced to the system. This behavior is shown in figure 4a on the left, when the mass of the quadcopter as seen by the controller is 1.4kg whereas the actual mass is 1.9kg. The effect of this disturbance is mostly seen as steady-state error in z-position and z-velocity.

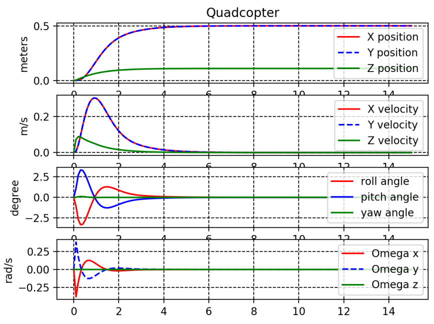
   To correct this error, integrator states for the x, y, and z-position of the Quadrotor were introduced as shown below :

$$i_x = r_x - x$$
$$i_y = r_y - y$$
$$i_z = r_z - z$$

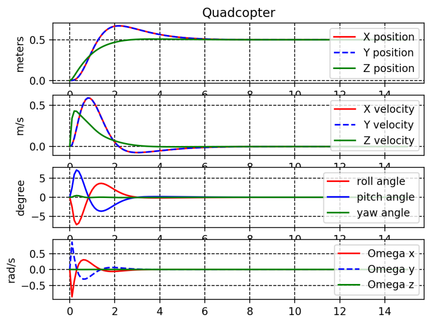
   Augmented system dynamics consisting of 15 states were created by adding 3 integrator states to the 12 system states. The discrete system matrices were appropriately populated to adjust the new augmented dynamics. The weight matrices, Q, R and the bounds were also changed to tune the new controller and get desired results.

```
Q = np.diag([1,1,1,1,1,1,1,1,1,1,1,1,1,1,1])
R = np.diag([1/(max_fz)**2, 1/(max_tau)**2,
              1/(max_tau)**2, 1/0.01**2])
P = np.matrix(scipy.linalg.solve_discrete_are(Ai,Bi,Q,R))

hrizon = 1
xlb=[-np.inf,-np.inf,-np.inf,-np.inf,-np.inf,-np.inf,-np.pi/2,-np.pi/2,-np.
    pi/2,-np.inf,-np.inf,-np.inf, -1,-1,-1 ] #lower bounds on states
xub=[np.inf,np.inf,np.inf,np.inf,np.inf,np.inf,np.pi/2,np.pi/2,np.pi/2,np.
    inf,np.inf,np.inf, 1,1,1]                #upper bounds on the states
```

   The errors were corrected as seen in the figure 4b.



(a) *Disturbed system without integral action.*



(b) *Disturbed system with integral action.*

Figure 4: Simulation results of the Quadrotor with disturbed mass $m' = 1.9\,\text{kg}$ *with and without integral action.*