# Final Project: Model Reference Adaptive Controller for Fault Estimation on an Unmanned Aerial Vehicle

Maxwell Cobar, Clifton Huett, and Carlos Montalvo

ME 494 – Nonlinear System Dynamics and Controls

May 7, 2020

## 1 Introduction:

The stability of a system is an important performance specification of a control system because it allows for the system to have a bounded output for a given bounded input. The stability of a system can depend on the parameters of the system [1]. The parameters such as mass and center of gravity of a system are known in most cases, however there are some systems where the parameters are unknown. A model reference adaptive controller (MRAC) is a control system that adapts to a controlled system with unknown parameters. A MRAC works by using a model of the system as a reference to the control system on the actual system. By comparing the model to the actual control system, unknown parameters can be estimated with great accuracy [2]. For this project, a MRAC is used to estimate faults that an unmanned aerial vehicle (UAV) may experience during flight.

## 2 Math Model:

Before the MRAC can be designed for the UAV, a free-body diagram (FDB) for the UAV needs to be developed, and from the FBD, the equations of motion (EOMs) of the UAV need to be derived. The FBD can be seen in Fig. 1.

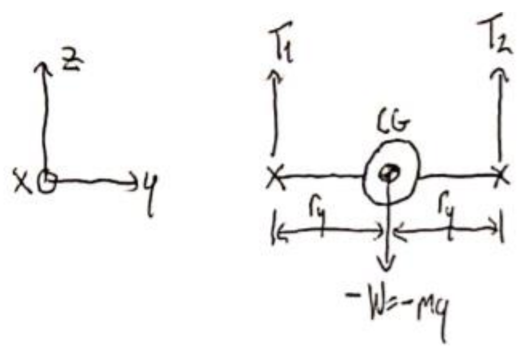


*Figure 1: A free-body diagram of the UAV with datum*

From the FBD, the EOMs are derived for the UAV by using Newton's Second Law of Motion. The EOM for the UAV is:

$$m\ddot{x} = f + gu \qquad (1)$$

Where $m$ is the mass of the UAV, $\ddot{x}$ is the acceleration of the UAV, $f$ is the weight of the UAV, $g$ is the unknown fault experienced by the UAV, and $u$ is the control law. Because the MRAC uses a model of the system as a reference, a reference model of the UAV needs to be developed. A reference model of the UAV is:

$$\ddot{x}_m = -2\zeta\omega_n\dot{x}_m - \omega_n^2 x_m + \omega_n^2 r \qquad (2)$$

Where $\ddot{x}_m$ is one state of the model, $\dot{x}_m$ is another state of the model, $x_m$ is another state of the model, $\zeta$ is the dampening factor for the UAV, $\omega_n$ is the natural frequency of the UAV, and $r$ is the reference signal input into the model. With the model reference and EOM of the UAV developed, the pseudo control law, control law, and adaptive control law need to be developed. The pseudo control law is:

$$v = \omega_n^2(r - x) - 2\zeta\omega_n\dot{x} - \frac{f}{m} \qquad (3)$$

Where $x$ is a measured state of the UAV and $\dot{x}$ is another measured state of the UAV. The control law is:

$$u = \hat{a}v \qquad (4)$$

Where $\hat{a}$ is the mismatch between the model and UAV due to faults experienced. Finally, the adaptive control law is:

$$\dot{\hat{a}} = -sign\left(\frac{g}{\omega_n^2 m}\right)ev \qquad (5)$$

Where $sign$ is a function that returns -1 if the value is smaller than 0, 0 if the value is equal to 0, or 1 if the value is greater than 0 and $e$ is the error from the measured state of the UAV subtracted from the model state.

## 3 Simulation Results:

For the simulation of the MRAC for the UAV, the initial conditions and actual parameters of the UAV are as follows: the initial state of the UAV is 0 meters, the initial state of the model is 0 meters, the initial value for the fault is 0.5, the initial mismatch between the model and UAV is 1, the mass of the UAV is 1 kilogram, the weight of the UAV is 9.81 newtons, and the reference signal input is 10 meters. With these parameters, the simulation is simulated for 200 seconds.

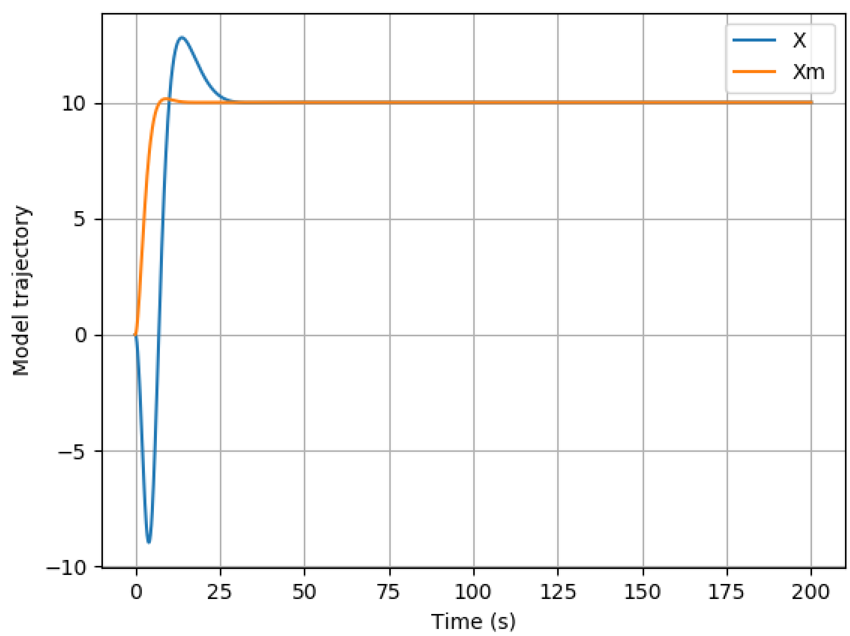


*Figure 2: The model and measured trajectory plotted against time*

In Fig. 2, the model and actual trajectory plotted against time. The model trajectory shown in orange starts at the initial state of 0 meters and increases to the reference signal input of 10 meters. There is a slight overshoot the model trajectory experiences showing the model is underdamped. The measured trajectory shown in blue starts at the initial state of 0 meters and decreases before increasing to the reference signal input. This decrease is due to the faults experienced by the UAV. The UAV experiences overshoot before settling at the value of the reference signal input. This overshoot shows the UAV is underdamped. The model and UAV both settle at the value of the reference signal input showing there is no steady state error, and both the model and UAV are stable.

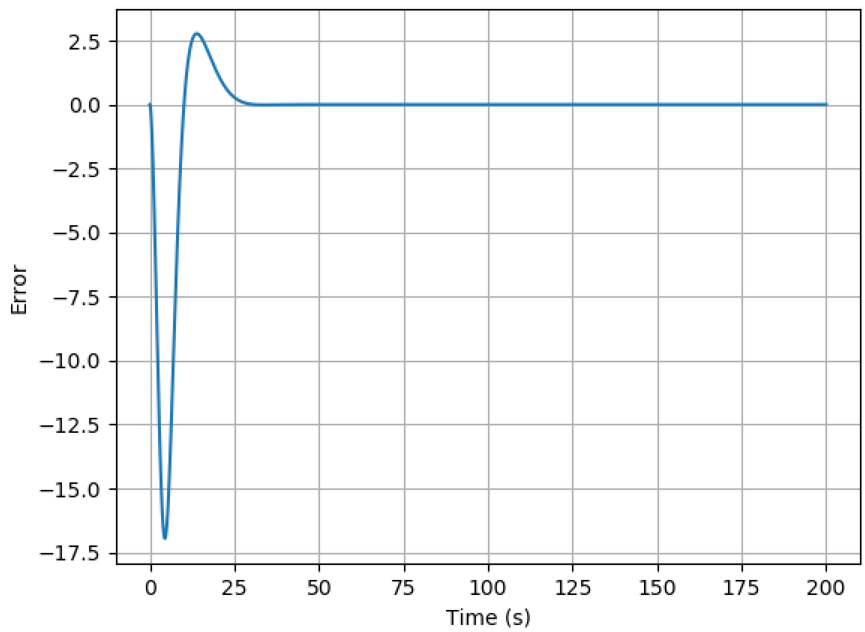


*Figure 3: The error between the measured state and the model state plotted against time*

In Fig. 3, the error between the measured state and the model state plotted against time. The error initially starts at 0 because both the model and UAV start at an initial state of 0. Since the UAV experiences faults that cause the model and measured states to deviate, the error is shown to decrease before increasing and settling at a value of 0. Because the error settles at a value of 0, the model and UAV both settle at the same value as shown in Fig. 2.

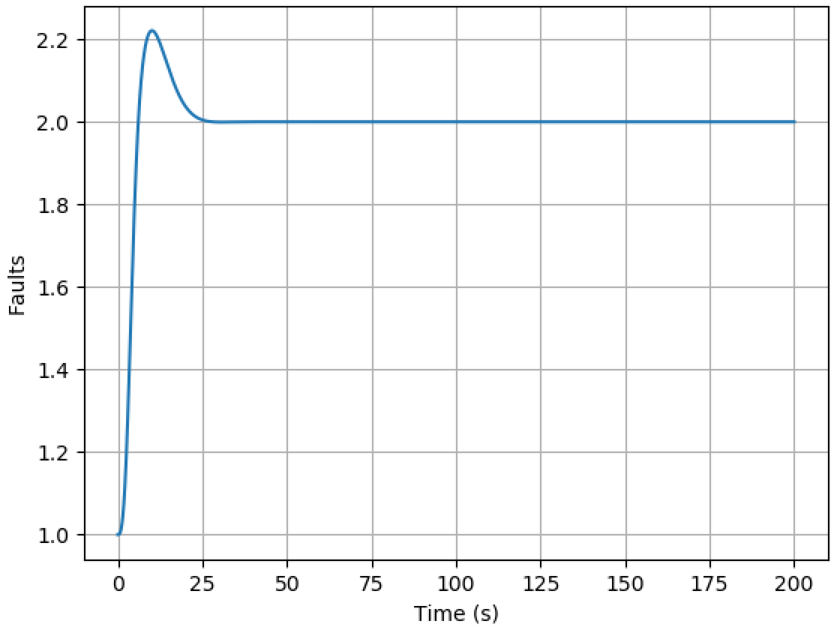


*Figure 4: The mismatch between the model and UAV due to faults experienced plotted against time*

In Fig. 4, the mismatch between the model and UAV due to faults experienced plotted against time. The mismatch is a factor at which the model states and UAV states misalign. For this project, the mismatch can be calculated by dividing the mass of the UAV by the initial fault experienced. This makes the mismatch to be a value of 2. Since the value for the mismatch settles at a value of 2, the system is shown to be working properly.

## 4 Conclusion:

The stability of a system is an important performance specification of a control system because it allows for the system to have a bounded output for a given bounded input. A model reference adaptive controller (MRAC) is a control system that adapts to a controlled system with unknown parameters. A MRAC works by using a model of the system as a reference to the control system on the actual system. For this project, a MRAC is used to estimate faults that an unmanned aerial vehicle (UAV) may experience during flight. As seen in Fig. 2, the modeled states of the UAV and actual states of the UAV both end at the reference signal input despite the actual states of the UAV experiencing faults during operation. Because both model and actual states settle at the reference signal input, neither experience steady state error. Both the model and actual states experience overshoot making them both underdamped, however with no steady state error or infinite oscillations, the system is stable.

## 5 Appendix:

### 5.1 Code for the MRAC on the UAV:

```
import numpy as np

import matplotlib.pyplot as plt

import scipy.integrate as I
```

```
def Derivatives(state,t):
    ##States
    x = state[0]
    xdot = state[1]
    xm = state[2]
    xmdot = state[3]
    ahat = state[4]
    ###Parameters
    f = -9.81 #N
    g = 0.5 #Represents half a fault
    m = 1.0 #kg
    r = 10. #m
    ###Error Signals
    e = x - xm
    ###Model Dynamics
    Ts = 10.0
    zed = 0.8
    wn = 4.6/(Ts*zed)
    xmddot = -2*zed*wn*xmdot - (wn**2)*xm + (wn**2)*r
    ###Control
    v = (wn**2)*(r - x) - 2*zed*wn*xdot - (f/m)
    gamma = 0.001
    ahatdot = -gamma*np.sign(g/((wn**2)*m))*e*v
    u = ahat*v
    ###Plant Dynamics and Kinematics
    xddot = (f/m) + (g/m)*u
    return np.asarray([xdot,xddot,xmdot,xmddot,ahatdot])
plt.close("all")
##Time vector
tout = np.linspace(0,200,10000)
```

```
##Initial Conditions
x0 = 0.
xdot0 = 0.
xm0 = 0.
xmdot0 = 0.
ahat0 = 1.
state_initial = np.asarray([x0,xdot0,xm0,xmdot0,ahat0])
state_out = I.odeint(Derivatives,state_initial,tout)
###Extract my states
xout = state_out[:,0]
xmout = state_out[:,2]
ahatout = state_out[:,4]
plt.figure()
plt.plot(tout,xout,label='X')
plt.plot(tout,xmout,label='Xm')
plt.xlabel('Time (s)')
plt.ylabel('Model trajectory')
plt.legend()
plt.grid()
plt.figure()
plt.plot(tout,xout-xmout)
plt.xlabel('Time (s)')
plt.ylabel('Error')
plt.grid()
plt.figure()
plt.plot(tout,ahatout)
plt.xlabel('Time (s)')
plt.ylabel('Faults')
plt.grid()
plt.show()
```

### 5.2 References:

[1] Franklin, G. F., Powell, J. D., and Emami-Naeini, A., *Feedback control of dynamic systems*, Upper Saddle River, NJ: Pearson, 2020.
[2] Slotine, J.-J. E., and Li, W., *Applied nonlinear control*, Taipei: Prentice Education Taiwan Ltd., 2005.