16.31 Project Report
Direct Model Reference Adaptive Control

Oliver Jia-Richards

December 8, 2017

1 Problem Statement

The goal of this project is to implement a direct model reference adaptive control (MRAC) scheme on a Parrot Rolling Spider minidrone to adapt the quadcopter's altitude controller. The altitude controller will initially start as an open-loop controller and be adapted to closed-loop.

For the adaption, the mass of the quadcopter is assumed to be unknown. Therefore, this scheme can also be used to adapt to midflight changes in the quadcopter mass or changes in the output thrust of the rotors.

2 Direct MRAC

Model reference adaptive control (MRAC) allows for the control of an unknown plant. In direct MRAC the controller parameters are updated such that the plant follows a reference system but the plant parameters are never estimated. This is in contrast to indirect MRAC where the plant parameters are estimated in order to update controller parameters [1].

MRAC has been used to account for nonlinearities in control design for helicopters [2] and for in-space assembly of satellites [3] where the plant parameters change as the satellite is assembled.

In both cases, MRAC updates the controller through comparing the system response to the response of a reference system. For this project, a direct MRAC scheme was implemented. Direct MRAC was chosen over indirect MRAC for implementation simplicity.

To learn how to implement a direct MRAC scheme we will start with a simple scalar case and then move to a simplified form of the quadcopter altitude dynamics. Using the results of the simplified altitude dynamics, the direct MRAC scheme will then be implemented on the full quadcopter model and tested in simulation and then in experiment.

2.1 Scalar Case

To begin with direct MRAC lets start with a simple scalar case of the form

$$\dot{x} = ax + bu \tag{1}$$

where only the sign of b is known. We then define a stable reference model

$$\dot{x}_{\text{ref}} = a_{\text{ref}} x_{\text{ref}} \tag{2}$$

that we would like our system to converge to. If we consider a full state feedback controller of the form

$$u = -kx \tag{3}$$

we can rewrite Equation (1) as

$$\begin{aligned}\dot{x} &= ax - bkx \\ &= (a - bk)x \end{aligned} \tag{4}$$

If our system does converge to the reference model then we know, based on Equations (2) and (4), that

$$a_{\text{ref}} = a - bk \tag{5}$$

It is clear that if a_{ref}, a, and b were given there has to exist some k that would solve Equation (5). Unfortunately, we only know a_{ref} and the sign of b. Therefore, we cannot calculate k before flying and instead must determine it online.

Since k cannot be calculated the implement controller is

$$u = -\hat{k}x \tag{6}$$

where $\hat{k}$ is the estimate for the ideal k. Equation (1) can then be written as

$$\begin{aligned}\dot{x} &= ax - b\hat{k}x \\ &= (a_{\text{ref}} + bk)x - b\hat{k}x \\ &= a_{\text{ref}}x - b(\hat{k} - k)x \end{aligned} \tag{7}$$

Clearly, if the estimated controller gain matches the ideal controller gain then the system will behave as the reference system. In addition, the tracking error of the system versus the reference system

$$e = x - x_{\text{ref}} \tag{8}$$

the derivative of tracking error can be written as

$$\begin{aligned} \dot{e} &= a_{\text{ref}} x - b(\hat{k} - k)x - x_{\text{ref}} \quad (9) \\ &= a_{\text{ref}} e - b(\hat{k} - k)x \quad (10) \end{aligned}$$

which further shows that if the estimated controller gain matches the ideal controller gain then the tracking error will asymptotically approach zero as a_{ref} was chosen in order to produce a stable system.

The only remaining step is to determine how to update $\hat{k}$ such that it converges to k. An inverse Lyapunov design approach is used in [4] and [5] to select

$$\dot{\hat{k}} = -\gamma x e \ \text{sgn}(b) \tag{11}$$

where γ is a chosen constant. It is shown in [4] that if such an update scheme for $\hat{k}$ is chosen then the tracking error will asymptotically approach zero.

To test this scheme a simple case was implemented in MATLAB Simulink with

$$\begin{aligned} a &= 1 \\ b &= 1 \\ a_{\text{ref}} &= -1 \end{aligned}$$

Based on the matching condition in Equation (5) the ideal gain for this system is $k = 2$.

Figure 1 shows the tracking performance of the scalar direct MRAC for an initial gain of $\hat{k} = 0$ and $\gamma = 5$. Initially, the system is unstable and does not follow the reference signal. However, over time the adaption scheme corrects the gain and by $t = 30$ the system and reference match quite closely.

Figure 2 shows the controller gain over time. The adaption scheme initially overshoots the ideal gain but over time the controller gain converges to the ideal gain. Changing γ will alter the convergence of the controller gain. Picking γ too high will increase the initial overshoot but picking γ too low will cause the convergence to be extremely slow.

Although in this situation the controller gain converged to the ideal gain this is not necessarily always the case. Returning to the gain update scheme in Equation (11) we can see that there are two situations in which $\dot{\hat{k}} = 0$. The first is that the tracking

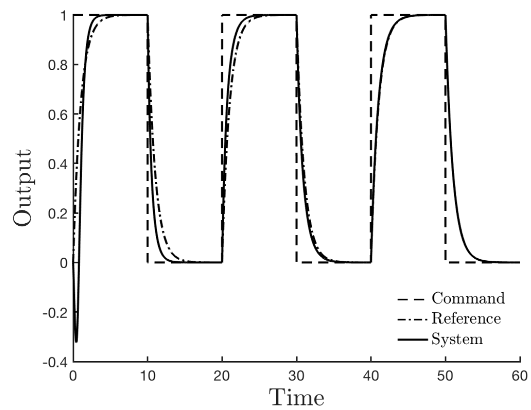

Figure 1: Scalar MRAC tracking performance.

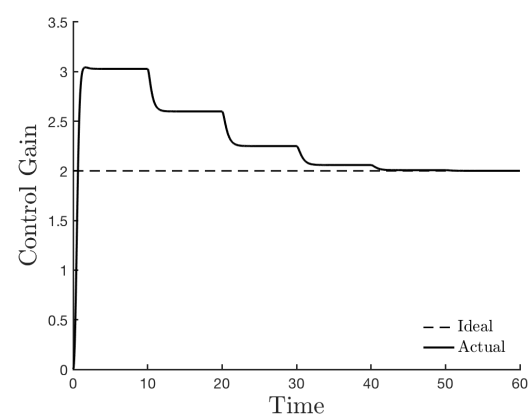

Figure 2: Controller gain for scalar MRAC.

error equals zero. The second is that the system is at equilibrium, $x = 0$.

If the system were to start at equilibrium then $\dot{\hat{k}} = 0$ for all t and the controller gain would not converge to the ideal value. This would also occur if the system converged to equilibrium faster than the reference system. Although $e \neq 0$, $x = 0$ and therefore the gain would not be updated.

This phenomena can be seen in Figure 2. The overshoot in controller gain from the adaption scheme causes the system to be faster than the reference system. Therefore, the system converges to the commanded value faster than the reference system and pauses any updates to the controller gain. It is only when the commanded value is changed that the controller gain is updated due to the transient in the system when both the tracking error and state error are non-zero.

2.2 Simple Altitude Dynamics Case

To use direct MRAC on the Rolling Spider minidrone the scalar case will have to be extended for use in systems with multiple states. To test the multi-dimensional case a simplified version of the quadcopter altitude dynamics was used. The governing ordinary differential equation for this system is

$$m\ddot{z}(t) = T(t) - mg \tag{12}$$

where T is the combined thrust from all four rotors, m is the quadcopter mass, and g is gravitational acceleration.

Taking the input function to be

$$u(t) = T(t) \tag{13}$$

the linearised form of the simplified quadcopter altitude dynamics is

$$m\ddot{z}(t) = u(t) \tag{14}$$

Two states are required to represent this system: the altitude, $z(t)$, and the vertical velocity $v(t)$. Using these two states results in the following state space system

$$\begin{bmatrix} \dot{z} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} z \\ v \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} u \tag{15}$$

where the time dependance of all variables, except for the mass, is implied.

Similarly to the scalar case, we assume that the A and B matrices are unknown but we do have knowledge of the "sign" of B. To account for this the B matrix is expressed as

$$B = B'\Lambda \tag{16}$$

where B' represents the known control matrix and Λ represents unknown parameters. For this case

$$B' = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$\Lambda = \frac{1}{m}$$

As before, we define a stable reference model

$$\dot{\mathbf{x}}_{\text{ref}} = A_{\text{ref}}\mathbf{x}_{\text{ref}} \tag{17}$$

and consider a full-state feedback controller

$$\mathbf{u} = -K\mathbf{x} \tag{18}$$

We can then rewrite our state-space model from Equation (15) as

$$\begin{aligned} \dot{\mathbf{x}} &= A\mathbf{x} - B'\Lambda K\mathbf{x} \\ &= (A - B'\Lambda K)\mathbf{x} \end{aligned} \tag{19}$$

From Equations (15) and (19) we know that if our system matches the reference system then

$$A_{\text{ref}} = A - B'\Lambda K \tag{20}$$

Since Λ is unknown, the ideal controller gain cannot be calculated. Therefore, the full-state feedback control that we implement is

$$\mathbf{u} = \hat{K}\mathbf{x} \tag{21}$$

where $\hat{K}$ is the estimated controller gain. This allows the state-space model in Equation (15) to be rewritten as

$$\begin{aligned} \dot{\mathbf{x}} &= A\mathbf{x} - B'\Lambda\hat{K}\mathbf{x} \\ &= (A_{\text{ref}} + B'\Lambda K)\mathbf{x} - B'\Lambda\hat{K}\mathbf{x} \\ &= A_{\text{ref}}\mathbf{x} - B'\Lambda(\hat{K} - K)\mathbf{x} \end{aligned} \tag{22}$$

As in the scalar case, if the estimated controller gain matches the ideal controller gain then the system will behave as the reference system. The tracking error can be defined as

$$\mathbf{e} = \mathbf{x} - \mathbf{x}_{\text{ref}} \tag{23}$$

the derivative of the tracking error can be written as

$$\begin{aligned} \dot{\mathbf{e}} &= A_{\text{ref}}\mathbf{x} - B'\Lambda(\hat{K} - K)\mathbf{x} - A_{\text{ref}}\mathbf{x}_{\text{ref}} \\ &= A_{\text{ref}}\mathbf{e} - B'\Lambda(\hat{K} - K)\mathbf{x} \end{aligned} \tag{24}$$

which shows that if the estimated controller gain matches the ideal controller gain then the tracking error will asymptotically approach zero as A_{ref} was chosen in order to produce a stable system.

The last step is to define an update scheme for $\hat{K}$ such that it converges to K. As with the scalar case, an inverse Lyapunov design approach is used in [4] and [5] to select

$$\dot{\hat{K}} = -\Gamma x e^T P B' \tag{25}$$

where Γ is a chosen matrix of constants and P is the solution to the algebraic Lyapunov equation

$$P A_{\text{ref}} + A_{\text{ref}}^T P = -Q \tag{26}$$

where Q is another chosen matrix of constants.

It is shown in [4] that if $\dot{\hat{K}}$ is chosen this way then the tracking error will asymptotically converge to the origin.

To test this case it was implemented in MATLAB Simulink with

$$A_{\text{ref}} = \begin{bmatrix} 0 & 1 \\ -2 & -2 \end{bmatrix}$$

which places the closed loop eigenvalues at $-1 \pm i$. Based on the matching condition from Equation (20) the ideal K is

$$K = \begin{bmatrix} 2m & 2m \end{bmatrix} \tag{27}$$

Figure 3 shows the tracking performance of direct MRAC for the simple altitude dynamics case with

$$\hat{K}_0 = \begin{bmatrix} 0 & 0 \end{bmatrix}$$
$$\Gamma = \begin{bmatrix} 1.6 & 0 \\ 0 & 0.9 \end{bmatrix}$$
$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 0.9 \end{bmatrix}$$

We can see that the system initially tracks the reference system quite poorly. However, over time the system converges to the reference system.

Figure 4 shows the controller gain over time. The adaption scheme initially causes the derivative gain to overshoot the ideal gain but over time both the proportional and derivative gains converge to the ideal gain of $2m$.

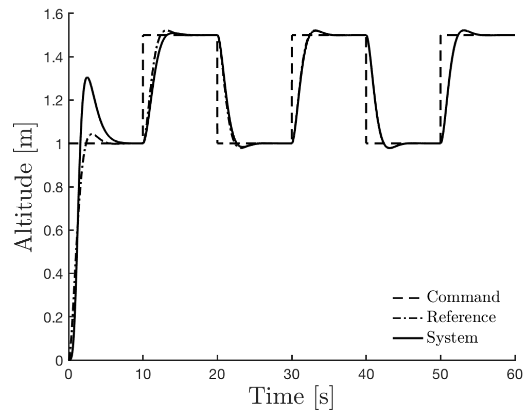

Figure 3: Direct MRAC tracking performance for simple altitude dynamics.

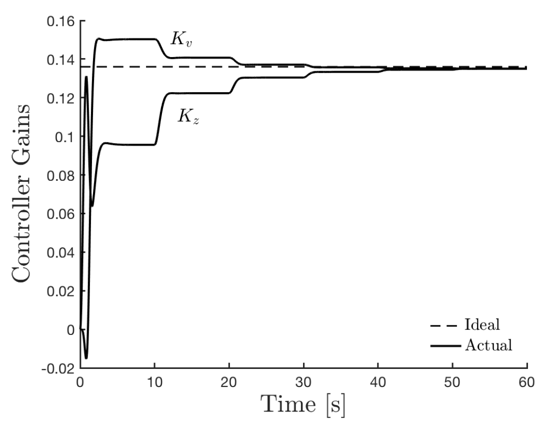

Figure 4: Controller gain for direct MRAC implemented on simple altitude dynamics.

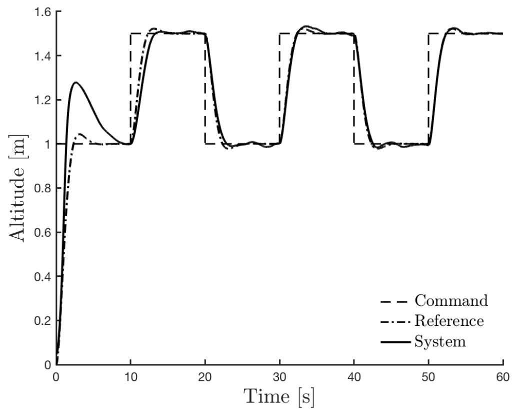

Figure 5: Direct MRAC tracking performance for full quadcopter model in simulation.

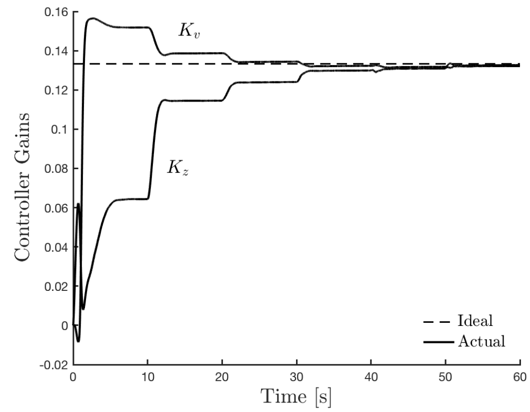

Figure 6: Controller gains for direct MRAC implemented on full quadcopter model in simulation.

3 Implementation

The direct MRAC scheme derived in the simplified altitude dynamics case was then implemented on the full quadcopter model in MATLAB Simulink. As with the simplified case, the parameters for the adaption scheme are

$$\hat{K}_0 = \begin{bmatrix} 0 & 0 \end{bmatrix}$$
$$\Gamma = \begin{bmatrix} 1.6 & 0 \\ 0 & 0.9 \end{bmatrix}$$
$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 0.9 \end{bmatrix}$$

This effectively starts the altitude controller as open-loop and the adaption scheme closes the loop by updating the controller gains.

3.1 Simulation

Figure 5 shows the tracking performance for the direct MRAC scheme implemented on the full quadcopter model. Compared to the simplified altitude dynamics seen in Figure 3 the full model performance matches quite well. The main differences occur from $t = 0$ to 10s.

This difference is most likely due to the full quadcopter model using a feedforward control for the take-off rather than the feedback controller. Therefore, although the controller gains are still being updated, the system response is not changing.

Figure 6 shows the controller gains for the direct MRAC scheme on the full quadcopter model. Again, besides from $t = 0$ to 1s where a feedforward takeoff controller is being used, the gain adaption is quite close to the simplified altitude dynamics case seen in Figure 4

3.2 Experiment

Figure 7 shows the tracking performance for the direct MRAC scheme implemented on the Rolling Spider minidrone. We can see that there is a noticeable difference between the experimental performance and the simulation performance for the first 30 seconds of flight. From $t = 30$ to 40s the experimental performance resembles both the simulation performance and the reference system.

The difference between the experimental data and simulation data is most likely due to disturbances on the physical system that are not accounted for in the simulation. These would include differences in the estimated and actual mass, variance in rotor thrust output, and the effect of battery voltage drops.

This hypothesis is supported by the controller gains seen in Figure 8. The proportional and derivative gains both converge to the same value. However, this value is higher than the ideal gain expected based on our provided reference model. This result indicates that to achieve the same behaviour as the reference system, the actual system has to provide more control authority. Two factors that could cause this are the quadcopter mass being higher than expected or the rotors outputting less thrust than expected.

Of note is that both the proportional gain and derivative gain converge to the same value. This is consis-

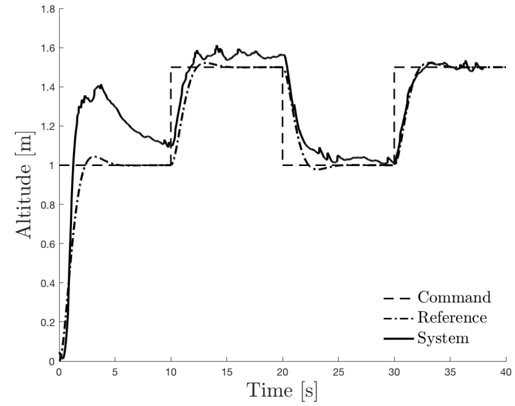

Figure 7: Direct MRAC tracking performance for Rolling Spider minidrone.

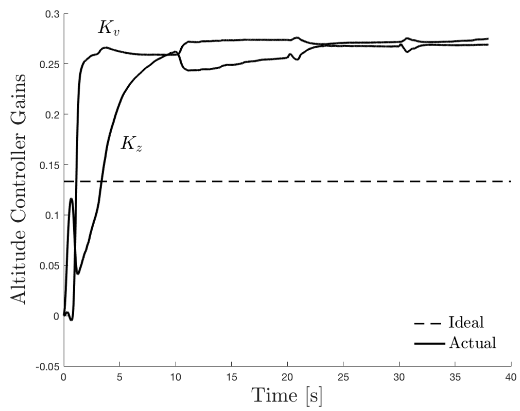

Figure 8: Controller gains for direct MRAC implemented on Rolling Spider minidrone.

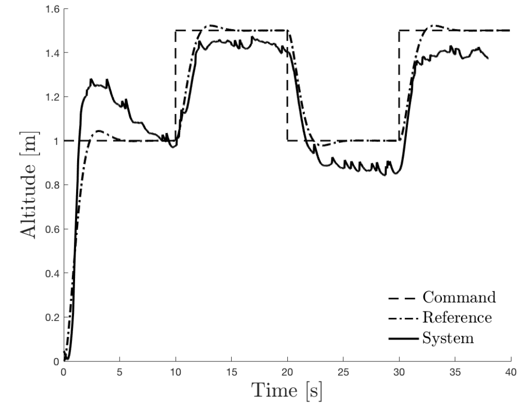

Figure 9: Direct MRAC tracking performance for "pre-drained" battery test.

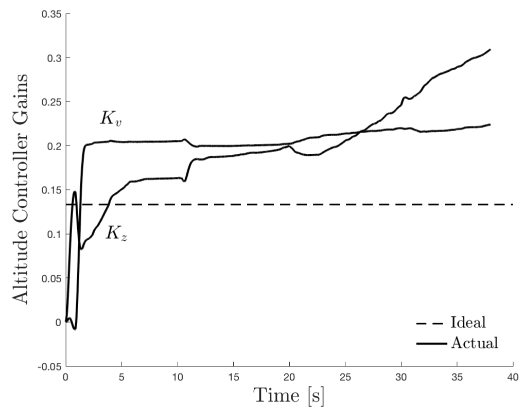

Figure 10: Controller gains for direct MRAC for "pre-drained" battery test.

tent with the hypothesis that the quadcopter mass is higher than expected or that the rotors are outputting less thrust than expected. Referring back to Equation (27), the ideal gains are both equal to $2m$. Therefore, if the mass of the quadcopter is higher, the ideal gains should be higher. Alternatively, if the rotors were outputting less thrust then that has the same effect as having a larger quadcopter mass and we would expect the same behaviour.

To see the effect of drained battery voltage further experiments were run using the same battery. This means that the battery voltage at the start of the test was the same as the battery voltage at the end of the test seen in Figure 7. The tracking performance for this "pre-drained" test can be seen in Figure 9. We can see a steady state error in the system that appears to grow during the test as the battery voltage drops further and further.

This ha a dramatic affect on the adaptive controller. Figure 10 shows the controller gains for this "pre-drained" test. We can see that towards the end of the test the proportional gain is continuously growing. This is due to the steady state error in the system. Since both the state error and tracking error are non-zero, even at equilibrium, the adaption scheme will constantly try to increase the proportional gain.

Based on the original and "pre-drained" test we can see that the direct MRAC scheme behaves quite well in the face of uncertainty in certain parameters such as the quadcopter mass or the rotor thrust output. However, there are some disturbances such as the battery voltage that cause the direct MRAC scheme to behave poorly. The controller gains no longer converge to an expected value but rather continuously increase over time due to the steady state error induced into the system.

References

[1] Narendra, K. S. and Valavani, L. S., "Direct and Indirect Model Reference Adaptive Control," *Automatica*, Vol. 15, 1979, pp. 653–644.

[2] Subramanian, R. G. and Elumalai, V. K., "Robust MRAC augmented baseline LQR for tracking control of 2 DoF helicopter," *Robotics and Autonomous Systems*, Vol. 86, 2016, pp. 70–77.

[3] Espinoza, A. T. and Saenz-Otero, A., *Probabilistic and Learning Approaches through Concurrent Parameter Estimation and Adaptive Control for In-Space Robotic Assembly (Masters Thesis)*, MIT, 2017.

[4] Lavretszky, E. and Wise, K. A., *Robust and Adaptive Control*, Springer-Verlag, London, 2013.

[5] Hokayem, P. A. and Gallestey, E., *Adaptive Control*, ABB Switzerland Ltd., 2015.