A brief[1] How-To on Lab circuitry for the Kitticopter

This doc will hopefully guide you on how to integrate circuits with the kitticopter lab setup. As you've already done the pre-prac, you should have a rough model of the system as well as a block diagram of how it's structured. If your block diagram looks a little different to mine don't worry (okay maybe worry a little if it looks nothing like mine and please go and ask a tutor to check that everything is fine).

Please note that this is simply a guide or a series of hints for how to go about constructing the circuits required for this lab. If you want to take the circuit design thinking out of the lab, then you can copy my circuit templates and do so. If you want to make your own adjustments or have your own approach, feel free to do so – both cases will need you to put your circuit diagrams and a picture of your constructed circuit in your final lab report. Also, you should recognise all the circuits I show here as you've covered them previously in your studies. They're compiled here to save you some revision time.

Part 1: System ID and a step test

The basic setup

We have the following block diagram to represent the system we're working with:

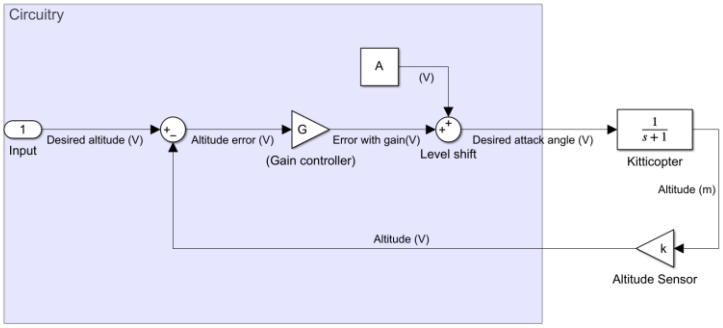


Figure 1: Block diagram of the kitticopter system showing where external circuitry is involved

Note that the level shift here is to counteract gravity within the kitticopter system. Also that's an arbitrary function representing the kitticopter system – not the actual TF!

The part that involves our circuitry (breadboard and then Veroboard) is shown in the big blue box. With this we need to show the interfaces with the kitticopter simulation via DACs and ADCs:

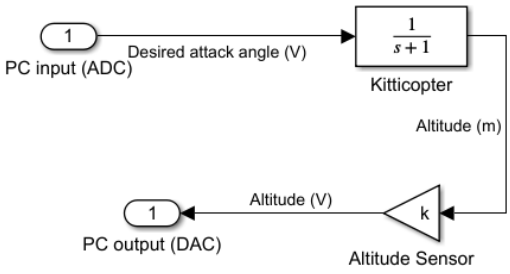


Figure 2: A diagram showing the PC interfacing with the kitticopter

[1] Whoops, this is quite long

We need to construct the rest of the setup ourselves via breadboard or Veroboard to interface with the kitticopter and do our step test as well as having closed-loop control.

Note that the ADC and DAC have input/output limits of ± 10V, and the kitticopter simulation only accepts inputs of 0-5V.

We also have a power supply available to us with +15V and -15V (non-adjustable).

Step-testing

Now that we know how to interface with the system, we can set up a basic step test. As there's no feedback loop for this section we only need to be concerned with the input (we'll be logging the output digitally).

For the step test we want to go from a baseline voltage *V1* which is when the helicopter is in the air and stationery and step to *V2*. The step size isn't important if we have met the following conditions:

1. We know our step size so we can normalise our measured output.
2. *V2* ≤ 5V, as above this our input saturates and what the ADC reads and what the kitticopter receives are two different values.

Condition 1 should be easy as we'll be recording both input and output values as we do the test.

Condition 2 requires some circuitry thinking as we must balance having a starting point of *V1* and not have our step go over 5V while using a supply of 15V. Note that exceeding a 5V input is viable if you're clever about it.

You're given access to potentiometers in the control lab and resistors are available from white lab, so we can construct the following step testing circuit.

Circuit 1: A simple pot voltage divider

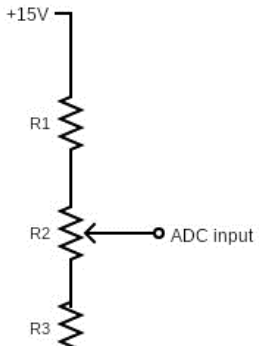


Figure 3: Pot based step test circuit

A standard voltage divider circuit. This circuit varies output voltage between $15\frac{R3}{R1+R2+R3}V$ and $15\frac{R2+R3}{R1+R2+R3}V$. Simply figure out the range you want (and what potentiometers are available), plug, and play.

To perform a step test simply swing the potentiometer from the lowest setting to the highest.

THIS IS THE END OF THE INFORMATION NEEDED FOR THE SYSTEM ID PART OF THE LAB

Part 2: Closing the loop and adding proportional control

It's now time for some feedback control. We need to construct a circuit that covers all the functionality required from Figure 1. As some of you may have feared, this means opamps, several opamps. Those of you adept in the analogue arts can manage this circuit with just a single opamp, and although this will gain you no extra marks it will gain you some bragging rights and my respect.

This tutorial/appendix/whatever we end up calling it will rather use multiple opamps to more easily show what we're up to.

Once again referring to Figure 1 we require the following components for our feedback loop:

1. An input voltage
2. A subtracting junction for the reference voltage and the feedback voltage
3. A gain amplifier (our controller)
4. A summing junction for the level shift

5. A level shift voltage

1 and 5 are simply solved by a potentiometer circuit and a voltage divider respectively, but 2, 3, and 4 require opamp functionality.

Subtracting junction (Differential amplifier)

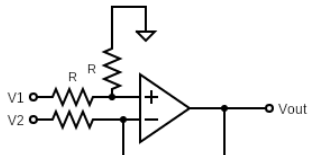
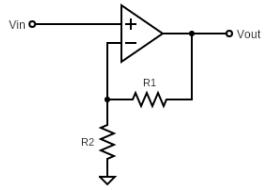


Figure 4: Differential amplifier

This circuit subtracts voltage according to the function $V_{out} = V_2 - V_1$ provided all resistors are equal. (They probably won't be, but more on that later)

Gain amplifier (noninverting amplifier)



Bog-standard noninverting amplifier. $V_{out} = V_{in}(1 + \frac{R_1}{R_2})$. Note that this only allows for gains above unity – if you need to have gain < 1, you'll need to alter your circuit to include an inverting amplifier (which you'll later invert again)

Figure 5: noninverting amplifier

Combining everything

Here's an example of how to put the whole setup together. In this example I've opted to use a differential amplifier for the level shifter (by just inverting the level, e.g., if you wanted to add 3V you'd input -3V instead), as this eliminates the hassle of summers inverting signals.

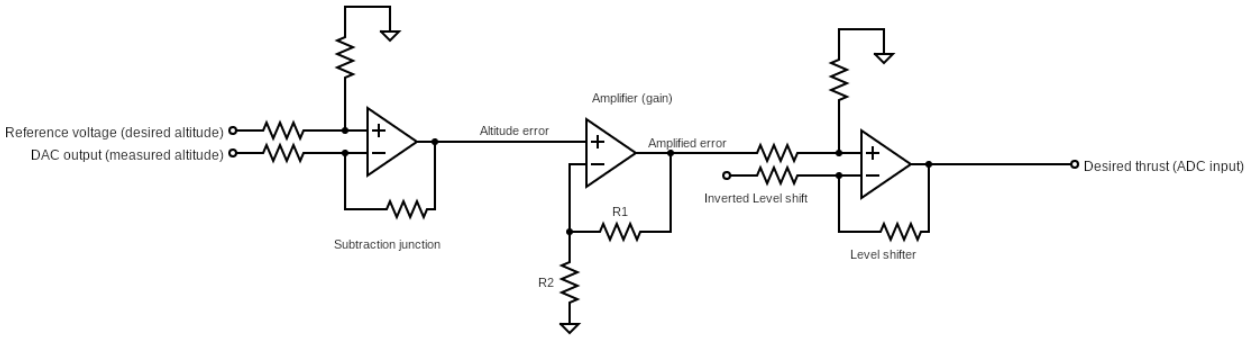


Figure 6: Example of an overall feedback circuit

A note on component tolerances

When constructing these circuits, you're bound to run into accuracy issues. This will show up with amplifiers with incorrect gains, voltage dividers giving incorrect errors, or any number of other ways. This is expected.

Most of the time this will be due to the tolerance of the resistors used. The standard resistors found in White Lab have a 10% tolerance, so the actual value will be between 90% and 110% of the listed value. This 10% explodes with opamp circuits, for example with a gain amplifier, the desired gain can vary between $\frac{0.9R_1}{1.1R_2} = 0.82\ \frac{R_1}{R_2}$ and $\frac{1.1R_1}{0.9R_2} = 1.22\ \frac{R_1}{R_2}$, which is now an almost 20% tolerance. Look at the 4 resistors involved in a differential amplifier, and you can imagine that being pedantic about accuracy makes your life very difficult.

I have 3 suggestions for how to deal with this:

Firstly, approach this problem with the specifications in mind – if it works it works. Your boss/client/instruction sheet asks for performance within a boundary. If the product works within said boundary you've succeeded in your task. If you want to spend extra time to upgrade it feel free, a better controller is still ideal, but don't sink your time trying to find the perfect resistor.

Secondly, pick what's worth fussing about. Controller not working? Tinker and measure around your circuit and figure out where the errors are happening, and where they matter. For example, with the level shifter you can either spend all day finding four matching resistors for the differential amp, or you can tune the level you're inputting to compensate.

Thirdly, control what you can control. The resistors have a 10% tolerance, sure, but you have potentiometers at your disposal, feel free to tune them to a 'perfect' value and glue them or whatever[2]. Additionally, there are fantastic bench multimeters available in White Lab where you can measure a set of resistors and find one closest to the desired value[3]. I'm sure there are other ways to add accuracy here but remember the second suggestion and try not get carried away being a perfectionist.

[2] I have heard wild stories regarding the methods professional technicians have fixed potentiometers in place – try top them!
[3] This can deteriorate into gambling very fast – I take no responsibility.