PDE 2420 project

2012-2013

One-degree-of-freedom helicopter control system design

Objective: This assignment is designed to test your knowledge and understanding of control systems.

You will be working in teams of two. You may chose who you wish to work with.

Your task is to design and build a one-degree-of-freedom helicopter. A motor with rotor will be attached to one end of a lever and the other end will have a counter-balance. The lever will pivot on a potentiometer which will provide feedback of the current position of the lever. You will use an Arduino as an I/O device to read the potentiometer and send it over the serial port to the host PC. The host PC will send the Arduino a value for the pulse width modulated output which will drive the motor via a MOSFET transistor.

The PID control system will be implemented in Simulink and will read and write values to the Arduino. Live data from the system will be displayed in Matlab.

You will be provided with:

- Motor & rotor
- 2x Potentiometers
- MOSFET
- Schematic and wiring diagram for the components and the Arduino
- Arduino code and Matlab/Simulink configuration sufficient to get you started

You should use the Arduino you already have.

Tasks:

1. Design and build the mechanical parts of the system. The base should solid and stable and the lever should be free to rotate around the potentiometer. Bear in mind the spinning rotor blades pose a safety risk so design a housing to enclose them as much as possible.
2. Put together the whole system as guided by your tutors. Ensure that all the parts are working correctly.
3. Carry out tests in order to write a project report detailed below.
4. Present your findings in an oral presentation and demonstration.
5. Prepare a video suitable for youtube/vimeo.

Project report

Your project report should follow this structure:

1. Title page, including your names and student numbers and module code and name
2. Introduction
3. Description of design and build of rig. Justify design choices. Include Solidworks rendering and 2D drawings. (Guideline page count: 5 pages + drawings)
4. Development of PID control system (Guideline page count: 20 pages including graphs)
    a. Describe (in engineering terms) the performance of the system when only P control is used. What happens over a range of P values?
    b. Describe (in engineering terms) the performance of the system when PD control is used. What changes compared to just P control? Select constants for P and D coefficients and justify your choices.
    c. Describe (in engineering terms) the performance of the system when PID control is used. How has the addition of the Integral term affected the behaviour of the system?
    d. Tune the coefficients for the PID control system. Justify the choices of the coefficient values. Test the performance of the system in a variety of situations including step changes in the desired input and external disturbance (someone knocking the lever). Describe and characterise the performance of the system using graphs and engineering terms (overshoot, damping, etc).
5. Conclusion

Assessment

The marks for the assessment will be divided as follows:

10% on the quality of build. Is the rig well designed? Is it stable? Is it safe? Does it allow free movement of the lever without excessive play?

65% on the project report.

15% on the project presentation and demonstration. Powerpoint presentation of no more than 10 minutes illustrating the key findings of your team. This will include a live demonstration of the system.

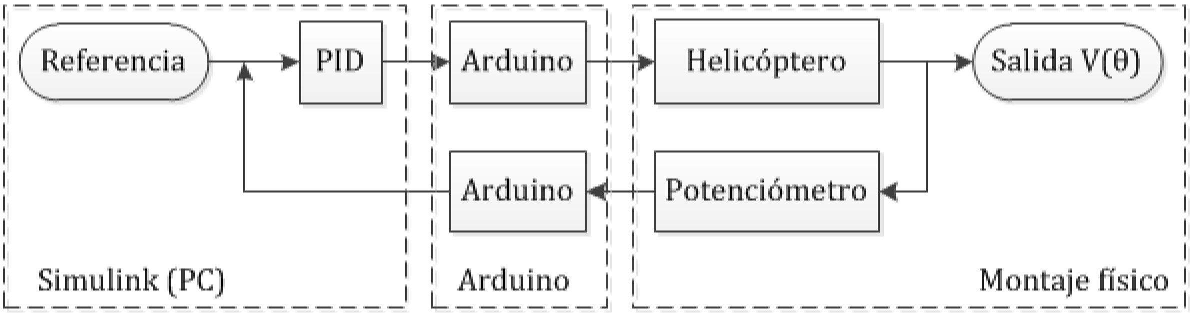
10% on a video illustrating the results of your project (not the build process of the rig). You should prepare a video of your system describing its behaviour. It should be in a suitable format for youtube or vimeo. (An example of an excellent student video is: http://vimeo.com/54337988)

Deliverables and Deadlines

Project report to be submitted electronically via Turnitin on OASISplus by Thursday 17th January at 12:00 noon. Presentation on Friday 18th January at 9:00am in C105 (normal teaching session).

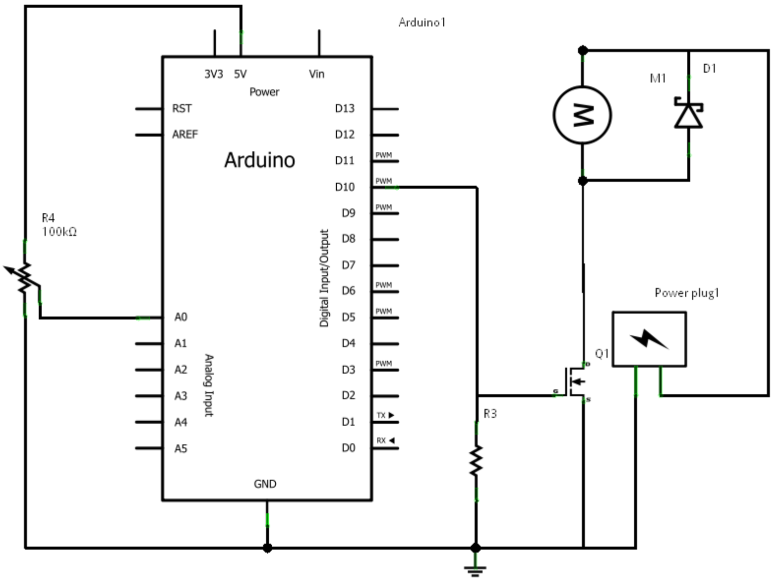
Project details

One-degree of freedom helicopter is an articulated arm with a motor at one end, connected to a propeller which provides more or less lift in function of the voltage supplied to the motor. In the joint's axis potentiometer returns a voltage $V(\theta)$ between 0V and 5V, depending on the angle θ . The control system scheme is as follows:

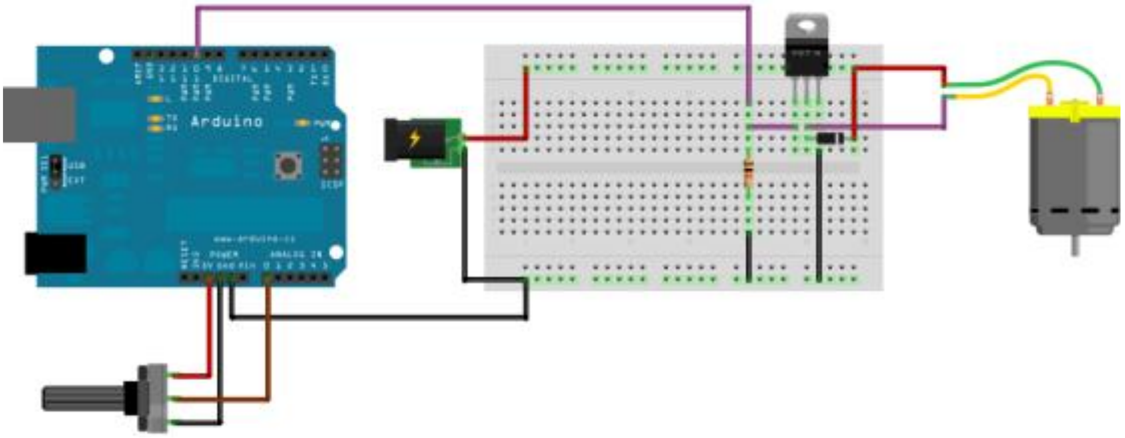


Thus, the **output variable** would be the **axis position** (voltage transmitted by the potentiometer) and the **control variable** would be the **lift propeller** (voltage provided to the motor). Therefore, a **control loop**, an **input variable** and a **physical interface** is required.

Physical interface: Arduino + PWM power circuit

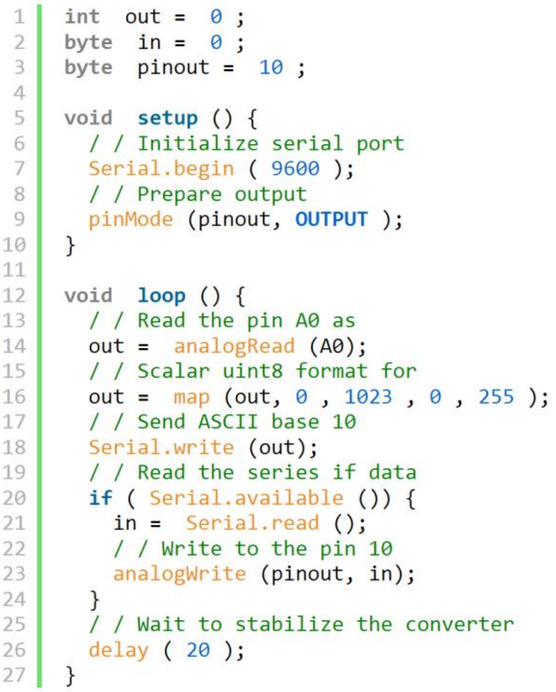


The Arduino is unable by itself to provide the amperage needed by the engine. Hence a power circuit consists of a MOSFET transistor is required. The microcontroller sends the PWM signal to the Gate of the transistor, and thus the same PWM signal will be received by the motor.



Arduino: reading and writing the physical variables

The Arduino should be able to (1) read the potentiometer $V(\theta)$; (2) send the data through the serial port the designed PID controller; (3) receive the new value provided by the PID controller from the serial port; (4) write the data in analog output. Note that the A/D converter returns a value between 0 and 1024 and needs to be mapped to 0 and 255 in order to send a single *unsigned* bit.



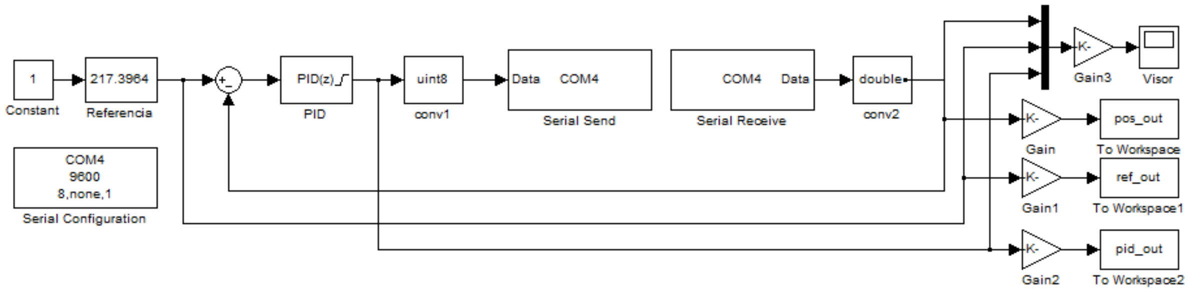
```
int  out =  0 ;
byte  in =  0 ;
byte  pinout =  10 ;

void  setup () {
  / / Initialize serial port
  Serial.begin ( 9600 );
  / / Prepare output
  pinMode (pinout, OUTPUT );
}

void  loop () {
  / / Read the pin A0 as
  out =  analogRead (A0);
  / / Scalar uint8 format for
  out =  map (out, 0 , 1023 , 0 , 255 );
  / / Send ASCII base 10
  Serial.write (out);
  / / Read the series if data
  if ( Serial.available ()) {
    in =  Serial.read ();
    / / Write to the pin 10
    analogWrite (pinout, in);
  }
  / / Wait to stabilize the converter
  delay ( 20 );
}
```

Simulink: reference and control loop

The Simulink program must receive the value of $V(\theta)$ through the serial port, compare to a reference voltage V_{ref}. Then a PID controller determines the control signal (motor power) required. This signal will be sent through the serial port in uint8 format, *8-bit unsigned integer* that takes values between 0 and 255, ideal for analogWrite () command in Arduino.



Within the PID block, you can edit the parameters P, I and D to achieve the optimum result. Watch the following video to see the overall performance of a model system, designed by Jorge García Tíscar : http://www.youtube.com/watch?feature=player_embedded&v=ttsKhNZZTHU

You can see how using the *slider* adjusts the reference (purple signal). The system, by adjusting the signal power delivered to the motor (blue signal) is adjusting the angle θ of the shaft until the voltage of the potentiometer $V(\theta)$ (yellow signal) matches the reference that has been identifed.

Matlab: post-processing of the data collected

It can be seen from the Simulink model that not only the data is presented in the display but also stored in the Matlab workspace. This allows us to post-process the results of the monitoring or export the data to Excel. Consider this feature for your future work.