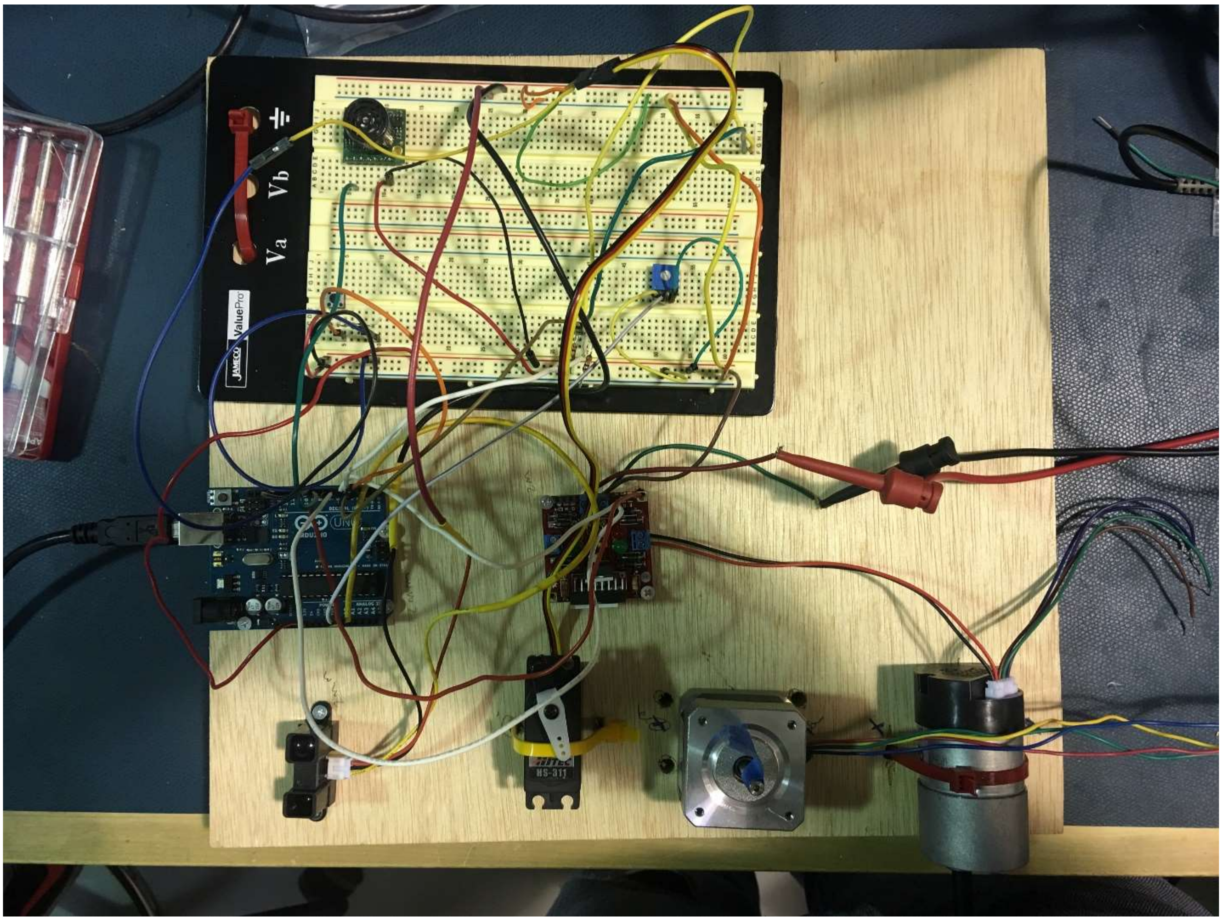
Fly Sense



Team C – ILR 1

Joao Fonseca Reis

13th October 2017

Individual Progress: Sensors and Motors Assignment

Shivang and Nick took the lead on the Sensors and Motors assignment, and instrumental for the success of this assignment. We tried to segment the work inside the team in a way that would be most efficient.

In that sense, at the beginning of the week I was focusing more on the case preparation for Friday class (Introduction to Robotics Business) and would join in the work on Monday.

Unfortunately, I spent more time than I wanted to finalize my Biomechanics homework and could not do too much until Tuesday.

Teamwork

I was supposed to do one of the motors as specified in the lab assignment description. However, by the time I could (finally) help my teammates, that was not the most urgent/important thing to do. The biggest challenge that we were having was not getting more things done but instead get them to work with each other properly. I than decided (for better or for worst) that instead of following the instructions of the assignment blindly, I should instead work to the success of the team.

In the end I worked a bit with almost everybody in the group, focusing mostly on making everything work together properly:

- With Hari: we did a set of paper pieces the size of the different components (they were being used in tests by the other team members), discussed the distribution in the wood board, got the holes drilled in the wood shop and fixated the components has they were released from testing

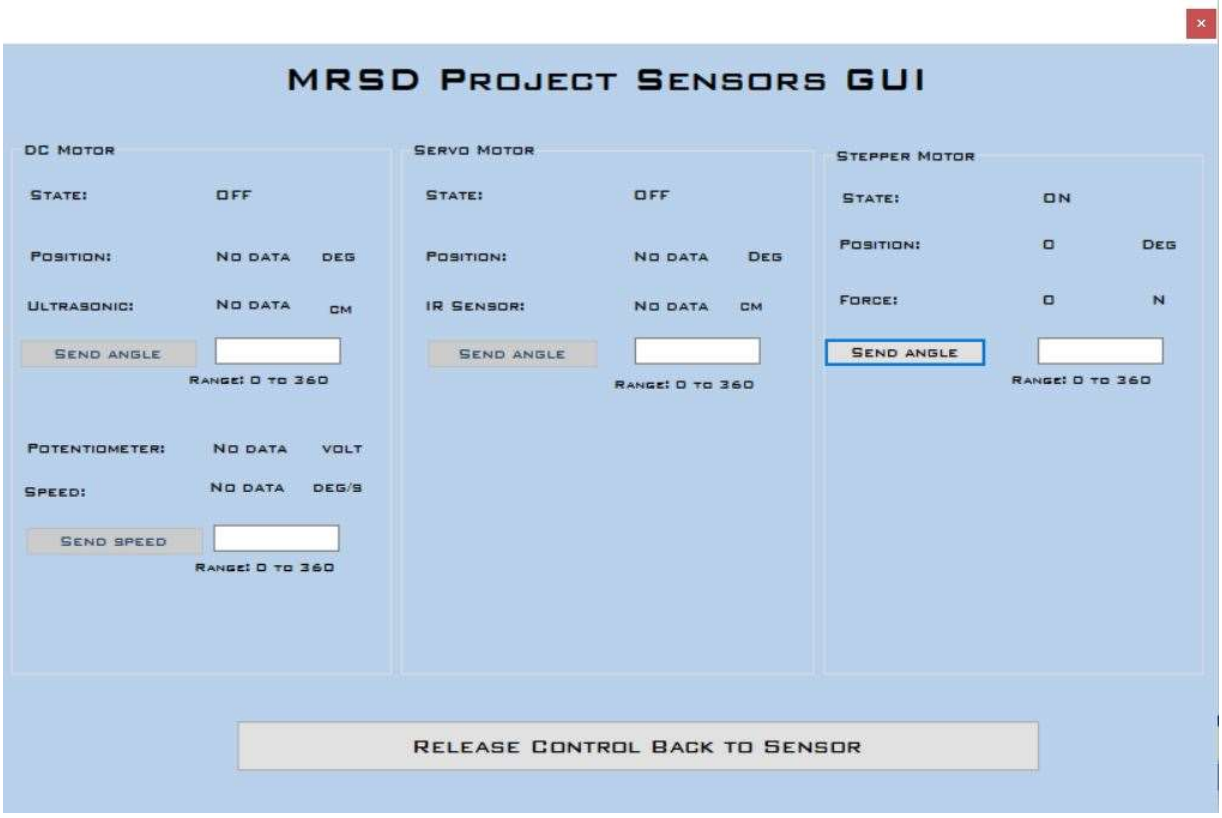
- With Nihar: we organized the screen consistently with the functions to make sure it was intuitive and well organized. Many requirements were removed (e.g. graphs that were making the GUI very slow with no material benefit), the code was redone and debugged jointly:
    - The screen was broken into three sections, one per motor (with the DC engine having two sub-sections: speed and position control, the other motors only position control)
    - The motors and the sensor readings were grouped together so that it was very intuitive to know which sensor was working with each motor
    - A simple text field with a send button was introduced to send commands to the motors in a simple way (manual override)
    - A release button was introduced to give the control back to the sensor after a manual override control had been sent

- With Nihar and Shivang: the interfaces between the GUI and the Arduino controller were made more resilient.
    - Garbage control was implemented at the GUI entry point so that non-compliant input would not abort the GUI. This was done at the GUI given the availability of higher level language (C#) and more computational power (laptop vs Arduino)

  - The input from the side of the Arduino was simplified to avoid heavy string processing before sending data to the GUI (e.g. the GUI would accept sensor readings of with one, two or three digits removing the addition of 00s to the left from the Arduino side)
  - The manual override command fields had filters installed so that non-complaint commands were not sent to the Arduino (that could otherwise hang or crash)
  - The bit transfer rate used to communicate between the Arduino and the GUI were optimized to ensure smooth transfer of data between the two systems

- With Shivang: we worked together on fine tuning the PID controllers for the manual override commands execution, as well as the sensor reaction calibration.

Figure 1: GUI interface



MRSD Project Sensors GUI

DC Motor
State: OFF
Position: No data deg
Ultrasonic: No data cm
Potentiometer: No data volt
Speed: No data deg/s

Servo Motor
State: OFF
Position: No data deg
IR Sensor: No data cm

Stepper Motor
State: ON
Position: 0 deg
Force: 0 N

I then initiated the preparation of the presentation with the description of how our motors/sensors prototype worked and its test guide. I also followed up with the rest of the team on the draft I prepared with the current project status.

While I was doing this, the other team members gradually covered all sensors and motors. Shivang and Nick did again a very good job in integrating the code Hari and Nihar had written for the different sensors and motors.

With hindsight I believe I decided correctly on what my priorities should be: ensuring things would fit together nicely, with an intuitive user interface and reliable enough that would not break down. But I feel I could have contributed more if, in addition, I had also owned a specific task end-to-end.

Current Project Progress and Future Plans

The project is currently broken down into five major blocks: Augmented Reality, User Interface, Sensing, System Integration and Testing, Project Management.

Key take aways so far:

1. Augmented reality: we have initiated Hololens testing and are now looking at how to tackle the difficulties arising from the current (lack of) maturity of this solution.

   Last Sunday Nihar and I initiated the preparation of an AR demo for the internal team. One of the tests that Nihar came up with was going to Robolounge, setting the volume into the maximum, put the noise of an helicopter in the background and test the Hololens voice recognition commands.

2. Sensing: Direct LIDAR (Velodyne) readings need to be processed in order to generate a “proper” map that can be presented to the pilot.

   Hari has initiated Velodyne testing with a borrowed Velodyne. This was an invaluable source of information and will help us guide the design of our prototype going forward.

In the annex, I present the detail of the project progress so far.

Annex

Augmented Reality Head Display: Nihar (primary owner)

- **Critical Milestone for PR1**: First visualization of sensor suite

- **Achievements so far**
  - Air Lab Hololens borrowed being for testing

- **Next step:** Test other AR headsets Finalize mock-up of Heads-up Display

Hololens testing: Pros	**Hololens testing: Cons**
• Easy to build applications • Excellent head tracking (gaze) • Resilience to lighting conditions (overlaid reality) • Excellent 3D Audio (with <u>no background noise</u>) • Excellent Voice Commands (with no background noise) • Works both in wireless and wired mode (voice commands only work online)	• Feels heavy after using it by 30/40 minutes • The headset can move in the head and the hologram loses the correct alignment • Hand gestures tracking does not work well enough to interact with tool • Voice commands do not work with helicopter background noise – tested in Robolounge) • Voice commands only work with online connectivity • Limited field of view (-30 to 30 degrees)

User Interface: Joao (primary owner)

- **Critical Milestone for PR1**: First draft of UI/UX

- **Achievements so far**
  - Analyzed the performance of Hololens in hand tracking, voice tracking and voice recognition identified gaps with against the required "ideal" performance

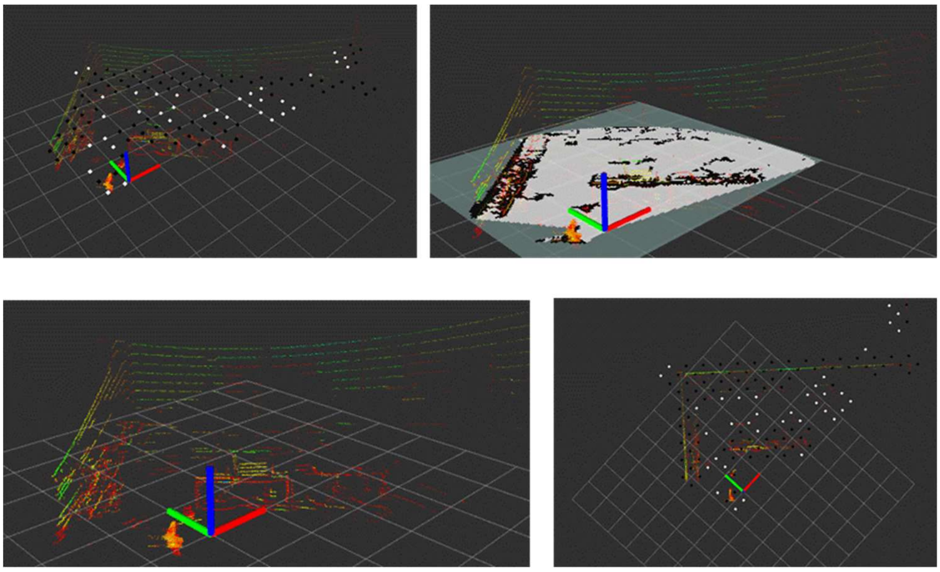
- **Next step:** Meet with CMU experts
  - Jean Oh: Meeting held today (good inputs received on Jean's User Interface experience in the Atlas program)
  - Aaron Steinfeld: meeting scheduled for next week
  - Richard Stern pending confirmation

Sensing: Hari (primary owner)

- **Critical Milestone for PR1**: Use data from Lidar data set to create and visualize point cloud

- **Achievements so far**
  - Lab LIDAR borrowed for preliminary testing (mapping functionality is needed)

- **Next step:** Improve maps produced
  - Retrieve 3D point cloud samples from NEA (gathered today afternoon)
  - Bridge with CMU/NEA experts on how to process Velodyne inputs into mapping



System Integration and Testing: Shivang and Nick (primary owners)

This stream is not part of the requirements for the Fall Validation Experiment. We are currently using our laptops and focusing on testing the key sub-systems (sensing and AR display).

Special care is currently being put into selecting sub-systems that can deliver well in the conditions present in the cockpit of a helicopter.

Jetson TX1 IMU is currently available for inclusion in the final prototype, but is not currently being worked on.

Project Management (all team)

Critical tasks are ensuring access to all assets needed to deliver the project. Given that we cannot by all the needed products and services (e.g. a helicopter to test the prototype), we have reached out to NEA.

We have signed a two-sided NDA agreement and hope the sponsorship agreement is ready by next week so that we can plan our work with them.

With NEA on board, things that were previously at risk can now materialize:

a) Testing with a quadcopter is now much easier (leveraging their expertise)
b) Testing with a helicopter is now possible
c) Invaluable support can be received in processing sensor data (e.g. LIDAR)

We will also have additional budget leeway to acquire different AR solutions and test them thoroughly, thus ensuring that we pick the most adequate sub-systems for our prototype.

Without NEA, there was a risk that we would pick the sub-systems based solely on the trade studies. While we tried to be as much fact based as possible on these, they still carry a lot of “opinions” that can only be confirmed/disproved after testing with a real unit.

Task 7 (Sensors and Motor Control Lab) Quiz

1. **Reading a datasheet. Refer to the ADXL335 accelerometer datasheet (https://www.sparkfun.com/datasheets/Components/SMD/adxl335.pdf) to answer the below questions.**

   - **What is the sensor's range?**

   Minimum specifications: Accelerations from Min = -3 g to Max = 3 g (with g=9.81 m/s^2)

   However, different accelerometers coming out of the factory may have higher operating ranges due to random inputs in the fabrication process. According to the datasheet, the typical range is from -3.6 g to +3.6 g (thus it is convenient to do a calibration test for the actual sensor being used).

   The calibration test would thus allow us to better derive the sensor's transfer curve: actual output voltage versus actual experienced acceleration.

   - **What is the sensor's dynamic range?**

   The minimum dynamic range is: Max measurable acceleration – Min measurable acceleration = 3g-(-3g) is 6 g (in order words: in this case is 2* maximum measurable acceleration before saturation occurs).

   The data sheet indicates that the window of measurable accelerations is always symmetric and centered around zero, regardless of the actual range of that specific sensor (that will change from sensor to sensor). This means the span or dynamic range will be always 2*maximum absolute accelerate that a specific sensor coming out of the factory can measure (the actual value will come from the calibration of the actual sensor being used).

   - **What is the purpose of the capacitor C_{DC} on the LHS of the functional block diagram on p. 1? How does it achieve this?**

   Stabilize the input voltage to the sensor to remove the ripple on the supply voltage.

   - **Write an equation for the sensor's transfer function.**

   The transfer function is:

   V_Bias + Sensitivity * Acceleration

For Z_{OUT}, x_{OUT}, y_{OUT}, in the typical case:

V_Bias = 1.5, Sensitivity = .3 (per each g of acceleration)

and so the transfer function is:

V_Output = 1.5 + .3 * g's (acceleration measured in g's)

- **What is the largest expected nonlinearity error in g?**

The largest expected non-linearity is +-.3 % of the full scale. This means that the error of the output signal will be within [-1,+1] * 3/1000 * (Vmax-Vmin) .

For the x_{OUT}, y_{OUT} axes: Vmax – Vmin = 1.65-1.35 = .3 and the error is -+ 0.0009 V. This converts into an error of 0.0009 / .05 = 0.018 g measuring the input signal.

For Z_{OUT} axis: Vmax – Vmin = 1.8-1.2 = .6 and the error is -+ 0.0018 V. This converts into an error of 0.0018/.1 = .018 g measuring the input signal.

The calculations above are made for the minimum specification in range (-+3g). They will have to be recomputed for each sensor using its actual range (typically -+3.6 g as per the datasheet).

- **How much noise do you expect in the X- and Y-axis sensor signals when the sensor is excited at 25 Hz?**

For x_{OUT}, y_{OUT} the typical noise level is 150 micro g / sqrt (Hz).

Thus, for 25 Hz the expected noise level should be = 150 * 10^-6 * Sqrt(25) = 0.00015 * 5 = .001 g {when measuring in g's}.

- **How about at 0 Hz? If you can't get this from the datasheet, how would you determine it experimentally?**

The datasheet only refers offsets incurred due to temperature variations (-+ .01 %/C) but does not discuss the impact of a noise signal that is fixed across time. We could do a static test to measure its impact (in a scenario of no acceleration, constant speed which is difficult to achieve to static test, the only sensor measurement would be external noise).

2. Signal conditioning

o Filtering

- **What problem(s) might you have in applying a moving average?**

The moving average performs well removing high frequency noise of low amplitude. When the amplitude and frequency of the noise increases, the number of points in the moving average window needs to be increased to filter those high amplitude/frequency noise artifacts.

The higher the moving average window the increased latency of the signal and thus the ability to react real time to the external world will fade away as the size of the moving average window increases. One solution would be to weight more recent more, but that may reduce the stabilization capabilities of the filter.

- **What problem(s) might you have in applying a median filter?**

The median filter is more effective than the moving average filter removing isolated high frequency noises with large/medium amplitudes. The median filter works by sampling the signal during a time window, sorting the measurements taken and extracting the median of that interval (after sorting the median will be the value in the middle of the array containing the time window measurements).

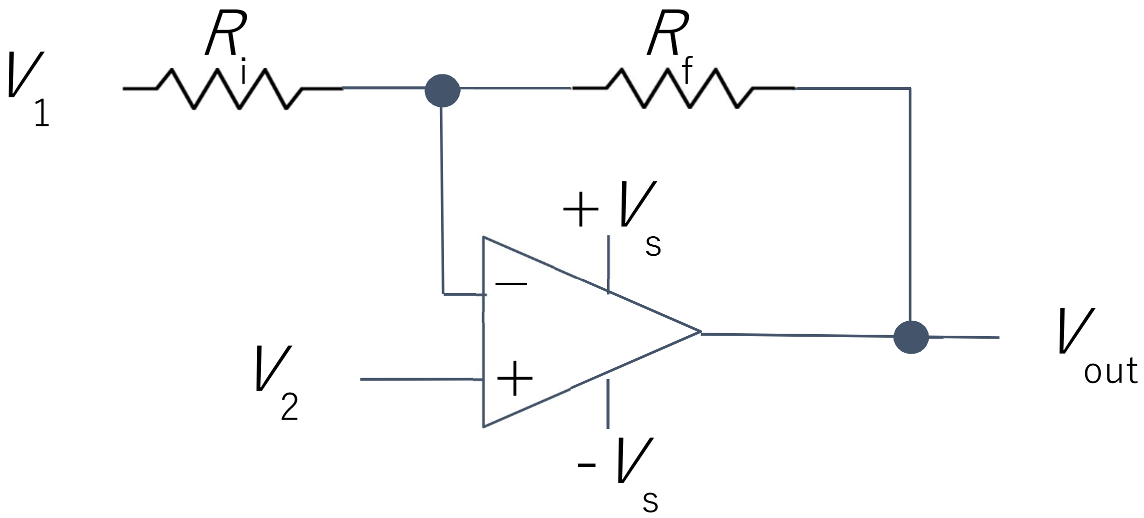
Thus, while the median filter performs very well removing isolated outliers from the measured data, it degrades its performance when the number of outliers increases. In particular, if for example two outliers are present on the measuring window time the median filter can "create" a measurement that does not make sense.

Computing a median based filter may also have a higher computation cost than computing a moving average based filter. The time interval over which the median is computed may be decreased, but in practical terms as the size of the window decreases the effectiveness of the median filtering decreases.

o Opamps

- **In the following questions, you want to calibrate a linear sensor using the circuit in Fig. 1 so that its output range is 0 to 5V. Identify which of V1 and V2 will be the input voltage and which the reference voltage, the value of the reference voltage, and the value of Rf/Ri in each case. If the calibration can't be done with this circuit, explain why.**

Fig. 1 Opamp gain and offset circuit



As per the equations reviewed in class:

$$V_{out} = (V_2 - V_1)\frac{R_f}{R_i} + V_2$$

- **Your uncalibrated sensor has a range of -1.5 to 1.0V.**

Selection made: V1 as the fixed reference voltage (value to be determined)

Given the limits of the sensor, substituting them in the equation above:

- Lower limit: Vout= (-1.5 – V1) * Rf/Ri – 1.5 = 0
- Higher limit: Vout= (1.0 – V1) * Rf/Ri +1.0 = 5

Computation yields: V1= -3 volts (fixed) and Rf/Ri = 1 (unitary gain). The ratio between the two resistances is computed but not the absolute value of the resistances, as intended.

Note: The actual value of the resistors could be determined based on the maximum current intended in a particular brunch of the circuit.

- **Your uncalibrated sensor has a range of -2.5 to 2.5V.**

This problem has no applicable solution. Most configurations will yield a mathematically impossible solution.

The closest we can get to a solution is trying an inverter:

- Lower limit: Vout= (-2.5 – V1) * Rf/Ri – 2.5 = 5
- Higher limit: Vout= (2.5 – V1) * Rf/Ri + 2.5 = 0

However, solving the system yields: V1= 1.25 volts (fixed) and Rf/Ri = -2 (double negative gain). The ratio between two resistances cannot be negative, so this configuration is mathematically possible but is impossible in “real life”.

3. Control

- **If you want to control a DC motor to go to a desired position, describe how to form a digital input for each of the PID (Proportional, Integral, Derivative) terms.**

Typical format of a PID controller

$$K_P e(t) + K_D \frac{d}{dt} e(t) + K_I \int_0^t e(x) d(x)$$

Each of the components of the PID controller is adequate for different types of situations:

	Proportional feedback	Integral Feedback (against proportional)	Derivative feedback (against proportional)
Rise time	Small	Bigger	Bigger
Settling time	Small	Bigger	Bigger
Overshooting	Potential issue	Bigger	Lower
Steady State error	Small	Not applicable	Small
Other	Same Input/output units	May saturate counters and circuits	Numerical instability due to noise levels.

A typical approach would be to gradually build the PID in the following order:

Create a proportional feedback for faster response

We would measure the deviation between the targeted angle and the current angle of the motor:

Error = (Angle_i – Angle_Desired)

We would then test several values of Kp to implement a proportional feedback sensor that would achieve the intended angle “fast enough”.

Introduce a derivative feedback for stability (damping)

The next step would be to test multiple levels of Kd to prevent the control from overshooting the targeted angle. As a cost the control would take longer time to settle/rise, but could in exchange have less/no material overshooting associated.

Introduce an Integrator feedback to eliminate steady-state error

The next step would be to test multiple levels of Ki to solve any steady state errors that may be identified.

- **If the system you want to control is sluggish, which PID term(s) will you use and why?**

If the system is slow to react (e.g. has inertia) the proportional feedback controller cannot be used without a derivative feedback component to prevent overshooting the target.

- **After applying the control in the previous question, if the system still has significant steady-state error, which PID term(s) will you use and why?**

The next step would be to introduce an integrator feedback to reduce the steady state error.

- **After applying the control in the previous question, if the system still has overshoot, which PID term(s) will you apply and why?**

If the system is still overshooting, the best course of action would be to increase the coefficient of the derivative feedback to further dump the oscillations around the target angle. Alternatively reducing the coefficient of the proportional feedback might be an option.