1. Hardware (0–100%)

• Robust construction

* Talk about how the bottom of the chassis is curved to maximise clearance
* Suspending the pcb in the middle of the chassis to maximise airflow and cooling and to center the mpu 6050 as close to where I estimated the center of mass would be
* A longer effective pendulum length (higher center of gravity) due to the larger battery allowed for lower control loop rates and was also found, in assignment 2, to be close to optimal ratio of 1:1 for this robot

• Cost effectiveness

Bit expensive, did not want to be limited by motor capabilities and wanted them quickly so bought from core rather than buying from china .

• Plant dynamics suitable for control loop rate

It was a large pendulum with a high center of mass and thus had a lower required control loop rate

• Plant design maximises available kinematic performance envelope

The center of mass is as high up as possible while keeping the robot compact, maximizing kinematic performance

• Plant design optimises fundamental control limitations

Limited by the waterbed effect, therefore had to optimize the plant to minise the bode sensitivity integral. Maximized effective pendulum length and minimized total mass as well as maximise the mass of the chassis vs the mass of the wheels

• Appropriate power electronics

The HC-05, motordrivers, current sensors and lcd were all the right tool for the job

2. Software (0–100%)

• Maintainability (e.g., serial interpreter uses command tables rather than if-else chains)

YES

• Code organisation (e.g., encapsulation, orthogonal functionality separated into C modules

(.c/.h), use data interfaces instead of global variables)

YOS

• Testing (e.g., coverage of unit tests, design for testing, TDD)

PROBS NOT, MAYBE

• Style (e.g., consistent indenting, function/variable naming convention)

YOS

• Documentation (e.g., appropriate use of code comments, online help via serial interface)

Implement function headers if time is there

• Safety features (e.g., watchdog, comms keepalive/timeout, unrecoverable actuator limits)

Chuck in some kill switches, a stop function etc

3. System identification (0–100%)

• Log experimental data using MCU

HECK YOS

• Use data to guide model structure selection

INDEED

• Fit actuator/plant models from gathered data (training set)

DONES

• Validate model fits against new data (validation set)

DONES

• Automation between Matlab and MCU

HECKIN YEA

Automate the results of the performance envelope to be the theta angle killswitch

4. Simulation model (0–100%)

• Sensor sampling and quantisation effects taken into account (e.g., encoder ISR, ADC)

Determine sample rate for MPU6050 and encoder and ADC

• Actuator model and control allocation

Heckin YOS

• Actuator limits (e.g., saturation and slew rate)

Chuck in a torque limiter in the model

• Model effect of inclined ground plane and transition

Modelled its effect by chucking in a “from workspace” block, did not control it

5. Control (0–100%)

• Model-based control design

Yos

• Robustness to disturbances (e.g., integral action and/or disturbance estimation + feedforward

cancellation)

nope

• Design consideration for actuator limits (e.g., anti-windup scheme for integral states,

closed-loop bandwidth)

• Account for uncertainty in plant structure/parameters and validate robustness via MonteCarlo

simulation

• Appropriate controller dynamics for implementation at control loop rate

Hindsight :

Motor

Robust:

* Equivalent components could have been sourced much cheaper from china
* Use stepper motors
* Designed a chamber for my battery so it would stay securely in place, was originally ziptied and is now superglued as well
* Better pcb design (tx to rx and the plethora of unused motor driver and lcd pins) and better testing to prevent fried components

REPORT STRUCTURE

Model based design

– Show how the mathematical model was derived and used, with the design constraints, to design

your mechanical components, choose sensors and actuators and design the controller.

– Mathematical models and discussion (backed by results) about validity and limitations.

– Divide and conquer – how the problem was broken down into smaller sub-problems and how they

were solved. For example, separation of controller and control allocation.

– Why do we force an integrator in the output of the controller when using a stepper motor?

• Design for testing

– What steps were taken to ensure that each subsystem was performing correctly and robustly?

• System identification

– Characterisation of actuators – were these as the manufacturer claimed?

– Experiment design

– Model calibration – structure, parameters and validation.

• Conclusion

– Reflect on the work done

– What would you do differently in terms of design if you could start again?