|  |  |
| --- | --- |
| Design Document & Report for MCHA3000's Project | University of Newcastle  University Drive  Callaghan NSW 2308  Australia |

|  |  |
| --- | --- |
| **Report Number:** |  |
| **Revision:** | . 1 |

|  |  |
| --- | --- |
| **Course** | MCHA3000 |
| **Project:** | Project |
| **Robot Type:** | Type I |

|  |  |  |  |
| --- | --- | --- | --- |
|  | *Name* | *Student ID* | *Contact* |
| **Prepared by:** | Christopher Caelli | C3206246 | C3206246@UON.EDU.AU |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | *Name* | *Role* | *Signature* | *Date* |
| **Approved:** |  |  |  |  |
| **Authorised:** |  |  |  |  |

|  |  |
| --- | --- |
| **FILENAME:** | **DesignDoc\_MCHA3000.docx** |

Change History

|  |  |
| --- | --- |
| Rev No. | Change Details |
| 1.0 | Initial Issue |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Distribution

Maintained copies of this document are only provided to organisations or individuals formally registered as a recipient for distribution of this document. All paper copies of this document are uncontrolled unless otherwise indicated on the front cover by a red stamp.

This document will be re-issued in full following any amendments

Executive Summary

* Defines the intention of the report.
* Places the report in context so the reader knows why it is important to read it.
* Why was the reported work undertaken?
* Why is it important?
* What problem is addressed?
* Briefly states the results.
* Briefly presents the implications and recommendations.

Table of Contents

[1. Introduction 6](#_Toc497927467)

[1.1. Identification & Objective 6](#_Toc497927468)

[1.2. System Overview 6](#_Toc497927469)

[2. Applicable Documents (Call out documents in reference to each stage) 7](#_Toc497927470)

[2.1. Project Documents 7](#_Toc497927471)

[2.2. Other Documents 7](#_Toc497927472)

[3. Abbreviations 7](#_Toc497927473)

[4. Definitions 7](#_Toc497927474)

[5. Overview of Required Work 9](#_Toc497927475)

[6. Model Based Design 12](#_Toc497927476)

[ 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. 12](#_Toc497927477)

[ Mathematical models and discussion (backed by results) about validity and limitations 12](#_Toc497927478)

[ 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 12](#_Toc497927479)

[6.1. Approach 12](#_Toc497927480)

[6.2. Results 12](#_Toc497927481)

[7. Design for testing 12](#_Toc497927482)

[What steps were taken to ensure that each subsystem was performing correctly and robustly? 12](#_Toc497927483)

[7.1. Approach 12](#_Toc497927484)

[7.2. Results 12](#_Toc497927485)

[8. System Identification 12](#_Toc497927486)

[ Characterisation of actuators – were these as the manufacturer claimed? 12](#_Toc497927487)

[ Experiment design 12](#_Toc497927488)

[ Model calibration – structure, parameters and validation 12](#_Toc497927489)

[8.1. Approach 12](#_Toc497927490)

[8.2. Results 12](#_Toc497927491)

[9. Conclusion 12](#_Toc497927492)

[Summarise the results 12](#_Toc497927493)

[reflect on the work presented 12](#_Toc497927494)

[make recommendations 12](#_Toc497927495)

[suggest future work or improvements. 12](#_Toc497927496)

[9.1. Reflect on the work done 12](#_Toc497927497)

[9.2. What would you do differently in terms of design if you could start again? 12](#_Toc497927498)

[10. Notes. 12](#_Toc497927499)

[11. Appendixes. 12](#_Toc497927500)

List of Figures

No table of figures entries found.

List of Tables

No table of figures entries found.

1. Introduction
   1. Identification & Objective

This report is written to document the design decisions, progress and achievements of the MCHA3000 Project. This document covers the all aspects of the work done throughout JUNE17-NOV17. It will focus on the detailed design of the Type-I cart robot named and analyse the results of the project.

* 1. System Overview

The MCHA3000 Project & prerequisite assessments seek to

“introduces the process of mechatronic system

design. It is a project-based course where a mechatronic system

for an electromechanical component is designed and built. The

course integrates tools and skills related to computer and

software, electronics, control, modelling and simulation. It also

develops the concepts of experimental modelling and

implementation of computer control systems. The course

provides a real-life experience related to the practice of

mechatronics engineering.”.

It achieves this through a personal project requiring the listed skills to achieve the end task: Produce a robot that can balance a pendulum. As such,

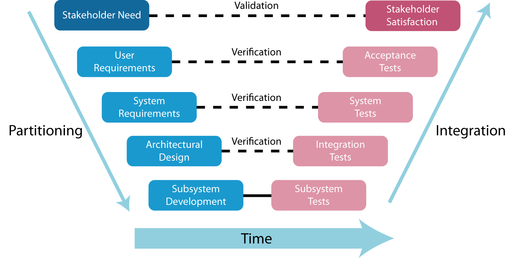


Figure 1 Work-Breakdown-Structure

[TODO: Add work breakdown structure here]

*[This paragraph shall briefly state the purpose of the system and the software to which the document applies. It shall describe the general nature of the system and software; summarise the history of system development, operation, and maintenance; identify the project sponsor, acquirer, user, developer, and support agencies; identify current and planned operating sites; list other relevant documents.]*

***V-Diagram here***



1. Applicable Documents (Call out documents in reference to each stage)

The following documents at the latest issue unless otherwise specified form a part of this document to the extent specified herein. In the event of conflict between the documents referenced herein and the contents of this document, the contents of this document shall be considered a superseding requirement.

*[This section shall list the number, title, revision, and the date of all documents referenced in this document. This section shall also identify the source for all documents not readily available.]*

* 1. Project Documents

| Project Identifier | Title | Rev. | Date |
| --- | --- | --- | --- |
| 001 | Type-I Robot |  |  |
|  | The Software Development Suite |  |  |
| 001.1 | The Firmware |  |  |
| Call out wbs | Test Rigs |  |  |
|  | Mechanical Design Documents |  |  |
|  | Test Data |  |  |
|  | Test Logs |  |  |
|  | Test Coverage & Design |  |  |
|  | Development & Engineering Plan |  |  |

* 1. Other Documents

| Document Identifier | Title | Rev. | Date |
| --- | --- | --- | --- |
| 2503Q–AVR–02/11 | ATmega32/L Datasheet | Q | 02/2011 |
|  |  |  |  |

1. Abbreviations

|  |  |
| --- | --- |
| Abbreviation | Expansion |
| NEMA | National Electrical Manufacturers Association |
| TI | Texas Instruments |
| MCU | Micro Controller Unit |
|  |  |

1. Definitions

|  |  |
| --- | --- |
| Term | Definition |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

1. Overview of Required Work

The requirements of the project can be broken down into 3 sections: Report Requirements, Project Requirements, and Course Requirements.

* 1. Project Report (Assessment 7)[[1]](#footnote-1)

**Type** Report

**Description** The project report meets the course objectives of knowledge acquisition and demonstrates assimilation of data, upon reflection and analysis, to produce articulate and document the design conducted in the project, which conveys evidence-based understanding of the concepts and topics.

* Abstract indicates objective, scope and results (10%)
* Report organisation (25%)
* Clarity of writing (20%)
* Clarity of figures and tables (15%)
* Analysis and discussion of results (30%)
  1. Project Demonstration Assessment (Assessment 6)[[2]](#footnote-2)

**Type**Project

**Description**The purpose of demonstration is to assess whether the project meets the required goals and specifications, and give students the opportunity to discuss orally the degree to which they attained the required goals

Project

Hardware (0-100%)

* Robust construction
* Cost effectiveness
* Plant dynamics suitable for control loop rate
* Suitable power electronics for selected actuators
* Adequate power regulation under load

Software (0-100%)

* Maintainability (e.g., serial interpreter uses command tables rather than if-else chains)
* Code organisation (e.g., encapsulation, orthogonal functionality separated into C modules (.c/.h), use data interfaces instead of global variables)
* Testing (e.g., coverage of unit tests, design for testing, TDD)
* Style (e.g., consistent indenting, function/variable naming convention)
* Documentation (e.g., appropriate use of code comments, online help via serial interface)
* Safety features (e.g., watchdog, comms keepalive/timeout, unrecoverable actuator limits)

System identification

* Log experimental data using MCU
* Use data to guide model structure selection
* Fit actuator/plant models from gathered data (training set)
* Validate model fits against new data (validation set)
* Automation between MATLAB and MCU

Simulation model

* Sensor sampling and quantisation (e.g., encoder ISR, ADC)
* Actuator model and control allocation
* Actuator limits (e.g., saturation and slew rate)
* Model effect of inclined ground plane and transition

Control

* Model-based control design
* Robustness to disturbances (e.g., integral action and/or disturbance estimation + feedforward cancellation)
* 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
* PIL/HIL testing of control system (controller + control allocation) implementation
  1. Course Outline

On successful completion of this course, students will be able to:

1. Formulate specifications for adopting/designing different components of a mechatronic system (mechanical, electrical, sensors, actuators).

2. Conduct a mechatronic design using a structured formal approach. Make decisions about component choice taking into account its effects on the choice of other components and the performance of a mechatronic system.

3. Design and implement software for a computer control system with sensor and actuator interfaces.

4. Design and implement communication interface with a computer control system for tuning.

5. Design and implement printed board circuit (electronic hardware) for a computer control system.

6. Implement a software-hardware verification using hardware-in-the-loop testing.

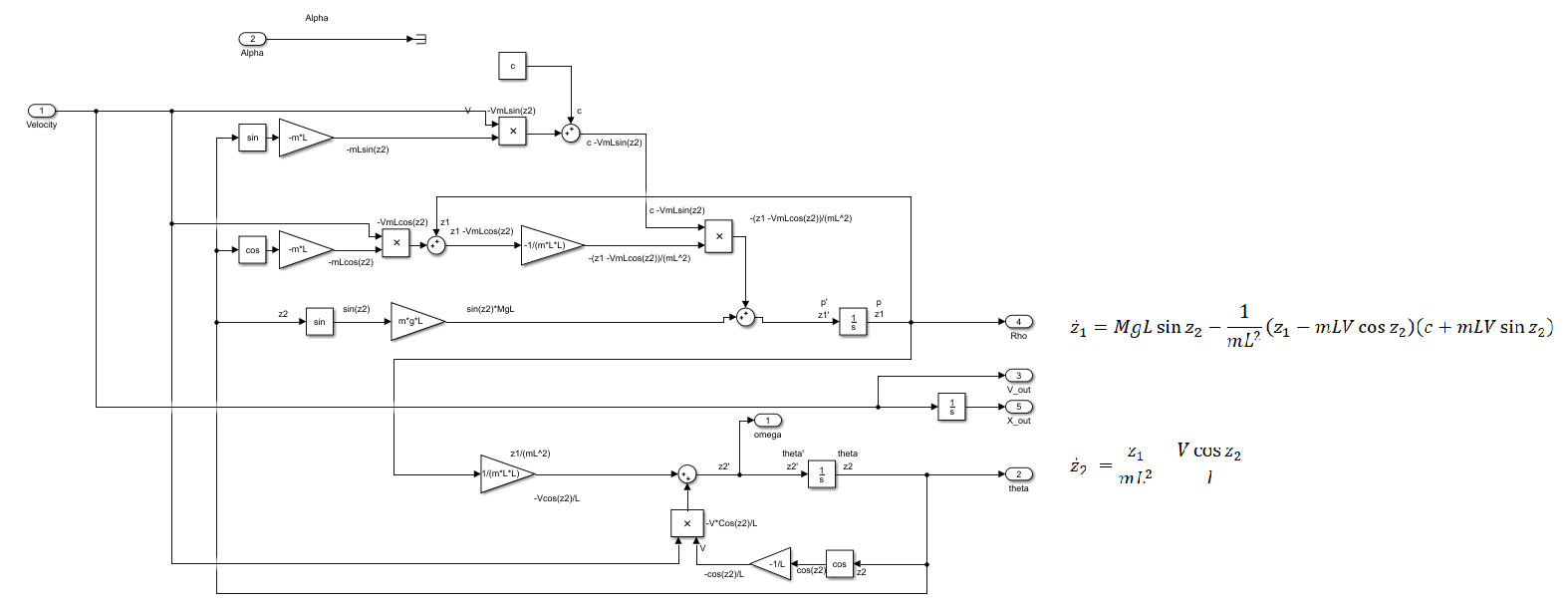
7. Conduct experimental modelling to assist in the design and tuning of control systems.

1. Model Based Design
   1. Brief

We utilised a Model-Based-Design, by employing both Bond-Graph techniques and traditional state space modelling to produce a state space we could simulate and model.

* 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
  1. Approach

The following is how we did it:



* 1. Results

1. Requirments a
2. Design for testing

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

* 1. Approach
  2. Results

1. System Identification
   1. Pendulum Friction
      1. Characterisation:

A combination of items effected the Pendulum Friction.

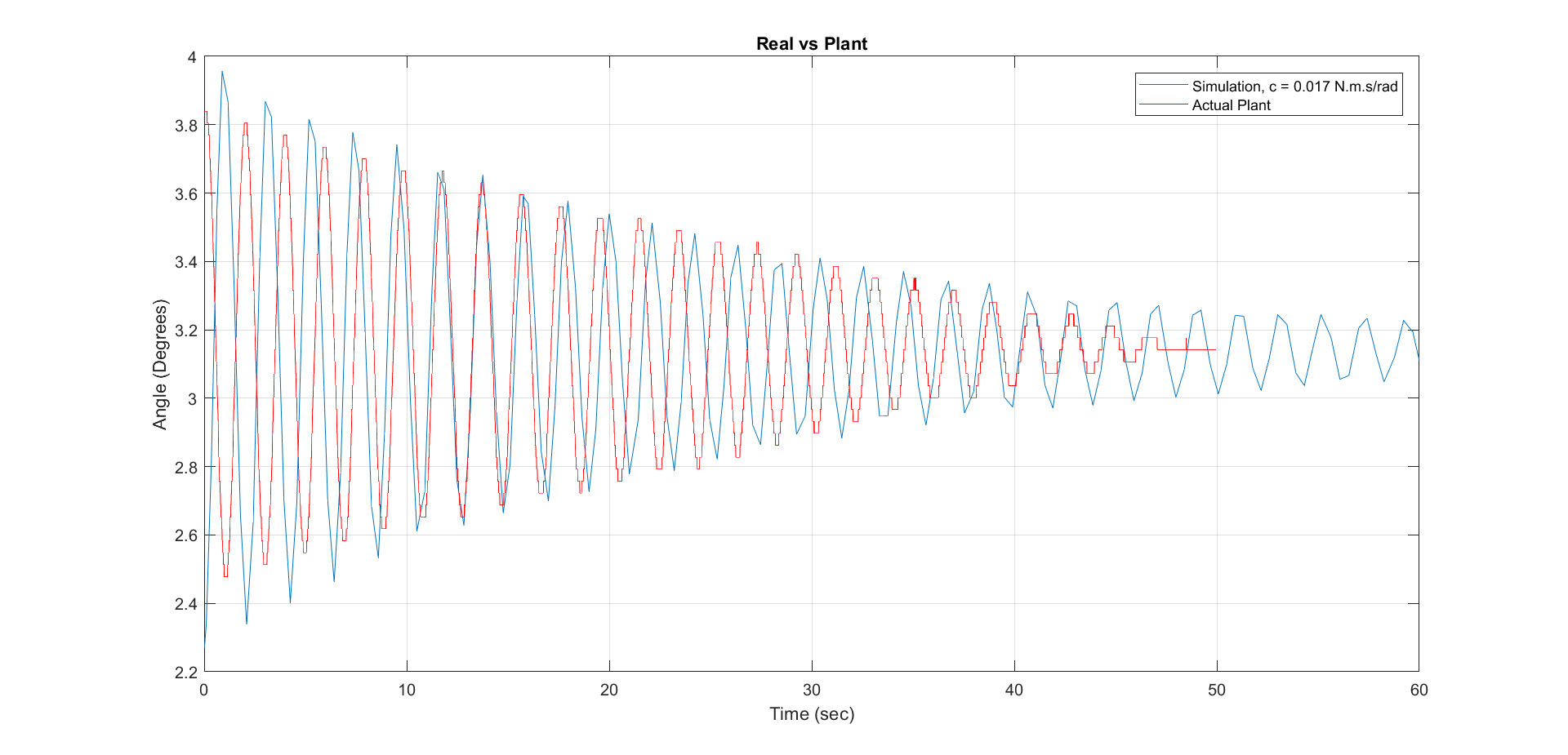
The 606-ZZ bearings, L-Bracket alignment, M6 Rod surface finish & tolerance, and potentiometer/encoder friction all contribute to the total friction moment occurring at the rotational axis of the pendulum.

* + 1. Experiment Design:

I decided to implement a single friction coefficient, and find a plant value experimentally. I logged real plant data and plotted this against an unpowered plant simulation. Using a binary search method, I was able to identify and estimate this value. A final value of 0.017 N.m.s/rads was chosen as best fit.

* + 1. Model calibration:

By experimentally fitting a value of friction from real plant data, crude calibration was already achieved. The friction was sufficiently small to not warrant looking at the specific datasheet for the supplied bearings, and no further action was taken.



* 1. Stepper Motors
     1. Characterisation:

The principle used to model and control the stepper motors was to find an operating envelope where the stepper motors did not slip, and add saturation limits to my input velocity, to ensure the steppers would not slip. As a result, the steppers are modelled as pure flow actuators, delivering the demanded velocity with no losses.

* + 1. Experiment Design:

The Data below helped me decide of a maximum of +-0.5Meters a second.  
Configuration:  
2\* 17H185H-04A

2 \* drv8825 (Vref @ 1.2V (Texas Instruments, 2017), ¼ microstepping)

2\* Pololu Wheel 80×10mm



**running torque: unknown**

**Holding torque of** 43.8 N·cm

* + 1. Model calibration:

The data was taken from the robot with full weight, and no input disturbances. Whist the final design was unable to balance, stress-testing during balancing would help further calibrate these saturation limits.

Testing with larger input disturbances would help identify the saturation points required. Further calibration could be achieved by characterising a maximum acceleration slew rate, which could help us further expand the stepper’s operating envelope.

* + Characterisation of actuators – were these as the manufacturer claimed?
  + Experiment design
  + Model calibration – structure, parameters and validation

1. Conclusion

Summarise the results

reflect on the work presented

make recommendations

suggest future work or improvements.

* 1. Reflect on the work done
  2. What would you do differently in terms of design if you could start again?

1. Notes.

[This section shall contain any general information that aids in understanding this document (e.g., background information, glossary, rationale). This section shall include an alphabetical listing of all acronyms, abbreviations, and their meanings as used in this document and a list of any terms and definitions needed to understand this document.]

1. Appendixes.
   1. MCHA3000 Requirement vs Design Matrix

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***MCHA3000 Requirement vs Design Matrix*** | | | | | | | | | | | | | | | | | | | | | | | | |
|
|  | **Hardware** | | | | | | | | **Software** | | | | | | | | | | | **Matlab / Simulink** | | | | |
| |  | | --- | | **Design Decision**  **Project Requirements**  **(or Activity)** | | **Aluminium Sheet Box** | **1000mm Alum. + Steel Pendulum** | **Polulu Wheels** | **Stepper Motors** | **DRV8825 Drivers** | **Encoder + Potentiometer** | **XFinity Plus Li-Ion 20V 2.0AH Battery** | **LM2596 3A Buck Converter + MC 7805CT Regulator** | **Command talbe** | **Separate Class Per Module** | **Test Matrix per Module** | **Timestamped Test Logs** | **Sublime Stylecop** | **Git Sync Make Integration** | **HIL Testing** | **Regression Testing** | **#include "<file>.h"** | **Log Data Command** | **Comms Handshake** | **Sample Data System** | **100Hz ZOH** | **Model Fitting to Real Data** | **Quantizer Float Convert** | **Saturation Block** |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1. Hardware (0-100%) |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Robust construction | **◊** | **◊** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | **Legend**:  Strong Link: ●  Weak Link: ◊ | | | | | |
| Cost effectiveness | **●** | **◊** | **◊** |  |  | **◊** | **◊** | **◊** |  |  |  |  |  |  |  |  |  |  |
| Plant dynamics suitable for control loop rate | **●** |  |  | **◊** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Suitable power electronics for selected actuators |  |  |  |  | **●** |  | **●** | **●** |  |  |  |  |  |  |  |  |  |  |
| Adequate power regulation under load |  |  |  |  | **●** |  | **●** | **●** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1.          Software (0-100%) |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Maintainability (e.g., serial interpreter uses command tables rather than if-else chains) |  |  |  |  |  |  |  |  | **●** | **●** |  |  | **◊** |  |  |  | **●** |  |  |  |  |  |  |  |
| Code organisation (e.g., encapsulation, orthogonal functionality separated into C modules (.c/.h), use data interfaces instead of global variables) |  |  |  |  |  |  |  |  |  |  |  |  | **●** |  |  |  | **●** |  |  |  |  |  |  |  |
| Testing (e.g., coverage of unit tests, design for testing, TDD) |  |  |  |  |  |  |  |  |  | **◊** | **●** | **◊** |  |  | **●** | **●** |  | **●** |  |  |  |  |  |  |
| Style (e.g., consistent indenting, function/variable naming convention) |  |  |  |  |  |  |  |  |  |  |  |  | **●** |  |  |  |  |  |  |  |  |  |  |  |
| Documentation (e.g., appropriate use of code comments, online help via serial interface) |  |  |  |  |  |  |  |  | **◊** |  |  | **◊** |  | **◊** |  |  |  | **◊** |  |  |  |  |  |  |
| Safety features (e.g., watchdog, comms keepalive/timeout, unrecoverable actuator limits) |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | **●** |  |  |  |  |  |
| 3. System identification |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Log experimental data using MCU |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | **●** | **●** |  |  |  |  |  |  |
| Use data to guide model structure selection |  |  |  |  |  |  |  |  |  |  |  |  |  | **◊** |  |  |  | **◊** |  |  | **●** |  |  |  |
| Fit actuator/plant models from gathered data (training set) |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | **●** |  |  |  |
| Validate model fits against new data (validation set) |  |  |  |  |  |  |  |  |  |  |  |  |  |  | **●** |  |  |  |  |  |  |  |  |  |
| Automation between MATLAB and MCU |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | **●** |  |  |  |  | **◊** |  |  |
| 4. Simulation model |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Sensor sampling and quantisation (e.g., encoder ISR, ADC) |  |  |  |  |  | **◊** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | **●** |  |
| Actuator model and control allocation |  |  |  | **◊** | **◊** |  |  |  |  |  |  |  |  |  |  |  |  |  |  | **◊** |  |  |  |  |
| Actuator limits (e.g., saturation and slew rate) |  |  |  | **◊** | **◊** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | **●** |
| Model effect of inclined ground plane and transition |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5. Control |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Model-based control design |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | **◊** | **◊** | **◊** | **◊** | **◊** |
| Robustness to disturbances (e.g., integral action and/or disturbance estimation + feedforward cancellation) |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | **●** |
| 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 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | **●** |  |  |  |
| PIL/HIL testing of control system (controller + control allocation) implementation |  |  |  |  |  |  |  |  |  |  |  |  |  |  | **●** | **●** |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

1. MCHA3000 Week 11 Lecture Slides, Dr Chris Renton [↑](#footnote-ref-1)
2. Project outline, Dr Chris Renton [↑](#footnote-ref-2)