Stabilisation and control of a Three-Dimensional inertial wheel based inverted pendulum system

Christan Hare
University of Sheffield
Chare1@sheffield.ac.uk

***Abstract* – This paper explores the simulation, stabilisation and physical design of an inertial-wheel based inverted pendulum (IP) system. Applying the principles of control systems and electrical engineering to construct state space matrices of the system, to allow for the design of a linear quadratic regulation (LQR) feedback system. The control algorithm designed is for affordable, low powered microprocessors applications for educational purposes and possible future developments within compact flywheel-based devices such as swarm robots. The final system will be placed within a cube housing, using the internal actuators to drive the system towards its equilibrium positions, along one of its edges, or on the point of one of its corners. This will be done by applying Newtons second law of motion, using inertial flywheels to apply controlling torques to stabilise the system. Ultimately this paper aims to explore the most appropriate control techniques available to be applied to each axis, to determine if individual stabilisation within each axis will provide global stabilisation for the whole system about all three axes of the cube. Although this paper is unable to answer the question proposed it successfully demonstrates how to use tools such as MATLAB and Simulink to construct a robust and stable LQR IP system, outlining the design process from simulation to component selection and physical design.**

I. INTRODUCTION

The Inverted pendulum

IP systems are naturally unstable, high order, under actuated and non-linear systems, hence have been used as an example within control theory when benchmarking and testing various control algorithms and techniques [1]. The algorithms used are an active area of research to this day. Applications range from personal transportation to stabilisation to drones [2, 3]. Each variation of the inverted pendulum presents a unique set of challenges. This paper focuses on the one, two and three-dimensional cases of the inertial wheel IP, using a control torque applied by the flywheel(s) to drive the system to equilibrium as seen in Fig. 1. Inertial flywheels have many applications, ranging from space exploration to swarm robots [4, 5].

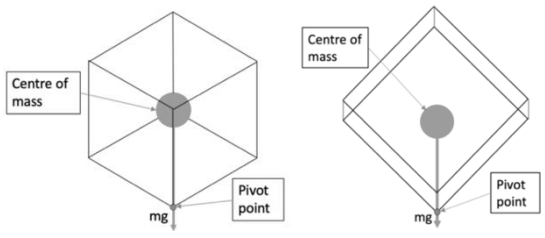


Figure 1, showing the three-dimensional IP system in equilibrium positions on corner (left) and edge (right)

The specific aim of this paper is to determine whether or not the principles used to stabilise the one-dimensional case can be independently applied to axes of the two and three-dimensional cases, resulting in global stabilisation of the system as a whole. In doing so this paper focuses on the physical hardware and mathematical models used to successfully design a stable and robust system, calling upon the large body of research available within this area to do so.

MATLAB and Simulink are powerful tools, allowing for the dynamic system to be accurately modelled in both continuous and discrete time by using a state space representation of the IP system. Linearizing this model between two points yields the state space matrices of the system. Manipulation of these matrices allows for the controllability of the system to be determined and used to accurately design of the linear quadratic regulation (LQR) control loop.

The aims and objectives are as stated:

- Research into control theory techniques.
- Research into appropriate physical hardware for construction of one two and three-dimensional test beds.
- Understand and use a state space representation of the IP system to accurately design a control algorithm, using MATLAB and Simulink.
- Determine appropriate testing techniques to accurately determine the parameters for design of the feedback system.
- Determine the most appropriate testing techniques for selection of hardware components and construction.
- Construct one, two and three-dimensional inverted pendulum systems for validation of the control algorithms developed and hardware selected.
- Determine if individual stabilisation of axes will provide global stabilisation.

Likely new developments of this technology

There are many applications when considering the control algorithms discussed as well as the electromechanical system of the inertial flywheel. State space modelling can be used for a variety of non-linear systems for design of LQR control algorithms. This approach has been used within the design of self-balancing personal transportation and drones, two technologies with huge scope and an increasing impact within people's everyday lives [2, 3]. More complex three-axis inertial wheel systems also provide a means of stabilisation and control within satellites and spacecraft, playing a key role in positioning of satellites within orbit, as well as avoidance of debris without the use of finite fuels such as compressed gas [6].

One key area where the inertial wheel-based propulsion systems may have the greatest future impact is within swarm robots. The most predominant example of this is the M-Block 2.0, using inertial actuators to transfer momentum to the cube body, allowing for it to move without any external actuators [5]. The more precisely this torque can be applied, the more accurate the movement of the cube itself. This actuation allows the cubes to self-assemble by moving along each other or jumping in order to form large structures. One of the most impactful application of this technology may be within space exploration, allowing for the construction of large structures on other planets without human intervention [7].

II. Background theory

DC motor model

In order to calculate the appropriate parameters for use within the state space model, the relationship between the angle of the IP (θ) and the applied armature Voltage (u) must be determined. This is done using both Newton's (second) and Kirchhoff's laws.

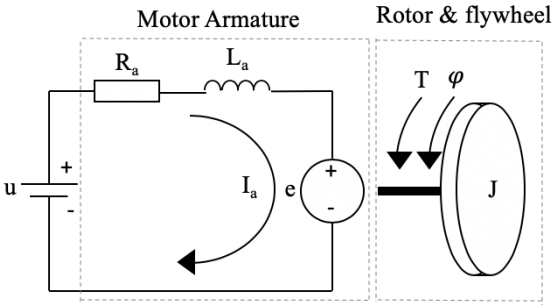


Figure [2], showing equivalent direct current (DC) motor model

Using the model of the DC motor seen in Fig. 2, the motor torque (T_{motor}) can be related to the armature current using equation (1). The back electromotive force (emf) can also be related to the angular velocity (*ω) using equation (2).*

$$T_{motor} = Ki_a \qquad (1) \qquad\qquad e = K\omega \qquad (2)$$

Where $K_t = K_e = K$, with K_e and K_t representing the back emf and torque constants of the motor respectively.

The dynamics of the DC motor can then be described by the coupled electro-mechanical equations (3) and (4) (where b_m is the friction within the motor bearing).

$$u = L_a\frac{di_a}{dt} + L_aR_a + L_a\frac{d\varphi}{dt} \qquad (3)$$

$$Ki_a = J_2\frac{d^2\varphi}{dt} + b_m\frac{d\varphi}{dt} \qquad (4)$$

As K, b_m, J and R_a are constants (where J is the moment of inertia of the flywheel) equation (5) can be derived.

$$K_1u = \frac{d^2\varphi}{dt^2} + K_2\frac{d\varphi}{dt} \qquad (5)$$

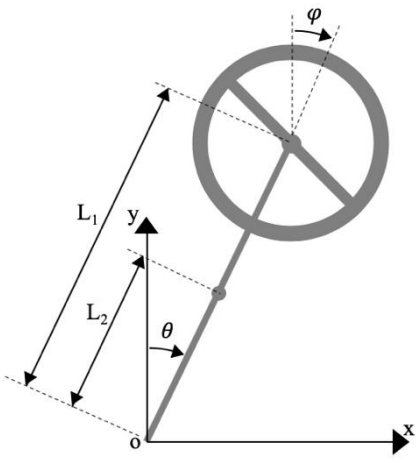


Figure [3], showing the one-dimensional inertial wheel inverted pendulum with its variables

The total rotating torque (T_{tot}) about the pivot can be calculated by considering the sum of the total external torques being applied to the IP system, from the DC motor, gravity and friction, as seen on right-hand side of equation (6) (where b is the friction about the pendulum's axis of rotation). The torque can then be equated to the angular position of the flywheel as seen in equation on the left-hand side of (6).

$$-J_2\frac{d^2\varphi}{dt^2} = -m_1gl_1sin\theta + m_2gl_2sin\theta + J_1\frac{d^2\theta}{dt^2} + b\frac{d\theta}{dt} \qquad (6)$$

The total inertia (J_{tot}) of the system about the pivot must also be considered, this is done by applying the parallel axis theorem as seen in equation (7). Where J_1 and J_2 represent the moment of inertia of the pendulum about the pivot and the flywheel moment of inertia about its centre respectively. m_1 and m_2 represent pendulum and motor/flywheel mass respectively, with l_1 and l_2 represent the distance from the origin to the pendulums centre of mass and the flywheels axis of rotation respectively.

$$J_{tot} = m_1l_1^2 + m_2l_2^2 + J_1 + J_2 \qquad (7)$$

By linearizing about the pendulum about its non-inverted position (where θ with 180+θ) equation (8) is obtained.

$$-J_2\frac{d^2\varphi}{dt^2} - m_1gl_1\theta + m_2gl_2\theta + J_2\frac{d^2\theta}{dt^2} + b\frac{d\theta}{dt} \qquad (8)$$

Considering zero initial conditions and Appling Laplace transforms to both (8) and (5) the following transfer function (9) & (10) can be derived respectively.

$$\frac{\theta(s)}{\phi(s)} = \frac{-J_2s^2}{J_{tot}s^2 + bs + mlg} \qquad (9)$$

$$\frac{\phi(s)}{U(s)} = \frac{K_1}{s(s+K_2)} \qquad (10)$$

Equating the two transfer functions (9) and (10) produces a third transfer function to describe the relationship between the pendulum angle with respect to the applied armature voltage, seen in equation (11) [11].

$$\frac{\theta(s)}{U(s)} = \frac{-\frac{J_2K_1}{J_{tot}}s^2}{s^4 + \left(K_2 + \frac{b}{J_{tot}}\right)s^3 + \frac{(mlg + K_2b)}{J_{tot}}s^2 + \frac{(mlgK_2)}{J_{tot}}s} \qquad (11)$$

Where $ml = m_1l_1 + m_2l_2$.

Construction of state space control matrices for LQR

LQR provides optimal control for dynamic systems comprise of linear quadratic equations, doing so at a minimal computing cost. When controlling a non-linear system with multiple inputs and outputs, such as the inertial wheel-based IP, the internal state of the system must be taken into account in order to effectively control the system. State space is a mathematical representation of the physical system, relating the input, output and state variables by first order differential equation. State space control allows for a system to be controlled by use of simple matrix manipulation. This is in contrast with proportional integral derivative (PID) control which relies on Laplace and Fourier analysis. State space LQR control allows for more effective control of a non-linear system when compared to PID, as the matrix manipulation produces dynamic control constants. A linear time invariant (LTI) state space system's state and output are represented by equations (12) and (13) [9].

$$x' = Ax(t) \times Bu(t) \quad (12)$$

$$y(t) = Cx(t) \times Du(t) \quad (13)$$

Where:

- X represents a vector of the state variables
- X' represents the time derivative of vector X
- U represents the input vector
- Y represents the output vector
- A represents the state matrix
- B represents the input matrix
- C represents the output matrix
- D represents the feedforward matrix

The state equation (12) describes the relationship between the current and future system states, with the output equation (13) describing the relationship between the systems current state and input to its output. The state space control block diagram can be seen in Fig. 4 [10].

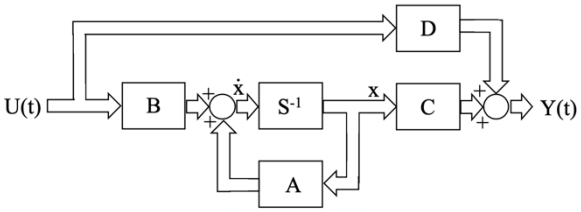


Figure [4], showing the state space LQR control block diagram

Using the state variables X_{IP} (17) with equations (10) and (11), the state matrix A_{IP} (19) and input matrix B_{IP} (18) for the inertial wheel IP were constructed [11]. (The angular position of the flywheel (φ) is not relevant in this case.)

$$x_{IP} = \begin{bmatrix} \theta \\ \dot{\theta} \\ \dot{\varphi} \\ \ddot{\varphi} \end{bmatrix} \quad (17) \qquad B_{IP} = \begin{bmatrix} 0 \\ -\frac{J_2 K_2}{J_{tot}} \\ 0 \\ K_1 \end{bmatrix} \quad (18)$$

$$A_{IP} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{mlg}{J_{tot}} & -\frac{b}{J_{tot}} & 0 & \frac{J_2 K_2}{J_{tot}} \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -K_2 \end{bmatrix} \quad (19)$$

DC motor driven inertial flywheel

This represents the control element within the system, with the flywheel being driven by the DC motor to provide a variable rotational torque, countering the torque produced due to gravity. The kinetic energy, inertia, mass, radius and angular velocity of the flywheel can be related using equations (20). With equations (21) and (22) describing the moment of inertia for a solid and hollow flywheel respectively.

$$Ke = \frac{1}{2} J\omega^2 \quad (20) \quad J_2 = mr^2 \ (21) \quad or \ J_2 = \frac{1}{2} mr^2 \quad (22)$$

Proportional integral derivative (PID) control

PID control is a common feedback control algorithm, used for applications requiring continuous modulation in order to be driven to a set point. PID control continuously calculates the error value with respect to the setpoint, applying correction based on the P, I & D terms seen in Fig. 5. The system response can by altering the constants K_p, K_i and K_d.

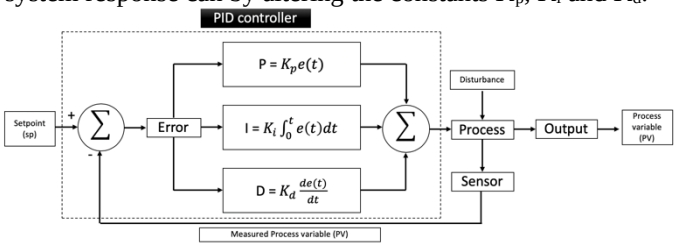


Figure 5, showing a block diagram of PID control loop

Microelectromechanical Inertial measurement unit (IMU)

The IMU device represents the sensing element within the system. In order to accurately calculate this angle (θ) in the one-dimensional system, or the Roll (about X axis) Pitch (about Y axis) and Yaw (about Z axis). In order to accurately calculate these parameters, the data from multiple devices must be combined by using an attitude and heading reference system (AHRS).

Accelerometer: This is a device used to measure the direction, strength or relative change of the acceleration experienced within three-dimensional space, due to gravity or movement (in ms^{-2}). In order to calculate the orientation of the accelerometer, the earths gravitational field is used as a reference ($9.81ms^{-2}$), with each axis of the accelerometer experiencing a different acceleration as a vector quantity with respect to g.

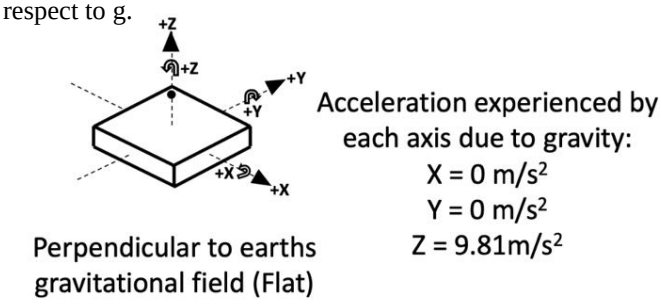


Figure 6, showing the acceleration experienced by each axis (X, Y & Z) of a MEMS accelerometer when placed in a stationary position on a surface perpendicular to the earths gravitational field

Magnetometer: This device is used to measure the direction, strength or relative change of the magnetic field experienced. The Magnetometer is able to act as a compass in order to determine the absolute orientation about the Z axis.

Gyroscope: The gyroscopes function is to measure the angular velocity experienced by the device (in $rads^{-1}$). Assuming the initial angle of the device is known (using the accelerometer data), the gyroscope data can be used to accurately calculate the new angular position of the device.

Attitude and heading reference system

The data produced by these individual devices must be combined in order to accurately and reliably calculate the heading and orientation of the device, its Roll, Pitch and Yaw. Examples of some of the most popular AHRS algorithms are the orientation based complimentary and Kalman filters, as well as the Bayesian estimation Mahony and Madgwick filters [12]. The Mahony filter was chosen due to its suitability for embedded hardware and easy implementation with Arduino [13]. This filter is based on a non-linear explicit complimentary filter applied to the accelerometer for low frequency attitude estimation, with integrated gyroscope data for high frequency attitude estimation. This data is combined with the magnetometer data in order to obtain attitude and orientation [14]. The orientation output by the AHRS algorithm provides the angle of the inverted pendulum about its axis, to be used within input matrix of the state space system in order to calculate the closed loop response required.

Pulse width modulation (PWM) control of H-bridge driver

In order for the flywheels torque to be accurately applied a H-bridge driver was used to drive the brushed DC motor. With the H-bridge allowing for a variable duty cycle PWM signal to control the armature current (and hence flywheel torque) of the DC motor. The larger the duty cycle, the larger the average voltage, as seen in Fig. 7, resulting in a higher flywheel torque. The H-Bridge and DC motor were chosen over a simple sensor-less brushless motor and three phase speed-controller, as the motor direction can more easily reversed and accelerated from low rpm without stalling. Reversal of the motor direction is done simply by applying a PWM signal to the complimentary MOSFETS within the H-bridge driver.

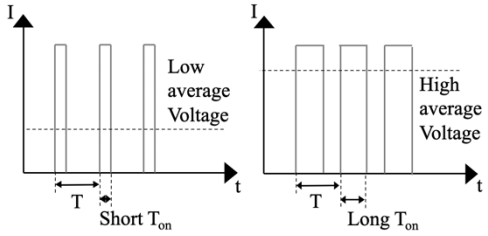


Figure 7, showing the PWM duty cycle vs output Voltage

III. Economic, Legal, Social, Ethical Environmental context

As stated, the inverted pendulum is a system that has been used as a benchmark when testing and teaching control theory. For this reason, the most significant impact of this project may be to do with the implementation and further development of the control theory presented. There have been many advancements within the technology related to different control algorithms, both theoretically and within real time applications [15, 16]. The advancements within control theory are closely related to education, due to the increasing interest by students to go into fields such as electronics, mechatronics and robotics due to the increasing importance of these fields within everyday life [17, 18, 19, 20].

Control theory will undoubtedly further the reach of automated processes within ‘smart factories’. Leading to positive impacts such as increased efficiency, profits and reduced greenhouse gas emission. Unfortunately, this may also lead to a reduction in the unskilled work force employed by these factories, with each new robot replacing on average 1.6 workers [21].

IV. Methodology

Construction of the one-dimensional inverted pendulum

In order to develop a stable system both in hardware and software, testing and simulations must be undertaken. Hardware tests provide the variables required for the simulations hence the more accurate the testing procedure, the more accurate the simulation, resulting in a more stable IP system. Prior to the initial design and construction of the one-dimensional IP, appropriate components must be selected in order to meet the system requirements. To ensure that the components selected were appropriate, unit tests were carried out to verify individual components and subsystems as seen in the interim report [1]. The construction of the one-dimensional IP system will provide a platform for the two and three-dimensional systems to be developed, sharing the same components.

One of the key assumptions made is that independent stabilisation of each IP axis will result in global stabilisation within the three-dimensional system. Assuming this is the case, by achieving stability within the one-dimensional case stability will also be achieved in the two & three-dimensional cases, applying the theory individually to each axis. This assumption was applied in order to reduce the complexity of the modelled system and controller design.

Selection of microprocessor

The Arduino Mega 2560 was selected for a number of reasons. The Arduino platform is designed to perform one specific task, being tailored to use a number of sensors, (analogue and digital) running a (relatively) simple C programme to process the data and provide an output. Due to the open source nature of the Arduino there are many libraries and example programmes within the Arduino IDE which may be utilised, modified and combined when writing the programme. The libraries for the FXOS8700 accelerometer/magnetometer, FXAS21002 gyroscope, state space controller and Mahony AHRS were used.

Lastly the Arduino Mega offers more than enough processing power and memory, allowing for the processing of the feedback algorithm (with many successful IP projects being based around Arduino). The Mega also provides a large number of input/output (I/O) pins to connect all the devices required for the three-dimensional system. Hence the Arduino meets all the criteria outlined in the interim report [1].

Selection of sensing elements

Initially the MPU6050 was selected, this device is a 6 degrees of freedom (DOF) gyroscope and accelerometer commonly used with Arduino. When testing the accelerometer large inaccuracies were found within the data seen within the interim report [1]. For this reason, the NXP 9-DOF breakout board was chosen in its place, utilising the FXOS8700 accelerometer/magnetometer and FXAS21002 gyroscope. This chipset also provides the advantage of offering an extra three DOF, using the magnetometer to allow for more accurate calculation of heading and orientation.

Selection of control elements

After initial testing with the Diatone 302XD electronic speed controller (ESC) and M2205 sensor-less brushless motor, it was found that the 4Tronix L298N H-Bridge driver and Mellor Electric 24v brushed DC motor would be best for this application. This is because the brushless motor was prone to

stalling when rotating from a stand still under load due to its lack of hall effect sensors, often resulting in the incorrect sequence of motor coils being energised for a given rotor position. This introduced further instability about the equilibrium point of the IP.

The hobby grade 302XD ESC also presented issues when reversing the motor direction, with the firmware not allowing for smooth transitions between clockwise and counter clockwise rotation of the motor. (This is as its intended application is within drones, hence only requiring rotation in one direction.) Despite this an appropriate sensored brushless motor system would provide some key advantages over a brushed system such as lower weight (hence reduced IP inertia), increased torque and efficiency as stated in the interim report. The use of the H-Bridge driver also provides the advantage of allowing for the armature voltage to be more closely controlled by varying the PWM duty cycle.

Determining the DC motor constants

As stated within the background theory the motor constants R_a, L_a and K must be obtained in order to construct an accurate state space model. K (in this case K_e) was calculated using the values of no-load speed and voltage provided on the datasheet [22]. This was done by dividing the motor voltage of 24V through by the no load speed of 366.5 rad/s to give K equal to 0.0655. R_a and L_a must be calculated using a wave generator and oscilloscope, applying a sinusoidal current to the motor phase, using a 1KΩ reference resistor (R_{ref}). The closer the actual value of the resistance to its stated value, the more accurate the calculated R_a and L_a will be. Hence a maximum of 1% tolerance should be used, verifying the resistance prior to conducting the test. (R_{fg} represents the resistance of the signal generator.)

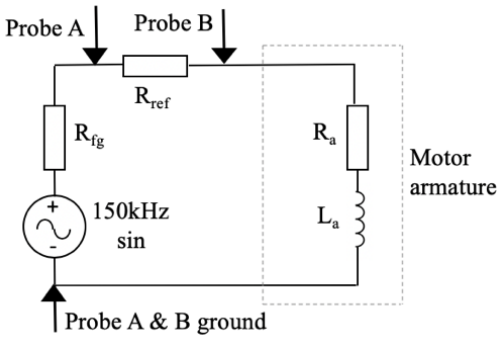


Figure 8, showing how to connect the oscilloscope probes when measuring R_a and L_a

Applying a sinusoid of frequency 10KHz (f) and amplitude 1.9v to the motor armature, the input (A) and output (B) waveforms can be obtained, placing the two oscilloscope probes as seen in Fig. 8. Using the measure tool within the oscilloscope, the amplitude (V_A & V_B) and phase difference (θ) between A and B can be measured directly. By using equations (23) to (26) L_a and R_a can be calculated from the impedance (Z).

$$Z_{polar} = \frac{V_B R_{ref}}{\sqrt{V_A{}^2 - 2V_A V_B \cos(\theta) + V_B{}^2}} \quad (23)$$

$$\alpha = \theta - \tan^{-1}\left(\frac{-V_B \sin(\theta)}{V_A - V_B \cos(\theta)}\right) \quad (24)$$

$$R_a = Z_{rectangular} \cos(\alpha) \quad (25)$$

$$L_a = \frac{Z_{rectangular} \sin(\alpha)}{2\Pi f} \quad (26)$$

This method (as outlined by Tektronix) will provide values for R_a and L_a within a tolerance of 3-5% [23]. Using these values, the maximum current throught the motor phase can be calculated, allowing for the PWM duty cycle for the H-bridge to be accurately mapped.

MATLAB & Simulink Modelling

Once the motor constants have been obtained and the IP parameters have been determined, the MATLAB & Simulink models can be constructed as follows. First, the motor and IP parameters must be defined, allowing for the construction of the state space matrices described within the background theory.

The inertia of the IP (j_1) and flywheel (j_2) can be obtained by using the 3D models created within Fusion360. This is done by selecting the appropriate material, ABS in this case and aligning the pivot point about the origin of the 3D workspace. (select body > right click > Properties > moment of inertia at origin.) (The parameters used are found in the appendix [8].)

Construction and evaluation of state space matrices

Using equations (18) and (19) the state space matrices were constructed. By using the Control System Toolbox within MATLAB the state space matrix properties can be evaluated, hence so can the behaviour of the IP system. In order to complete the system, we must construct the output (C) matrix. By setting the value of a column within the C matrix to one, defines the variable output by the system with the variables seen in equation (17). Setting the output matrix as seen in equations (27) will allow for the output of θ, for use within the LQR feedback loop.

$$C_{IP} = [1\ 0\ 0\ 0]^T \quad (27)$$

The system is then built within MATLAB by using the command sys=ss(A, B,C,D) and can be seen in Fig. 9.

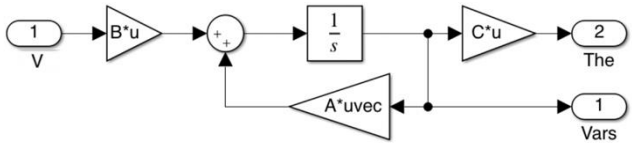


Figure 9, showing IP state space block diagram

As the feedback system is time discrete the poles of the system must be within the unity circle in order for the system to be stable. This was done using the command des_pole_d = [0.3; 0.3; 0.3; 0.3], once these poles have been set the LQR feedback matrix K_lqr can be calculated using the command K_lqr=lqr(A,B,Q,R) to construct the feedback loop seen in Fig. 10 where Q = 5*eye(4) & R = 0.1.

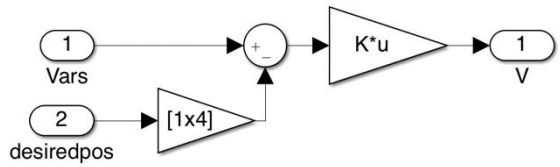


Figure 10, showing the LQR feedback block diagram

The behaviour of the LQR controller can be changed by changing the pole locations of the system within MATLAB. By Increasing the values of the poles, we are able to improve the response time of the system, with a trade-off between input effort and performance of the system. The lower the value of R, the less effect input effort has on the response, with a larger unity matrix Q correlating to better performance.

Construction of Simulink model

DC motor Simulink model and control

Using the transfer function (11) derived with the parameters stated in the appendix, a closed loop control model of the motor was constructed as seen in Fig. 11. The transfer function describes the relationship between the armature voltage applied and the angular displacement of the pendulum as a result of the applied flywheel torque.

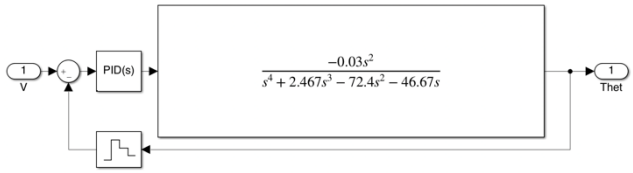


Figure 11, showing DC motor transfer function and PID control loop

It is well known that the applied armature current is directly related to the torque output by a DC motor by the constant K_t, hence using the inner PID control loop the armature voltage can be managed in order to manage the motor output torque.

Construction of the final system

Using the DC motor model seen above, the entire IP system and control system can be modelled (as seen in Fig. 12). By modulating the armature voltage, the motors output torque is controlled, hence so is the total torque experienced by the IP system. The applied voltage is related to the IP angle by equation (11), hence the IP response can be controlled by modulating the applied armature voltage according to the error value produced. The control architecture can be seen in Fig. 12, showing the cascade control architecture used, the inner loop seen is controlling armature current. With the outer loop using and LQR controller to control the angle of the IP with respect to the vertical axis, driving it to equilibrium.

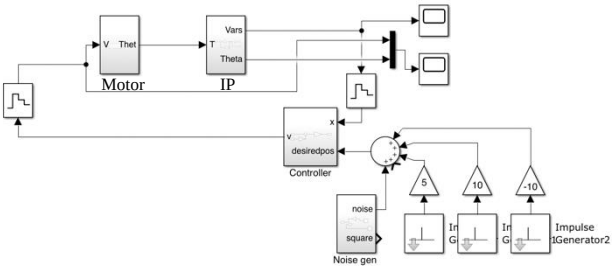


Figure 12, showing the proposed cascade LQR PID control architecture (within Simulink)

To improve the accuracy of the model it was broken up into two sections, continuous and discrete time. This is as the outer loop feedback will be digital, taking samples of the continuous time IP angle at a frequency of 50Hz. (The sample time is represented by the sample & hold blocks seen in Fig. 12.) It is worth noting that the system remained stable even with a sample as low as 10Hz.

Proposed design of one two and three-dimensional test bed

Using the dimensions stated within the appendix the one-dimensional test bed was constructed as shown in Fig. 13. The two/three-dimensional test bed can be seen in Fig. 14. Both of these designs have been optimised for 3D printing with a fused deposition modelling (FDM) printer. The flywheel designed to include holes for the addition of mass around its perimeter using bolts, hence increasing its moment of inertia.

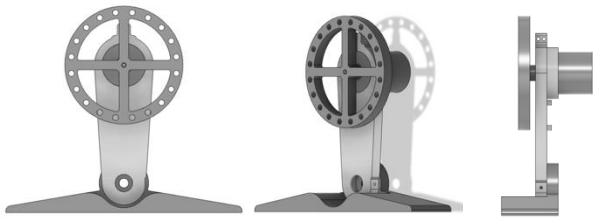


Figure 13, showing the one-dimensional IP test bed CAD model (within Fusion360)

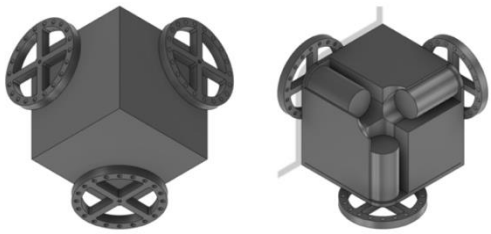


Figure 14, showing two/three-dimensional IP test bed CAD model (within Fusion360)

One two/three-dimensional circuit diagram

The circuit diagram for the one, two and three-dimensional IP systems can be seen in Fig. 15, showing the final component selection, with the addition of an extra H-bridge and motor for each additional axis.

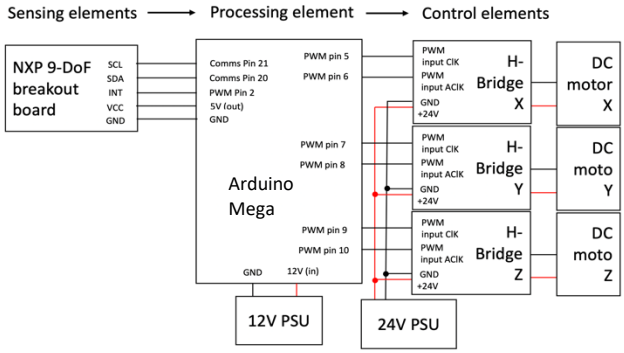


Figure 15, showing the circuit block diagram for the one, two and three-dimensional IP systems

Implementation of the LQR controller using Arduino

The proposed structure for the Arduino code can be seen in Fig. 16. In order to apply the theory explored within this paper an 'observer' must be used in order to reconstruct all system states within the IP state matrix from the angle theta measured by the IMU and calculated using Mahony AHRS. This should not be an issue, as all states within the matrix can be reconstructed simply by implementing an integrator. All of the necessary libraries are available for use with example code within the Arduino integrated development environment (IDE) for the Adafruit's sensor suite, Mahony AHRS and state space controller. (The equilibrium offset orientation must be measured to correctly configure AHRS.) Due to the open source nature of Arduino, all libraries and hardware (Intellectual Property) used would be suitable for mass production of a final product without further licencing.

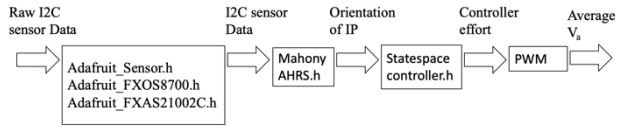


Figure 16, showing a flowchart of the proposed Arduino C programme structure and libraries

V. RESULTS & EVALUATION

Testing within MATLAB and Simulink

Determining state space properties

Using the Sc=ctrb(sys) command within MATLAB the system controllability was found to be of the 4th order. Hence with the use of a feedback controller the system is in theory fully controllable. Using the So=obsv(sys) command the system observability was found to be of the 3rd order. Hence all of the internal states of the system cannot be directly inferred via the output. This should not present any issues as the internal states of the system can be inferred via other means, in this case integrating $\dot{\varphi}$ to give φ hence allowing for the reconstruction of all states. Using the command rlocus(sys), a root locus diagram of the system was produced within MATLAB. As seen in Fig. 17, the system has poles to the right-hand side of the imaginary axis, showing the system to be unstable with there being no value of proportional gain that will successfully drive the system into equilibrium. It is for this reason we will be using a state space feedback controller.

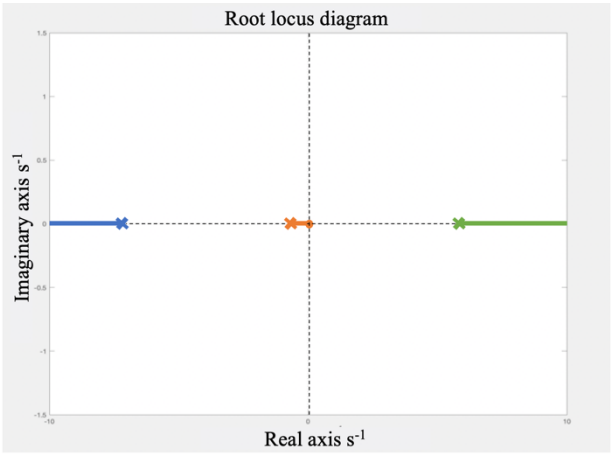


Figure 17, showing the IP system root locus diagram

Control of the DC motor

Using the control architecture seen in Fig. 12 the applied armature voltage can be more accurately controlled, hence the motor torque and angular position of the pendulum. Using the optimised PID constant calculated in MATLAB K_p=-38000, K_i=-14800, K_d=-2333 the step response of the motor was greatly improved as seen when comparing Fig. 18 & 19, allowing for tighter control of the pendulum angle, with little overshoot or oscillation seen. The large values of gain provide high performance but show it is difficult to accurately drive this system to a set point.

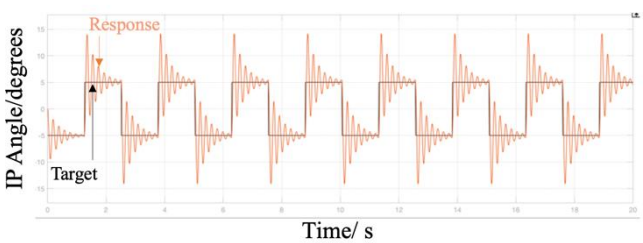


Figure 18, showing resultant IP angle for an applied armature voltage, without optimised PID constants

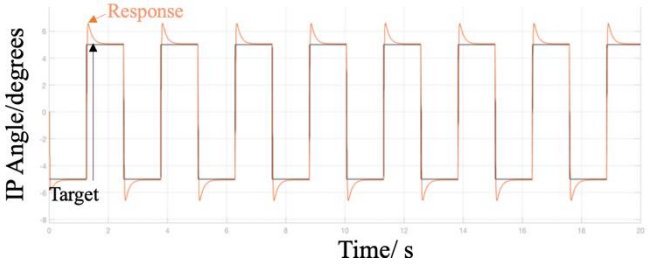


Figure 19, showing resultant IP angle for an applied armature voltage, with optimised PID constants

IP stabilisation using LQR feedback control

Using the model shown in Fig. 12 the entire system was tested to determine if stability could be achieved. To ensure the LQR feedback was as robust as possible, the disturbance placed upon the system consisted of impulses of various size and duration. In addition to this random noise was added to approximate real-world disturbances and variation within sensors reading. Fig. 20 shows the internal states of the IP system when a disturbance is applied, showing the controller to be working as expected, exerting a response to oppose that added to bring the state space variables to zero, hence driving theta towards zero at equilibrium. The control matrix K_lqr calculated has the value [-1935.97; -268.21; -12.25; -27.32].

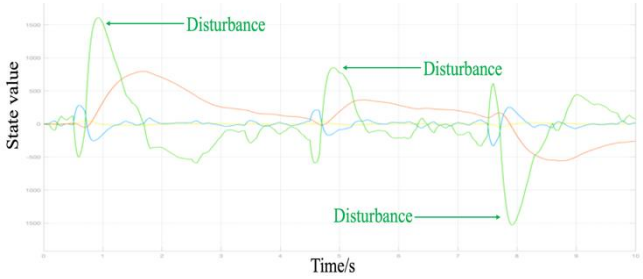


Figure 20, showing the LQR controller effort compared to the disturbance effort applied

Fig. 21 shows the disturbance effort applied to the IP (orange), along with the controller effort (blue) and angular position of the pendulum (purple). The figure shows the IP system successfully rejecting the disturbance applied, with the controller effort opposing the disturbance effort to bring the angular IP position to zero degrees. Despite the response being appropriate, maintaining equilibrium it may be further improved, to provide a more critically damped response with reduced oscillation by alteration of R and Q when calculating K_lqr.

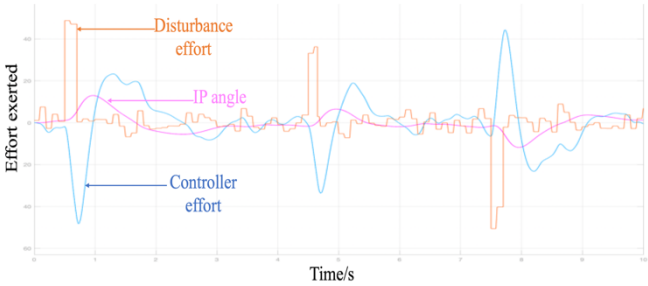


Figure 21, showing the LQR state space control variables when compared to the input disturbance applied

Despite the simulated system showing stability, many parameters were assumed, in conjunction with this the control system could not be tested in hardware for comparison (due to unforeseen circumstances, see appendix). Due to these factors the real system behaviour may be very different to that seen in the simulation.

VI. Discussion

This paper successfully demonstrates the tools, methods and considerations required when modelling and designing an LQR state space control system for an inertial wheel-based IP. As shown within the results of this project the one-dimensional IP was successfully modelled and stabilised within MATLAB & Simulink. This success is based on prior work to derive the transfer functions and state space representations, for the DC motor and IP systems, along with the cascade feedback architecture used [8, 11].

The results show how to implement PID to improve the motors response as seen in Fig. 19, hence improving the response of the system as a whole. In conjunction with this Fig. 20 & 21 show the robust LQR feedback system successfully driving the pendulum to its equilibrium position (upright), over a range of disturbance amplitudes, durations and noise levels. Furthermore, the results shown are within the specifications such as sample time and current draw of the components selected. Many of the project objectives were successfully met, with this report exploring a wealth of relevant resources to develop the understanding necessary to model the system, as well as to design an appropriate method of feedback by implementing the necessary engineering, mathematic and scientific principles to do so.

Despite the paper successfully demonstrating the methods required to simulate and construct an inertial wheel IP system, it is ultimately unable to verify the models created. With the project being on track this is due to the extenuating circumstances seen within the Gannt chart (within the appendix), leading to a closure of project labs, hence preventing final tests of the DC motor and construction of the one two and three-dimensional test beds for comparison to the simulation.

VII. Conclusion

Unfortunately, due to the unforeseen circumstances (seen in the appendix) not all of the project aim and objective could not be met. This is due to the reliance of the project on both physical testing and requirements for further hardware. Because of this it was not possible to conclude weather or not the models were accurate, or if the feedback system implemented within Simulink would provide stabilisation of individual axes when applied to the physical test bed. Furthermore, the main research question remains unanswered, as this paper cannot conclude weather or not stabilisation of individual axes will provide global stabilisation in the two and three-dimensional cases.

Despite the research question remaining open, this paper successfully demonstrates how to model a DC motor and IP system to produce a robust cascaded PID LQR feedback system, achieving stabilisation within Simulink. Due to lack of physical testing the model is limited as certain system parameters could not be obtained, hence the model and feedback mechanism cannot be verified. Regardless, the method demonstrates how knowledge from many different disciplines must be combined and applied to create a complete physical and electrical system, in simulations and hardware.

This paper demonstrated the power of tools such as MATLAB, Simulink and Fusion360 when designing complex non-linear systems. Furthermore, it demonstrates the power of open source devices such as Arduino for testing and implementing bespoke programmes. The use of Arduino emphasises the importance of open source libraries, allowing for the IMU to be easily integrated alongside an AHRS and state space controller. For these reasons the Arduino has allowed for flexibility within the design approach and component selection. Arduino also allows for the libraries and microcontrollers used to be implemented into the final device production, without infringement on intellectual property, encouraging innovation.

Although the work outlined doesn't demonstrate any new findings, it successfully builds on previous works to demonstrate the importance of control theory and resultant algorithms such as LQR state space control. This importance is as a result of the breadth of possible applications within modern technology, from personal transportation to manufacturing to space exploration. The largest impact of this work may be within education, providing an outline for students to understand and develop control algorithms. Allowing for physical implementation and testing within affordable hardware such as Arduino, furthering the development of control techniques.

VIII. Future Work

There is a large amount of work remaining, in part due to the extenuating circumstances outlined. The limitations and future work within this report revolve around physical testing to obtain motor and IP constants. Final construction of the one, two and three-dimensional testbeds remains. Allowing for testing and verification of the model as well as refinement of feedback method and constants. This would allow for the main research question to be answered, determining if individual stabilisation of each axis would provide global stabilisation in two and three dimensions. Further physical testing would also allow for the LQR state space and PID control constants to be further refined. The inertial wheel system within the IP system also has room for improvement, with future iterations utilising sensored brushless motors and rotary encoders to improve efficiency, reduce overall inertia and increase the accuracy of the feedback.

Outside of the physical testing required there is a large scope for the project to be developed. For example, the implementation of swing up control would allow for the pendulum to reach the equilibrium position without first being placed there. This can be seen within the Cubli and would provide a direct use case within the M-block 2.0 when self-assembling into larger structures [24, 10]. Furthermore, the hardware may be improved by the addition of a rotary encoder on the motor's rotor. Allowing the angular velocity, hence motor torque to be more accurately measured to provide a more robust method of feedback when compared to the DC motor transfer function.

Development of this project into a final self-contained device would allow for research to take place. Allowing for limitations and further possible use cases of this technology to be explored. More specifically when considering the control of linear or rotational torques to an objects body, in order to control its motion without external actuation. Possible testing procedures may be used to evaluate the behaviour of a system such as this within alternative environments such as water, vacuum or zero-g. Construction of a complete three-dimensional control model would greatly aid the development of this technology.

REFERENCES.

[1]. C. M. Hare, "EEE Individual Design Project Interim Report", University of Sheffield, Sheffield, UK. Paper Unknown, 2020.

[2]. R. Babazadeh, A.G. Khiabani, H. Amzi, "Nonlinear Modeling and Optimal Output Control of Two Wheeled Balancing Transporter", University of Technology, Tehran, Iran. Paper Unknown, Unknown.

[3]. Z. Tahir, W. Tahir, S.A. Liaqat, "State Space System modelling of a Quad Copter UAV", Unknown. Paper Unknown, Unknown.

[4]. T. Letcher, "Storing Energy", 1st Edition, Netherlands, Amsterdam, Elsevier. Paper 9781483145488, January, 1980.

[5]. R. Gordon, "Self-transforming robot blocks jump, spin, flip and identify each other", MIT News, October, 2018. [Online] Available: http://news.mit.edu/2019/self-transforming-robot-blocks-jump-spin-flip-identify-each-other-1030 (Accessed: May, 05, 2020).

[6]. M. Hulsmann, R. Kahle, M. Schneller, S. Aida, M. Hahn, E. Panzenbock, A. Sporl, T. Terzibaschain, W. Halle, "Debris collision avoidance by means of attitude control-in-flight demonstration with TET-1", Journal of Space safety Engineering, vol 6. Issue 4. Paper 284-290, December, 2019.

[7]. E. Gibney, "How to build a moon base", Nature, [Online] Available: https://www-nature-com.sheffield.idm.oclc.org/articles/d41586-018-07107-4 (Accessed: May, 05, 2020).

[8]. M. Olivares, P. Albertos, "On the linear control of underactuated systems: the flywheel inverted pendulum", IEEE International Conference on Control and Automation, Hangzhou, China. Paper Unknown, June, 2013.

[9]. D. Collins, "What is state space control?", Motion Control Tips, Febuary, 2018. [Online] Available: https://www.motioncontroltips.com/what-is-state-space-control/ (Accessed: May, 05, 2020).

[10]. D. Rowell, "State-space Representation of LTI Systems", Analysis and Design of Feedback Control Systems, October, 2002. [Online] Available: http://web.mit.edu/2.14/www/Handouts/StateSpace.pdf (Accessed: May, 05, 2020).

[11]. J. R. C. Vasconcelos, M. E. Astorga Gonzalez, P. M. Del Foyo, "DESIGN AND CONTROL OF A FLYWHEEL INVERTED PENDULUM SYSTEM", University Federal de Pernambuco, Recife, Brazil. Paper 50670-901, Unknown.

[12]. S. A. Ludwig, K. D. Burnham, A. R. Jimenez, P. A. Touma, "Comparison of Attitude and Heading Reference Systems using Foot Mounted MIMU Sensor Data: Basic, Madgwick and Mahony", North Dakota State University, Fargo, USA. Paper Unknown, Unknown.

[13]. E. M. Diaz, F. de Ponte Muller, A. R. Jimeneza, "Evaluation of AHRS Algorithms for Inertial Personal Locolization in Industrial Environments", German Aerospace Centre, Wessling, Germany. Centre for Automation and Robotics, Madrid, Spain. Paper 978-1-4799-7799-4/15, Unknown.

[14]. M. Euston, P. Coote, R. Mahony, J. Kim, T. Hamel "A Complementary Filter for Attitude Estimation of a Fixed-Wing UAV", Australian National University, Canberra, Australia. Paper Unknown, Unknown.

[15]. R. Bars, P. Colaneri, C. E. de Souza, L. Dugard, F. Allgöwer, A. Kleimenov, et al. Theory, "algorithms and technology in the design of control systems," Annual Reviews in Control, Vol. 30, Paper 19-30, 2006.

[16]. R. Bars, P. Colaneri, L. Dugard, F. Allgower, A. Kleimenov, C. W. Scherer, "Trends in theory of control system design - status report by the IFAC coordinating committee", Proceedings of the 17th IFAC World Congress, Séoul, Vol. 17, Paper 93-114, 2008.

[17]. P. Horác9ek, "Laboratory experiments for control theory courses: A survey," Annual Reviews in Control, vol. 24, Paper 151-162, 2000.

[18]. N. A. Kheir, K. J. Åström, D. Auslander, K. C. Cheok, G. F. Franklin, M. Masten, et al. "Control systems engineeringeducation," Automatica, Vol. 32, Paper 147-166, 1996.

[19]. A. Ollero, S. Boverie, R. Goodall, J. Sasiadek, H. Erbe, D. Zuehlke, "Mechatronics, robotics and components for automation and control, IFAC milestone report.," Annual Reviews in Control, Vol. 30, Paper 41-54, 2006.

[20]. J. B. Weinberg, G. L. Engel, K. Gu, C. S. Karacal, S. R. Smith, W. W. White, et al., "A multidisciplinary model for using robotics in engineering education", Proceedings of Annual Conference & Exposition of the American Society for Engineering Education, Albuquerque, Paper 693-701, 2001.

[21]. R. Cellan-Jones, "Robots 'to replace up to 20 million factory jobs' by 2030", BBC News, June, 2019. [Online] Available: https://www.bbc.co.uk/news/business-48760799 (Accessed: May, 05, 2020).

[22]. Mellor Electrics Limited, "SC9 Series Motor", Mellor Electrics Limited, Unknown. [Online] Available: https://docs.rs-online.com/6f54/0900766b815dee2a.pdf (Accessed: May, 05, 2020).

[23]. Tektronix Inc, "Capacitance and Inductance Measurements Using an Oscilloscope and a Function Generator", Tek, Unknown. [Online] Available: https://www.tek.com/document/application-note/capacitance-and-inductance-measurements-using-oscilloscope-and-function-ge (Accessed: May, 05, 2020).

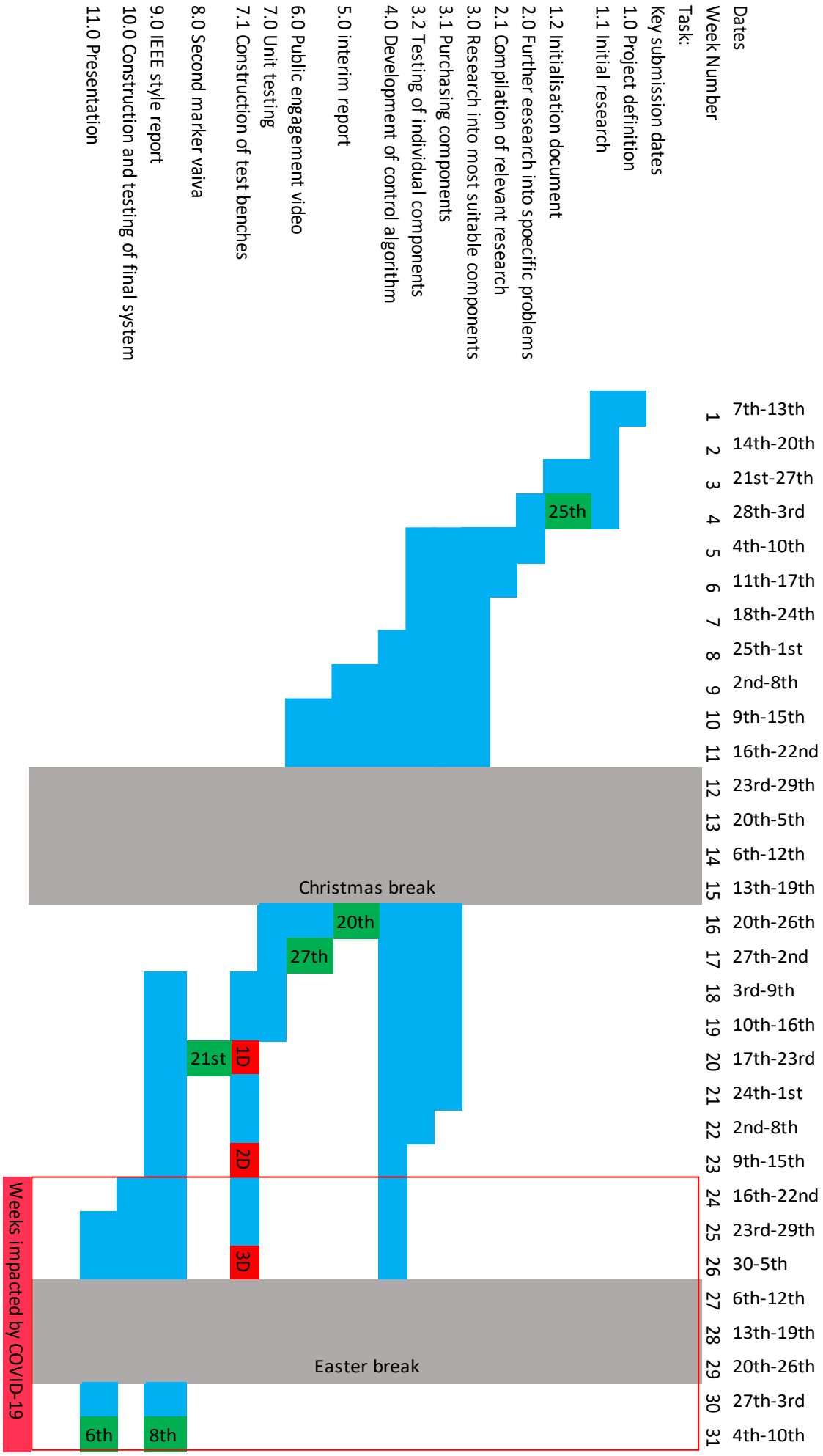
[24]. M. Gajamohan, M. Merz, I. Thommen, R. D'Andrea "The Cubli: A Cube that can Jump Up and Balance", IEEE/RSJ International Conference on Robota and Systems, Vilamoura, Portugal. Paper Unknown, 2012.

APPENDICIES.

Table 1 , Parameters used

Parameter name	value
b (Nm/rad), Friction about IP pivot	0.02
b_m (Nm/rad), Friction about motor shaft	3e-04
L_1 (m), length to IP centre of mass	0.063
L_2 (m), Length to flywheel centre	0.125
m_1 (Kg), Mass of IP	0.1
m_2 (Kg), Mass of motor & flywheel	0.42
j_1 (Kg.m^2), Inertia of IP	0.47e-03
j_2 (Kg.m^2), Inertia of flywheel	6.4e-03
K, Motor constant	18.2e-03
R_a (Ω), Armature resistance	0.83
L_a (H), Armature inductance	3e-03
G (m/s)	9.82
K1	0.14
K2	0.63

Project Gantt Chart



MATLAB code

```
%define parameters
% inertia of flywheel - J
J = 3.2284E-6;
x0 = [0; 5*pi/180; 0; 0];
xm0 = [0; 0; 0; 0];
x1 = [0 0 0 0];
x2 = [1 0 0 0];

ke = 18.2e-03;
ra = 0.83;
g = 9.82;
la = 3e-03;
bm = 3e-04;
m1 = 0.1;
m2 = 0.42;
l1 = 0.063;
l2 = 0.125;
j1 = 0.47e-03;
j2 = 6.4e-03;
k1 = 0.14;
k2 = 0.63;
j = (m1*l1*l1)+(m2*l2*l2)+j1+j2;
b = 0.02;
ml = (m1*l1)+(m2*l2);
Ts= 0.02;

A = [0              1              0
0;
     (ml*g)/j       -b/j           0
((j2*k2)/j);
     0              0              0
1;
     0              0              0
-k2];

B = [0; (-j2*k1)/j; 0; k1];
B1 = [0 (-j2*k1)/j 0 k1];

%output
% C =  [0; 1; 0; 0] %q2 qs output
C = [1; 0; 0; 0]; %q1 as output
D = 0;
%build the system in state space
sys = ss(A,B,C',D);
%analysis of system
Sc = ctrb(sys) %controllability matrix
So = obsv(sys) %observability matrix
eig(A)
pole(sys)
rlocus(sys)
%if pole on rhs of s plane - not stable
%control of system
des_pole_d = [0.3; 0.3; 0.3; 0.3];
k_d = acker(A,B, des_pole_d)
Q = 20*eye(4);
R = 0.1;
K_lqr=lqr(A,B,Q,R);

dist_cl = ss(Aa-Ba*Ka,[0 ; 1/J ; 0;
0],Ca,Da);
step(dist_cl,t)
Kc = place(A,B,[p1, p2, p3, p4])
```

Risk matrix

Risk Number	Description of Risk	Mitigation of Risk	Risk evaluation (L/M/H)	Chance of risk occurring (L/M/H)
~~1~~	~~Unable to program a successful control algorithm (H/M)~~	~~Use of adequate research from peer reviewed sources, and programming in a familiar language (C/C+) (M/M)~~	~~M~~	~~M~~
~~2~~	~~Unable to successfully create an appropriate mathematical model of the inverted pendulum with chosen software (H/M)~~	~~Use of familiar software and adequate research from peer reviewed sources containing appropriate case studies (H/L)~~	~~H~~	~~L~~
~~3~~	~~Unable to find source suitable and compatible electronics (motor, speed controller, sensors, microprocessors) (H/H)~~	~~Use of well understood, well documented, open source, universally compatible electronics (I.e Arduino) (H/L)~~	~~H~~	~~L~~
~~4~~	~~Loss of critical project data due to catastrophic failure of data to the laptop (H/M)~~	~~Regularly backing up project of data to the cloud (L/L)~~	~~L~~	~~L~~
5	Unable to source all components through university approved suppliers within given timeframe (M/M)	Discussion with tutor and use of the provided system to source components from other suppliers (L/L)	L	L
6	Unable to expand and apply theory used for one dimensional test to two and three-dimensional cases (M/M)	Use of adequate peer reviewed research on similar three-dimensional inverted pendulum systems (L/L)	L	L
7	Unable to produce a final system that meets all project specification within the given timeframe (M/M)	Breaking the project up in-to individual sections & use project management tools such as Gantt chart (L/L)	L	L
8	Unable to produce test beds and final system within the set project budget (H/L)	Discussion with tutor when purchasing components & use of components already available within the department (parts- bin) (L/L)	L	L
9	Global pandemic (i.e COVID-19) preventing access to project labs.	Reduce reliance on physical testing by using simulation - based techniques in MATLAB and Simulink to evaluate behaviour of system.	L	H