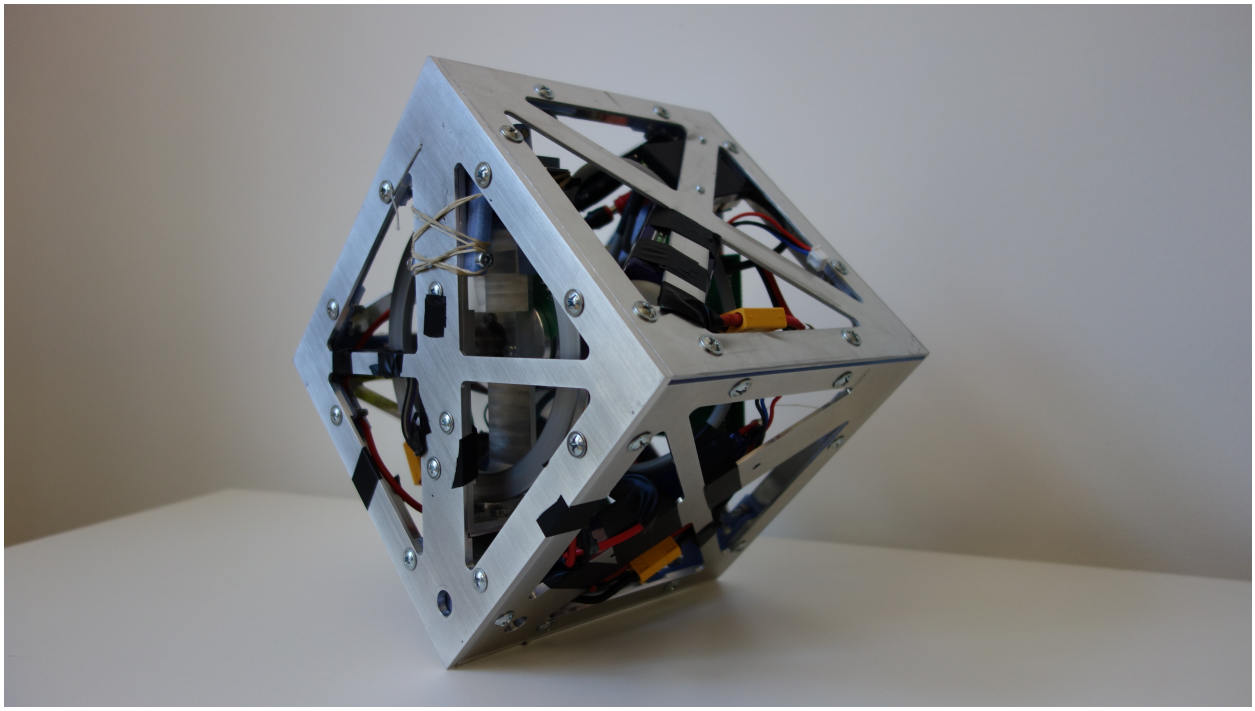
CHALMERS



Development of a Nonlinear Mechatronic Cube

The Jumping and Balancing Cube
Master of Science Thesis

ERIK BJERKE
BJÖRN PEHRSSON

Department of Signal and Systems
Division of Automatic control, Automation and Mechatronics
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2016

Master's thesis 2016:02

Development of a Nonlinear Mechatronic Cube

The Jumping and Balancing Cube

Erik Bjerke
Björn Pehrsson

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Signals and Systems
Division of Automatic control, Automation and Mechatronics
Chalmers University of Technology
Gothenburg, Sweden 2016

Development of a Nonlinear Mechatronic Cube:
The Jumping and Balancing Cube
Erik Bjerke
Björn Pehrsson

Supervisor: Thomas Hedberg, Combine Control Systems AB
Examiner: Petter Falkman, Signals and System

Master's Thesis 2016:02
Department of Signals and Systems
Division of Automatic control, Automation and Mechatronics
Chalmers University of Technology
SE-412 96 Gothenburg

Cover: Picture of our system

Printed by [Chalmers Library, Reproservice]
Gothenburg, Sweden 2016

Development of a Nonlinear Mechatronic Cube
The Jumping and Balancing Cube
Erik Bjerke
Björn Pehrsson
Department of Signals and Systems
Division of Automatic control, Automation and Mechatronics
Chalmers University of Technology

Abstract

The goal of the project is to develop and construct a nonlinear mechatronics cube which is able to jump up from the surface to one of its edges and balance. The mechanical part of the cube was developed using the softwares CATIA V.5 and MATLAB/Simulink. The final cube is controlled by an LQR-controller with added integral states. The code was implemented using the software Waijung, which translated MATLAB/Simulink code to C-code.

The cube was able to balance on any surface and could resist small pushes up to 2 degrees. It also managed to successfully jump up and balance 4 out of 25 times.

Keywords: LQR, balancing cube, SIMULINK, Control theory, embedded systems

Acknowledgements

This report documents the development of a nonlinear mechatronics cube which is able to jump up and balance on one of its edges. The project was part of a master thesis at the Department of Signals and Systems, Division of Automation Control, Automation and Mechatronics at Chalmers University of Technology. The development of the cube took place at the company Combine AB, Sweden, Järntorget.

We would like to thank **Combine AB** for giving us the opportunity to develop our cube at their facilities. This project highly influenced by the people working at the company. Combine gave us economic support as well as expert help in several areas during development.

We would like to thank the following people for their contribution in specific areas:

Annie Johansson, Stefan Holmström, 钟雪芬 **, Samuel Johnsson** and **Ismael Shojai**, for all the help with modeling the cube in CATIA and Solidworks. This gave us the opportunity to model the cube in Simmechanics and also some of the pictures for the presentation movie.

Foad Mohammedi, **Mattias Henriksson** and **Fredrik Olin** for all the help during the physical modeling of the cube and selection of components.

Göran Stigler, Reine Nohlborg and **Jan Bragee** working at Prototyplabbet at Chalmers, for helping us construct early 3D printed models of the cube, and also laser cut three of our sides on the cube.

Petter Falkman, our examiner, for all the help with the report and support during the thesis project.

Thomas Hedberg, our supervisor. First of all thank you for believing that this project was possible and continuously pushing the project forward by supporting with ideas, material and important contacts.

Erik and Björn
Gothenburg Feb 2016

Contents

1
Introduction

1.1 Background

Humans have always been interested in exploring new areas. Since developing the first telescope, we have been able to see further out and a new step of exploration started. In 1959 the first spacecraft, Luna 2, was able to land and explore the moon [1]. The moon was in focus for decades but from the later part of 1990's asteroids, comets and meteorites had started to draw attentions across the globe [2]. In a space exploring mission, different areas are covered. Everything from launch to the mobility system technology in charge of exploring new asteroids, comet or meteorites. A lot of different research in developing mobility systems have been done and today we have wheel-, leg-, track-enabled systems as well as hybrid, sliding, aerial, rolling and hopper systems. In all of these areas, Developments have been made, highlighting advantages and disadvantages with different methods [3]. A big challenge when exploring asteroids is the gravitational force that can be as low as 10 μg, whereas earth has 1 g. In this area a lot of the developed systems were not working, but in 2005 the Institute of Space and Astronautical Science of Japan, ISAS, launched a hopper system named *HAYABUSA* [2]. It used a reaction wheel to move forward and backward. With a small acceleration on the reaction wheel it could just move shortly but with a larger acceleration it could be able to move over bigger objects .

This thesis focuses on the same method that was used in the *HAYABUSA* system, but instead of using just one reaction wheel, the developed product consists of three reaction wheels. The extra reaction wheels allow the system to move in all possible directions and with high accuracy.

1.2 Previous work

Inverted pendulum systems have been a heavily researched area in the control community for the last century. They has been widely used within the area of dynamics and control theory for bench-marketing different control algorithms. The balancing cube which is developed in this thesis will have the most common features with the double pendulum [4] which could be considered as a system with two masses which have 4 degrees of freedom and the 3D-pendulum [5] with 6 degrees of freedom.

Earlier projects on cubes have been done by a research team, led by Michael Muehlebach and Prof. Raffaello D'Andrea where a 15x15 cm cube was able to go from resting position up to balancing on one of the edges, and then jump up to one of the corners. The cube has three features; Jumping up, balancing and controlled falling, walking, [6].

Another related project is M-Blocks, which are momentum driven magnetic cubes [7]. They use a reaction wheel and a brake system to move around. They can move freely in one direction using the reaction wheel, but as a group they move in more directions.

1.3 Purpose and aims

The main purpose was to develop a cube which is able to jump up from being on its surface to one of its edges and balance there for at least 25 seconds. A cube which is positioned with one of its faces at the surface is denoted resting position, while cube which is standing at one of its edges is denoted balancing position. The plan is to have the system built as robust as possible. The cube should be able to resist small software and hardware disturbances and also external disturbances. One additional goal is that the cube should have good repetitiveness achieving the main goal, i.e. jump up and balance several times in a row. To verify and achieve this purpose the project had the following milestones:

1.3.1 Balance with external disturbance

The first aim is that the cube, while balancing, could resist small external disturbances. For example adding pressure to a button on an interface of the cube, or a person pushing on the side of the cube. The disturbance will be measured and validated as a change of the angle from the balancing position of the cube. The cube should be able to balance more than 25 s, and the repetitiveness aim is that the cube should manage not to fall 25 times in a row.

1.3.2 Balance on various surfaces

The second aim is that the cube should be able to balance on different surfaces, with various friction and be able to balance on surfaces which are tilted up to 45 degrees.

1.3.3 Jump up within a angle space close to balancing angle

The third aim is that the cube should be able to jump up from -45/+45 degrees starting in resting position, up to a balancing point, which limited to +-2 degrees from balancing point. The cube should be able to achieve a jump within this area 25 times in a row.

1.3.4 Walking ability

The forth aim is that the cube should be able to walk, i.e. jump up from all four sides and fall down without damaging itself.

1.3.5 Further goals

This thesis is the first stage of a bigger project. The aim is that the mechanical models, electrical components and software are designed in a way that all parts could be used in the bigger project which would involve balancing on one corner.

Jump up and balance on one of its corners

Later goal, outside this thesis is to develop a cube which would be able to jump up and balance any corner, with help of three reaction wheels.

1.4 Problem

The main scientific challenge is the development of system design. The overall system could be divided into two subsystems, jumping up from lying on one of the sides to the edge, named Jumping Up action and balancing on the edge named Balancing action. The system design could be divided into hardware and software parts.

The hardware design include construction and design selection of all the mechanical components such as reaction wheels, brake-system and selection of all electrical components such as actuators, sensors and controller cards.

As for the software design part, it includes development of a control algorithm for both Jumping Up action and Balancing action using the modeling software MATLAB, Simulink and Waijung Blockset. It also includes design and encoding of communication protocol.

The project consisted of several parts; Modeling, Sensors, Control and Braking-dynamic. The systems were not linearly connected to each other, which would give rise to several of optimization problems in-between the subsystems. The most critical optimization problem was in between Jump Up action and Balancing action according to placement of center of gravity, CoG, point in the cube. Jumping action strives to have CoG point as low as possible reducing the work to be done, and Balancing action strives to have CoG as high as possible.

1.5 Scope

This thesis focused both on the theoretical and the physical development of a cube. Developing different control methods for the balancing cube, which would be validated both by theoretical simulation models and tested at the constructed cube.

The main objective was to have a system which was able to jump up and balance. As well as achieve a stable system, i.e. keep the cube balancing on one of the edges for more than 25.

1.6 Method

The work flow is summarized in several steps, starting with a brief literature study of earlier projects which were similar. This involving a detailed understanding of all included sub problems and methods for the system. Understanding the main part of the physics of the system, and the relationship between hardware and software design will give a requirement specification, based on main functionality and performance of the final product. The next step includes constructing theoretical models of the whole system iteratively using CATIA and MATLAB. The hardware includes modeling of each part in CATIA and the software part included developing control algorithms and sensor fusion. The goal is to select all the mechanical and electrical components and validate it by a theoretical controller that met all the requirements in the specification. This is a process, where each component selected would give more understanding of the overall system, and how each sub components effects the final product. Therefore the component selection is an iterative process, where an earlier selected component could be changed in later iteration. Prototype of some selected components would be evaluated by test construction in plastic material. The following step is to produce a test rig, where the selected sensors, actuators and different mechanical solutions can be tested separately from each other and in this way validation of each component could be done. The last step is to combine all components together, where a comparison in between simulation models and real-product could be done.

1.7 Outline

The report starts with a modeling part where each of the systems, edge balancing and corner balancing, equations of motion were derived.

The third and the fourth chapters use the equations of motion to continue developing the functionality of the cube, both the Balancing action as well as the Jump Up action.

In order to balance an unstable system, a controller needs to be implemented. In chapter five, different control methods will be described and validated in MATLAB.

In order to test if the equations of motions, the developed system and the control work properly, a physical cube has been developed. In chapter six, the selected components which include hardware, construction material and software are described.

Chapter seven includes test results of the real system. It contains different controllers and plots that demonstrate how well the system behaved.

2
Modeling

This chapter explains the theoretical modeling and dynamics of the cube. It starts with a theoretical part about the methods involved. The next parts include the implementation of the theory and development of the equations of motion for the Edge balancing system.

2.1 Modeling theory

This section summarizes the background theory used while modeling the cube.

2.1.1 Lagrangian

There are different methods for modeling mechanical systems; Newtonian mechanics, Hamiltonian and Lagrangian method. The Lagrangian and Newtonian approach to modeling dynamic systems end up in the same equations of motion, but since the systems become more complex, i.e. several degrees of freedom and more than one mass which rotate with constraints of motion. The Lagrangian method gives an easier way to find the solution of equations of motion [8].

The Lagrangian method can be formulated by selecting a set of generalized coordinates q_i which describe the configuration in all degrees of freedom, DOF, relative to a reference system. The numbers of coordinates chosen equal to the number of DOF. Then the lagrangian is formulated by form kinetic energy, $T(\dot{q}_i)$ and potential energy $V(q_i)$ which gives $L = T - V$ [8]. To solve the Lagrangian, the Euler-Lagrange equations is formulated as:

$$\frac{d}{dt}(\frac{\partial L}{\partial \dot{q}_k}) - \frac{\partial L}{\partial q_k} + \frac{\partial R}{\partial \dot{q}_k} = \tau_k \tag{2.1}$$

Where τ_k is the nonpartial torques and the $\frac{\partial R}{\partial \dot{q}_k}$ models the dissipative forces.

2.2 Derivations of mathematical models

This section derives the equations of motions for the edge balancing model and derives a state-base model.

2.2.1 Potential and Kinetic Energy

The potential energy of the whole system can be modeled as

$$P = L_t m_{tot} \mathbf{g} Cos\theta_{cube} \tag{2.2}$$

where L_t is the length of a vector starting at the edge of the cube to the cubes center of mass. The total mass of the cube is represented as m_{tot} and the gravitational force represented as **g** and where θ_{cube} is the angle between cube and surface seen in *Figure 2.1* and θ_{wheel} representing the reaction wheel angle, which is in relation to the cubes coordinate system.

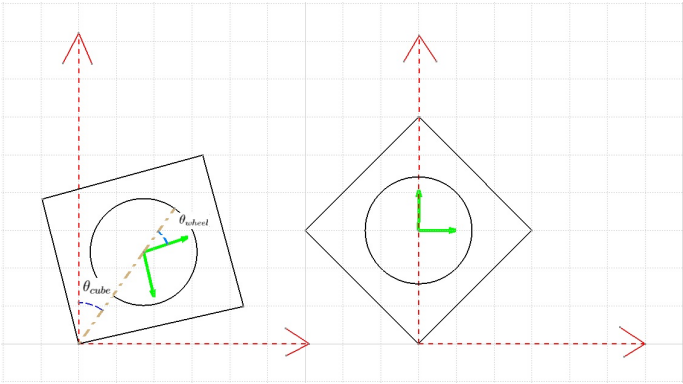


Figure 2.1: Defining the variable θ_{cube} and θ_{wheel} of the cube. The right part of the figure represents the balancing position, where both of the variables are zero.

Kinetic energy T, is formulated as the moment of inertia, MOI, around the systems rotational point times an angle velocity of an object moving around this point. The two objects in this case are the cube and the reaction wheel. The angle velocities around this point for the cube equals $\dot{\theta}^2_{cube}$ and the velocity of the reaction wheel equals $(\dot{\theta}_{cube} + \dot{\theta}_{wheel})^2$.

$$T = \frac{1}{2} I_{frame} \dot{\theta}^2_{cube} + \frac{1}{2} I_{wheel} (\dot{\theta}_{cube} + \dot{\theta}_{wheel})^2 \tag{2.3}$$

where I_{frame} represents the moment of inertia for the cube shell, including all parts except from the reaction wheel and the variable I_{wheel} represents the inertia of the reaction wheel.

2.2.2 Lagrangian

Combining the kinetic energy and the potential energy from *Equations 2.2 and 2.3* the Lagrangian can be formulated as

$$L = \frac{1}{2} I_{frame} \dot{\theta}^2_{cube} + \frac{1}{2} I_{wheel} (\dot{\theta}_{cube} + \dot{\theta}_{wheel})^2 - L_t m_{tot} g Cos\theta_{cube} \tag{2.4}$$

the generalized momenta which is formulated by derivatives of the Lagrangian with respect to the angle rate states. The following equations give the generalized

momenta derivatives with respect to time.

$$\frac{d}{dt}(\frac{\partial L}{\partial \dot{\theta}_{cube}}) = I_{frame}\ddot{\theta}_{cube} + I_{wheel}\ddot{\theta}_{wheel} = M_{\theta_{cube}} \tag{2.5}$$

$$\frac{d}{dt}(\frac{\partial L}{\partial \dot{\theta}_{wheel}}) = I_{wheel}(\ddot{\theta}_{cube} + \ddot{\theta}_{wheel}) = M_{\theta_{wheel}} \tag{2.6}$$

By formulating the derivatives of Lagrangian with respect to angles states and adding frictional forces on both cube, F_{cube} and reaction wheel, F_{wheel} the Euler Lagrange equations are calculated implementing T_{motor} as a non-potential force generated from the engine spinning the reaction wheel.

$$\frac{\partial L}{\partial \theta_{cube}} = L_t m_t g \sin\theta_{cube} + F_{cube}\dot{\theta}_{cube} \tag{2.7}$$

$$\frac{\partial L}{\partial \theta_{wheel}} = T_{motor} - F_{wheel}\dot{\theta}_{wheel} \tag{2.8}$$

The frictional forces are added to the Euler Lagrange equations by state the derivatives of R with respect to angular rate states.

$$\frac{\partial R}{\partial \dot{q}_k} = F_{cube}\dot{\theta}_{cube} \tag{2.9}$$

$$R = \frac{1}{2}F_{cube}\dot{\theta}^2_{cube} + \frac{1}{2}F_{wheel}\dot{\theta}^2_{wheel} \tag{2.10}$$

Finally, by combining the generalized momenta and Euler Lagrange equations the equations of motion can be formulated as

$$\ddot{\theta}_{cube} = \frac{L_t m_t g \sin\theta_{cube} - T_{motor} + F_{wheel}\dot{\theta}_{wheel} - F_{cube}\dot{\theta}_{cube}}{I_{frame}} \tag{2.11}$$

$$\ddot{\theta}_{wheel} = \frac{T_{motor}(I_{frame} + I_w) - F_w\dot{\theta}_{wheel}(I_{frame} + I_w) - L_t m_t g \sin\theta_{cube} I_w + F_{cube}\dot{\theta}_{cube} I_w}{I_{frame} I_w} \tag{2.12}$$

which describes the dynamics of the cube.

2.2.3 State-space model

The derived equations of motions, *Equation 2.11* and *Equation 2.12*, are used to form a state space model. A regular state-space model is defined as

$$\dot{x} = Ax + Bu \tag{2.13}$$

$$y = Cx + Du$$

However to be able to form the A and B matrices for the system, the equations have to be linearly dependent. By selecting the unstable equilibrium point, when the cube is standing on the edge, would give a controller that aims to keep the cube standing in a balanced position. Linearization around the states $(\theta_{cube}, \dot{\theta}_{cube}, \dot{\theta}_{wheel}) = (0, 0, 0)$ and by substituting T_{motor} to $K_m u$, where K_m is the motor constant and where u is the motor control signal, the system matrices is given by

$$A = \begin{bmatrix} 0 & 1 & 0 \\ \frac{L_t m_t g}{I_{frame}} & \frac{-F_{cube}}{I_{frame}} & \frac{F_{wheel}}{I_{frame}} \\ \frac{-L_t m_t g}{I_{frame}} & \frac{-F_{cube}}{I_{frame}} & \frac{-F_{wheel}(I_{frame}+I_{wheel})}{I_{frame} I_{wheel}} \end{bmatrix} \tag{2.14}$$

$$B = \begin{bmatrix} 0 \\ \frac{-K_m}{I_{frame}} \\ \frac{K_m(I_{frame}+I_{wheel})}{I_{frame} I_{wheel}} \end{bmatrix} \tag{2.15}$$

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.16}$$

$$D = \begin{bmatrix} 0 \end{bmatrix} \tag{2.17}$$

3

Balancing action

This chapter starts to explain what main components, hardware and software, that will effect the stability of a balancing cube. In the end, a calibration method for incorrectly estimated angles is described.

3.1 Recovery angle

The recovery angle is specified as the maximum angle relative to the balancing angle which the system can have before falling. The robustness to disturbance is directly effected by the recovery angle. The recovery angle is both hardware and software dependent.

In the hardware part the recovery angle is dependent on how fast the motor is and how much torque that the motor produce relative to the torque needed to rise the cube from a certain angle. The maximum angle can be calculated by

$$T_{motor} > g(m_{tot} - m_{wheel})\nabla_{cube}\sin(\theta_{cube}) + gm_{wheel}\nabla_{wheel}\sin(\theta_{cube}) \tag{3.1}$$

where ∇_{cube} is the relative change of center of gravity, CoG, of the cube and ∇_{wheel} is the relative change of CoG of the reaction wheel. If the reaction wheel is optimally balanced, i.e CoG is exactly in the middle of the reaction wheel, the ∇_{wheel} is equal to zero.

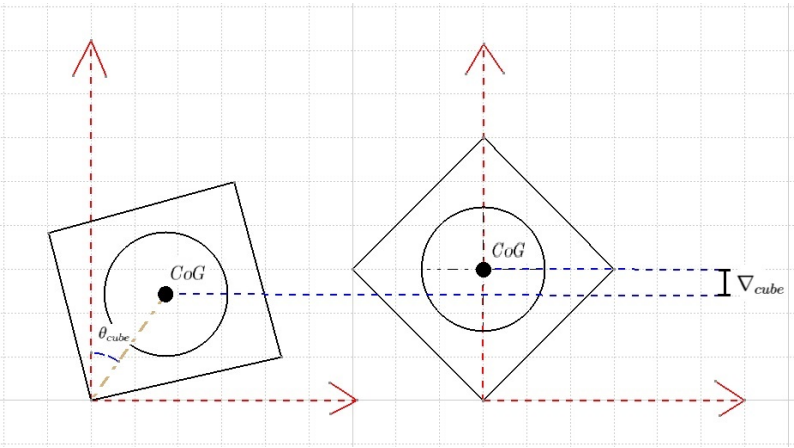


Figure 3.1: The figure demonstrates the relation between ∇_{cube}, the angle θ_{cube} and the center of gravity CoG.

Equation 3.1 is a simplification of the real system, where the engine is assumed to directly give full torque. A better approximation of the recovery angle will be discussed in Chapter *5 Control*.

In the software part, the recovery angle is affected by a limited sampling time of the processor. It is also affected by the amount of noise in the sensors that estimates the angle of the cube.

3.2 Adaptation to surfaces

The balancing point and the center of gravity, changes depending on how the surface is tilted. In some cases the reference angle, calculated during initialization, has an offset to the real balancing point. By adding an algorithm which update the reference angle while balancing the cube, the real and the estimated one would fit each other better. A better fit lowers the velocity of the reaction wheel due to that this offset will affect the output of the controller. An advantage with lower velocity is the saving of the battery, which increases the balancing time of the cube.

In the physical cube, this algorithm is implemented to start 1.5 s after the cube starts to balance. The algorithm starts to calculate the average measured angle during a period of 80 ms, which gives approximately how much the cube is tilted compared to the balancing edge. The calculated average is late on subtracted from the new angle that the sensor estimates.

$$angle_{updated} = angle_{original} - average_{measured} \tag{3.2}$$

Figure 3.2 on page *13*, illustrates the effect on the implemented algorithm in the real cube. The cube starts to balance after 9 s, approximately 1.5 s later, the algorithm starts to calculate the average angle offset during a period of 80 ms. As seen in the *Figure 3.2*, the reference angle where initially calibrated 0.5 degrees off and there can also be seen that this effect set the velocity of the reaction wheel close to 1500 rpm. One can also see that after the algorithm is finished both the angle of the cube, and the reaction wheel velocity enclosing zero.

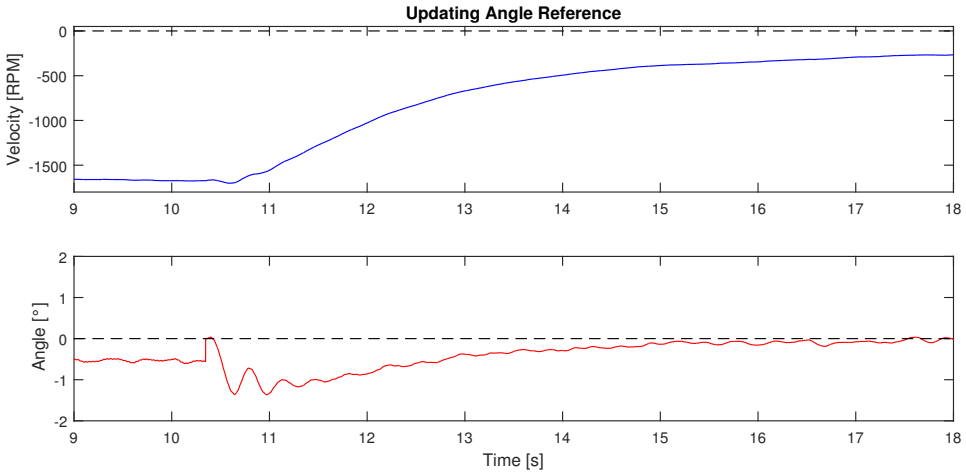


Figure 3.2: Illustrates a change of reference angle while balancing. The blue curve representing the measured velocity of the reaction wheel in rpm and the red curve represent the measured angle of the cube.

4

Jump Up action

The second behavior of the cube is to move from resting position up towards the edge balancing position, the so-called Jump Up action. This chapter starts with explaining previously used braking systems and the final selection of the final braking system implemented. Section continues with the calculations of the components that are included in the selected brake system. The last section includes a method to improve the accuracy rate of the brake system.

4.1 Selection of the Jump Up action system

The Jump Up action can be done by changing the rotational direction of the motor, i.e engine braking. This limits the maximum brake torque, which will be used to move the cube upward, to the maximum momentum the current engine can produce.

The amount of torque needed to move the cube up to balance position can be calculated, using Lagrangian Equation *2.4*. If assuming that the acceleration is equal to zero and no friction considered, either on the reaction wheel or on the cube the torque on the motor would need to be

$$T_{motor} > L_t m_t g \sin \theta_{cube} \tag{4.1}$$

where L_t is the length to the CoG, m_t the total weight of the cube and θ_{cube} the angle of the cube from the balancing position.

The maximum torque generated from a motor, and total mass of the motor are often related and higher requirements of torque follow with a heavier motor. A heavier motor effects the total mass of the cube, m_t, and a heavier cube gives higher requirements on the motor torque. In the 3D case, three motors are needed, a small increment of the mass of one motor increases the weight of the cube three times.

To enhance the performance, the energy from the reaction wheel has been considered. This can be done by braking the reaction wheel from a high velocity down to a lower velocity within a short amount of time. The torque generated during a brake procedure can be calculated according to

$$\tau_{wheel} = I_{wheel} * \ddot{\theta}_{wheel} \tag{4.2}$$

where τ_{wheel} is the braking torque and where the average deceleration,$\ddot{\theta}_{wheel}$ can be approximated to

$$\ddot{\theta}_{wheel} = \frac{\Delta\dot{\theta}_{wheel}}{60 * \Delta t_{braketime}} \tag{4.3}$$

One way to increase the energy using the theory from Equation 4.2 and 4.3, is to use a rubber belt brake system [7]. In this brake system a rubber belt is placed around the reaction wheel. When the belt tension is released, as represented in Figure 4.1, to the left, the reaction wheel can move freely. When the rubber belt is tightened, right side of Figure 4.1, the friction between the reaction wheel and the rubber belt will decrease the velocity of the reaction wheel. An advantage with this brake system is that the energy can both come from the force generated from the motor as well as from the rubber brake. One disadvantage though is that there will be a high force on the motor axis and the brake system can only brake the reaction wheel in one direction [7]. This means that with the one direction brake it will only allow the cube to move from 45 degrees to 0 degree or -45 degrees to 0 degree depending on the implementation.

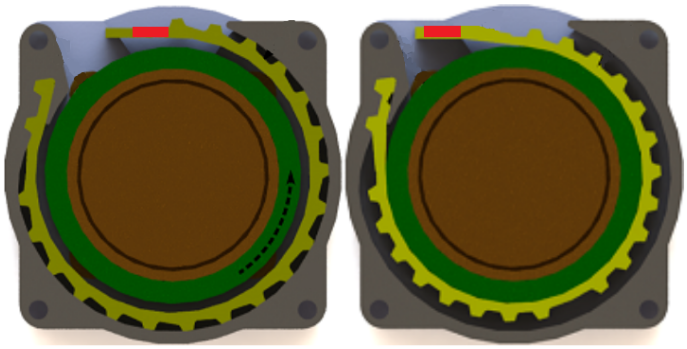


Figure 4.1: Rubber belt brake system [7]

In order to allow the cube to move in both directions during braking, a brake system with nonspecific direction has been implemented.

According to *Equation 4.2* and *Equation 4.3*, the reaction wheel deceleration, which depends on time and velocity has high influence on how much torque that will be generated and used to rotate the cube. In earlier projects different approaches have been established to brake the reaction wheel.

One approach is to attach a metal piece to the reaction wheel, in an area where the reaction wheel is enhanced. According to the project; The Cubli: A cube that can jump up and balance, the authors use an RC-servo to quickly collide with a metal barrier with a bolt head attached to the reaction wheel, see *Figure 4.2* [9]. The advantage with this implementation is that the reaction wheel will more or less stop instantly which gives a high moment for the cube compared to the case when the motor changes rotation direction while breaking. One disadvantage is that the break system is not possible to control between full brake or no brake. In further research, Cubli mentioned that this brake system worked properly in the

1D prototype but due to the high mass in the 3D case the large instant impulse force caused the structure of the reaction wheel to deform [10].

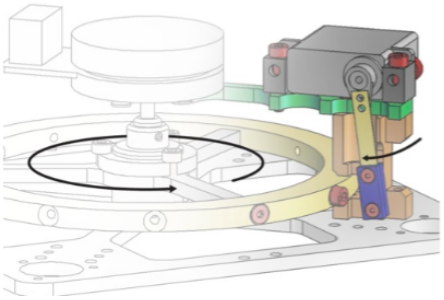


Figure 4.2: Metal barrier brake system [9]

To solve the deformation problem, the brake system can be designed according to Kris Temmermans, Rolling Cube videos [11], where a servomotor pushed a brake pad against the side of the reaction wheel, see *Figure 4.3*. The brake pad friction would slow down the reaction wheel and due to the high force from the servo, slightly bend the reaction wheel. The bending would move the reaction wheel so the second brake pad, placed on the other side, which helped the braking procedure. In this way, the servomotor can be selected with half of the calculated vertical brake force due to the normal force, according to

$$F_{brake} = 2F_{brakepad} * \mu \tag{4.4}$$

Where μ is the friction coefficient between the reaction wheel and the brake pad and where $F_{brakepad}$ is the force which presses against the brake pad. A disadvantage with this brake system is that the reaction wheel becomes slightly bent which would end up in an radial load on the motor. The radial load depends on how much the reaction wheel needs to bend before hitting the second brake bad. This depends on the reaction wheels construction in relation to wobbling. If the wobbling is high, the bending becomes larger due to the distance between the second brake pad and the reaction wheel. A small calibration mistake or movement in the structure would end up in a destroyed motor axis.

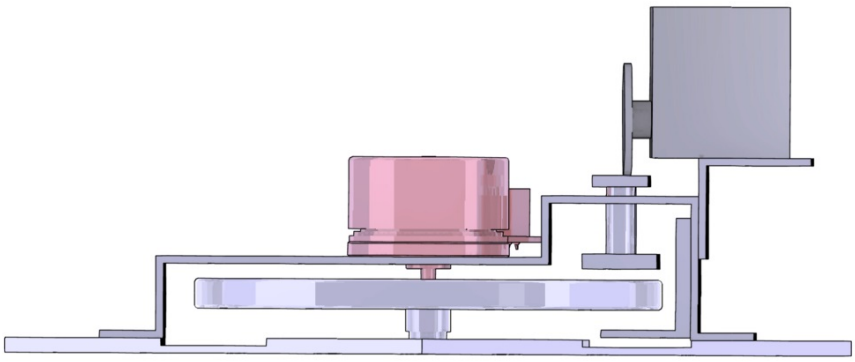
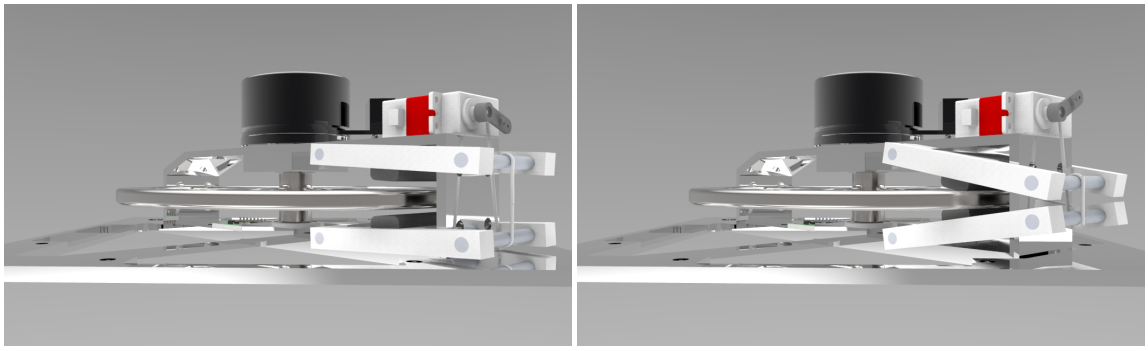


Figure 4.3: One side brake

The selected brake system is a combination of the previously mentioned systems, where all the disadvantages and advantages have been taken into consideration. The final brake system consists of a servo with a servo-wheel attached in plastic. Fixed to the servo-wheel runs a wire, which envelopes two braking poles. The double wiring method results in an equal force on both braking pads generated from the servo. When the poles moves, the attached brake pads will hit the reaction wheel, see *Figure 4.4*. Where the (a) part represent a non active brake and the (b) part a activate brake system. When the servo rotates back to the original position, after the brake procedure, two rubber bands will separate the poles and brake pads from the reaction wheel and within a short period of time the reaction wheel can move freely once again.



(a) Brake is not active **(b)** Brake is Active

Figure 4.4: Final brake system

The advantage with this brake system is that the moment applied on the cube, according to *Equation 4.2* and *Equation 4.3*, can be controlled with both the brake time and the velocity of the reaction wheel. A disadvantage with this method is that the poles and brake pads need to be pushed away using rubber bands. The rubber bands increase the force from the servo which increases the weight and normally the size. Another disadvantage is that the rubber brake pads can be worn out, which makes the system behave differently between each brake sequence.

4.2 Decision of actuator

The selected brake system requires high accuracy on the final position of the brake pad, which motivates the decision to choose a servo motor as actuator to the brake system. The brake time and the brake force are two critical parameters for the complete Jump Up action. The brake time can be calculated as

$$Brake_{tservo} = \frac{d_{fullbrake} * Speed}{d_{servo}\pi} \tag{4.5}$$

where $d_{fullbrake}$ is the distance that the wire connected to the brake pad travels, from open position to full brake position. $Speed$ is the servo speed and d_{servo} is the diameter of the plastic servo-wheel. The brake force needed to brake the system

is calculated as

$$Brake_{Fservo} = \frac{Torque}{d_{servo}} \tag{4.6}$$

where $Torque$ is the torque specified according to datasheets from a servo.

4.3 Feed forward braking signal

The brake action performance can be improved by adding a feed forward system. This can be done by feedback from either the reaction wheel deceleration or the accelerometer signal at the same time as the brake action is performed. By developing an optimal trajectory the brake time can be changed; decreased, increased or remained the same, according to how the reaction wheel deceleration or accelerometer signal appear during the new brake procedure. *Figure 4.5* represents an example of how the reaction wheel deceleration feed forward works. The black signal represents the a regular braking time and the blue signal represents the velocity of the reaction wheel. Within the red area, the reaction wheel deceleration is calculated and compared to the optimal trajectory of the deceleration. If the deceleration is too low, the black brake signal would be increased but if the value is too high the brake signal would be decreased. The shorter the decision period is, the larger changes in the brake time there will be, which is then also able to compensate for larger differences in the brake action. Although a shorter decision period gives less sampled data to make the right decision, which raises the chance of errors in the decisions.

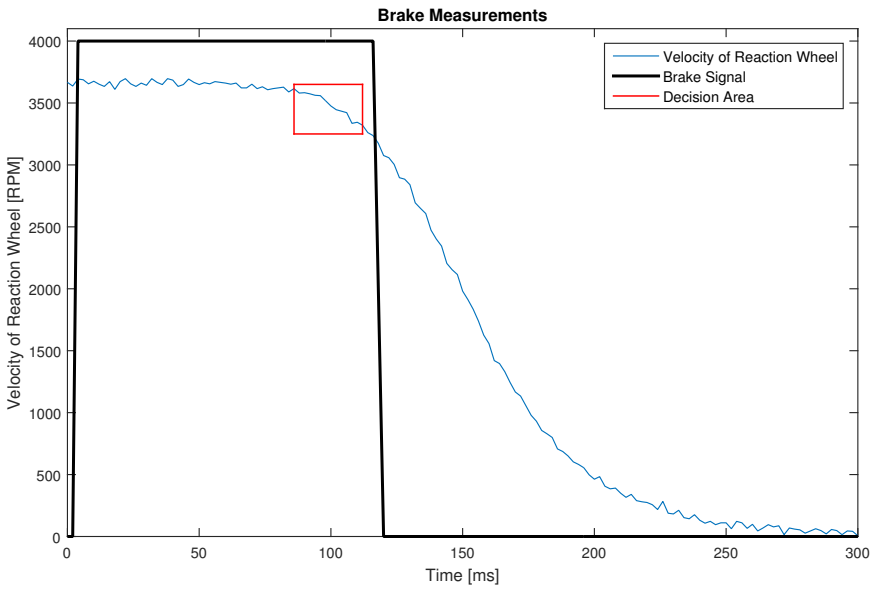


Figure 4.5: Feed forward using reaction wheel deceleration.

5
Control

The first part of this chapter contains the theory of control methods. The second part contains the implementation of the theory into MATLAB and Simulink in order to evaluate different control methods in simulation environment.

5.1 Control theory

5.1.1 Proportional-Integral-Derivative, PID

PID-controllers are often used when there is no deeper understanding of the physical system. A controller strives to minimize the present error, past error and future work. The P stand for proportional to error at the time, I is the proportional to Integral of present time and D is proportional to Derivative of present time.

The controller could be described as

$$u(t) = K_P(e(t) + \frac{1}{K_i}\int_0^t e(\tau)d\tau + K_d\frac{de(t)}{dt}) \tag{5.1}$$

where K_p, K_i and K_d are changeable variables [12].

5.1.2 Linear-Quadratic-Regulator, LQR

An infinite horizon, Linear Quadratic Regulator, LQR, given a multi-input linear system, can be described as

$$\dot{x} = Ax(t) + Bu(t) \tag{5.2}$$

$$y = Cx(t) + Du(t) \tag{5.3}$$

$$x\epsilon\ \mathbb{R}^n, u\epsilon\ \mathbb{R}^p$$

which strives to minimize the quadratic cost function

$$J = \int_0^\infty (x^T Q_r x + u^T Q_u u)dt \tag{5.4}$$

where $Q_r \geq 0$ and $Q_u > 0$ are symmetric, positive semi-definite matrices. Their structure is shown in following matrices.

$$Q_r = \begin{bmatrix} q_{r1} & & 0 \\ & \ddots & \\ 0 & & q_{rn} \end{bmatrix} \tag{5.5}$$

$$Q_u = \begin{bmatrix} q_{u1} & & 0 \\ & \ddots & \\ 0 & & q_{up} \end{bmatrix} \tag{5.6}$$

The solution can then be given by the control law

$$u = -Q_u^{-1} B^T P x = Kx \tag{5.7}$$

where P is a positive definite, symmetric matrix that satisfies the algebraic Riccati equation

$$PA + A^T P - PBQ_u^{-1} B^T P + Q_r = 0 \tag{5.8}$$

which gives the controller feedback optimal gain K [13]. In order to implement the controller in a real-time system a discrete-time LQR is needed.

5.1.2.1 Discrete-time LQR

Given the discrete-time system model as

$$x[k+1] = Ax[k] + Bu[k] \tag{5.9}$$

$$y[k] = Cx[k] + Du[k] \tag{5.10}$$

the quadratic cost function which the aim is to minimize can be formulated as

$$J = \sum_{k=1}^{\infty} (x[k]^T Q_x[k] + u[k]^T Q_r u[k] + 2x[k]^T N u[k] \tag{5.11}$$

The last term in the cost function has generally small effect of the outcome compared to the two terms in front. The effect of the last term in this equation is often small compared to the first two terms, therefore usually N is set to zero to simplify the calculations.

Given the solution to S in association with discrete-time Riccati equation

$$A^T SA - S - (A^T SB + N)(B^T SB + Q_r)^{-1}(B^T SA + N^T) + Q_u = 0 \tag{5.12}$$

the optimal gain matrix K can be calculated as

$$K = (B^T SB + Q_r)^{-1}(B^T SA + N^T) \tag{5.13}$$

By changing the weight in the matrices Q_r the optimal controller gain K is calculated, based on robustness and speed criterion of the system [13].

5.1.2.2 Integral action

The LQR allows good performance if the dynamical model is exactly the real model, if not there will be a steady-state error. By implementing an integral feedback to the LQ-controller, the steady-state error could be driven to zero. The approach is to add a state $z_I = \int_0^t (r - y) d\tau$ within the controller which computes the integral of the error signal [13]. This will result in an augmented state-space model represented as

$$\begin{bmatrix} \dot{x} \\ \dot{z}_I \end{bmatrix} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix} \begin{bmatrix} x \\ z_I \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ I \end{bmatrix} r \tag{5.14}$$

The LQ regulator with added integral state can be calculated by minimizing the quadratic cost function

$$J = \int_0^\infty (\begin{bmatrix} x^T & z_I^T \end{bmatrix} \begin{bmatrix} Q_r & 0 \\ 0 & Q_I \end{bmatrix} \begin{matrix} x \\ z_I \end{matrix} + u^T Q_u u) dt \tag{5.15}$$

5.1.3 Look Up table

In systems which are piecewise linear and nonlinear, a controller could be optimal in one area of the system, while its performance could be poor in another area. There are several different nonlinear methods modelling a controller to a nonlinear system. One of them is making a Look Up table [14]. The Look Up table changes the control gain parameters feed to the regulator, in steps according to a reference, while running. Look Up tables can be implemented if the control target varies, i.e. in one area the robustness against high frequent noise is most important and in another area the controller has to be as responsive as possible.

5.2 Implementation and Simulations

The theoretical model of the edge balancing system is divided in two separate systems, Jump Up action and Balancing action. The two systems are dependent of each other but are controlled separately. The overall structure of the model can be seen on Figure *5.1*.

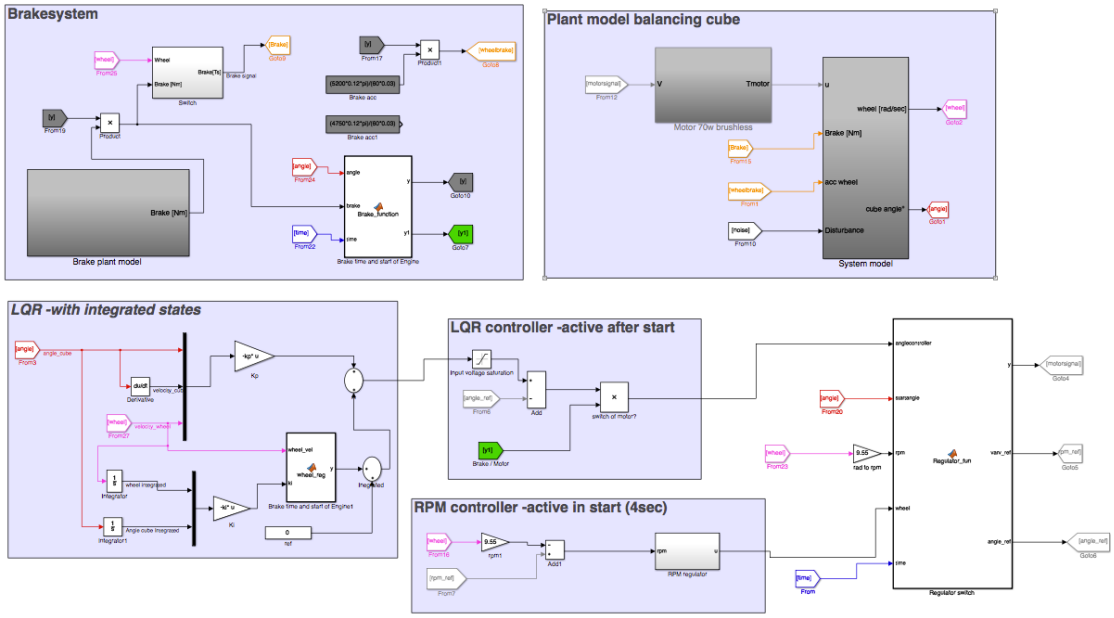


Figure 5.1: Model structure of the edge balancing system.

Figure 5.1 represents the Simulink model describing the system. The plant is divided in two main blocks; Brake system and Plant model. The Brake system describes the mechanics related to moment generated when braking pads attach to the wheel surface. The plant model describes the cube and wheel using the *Equation 2.12* from the Modeling chapter. The controller part implements a LQR controller with integrated states, as described in *5.1.2 Linear-Quadratic-Regulator, LQR*.

5.2.1 Balancing action

The controller aims are to keep in balance position with criterion at motor signal activity and reaction wheel velocity. The engine is sensitive against high frequent switches in between positive/negative rotation of the axis, which implies that the signal activity should be minimized. The second criterion is that the reaction wheel velocity should be kept as low as possible to minimize the power consumption. Those criteria are met by implementing an LQR-controller with added integral action at the states θ_{cube} and θ_{wheel}. Since the real time model gives sampled output data, the model needs to be discretized. This is done with help of the MATLAB command

$$sys_{Discrete} = c2d(sys_{Continous}, T_s) \tag{5.16}$$

where $sys_{Continous}$ is the continous-time system model, from *Equation 2.14*, and T_s the sampling time, set to 2 ms. A discrete-time LQR gain matrix K is then calculated as described in *5.1.2.1 Discrete-time LQR* where the weight matrices Q_r and Q_u are set to

$$Q_r = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0.8 & 0 & 0 & 0 \\ 0 & 0 & 0.4 & 0 & 0 \\ 0 & 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 0.1 \end{bmatrix} \tag{5.17}$$

$$Q_u = 1 \tag{5.18}$$

The final controller used in simulations where the controller gain matrices K_P and K_I are

$$K_P = \begin{bmatrix} -3.7813 & -1.0150 & -0.0078 \end{bmatrix} \tag{5.19}$$

$$K_I = \begin{bmatrix} 0 & -0.0030 \end{bmatrix} \tag{5.20}$$

Here K_P corresponds to the system states from state-base model in *Equation 2.13* and K_I corresponds to the integrated states $\frac{\theta_{cube}}{S}$ and $\frac{\dot{\theta}_{wheel}}{S}$.

5.2.1.1 Balancing tolerance external push

The simulated model together with controller is used to validate how much outer disturbances, the system could handle before falling.

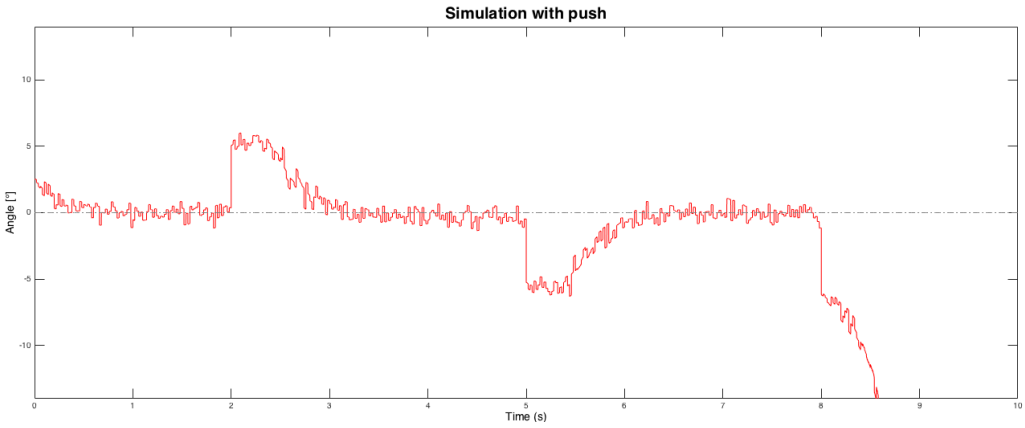


Figure 5.2: Simulated balancing with external push where red is the angle of the cube

Figure 5.2 shows that the simulated system can handle an external push, i.e a person pushes the cube with a force up to 5 degrees from its balancing position. If the push is larger than 5.5 degrees, the controller fails and the cube falls. The cube is pushed at time instance 2 and 5, with the same force in opposite direction. At time instance 8, the cube is pushed with a greater force and falls.

5.2.1.2 Evaluation of physical limits

Assume an optimal controller. The physical limitations of the cube can be analysed by starting the cube in a tilted position and allowing the controller to bring the cube up to the balancing position without any external forces.

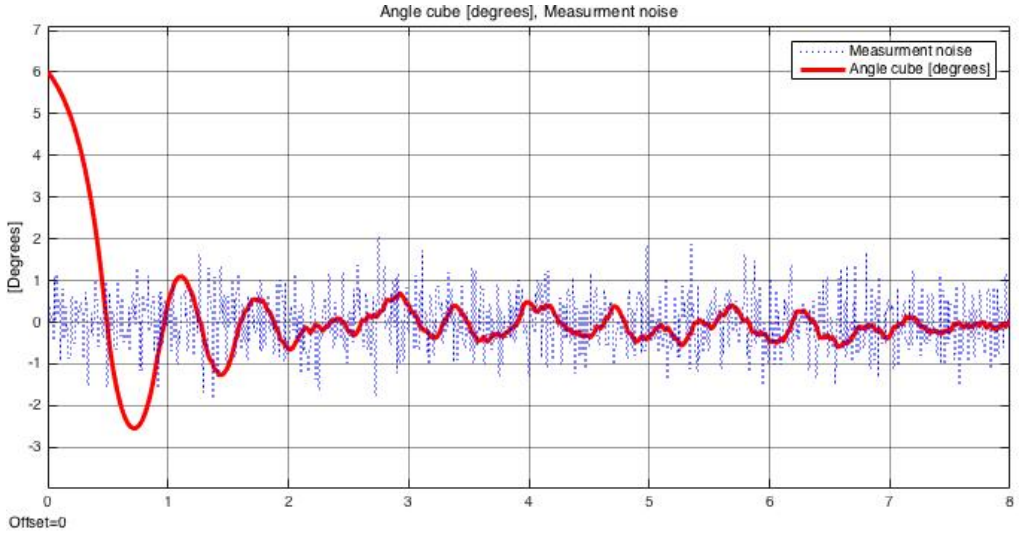


Figure 5.3: Balancing action starting from 6 degrees off from balancing point.

Simulation shows that the regulators can bring the cube to balancing position, without braking action if the starting position of the cube is within the span of +- 6 degrees, with a Gaussian distributed disturbance added at the angle measured signal of +-2 degrees. An example of this behaviour can be seen on Figure 5.3, where the red line represents the angle of the cube and the blue line represents the Gaussian noise.

5.2.2 Jump Up action

The Jump Up action aims to bring the cube within the balance space, i.e. within +- 5 degrees from the main balance position. The aim is also that the velocity of the reaction wheel after brake action would be as close to zero as possible, to save as much as energy as possible. With those parameters in mind, the best brake action is modeled to find a velocity of the wheel which gives enough moment to bring the cube into the balancing space. The theoretical model has limitations, the friction in between the brake pad and reaction wheel is modeled as constant, where $\mu = 0.3$. In reality this friction factor will change during the braking procedure and it will also differ from time to time. An example of a simulated brake procedure is shown on Figure 5.4, where the brake action starts at 4 s.

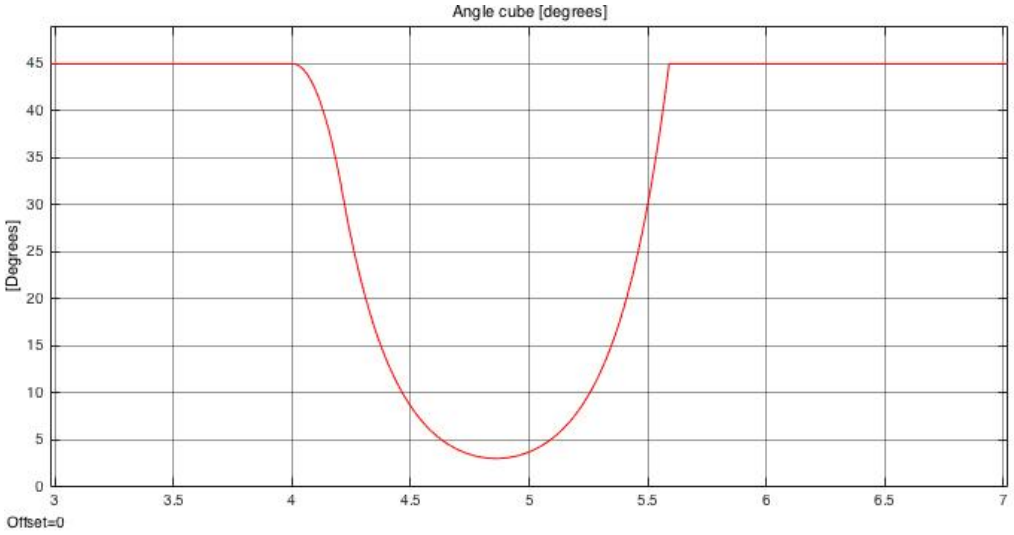


Figure 5.4: Demonstration of a brake action, where the starting velocity of the reaction wheel is 2600 rpm. The cube starts from 45 degrees and reaches 3 degrees.

5.2.3 Jump and Balance

The last action is to combine the two systems, Balancing and Jump Up. One of the issues here is that there are two different control systems which have to be synchronized. In this case the Jump Up controller can be used between 45 to 12 degrees from balancing point, and the Balancing controller is allowed to work within the area of +-10 degrees from balancing point. This can prevent the possibility that the brake system tries to stop the wheel and the balancing system tries to accelerate the wheel.

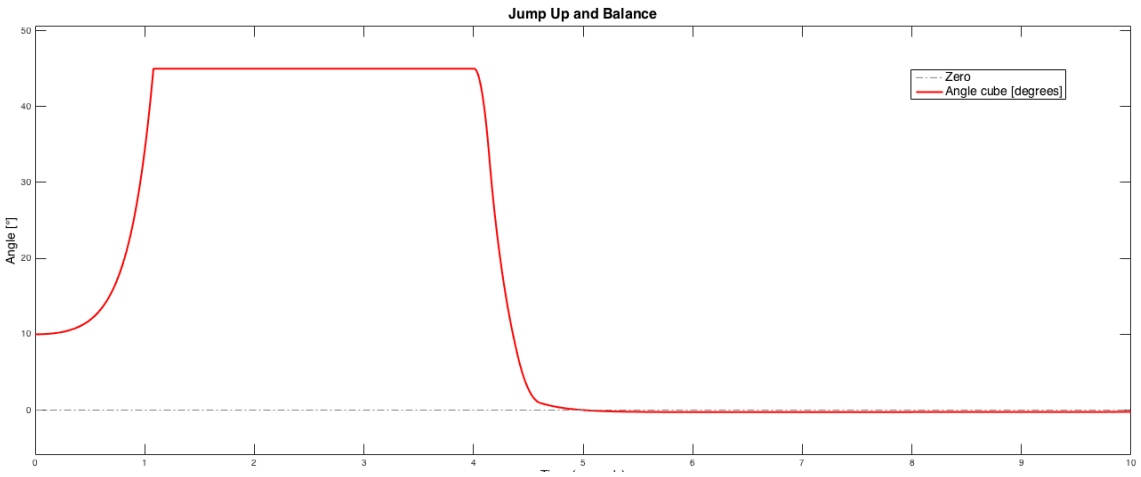


Figure 5.5: Angle of the cube in a Jump Up And Balance sequence

The tested scenario, represented on *Figure 5.6* and *Figure 5.6*, is a test where the cube is released at 10 degrees and due to that the engine is not able to produce enough momentum, it falls to the ground. At time unit 4.0 the velocity of the wheel has reached 2600 rpm, and the braking action is switched to on. At time

unit 4.22 the brake system is switched off and the second controller, the LQR, is activated and the cube can balance on the edge.

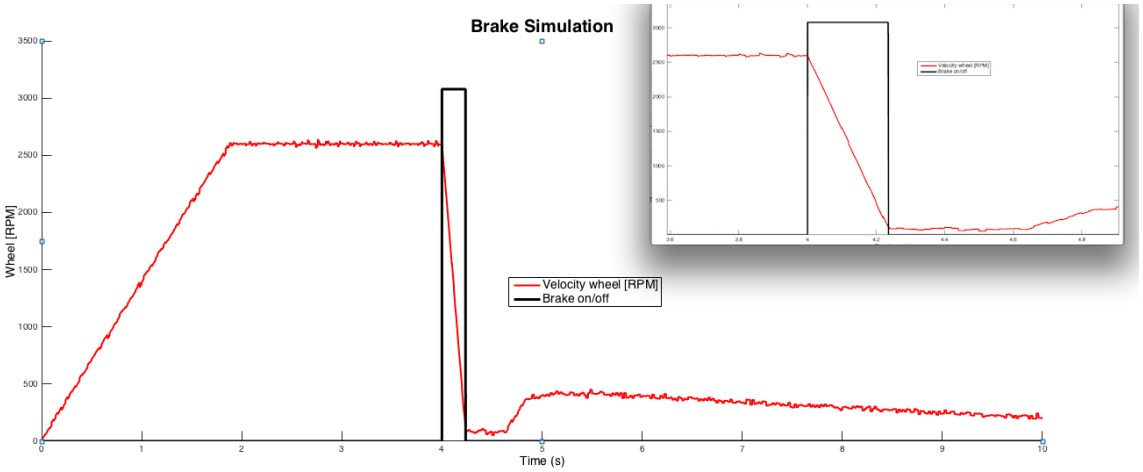


Figure 5.6: Jump Up And Balance sequence, with black as brake signal and red as reaction wheel velocity.

6

Design of software and hardware

The mass and size of the cube are two factors that have an huge effect at the final systems ability to Jump Up and Balance. The balancing action benefits from a larger cube, while the Jump Up action's performance decreases by the size of the cube. This is due to that the location of the center of mass changes, and affects the ability to Jump Up. The conclusion is that the cube should be big enough to contain all the components necessary, but with a lower center of mass, it has a higher chance of managing the Jump Up action. This chapter describes the most important part in relation to trade offs and design parameters.

6.1 Hardware

This section contains the hardware design, both the theory about the components, and also the reasoning about the selection process of the components. The design of the final cube is presented in *Figure 6.1*. In this picture, three sides of the cube are open to enhance the understanding of each of the components position.

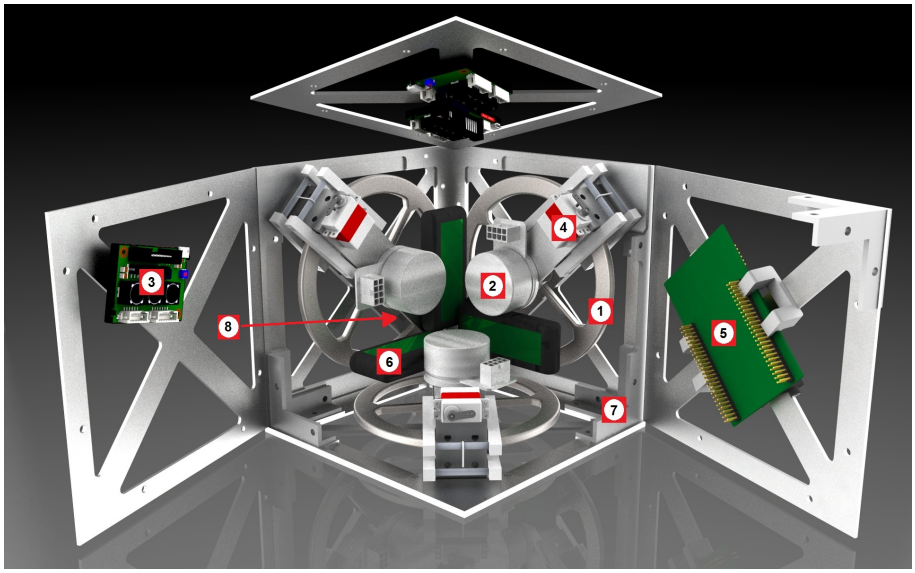


Figure 6.1: Hardware components of the cube; 1 Reaction wheel, 2 Reaction wheel actuator, 3 Motor controller, 4 Brake system actuator, 5 Microcontroller, 6 Batteries, 7 Corners, 8 IMU Sensor.

6.1.1 Reaction wheel

The moment of inertia for an object with a constant density is defined as

$$I = \sum_{i=0}^{\infty} m_i r_i^2 \tag{6.1}$$

m_i is the mass of number i particle on the object and the r_i is the distance from the rotational axis to the particle i [15]. To minimize the mass of the reaction wheel and increase the moment of inertia, the mass should be placed as far away from the middle point as possible.

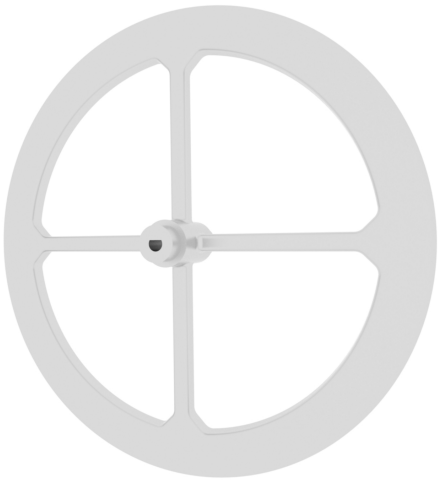


Figure 6.2: Reaction wheel

Simulations in MATLAB and Simulink have shown that the reaction wheel needs a moment of inertia as big as $8.4 * 10^{-4}\ Kgm^2$. The material of the reaction wheel has been selected to steel 21/72 instead of aluminum, due to the reason of keeping the weight as low as possible as well as the size of the reaction wheel. *Figure 6.2* represents the used reaction wheel with a mass of 273 g and a diameter of 13.5 cm. The non-circle hole in the middle of the reaction wheel is designed to allow the engine axis fit into it. The reaction wheel is manufactured in a CNC machine.

6.1.2 Bearing

The axis of the current motor is four millimeters. Calculations of durability of the axis while running, the system would make the axis vulnerable to wobbling of the reaction wheel. If the reaction wheel attaches directly to the motor, the risk that the axis cracks and deforms during the Jump Up action would be high.

To minimize the risk of a deformed axis, bearings have been attached to each side of the reaction wheel. The choice of the bearings have been selected with consideration to the dynamic load that appears if there is a displacement of the center of gravity. This load can be calculated using

$$F_{Dyn\ Load} = m_{wheel}(r + e)\dot{\theta}^2_{wheel} \tag{6.2}$$

where e is the distance between the rotational axis and the center of gravity of the reaction wheel. The r represents the distance of the axis deflection and can be calculated according to

$$r = \frac{\dot{\theta}^2_{wheel}\, e}{1/(\alpha_F m_{wheel}) - \dot{\theta}^2_{wheel}} \tag{6.3}$$

where α_F can be defined as

$$\alpha_F = \frac{4}{243} * \frac{L^3_{shaft}}{EI} \tag{6.4}$$

where E is the modulus of elasticity and I is the Area Moment of Inertia [16]. In this case I is defined according to $I = \frac{\pi(d^4_{outside} - d^4_{inside})}{64}$, due to the hole in the axis [17].

With the assumption of a maximum e defined as 10 millimeter and a maximum velocity of the wheel to 4800 rpm, the bearings needs to handle a dynamical force bigger than 700 N. To ensure the stability, bearings with a dynamical strength of 1300 N have been selected, for further information regarding the bearings, see *Appendix A.1*.

6.1.3 Reaction wheel holder

The reaction wheel holder is made of aluminium with a total weight of 111 g. Due to the two bearings, we also need to implement and install mountings/attachments. To enhance the stability and accuracy of the attachments they are connected with one millimeter wide guide-pins.

In the corner where all the reaction wheel holders are together, there is some unused space. If the cube's reaction wheel is placed in the middle of the frame, there would be a second unused space located on the right side of the holder. To decrease the weight and the size, the reaction wheel has been moved slightly to the right, see *Figure 6.3*.

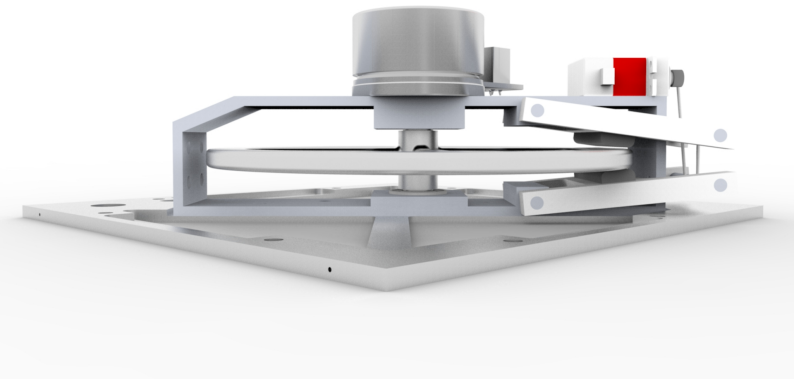


Figure 6.3: Reaction wheel moved slightly to the right

6.1.4 Frame

As presented in *Figure 6.1*, the cube's sides are constructed with three thicker metal frames and three thinner frames. The reaction wheel where attached on the thicker frames together with the reaction wheel holder and the brake system. The torque from the motor and the forced conserved during the brake required stronger construction and the selection of the material has become metal. To keep the weight down the metal selected was aluminum. The weight of the thicker frames are 152 g, 123 g and 136 g and the smaller ones are 98 g.

To be able to balance on one of the corners as well as on the edges, the three metal parts need to be attached with high accuracy. To get high accuracy, one millimeter wide guide pins have been attached between each of the sides. Drilled holes for this guide pins are shown on the left side of *Figure 6.3*.

The thicker frames have been produced in a CNC milling machine and the thinner frames have been constructed in a laser cutting machine. To keep the manufacturing cost as low as possible, the components have been designed to lower the manufacturing time and cost.

6.1.5 Brake pad holder

The final brake pad holders used in the cube can be seen on *Figure 6.4* and they are positioned over and under the reaction wheel, see *Figure 6.3*. Some part of the holders are made of plastic, and their length are decided with respect to the leverage force in relation to the servo motor force. The rubber is wedged to the plastic holders where the two parts are designed to perfectly fit with each other. This solution is to keep the weight of the parts as low as possible, by not adding any screws or glue to hold the parts together. This solution also makes it possible to change the rubber in the future. The metal bar inside the plastic holders is selected due to the slippery surface where both the wire and the reaction wheel holder should rotate around.

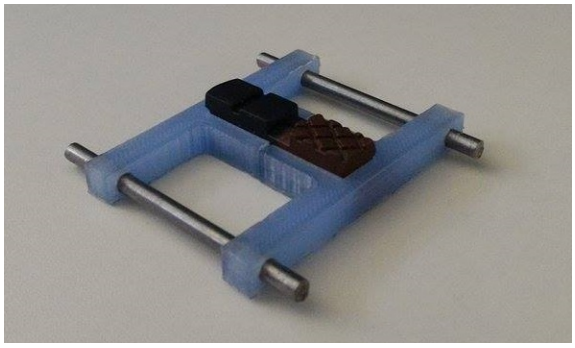


Figure 6.4: Brake pad holder

6.1.6 Corners

The frames are connected to each other with help of eight corners. The corners were manufactured in a 3D plastic printer, due to the low weight in relation to stability and price range.

6.1.7 User interface

The cube has different functions and with help of a user interface panel, the users would be able to interface with the cube. The user interface panel contains five buttons, four push buttons and one on/off switch, see *Figure 6.5*. The left button is a sensor calibration button, the second to the left decides which side the cube should rotate around. The up and down buttons are used for changing the reaction wheels starting velocity in the Jump Up procedure. The on/off switch is a safety button which starts the jump up procedure after switching to on. No current will be sent to any motor if the switch is set to off.

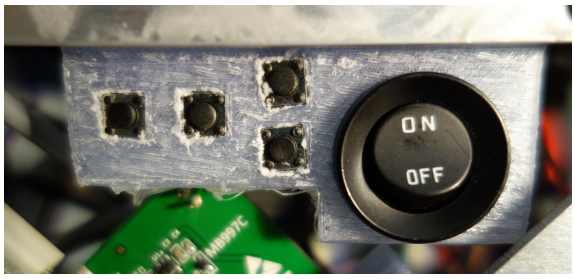


Figure 6.5: User interface panel

6.1.8 Reaction wheel Actuator

There are many different types of DC actuators and the most commonly used one is the brushed DC motor which has the advantage of low price compared to other actuators. However the disadvantage of using brushed motors is that the brushes heat up while running, which limits the maximum speed and therefore a brushless DC motors is more commonly used for high speed applications [18].

Brushless EC motor model

The selected motor, Brushless electrical commutated, BLEC, are the commutation done by electronic control and there is no contact in between stator and rotor. Stator consists of stator windings organized in sequence giving 3-phase [19]. The rotor consists of 8-polepairs and three symmetrically attached hall sensors, which give rotor position feedback.

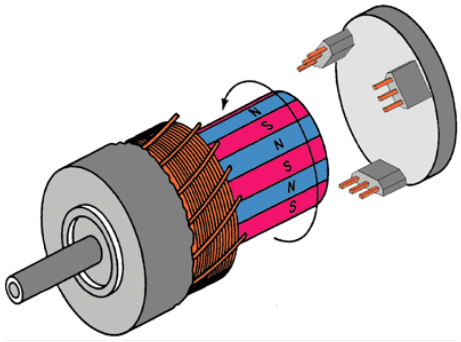


Figure 6.6: Brushless 3-phase motor with 8 polpairs [20]

The BLEC motor is modeled as a common DC motor model, with the addition of three phases which effect the numbers of resistors and inductance. A BLEC engine can be modeled as:

$$T = K_T \phi I \tag{6.5}$$

$$E_a = K_C w \tag{6.6}$$

where T is the torque generated by the motor, w is the velocity of the motor axis, E_a is the back EMF, counter-electromotive force, K_T is a motor constant given for the motor and K_C is Voltage constant calculated by

$$K_C = \frac{3RI_{motor}}{\tau_{mec} K_T} \tag{6.7}$$

where τ_{mec} is the mechanical time constant given for motor and I_{motor} is the inertia of the motor [19].

The reaction wheel actuator is a EC 45 flat Maxon 70 W motor and the motor controller is an ESCON-36-3-EC. The motor has a maximum torque of 0.128 Nm and a maximum rotational velocity of 4860 rpm .

6.1.9 Brake system Actuator

Servo motors are commonly used in applications requiring high accuracy of position. The servo motor is fed by a pulse width modulated, PWM signal, which used as a control signal for positioning a pre-specified angle. The four main parameters selecting a servo motor are rotating speed, maximal torque, weight and working frequency.

Given the requirement, the digital servo TGY-616MG is selected and the specifications can be seen on *Table 6.1 Servo specification*.

According to the specification, the servo motor has the highest torque and operating speed with a power supply of 8.4 V and by using Equations 4.5 and Equations 4.6 the values of the critical constants can be calculated to

Table 6.1: Servo specification

	TGY-616MG		
Operating Voltage	6.0 V	7.4 V	8.4 V
Operating Speed	0.07 sec/60°	0.06 sec/60°	0.05 /sec60°
Stall Torque	4.6 kg.cm	5.5 kg.cm	6.6 kg.cm
Weight	34 g	-	-
Working frequency	333 Hz	-	-

$$Brake_{tservo} = 28.6ms \tag{6.8}$$

$$Brake_{Fservo} = 26.5N, \tag{6.9}$$

This fulfills the first three specified requirements to simulation in MATLAB. The last one, which is the working frequency, decides how fast the servo reacts on a change from the micro processor. The frequency is 333 Hz, 3 ms, which is the standard for a digital servo [21]. The working range for the TGY-616MG servo is between zero degree and 45 degrees in the area of 800 us to 2200 us out of 3000 us.

6.1.10 Sensors and sensor fusion

In order to use the controller developed in chapter *5 Control*, information regarding velocity and position is needed. That information can be done using camera motion detections, GPS, encoders, accelerometer, gyro and Ampere meter. With all of these in consideration and the price in relation to the functionality the sensors, MPU 6050, has been selected. The MPU is a sensor which measure both accelerometer and gyro changes[22]. The sensor is mounted on the reaction wheel holder, 9 cm from the balancing edge, see *Figure 6.1*. An alternative sensor that may be used in the future is the MPU 9250 which is similar to the MPU 6050, but with faster communication protocol and with a higher price [23]. On account of the price difference, the MPU 6050 is selected in the first place.

This coming subsections describe the IMU sensors and their main functionality.

6.1.10.1 IMU

The velocity, acceleration and geographical positioning of the system are measured using Inertial Measurement Units, IMU, which is an electrical unit generally consisting of a three-axis accelerometer and a three-axis gyroscope and in some cases also a three-axis magnetometer. The numbers of axis measured give the number of maximum states to fully measure. The 3-dimensional case of balancing the cube on one corner gives a system of 6 degrees of freedom, which motivates the selected 6 degree IMU, consisting of accelerometer and gyroscope. The output of the sensors is communicated via an Inter-Integrated Circuit, I^2C, protocol. The raw data from the sensors is treated with filtering method to combine

the measurements, using a complementary filter discussed in the section *6.1.10.2 Complementary filter*.

Three-axis accelerometer

The accelerometer measures the acceleration within the 3-dimensional space. The measuring technique differs from other accelerometers. Some using a technique of piezoelectric effect, containing microscopic crystal structure, effected by acceleration forces which in their turn generates a voltage output. The selected IMU, MPU 6050, uses another technique which is based on measuring changes in the capacitance.

The sensor (accelerometer) consists of: three masses that are attached to a spring, which is mounted orthogonal to each other in the 3-dimensional space. During acceleration, the displacement of the mass is sensed by a difference in the capacitance. The system using this technique is often referred to micro electro-mechanical systems,(MEMS)[22]. The sensor has adjustable sensing areas between +-2g and +-16g. The selection of a sensor is a trade-off between sensitivity and the maximum measurable acceleration.

Three-axis gyroscope

Gyroscopes measures the angular rate around an axle. They consist of several different mechanical approaches. One of the most common gyroscope, where the units becomes very small, is the vibratory "MEMS" gyroscope used in this thesis. The sensor used consists of three gyroscopes, which all rely on MEMS technique. The method use the Coriolis Effect, which causes a vibration when the gyroscope is rotated around its axle. The vibration is then sensed by the difference in the capacitance (output). By conditioning the output, i.e. amplify and filter the capacitance-signal, the output of the sensor is voltage proportional to angular rate[22].

6.1.10.2 Complementary filter

Gyroscopes are known for the drift, which increases by time. Accelerometers in general have stable steady-state behavior, but are relatively slow at following fast changes and very sensitive to measurement noise [24]. The technique of low pass filtering the accelerometer value and high pass filtering the gyroscope value and combining them gives a complementary filter described by the transfer function

$$\theta = \frac{1}{1+Ts}\psi_{acc} + \frac{Ts}{1+Ts}\frac{1}{s}\psi_{gyro} = \frac{\psi_{acc} + T\psi_{gyro}}{1+Ts} \tag{6.10}$$

where the estimated angle, θ, is calculated using a filter cut-off frequency T, and the angle estimated accelerometer measurements ψ_{acc} and gyro rate ψ_{gyro} [25].

Using backward difference equation

$$s = \frac{1}{\Delta_t}(1 - z^{-1}) \tag{6.11}$$

and the formulation

$$1 + Ts = (1 + \frac{T}{\Delta_t}) - \frac{T}{\Delta_t} z^{-1} \tag{6.12}$$

the discretized estimated angle θ_d could be formulated as

$$\theta_d[k] = \alpha(\theta_d[k-1] + \psi_{gyro}[k]\Delta_t) + (1-\alpha)\psi_{acc}[k-1] \tag{6.13}$$

with

$$\alpha = \frac{\frac{T}{\Delta_t}}{1 + \frac{T}{\Delta_t}} \tag{6.14}$$

In the real system, the angle is estimated using the Complementary function described in Equation 6.1.10.2 where ψ_{acc} is estimated to

$$\psi_{acc} = \arctan 2(Y_{acc}, X_{acc}) \tag{6.15}$$

Where Y_{acc}, X_{acc} is accelerometer measurements in y respectively x direction. Due to that the accelerometer in z direction from the IMU is highly influenced by the rotation of the reaction wheel which makes the estimated angle less accurate. Due to the position of the IMU, ψ_{gyro} can be estimated accurately enough just by using the z direction of the gyro signal. The most accurate test, verified with an encoder connected to the cube, was when the α value is set to 0.99.

6.1.11 Microprocessor

The main processor is the Discovery kit STM32F4 with a 32-bit ARM Cortex-M4F. The STM32F4 have for analog and digital outputs: digital to analog converter and vice verse as well as a USB OTG FS, with the ability of data logging [26]. STM32F4 works perfectly with MATLAB and Simulink using the Waijung Blockset, 6.2.2 Waijung Blockset.

6.1.12 Battery

The cube contains three 2 cell Lipo batteries. A Lipo battery is normally used in applications like radio controlled airplanes and helicopters due to its light weight, large capacity in small package and the ability to deliver a high discharge current [27]. The reaction wheel actuator is preferably driven with a 24 V. According to the datasheet, it has a nominal current of 3.21 Ampere and a stall current up to 39.5 Ampere. The selected Lipo batteries is specified as *Turnigy nano-tech 1500 mAh 2S 35-70C*. They are series connected to deliver a voltage between 19.8 to 25.2 depending on the time it have been used since full charged [28]. Due to the specification, the batteries discharge current is between 52.5 to 105 Ampere. The first battery in the series, is also connected to the brake system actuator. Due to the voltage range of this battery, 6.6 to 8.4, the Stall Torque and operating speed of the brake system actuator will change over time, but the voltage is within the working range of the actuator, see *Table 6.1 Servo specification*. The same battery is also connected with a five volt converter which provides power to the Microprocessor.

6.2 Software

With assistance of all the hardware described in subsection 6.1 Hardware, the cube should be able to handle the main goals. In order to effectively use the hardware, different software are implemented.

6.2.1 Communication Protocol

Communicating information between the microprocessor and the IMUs is done using a pre-designed protocol. There are several different designed protocols, where the main factors are communication speed, integrating simplicity in relation to cost.

6.2.1.1 I^2C

Inter-Integrated Circuit, I^2C, is a protocol with a serial data cable, SDA, and serial clock, SCL. I^2C supports multiple slaves and arranges the information to each slave, with help of 7-bits address. Normal selected bitrate are 100 kbps, 400 kbps and 3.4 Mbps, depending on what supports components [29].

6.2.1.2 SPI

Serial Peripheral Interface is a protocol with two separate data cables, receiving and transmitting. A clock signal synchronizes the master and the slave to decrease communication errors. It contains more connected cables compared to a I^2C protocol. A SPI supports multiple slaves and a normal bitrate is around 10 Mbps [29].

6.2.2 Waijung Blockset

Waijung blockset is a Simulink embedded target blockset which was developed by academic researchers who were sponsored by the Thailand Research Funds and Royal Thai Naval Academy [30]. The blockset minimized the time on pure microcontroller programming. Instead of converting the Simulink blockset into C, using computer written C-code, the blockset compiles and transfers the Simulink blocks into C-code automatically to the STM32F4 board.

7

Results

This chapter starts with a description of the end product. Further subsections describe different test that have been performed on this product. There are four main tests; Balance Action, Jump Up Action, Walking ability and in the end the final Jump Up and Balance.

7.1 Final product

According to section *6.1 Hardware* the cube should have three sets of reaction wheels, reaction wheel holders, motors, brake system motors and motor controllers. Due to delivery problem, only one of the sets is installed in the cube. One set is enough to evaluate the thesis goals, but to accomplish future goals, mentioned in subsection *1.3.5* , all of the sets needs to be implemented. The evaluated cube which is developed is shown on *Figure 7.1* with a total weight of 2083 g. The missing two sets would have increased the weight of the cube to approximately 3487 g.

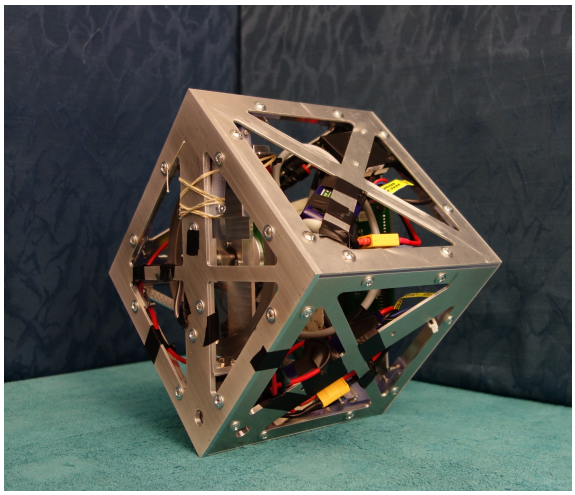


Figure 7.1: Final product with one set of reaction wheel

7.2 Balance

The Balance Action is evaluated by several tests. In the following subsections, different tests are described and results of each of the tests are shown.

7.2.1 System limits

The physical system limits of the combined solution with the current engine together with the reaction wheel was evaluated during a test. In the test the cube is tilted to a position and by giving maximum current to the actuator, +-6 Ampere, we can see if the torque manages to move the cube up to balancing position. The cube has been able to move up to balancing position from +-3.5 degrees.

7.2.2 Balancing repetitiveness

To validate the repetitiveness of the balancing action, 25 trials have been done. If the cube manages to balance more than 25 s, the test will be considered as successful. The test results of the system have shown that the cube can balance with a success rate of 100 %.

7.2.3 External push while balancing

Evaluation of the system's robustness against external disturbances, such as external push. The test has been done using different controllers to evaluate the differences. Starting in a balancing position and then give a small push by poking the cube a little bit to the left or right. The maximum angle that the controller can manage, without falling, has been recorded. The tests have been done 50 times in a row.

7.2.3.1 PID controller

By using a double loop PID controller, the inner loop with angle of the cube as a reference and an outer loop with the velocity of the reaction wheel, the cube is able to balance. During the test series of 50 trials the cube has been able to handle a disturbance of maximum +-1.7 degrees.

7.2.3.2 LQR controller

With a LQR controller the cube has been able to handle up to +-1.7 degrees before falling.

7.2.3.3 LQR-controller with Look-Up table

With the same initial LQR controller parameters added to a look-up table, where the proportional gain changes depending on the measured angle, the cube has been able to handle disturbances up to +- 2.1 degrees before falling, see *Figure 7.2*.

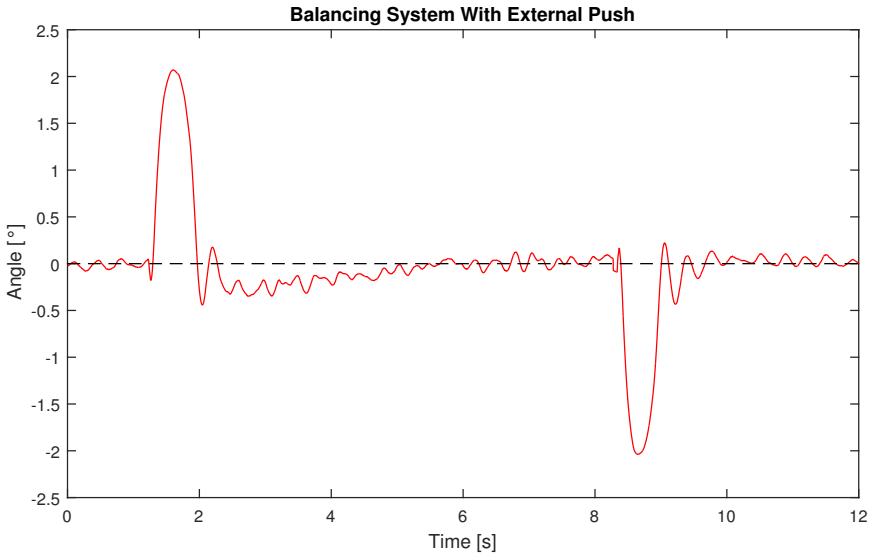


Figure 7.2: LQR-controller with Look-Up table and external push

7.2.4 Various surfaces

With both the PID, LQR and the LQR-controller with look-up table, the cube can balance on a high frictional surface as well as on a low friction surface. High friction surface is defined as material similar to rubber, and low frictional material is defined as wooden or plastic table. The test also shows that if the cube slides slowly on the surface, the controllers can manage to keep the cube balanced, but sliding will decrease the robustness against external push.

7.3 Jump Up Action

Jump Up Action test is done to analyse repetitiveness and accuracy of the designed brake system. The first test evaluates the repetitiveness of reaching the same angle while braking at the same initial velocity of the reaction wheel. The second test evaluates if there is a clear difference of the sampled data using forward feedback, in between two special cases i.e. jump up and fall back or jump up and fall over.

7.3.1 test of repetitiveness

The repetitiveness of the brake action is evaluated during a test sequence of 20 trials. A reference velocity is given to the PID controller, which brings the reaction wheel to the reference velocity. To avoid that the motor is still on while starting the braking action, a delay of 50 ms is applied before the system brakes. In the 20 trials the same reference speed has been applied and the ability of bringing the cube to the same level, angle, every time has been evaluated.

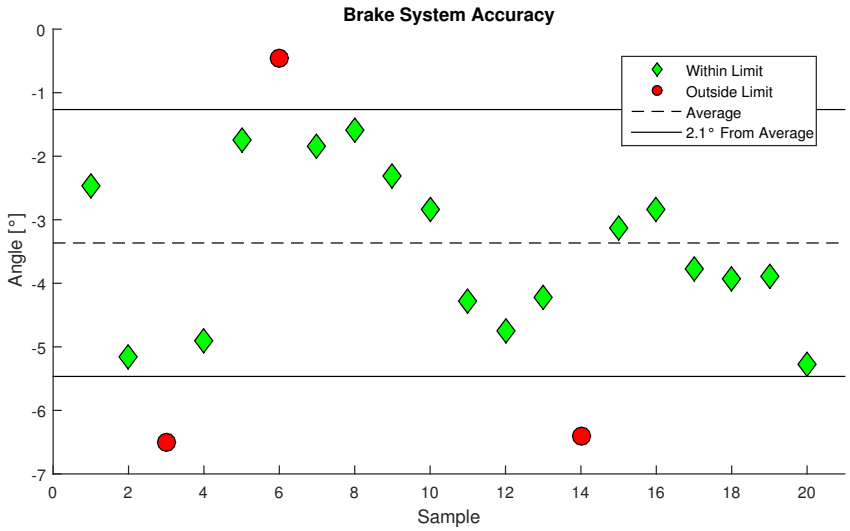


Figure 7.3: Brake system test

The test result of the accuracy and repetitiveness of the brake system is shown on *Figure 7.3*. In this case, the starting velocity before brake is 3630 rpm. The cubes starts at -45 degrees and with the use of the brake system, the highest angle it reaches before falling back is represented in the Figure. The green diamonds represent test cases where the balance LQR controller with look-up table probably would make the whole system work, according to test *7.2.3.3 LQR-controller with Look-Up table*, if the average line could be pushed up to zero degree. The red circle represents the cases where the LQR controller probably would fail due to outside the controller limits, which is between +-2.1 degrees.

It can be seen that the current brake system is able to bring the cube up within the allowed space 17 out of 20 times.

7.3.2 Improvements of brake system

To enhance the repetitiveness of the Jump Up action, see test result from *7.3.1 test of repetitiveness*, the feed forward system could be implemented. According to *4.3 Feed forward braking signal*, the wheel deceleration and the accelerometer signal could be fed forwarded into the system to increase the accuracy rate of the brake procedure.

To evaluate the possibility of feed forward implementation, 50 different Jump Up action tests have been carried out. In 25 tests, the cube jumps up and falls back to the original position and in the other 25, the cube falls over to the other side. In the reaction wheel deceleration test an average deceleration has been calculated within the decision area, see *Figure 4.5* on page *19*, where the brake signal can be changed +-4 ms. The result is represented in *Figure 7.4*.

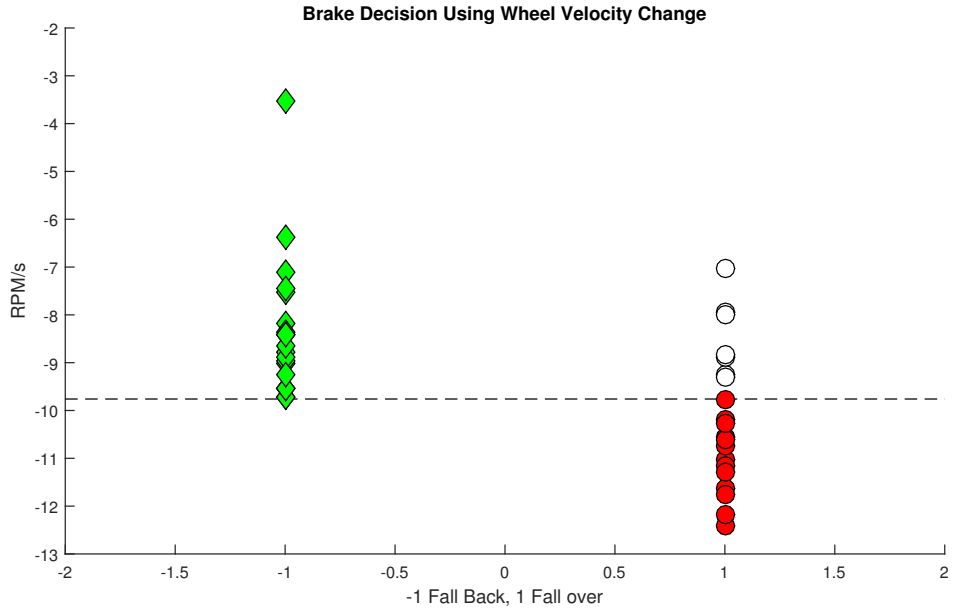


Figure 7.4: Brake system test using Velocity deceleration

In the accelerometer test, the sum of the accelerometer values within the decision area has been calculated. This test results are represented in *Figure 7.5*.

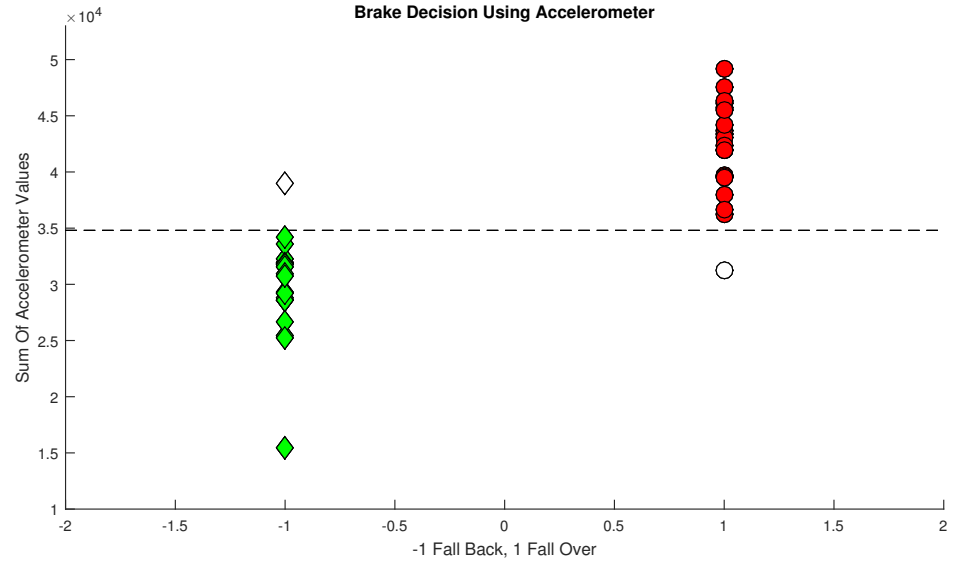


Figure 7.5: Brake system test using Accelerometer

In the test result, the velocity that the initial speed of the reaction wheel has before brake fluctuates within the span of 20 rpm. Delay time of the brake signal between the microprocessor STM32F4 and the servo motor is between 10 and 20 ms.

7.4 Walking ability

To evaluate the walking ability, the cube should be able to roll around four sides without any interruptions. After ten test series, the evaluation should also reveal if there is any damage. If they appear, the walking ability test fails.

During the test, the cube rolls around without any problem on all four sides and with visual inspections, no damage has been found on the cube. The test is successful.

7.5 Jump Up action and Balance action

To evaluate the final goal as the combination of the balance and the Jump Up action, a receptiveness test have been done. In this test the cube is placed on the table without any external help, the cube should be able to jump up and balance on its edge. The balancing should continue for at least 25 s before the trial is approved. The test should be done with a number of 25 test series.

The test has been done with the LQR controller with look-up table and the cube has successeded 4 times out of 25 which gives a success rate of 16 %. Three of these successful tests are represented in *Figure 7.6*.

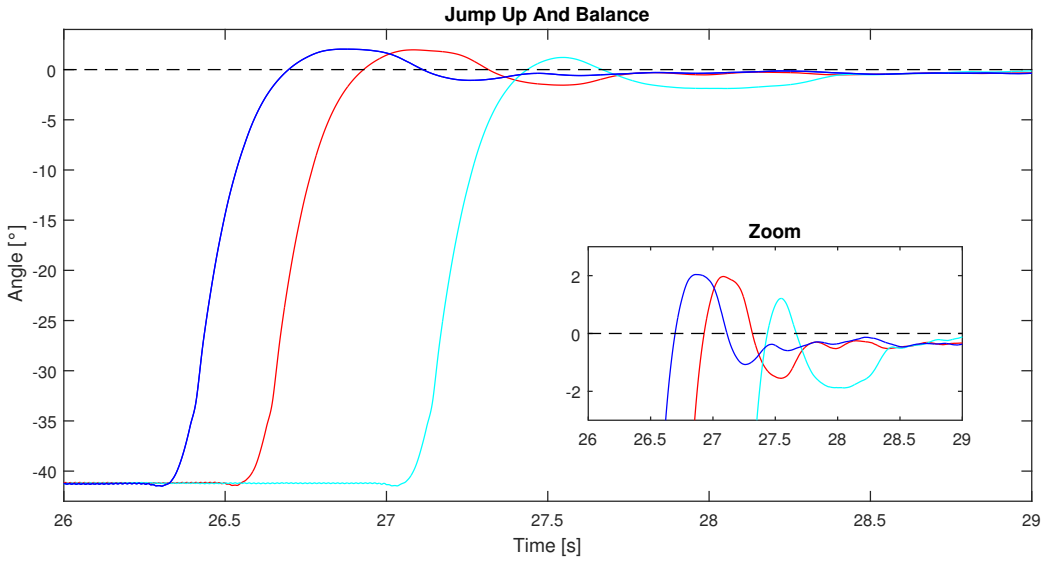


Figure 7.6: Jump Up and Balance action test where each of the colours represents a test scenario when the cube jumps up and balances on its edge.

8
Discussion

This chapter discusses the project development and also discuss the results achieved.

8.1 Project development

During the project the initial goal has been modified more than one time. The initial plan was to build a cube which was able to jump up and balance on one of the corners. It was later discovered that it was too ambitious with in the limited time of the master thesis project. The modeling as well as the hardware/ software design were two parts that took much longer time than expected. Every single component, from the weight of the senors to speed of engine, is optimized to give the best possible combination to finally have a system which can jump up and balance. Another main initial goal was to focus the project on evaluating different control methods. Due to the time limit, there were not much focus on developing our own controller, so instead we make use of controller methods which have been proven to work well, namely LQR and PID- controller.

8.2 Material selections

The material selection of the different parts is based on several parameters. For example the sides and frames is of high priority to reduce the weight while keeping the durability to handle repetitive falls from balancing position. One initial guess was that the plastic material would not be able to keep that high durability but evaluating the manufactured corners, there could be seen that stability was much more then enough. Due to this, it would be interesting to produce more parts in plastic. In general we have found that each aluminum produced part can reduce its weight by approximately 50 % by changing material to the 3D printed plastic. In this case the total weight of the cube can be reduced by around 500 g. Having a lighter cube would decrease the maximum velocity of the reaction wheel needed during brake action. In addition to this, there would be a opportunity to install a smaller reaction wheel, which contribute to make the cube smaller.

3D printer materials also have limitations on accuracy, and all the components that attach to reaction wheel are highly sensitive against unbalanced configuration. Today the reaction wheel holders are constructed with maximum positive uncertainty of 0.002 mm and the 3D printer used in this project is able to produce

parts with accuracy rate of 2 % from the specified CAD file.

8.3 Balance action

The system manages to balance with 100 % repetitiveness and is also able to handle external disturbances up to +-2.1 degrees. This can be compared to the evaluated physical limits of the cube given the current engine and reaction wheel which according to *7.2.1 System limits* are +-3.5 degrees. With help of an optimal controller, in a sense of handling low frequent disturbance, the cube should be able to handle disturbances up to 3.5 degrees. Although one should note that the optimization is a trade off in between handling low and high- frequency disturbance. So, optimizing the system to be able to handle low frequency disturbance up to 3.5 degrees would effect the sensitivity against high frequency noise.

8.4 Jump Up action

According to *7.3.1 test of repetitiveness*, the system manages to jump up within the balancing area 85 % of the time. Important to mention is that some calibrations needed to be done before the system could jump up one time. These calibrations need to be done before every test series to make sure that a perfect starting velocity has been set. In the following subsection parameters that affect the repetitiveness behavior and possible improvements will be discussed.

8.4.1 The servo power varies

The power to the brake servo motor is generated from one of the Lipo batteries. According to *Table 6.1 Servo specification* the operating speed and the stall torque from the servo changes due to the voltage that are given to the servo. Different speed and different stall torque affect the braking force generated on the reaction wheel which changes the momenta generated during a braking procedure. In a test series of jump up tests the voltage is only dependent on the servo, but when the systems are combined the battery to the servo motor is also used during the balancing action which can affect the braking force due to the battery level. One solution to this problem would be to add a feedback controller on the battery power signal and take this into consideration when braking action is finished.

8.4.2 Friction between brake pad and reaction wheel

The friction in between the brake pads,made of rubber, and reaction wheel, made of steel, affects the momenta generated during the braking procedure. The friction changes due to temperature as well as the total area of the rubber which is attached to the wheel surface. Both of these parameters give uncertainty to the total momenta generated from the brake.

8.4.3 Rubber bands variations

The braking has been done with help of two brake pad holders. The holders are pushed away from the reaction wheel with help of two rubber bands and are pressed together with the servo motor. Depending on the force difference between the two rubber bands, the brake force from the upper performs differently compared to the under one. Different brake force effects change the friction of both the brake pads which can bend the wheel. Another thing is that one brake pad can connect to the reaction wheel before the first one, which lowers the velocity of the reaction wheel a little bit before both of the brake pad hit the reaction wheel and can stop the reaction wheel much faster. Among several tests, the rubber band does not follow a specific line and can also slide little bit to left or right which would change the brake force. The difference was not measured by the authors, but would be one thing to improve.

8.4.4 Initial braking speed varies

The speed that the reaction wheel has before the braking procedure starts varies up to 20 rpm during a test series. This gives a small change between the tests brake momentum which change the behaviour of the Jump Up Action.

8.5 Jump Up and Balance Action

Evaluating Jump Up action and Balance action separately give a hint on how good the whole system would work. The separate braking system has succeed to jump up within the balancing space 85 % of the times but the total success rate of the whole system is 4 times out of 25. There are several factors which could affect the final system and in the following subsections they will be discussed.

8.5.1 Switch between Jump Up Action and Balance Action

According to *5.2.3 Jump and Balance* in the Control Chapter, the two systems, Jump Up and Balance, are connected to each other by switching off one controller and switching on the other one. In the real system, this switching is done when the cube is 10 degrees away from the balancing point. Due to that there exists some energy after the braking is done, the cube continues to move up to the balancing position while it is in the switching mode.

When the balancing controller becomes activated, the cube is still on the way up and due to this position the regulator assumes that the cube may need more power to move to balancing position. In this way the balancing regulator can catch the cube if the jump up force is too low. But there is a problem with this procedure when the Jump Up action is perfect or too hard. In that cases, the balancing regulator will, in the beginning, try to compensate the 10 degrees by adding high torque. This torque together with the remaining energy from the braking system will end up in a torque that is too big for the controller to handle when the cube

reaches zero degree with a high velocity. The cubes motor, is then to weak to be able to compensate for the high velocity and the cube will fall over to the other side.

This could be one of the reason why the Jump Up and Balancing Action is not as well as the action when they are separate. An improvement would be to evaluate a new regulator method that would take consideration of this. For example by feeding forward the accelerometer value of the cube and estimate if the regulator should start in 10 degrees, or if it should wait until the cube is closer to the balancing position.

8.5.2 Feed forward at braking system

One way to improve the Jump Up action would be to feed forward measurements during the breaking and use this data to decide whether we should brake longer or shorter time. Given the result seen in *Figure 7.4* and *Figure 7.5* which evaluates if there is any clear differences between the case where the cube falls back, and the case where the cube falls over. That could be seen in the Figures that by use accelerometer measurements as feedback signal and a boundary at $3.5 * 10^4$, we are able to cluster the two different cases in separate clusters except the two tests which would be clustered wrong. Using this measurements as feedback to breaking action has been tested with no visible effect. The activation time of the servo combined with brake is longer then the time which is left in the braking action. One way to overcome this is to move the sampling of data earlier in the braking process, evaluate similar tests as in *7.3.2 Improvements of brake system* at earlier data given that both cases where clustered together, and could not be separated at all.

8.6 Simulation vs reality

Comparing the early results given in simulations with the final results of the real system, it could be seen that all the simulation results are optimistic, and give better result in each area. For example the maximum angle at which the engine alone would be able to bring up the cube is in simulations +-6 degrees, where in real system the limit is tested to only +-3.5 degrees. The simulation is calculated in an optimal case, where voltage level from batteries is constant max, engine produces maximal moment according to data sheet constant, and wheel does not wobble. How much momentum is lost has not been evaluated.

8.7 Jump Up and Balance on a corner

As mentioned in *1.3.5 Further goals* this thesis is the first stage of a bigger project. In the end the cube should contain three reaction wheels and be able to jump up and balance on a corner.

8.7.1 Jump Up possibilities

To be able to jump up to one of its corners, three more reaction wheels need to be implemented. When the cube should go between balancing on a side to balancing on a corner, two reaction wheels will work together to manage a jump up procedure. For the moment one reaction wheel manages to jump up 2083 g and with the estimated final weight as 3487 g, two reaction wheels should be able to handle this jump without any problem.

The reaction wheel motor can increase the speed of the reaction wheel by 1180 rpm. This extra speed should be able to move 1404 g more than before. To make this possible some of the components would have to be moved down to lower the CoG, but we believe that this is just an optimisation problem that could be solved.

8.7.2 Sensors

The selected IMU sensor, MPU-6050, used a I^2C protocol. The maximum bit rate within the selected IMU is 400 kbps. With one sensor connected the capacity speed is high enough, but to be able to handle in total three to four sensors, when balancing on a corner, the data speed needs to be increased. Otherwise the sample time would increase. To enhance the performance, while still being able to use as much as the already generated code as possible, similar sensor with a different protocol could be used. There are sensors which uses a faster protocol, SPI.

9
Conclusion

A balancing cube has been developed during this project, the development includes modeling, simulations, software design, hardware design and construction. The main goal is to build a mechatronic system, i.e. cube, which is able to jump up without any external force, up to one of its edges and then balance there for more than 25 s. We accomplish the goal for 4 out of 25 times using an own designed mechanical brake together with a designed reaction wheel manufactured in a CNC machine. The control system is divided into two parts, Jump Up Action and Balance Action. During the different actions, there are two separately controllers which are active. The system makes use of both PID controller and a look-up tabled LQR-controller with added integrated states.

Bibliography

[1] National Geographic Society. Moon exploration, http://science.nationalgeographic.com/science/space/space-exploration/moon-exploration-article/.

[2] T. Yoshimitsu T. Kubota. Intelligent unmanned explorer for deep space exploration. Technical report, Institute of Space and Astronautical Science, Sahamihara, Japan, 2007, Available at http://arxiv.org/ftp/arxiv/papers/0804/0804.4717.pdf.

[3] Michael McHenry et al. Marco B. Quadrelli. Guidance, navigation, and control technology assessment for future planetary science missions. Technical report, National Aeronautics and Space Administration, California, USA, 2013, Available at https://solarsystem.nasa.gov/docs/GNC%20Tech%20Assess_Part%20III_Surface%20GNC_20130402_soo.pdf.

[4] Carl W. Akerlof. The chaotic motion of a double pendulum. notes, Department of Physics, University of Michigan, September 2012.

[5] Nalin A. Chaturvedi Dennis Bernstein Harris McClamroch Jinglai Shen, Amit K. Sanyal. Dynamics and control of a 3d pendulum. In *43rd IEEE Conference on Decision and Control*, pages 323–328, University of Michigan, December 2004.

[6] T. Widmer M. Gajamohan, M. Muehlebach and R. D'Andrea. The cubli: A reaction wheel-based 3d inverted pendulum. In *2013 European Control Conference*, pages 268–274, Switzerland, July 2013.

[7] Kyle Gilpin John W. Romanishin and Daniela Rus. M-blocks: Momentum-driven, magnetic modular robots. *IROS*, 2013.

[8] David Morin. *Introduction to CLASSICAL MECHANICS With Problems and Solutions*. Cambridge University Press, 2008.

[9] Igor Thommen Mohanarajah Gajamohan, Michael Merz and Raffaello D'Andrea. The cubli: A cube that can jump up and balance. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3722–3727, Portugal, October 2012.

[10] Gajamohan Mohanarajah. *The Cloud, Paper Planes, and the Cube*. PhD thesis, Tokyo Institute of Technology, Japan, 2014.

[11] Kris Temmerman. Making a rolling cube, Mars 2015, https://www.youtube.com/watch?v=-1PDWG4GqNE.

[12] K. Åström and T. Hägglund. *PID Controllers: Theory, Design and Tuning*. Instrument Society of America, USA, 1995.

[13] Karl Johan Aström and Richard M. Murray. *Feedback Systems*. Princeton University Press, USA, February 2009.

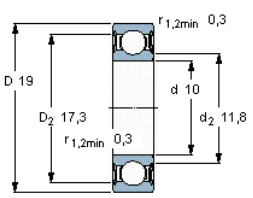
[14] Michio Sugeno Tadanari Taniguchi, Luka Eciolaza. Look-up-table controller design for nonlinear servo systems with piecewise bilinear models. 2013, Available at http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6622512.
[15] Ragnar Grahn and Per Åke Jansson. *Mekanik*. Studentlitteratur, Lund, Sweden, 2002.
[16] Kjell Melkersson Mart Mägi. *Lärobok i Maskinelement Del A*. EcoDev International AB, Sweden, 2012.
[17] Engineeringtoolbox. Area moment of inertia, http://waijung.aimagin.com/.
[18] ZEITLAUF. Brushless dc machinery, 2015, http://www.zeitlauf.com/product_info/technology/motoren/ec_motor.html.
[19] Oludayo John Oguntoyinbo. Pid control of brushless dc motor and robot trajactory planning and simulation with matlab/simulink. Master's thesis, VASA YRKESHÖGSKOLA UNIVERSITY OF APPLIED SCIENCES, February 2009.
[20] Josh Edberg. Selecting hall-effect sensors for brushless dc motors. *ECN ECN Newsletter*, 2012, Available at http://www.ecnmag.com/article/2012/10/selecting-hall-effect-sensors-brushless-dc-motors.
[21] Giorgos Lazaridis. How rc servos works, June 2009.
[22] InvenSense Inc., USA. *MPU-6000 and MPU-6050 Product Specification Revision 3.3*, May 2012.
[23] InvenSense Inc., USA. *MPU-9250 Product Specification Revision 1.0*, January 2014.
[24] Shane Colton. The balance filter a simple solution for integrating accelerometer and gyroscope measurements for a balancing platform. Massachusetts Institute of Technology, June 2007.
[25] OlliW's Bastelseiten. Imu data fusing: Complementary, kalman, and mahony filter. *Grundlagen*, 2015.
[26] STMicroelectronics. Stm32f4discovery, 2016, http://www.st.com/web/catalog/tools/FM116/SC959/SS1532/PF252419.
[27] John Salt. Understanding rc lipo batteries big page - big subject!, http://www.rchelicopterfun.com/rc-lipo-batteries.html.
[28] Cheetah62. Lithium-polymer (lipo) batteries faq, 2010, http://www.helifreak.com/showthread.php?t=194395.
[29] Byte Paradigm sprl. Introduction to i²c and spi protocols, 2015, http://www.byteparadigm.com/applications/introduction-to-i2c-and-spi-protocols/.
[30] AIMAGIN CO. Waijung blockset, April 2015, http://waijung.aimagin.com/.

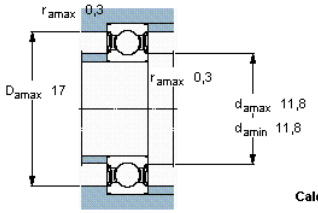
A
Appendix

A.1 Bearing

SKF

Principal dimensions			Basic load ratings		Speed ratings		Designation
			dynamic	static	Reference speed	Limiting speed	
d	D	B	C	C0			* SKF Explorer bearing
mm			kN		r/min		-
10	19	5	1,38	0,585	-	22000	**61800-2RS1**



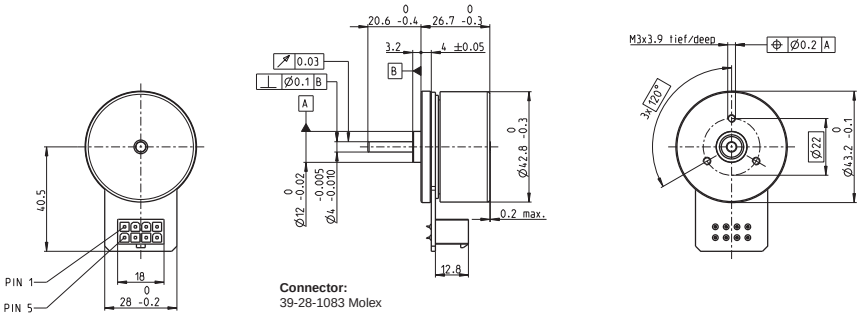


Calculation factors

k_r 0,015
f_0 9,4

A.2 Reaction Wheel Actuator

EC 45 flat ∅42.8 mm, brushless, 70 Watt



Connector:
39-28-1083 Molex

M 1:2

maxon flat motor

Stock program
Standard program
Special program (on request)

Part Numbers

	with Hall sensors	397172	402685	402686	402687
Motor Data (provisional)					
	Values at nominal voltage				
1	Nominal voltage	V	24	30	36	48
2	No load speed	rpm	6110	6230	6330	3440
3	No load current	mA	234	194	166	48.1
4	Nominal speed	rpm	4860	4990	5080	2540
5	Nominal torque (max. continuous torque)	mNm	128	112	108	134
6	Nominal current (max. continuous current)	A	3.21	2.36	1.93	0.936
7	Stall torque	mNm	1460	1170	1100	915
8	Stall current	A	39.5	25.8	20.7	6.97
9	Max. efficiency	%	85	84	83	84
	Characteristics					
10	Terminal resistance phase to phase	Ω	0.608	1.16	1.74	6.89
11	Terminal inductance phase to phase	mH	0.463	0.691	0.966	5.85
12	Torque constant	mNm / A	36.9	45.1	53.3	131
13	Speed constant	rpm / V	259	212	179	72.7
14	Speed / torque gradient	rpm / mNm	4.26	5.44	5.85	3.82
15	Mechanical time constant	ms	8.07	10.3	11.1	7.24
16	Rotor inertia	gcm²	181	181	181	181

Specifications

Thermal data

17	Thermal resistance housing-ambient	3.56 K/W
18	Thermal resistance winding-housing	4.1 K/W
19	Thermal time constant winding	29.6 s
20	Thermal time constant motor	178 s
21	Ambient temperature	-40 ... +100°C
22	Max. winding temperature	+125°C

Mechanical data (preloaded ball bearings)

23	Max. speed	10000 rpm
24	Axial play at axial load < 4.0 N	0 mm
	> 4.0 N	0.14 mm
25	Radial play	preloaded
26	Max. axial load (dynamic)	3.8 N
27	Max. force for press fits (static)	50 N
	(static, shaft supported)	1000 N
28	Max. radial load, 5 mm from flange	21 N

Other specifications

29	Number of pole pairs	8
30	Number of phases	3
31	Weight of motor	141 g

Values listed in the table are nominal.

Connection

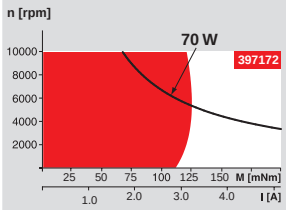
Pin 1	Hall sensor 1*
Pin 2	Hall sensor 2*
Pin 3	V_{Hall} 4.5 ... 18 VDC
Pin 4	Motor winding 3
Pin 5	Hall sensor 3*
Pin 6	GND
Pin 7	Motor winding 1
Pin 8	Motor winding 2

*Internal pull-up (7 … 13 kΩ) on pin 3
Wiring diagram for Hall sensors see p. 35

Cable

Connection cable Universal, L = 500 mm	**339380**
Connection cable to EPOS, L = 500 mm	**354045**

Operating Range



Comments

Continuous operation
In observation of above listed thermal resistance (lines 17 and 18) the maximum permissible winding temperature will be reached during continuous operation at 25°C ambient.
= Thermal limit.

Short term operation
The motor may be briefly overloaded (recurring).

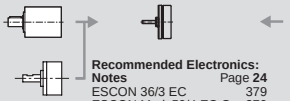
Assigned power rating

maxon Modular System

Overview on page 20–25

Planetary Gearhead
∅42 mm
3 - 15 Nm
Page 316

Spur Gearhead
∅45 mm
0.5 - 2.0 Nm
Page 317



Encoder MILE
256 - 2048 CPT,
2 channels
Page 342

Recommended Electronics:

Notes	Page **24**
ESCON 36/3 EC	379
ESCON Mod. 50/4 EC-S	379
ESCON Module 50/5	379
ESCON 50/5	380
DEC Module 50/5	382
EPOS2 Module 36/2	386
EPOS2 24/5, 50/5	387
EPOS2 P 24/5	390
EPOS3 70/10 EtherCAT	393
MAXPOS 50/5	396

Option
With Cable and Connector
(Ambient temperature -20 ... +100°C)