AALBORG UNIVERSITY
STUDENT REPORT

ED4-2-F16

# Cube based balancing system

*Students:*
Mihkel Soolep

*Supervisor:*
Christian Mai
Leif Hansen

November 23, 2016

**AALBORG UNIVERSITY**
STUDENT REPORT

**School of Information and Communication Technology**
Niels Bohrs Vej 8
DK-6700 Esbjerg
http://sict.aau.dk

**Title:**
Cube shaped balancing system

**Theme:**
Scientific Theme

**Project Period:**
Spring Semester 2016

**Project Group:**
H107

**Participant(s):**
Mihkel Soolep

**Supervisor(s):**
Christian Mai
Leif Hansen

**Copies:** 3

**Page Numbers:** 35

**Date of Completion:**
November 23, 2016

**Abstract:**

In the paper at hand we introduce a cube shaped system, what can jump up from the edge and then balance itself on the corner. Movement of the system has been made possible by an embedded momentun wheel what can rotate on high angular velocities to move the cube. Mainly we can see the development of the prototype for one side of the cube. When the prototype is made to jump up it starts to balance itself on the edge by controlling the motor torque with a nonlinear controller. Mainly we focus on the theoretical side of the controller.

# Contents

# Preface

P4 project

From hereby on, every mention of 'we' refers to the author listed below.

Aalborg University, November 23, 2016

Mihkel Soolep
<msoole14@student.aau.dk>

# Chapter 1

# Introduction

## 1.1 Introduction

In this project we will try to understand and implement controlling of an unbalanced device via using non-linear control. Our aim is to build a prototype for a cube what would be able to move and balance on its edges without external forces. The cube would move and balance itself thanks to the embedded reaction wheel. Goal of our project is to make one side of the cube and try to implement the nonlinear controller that we develop during this project. The idea was adapted from an older work done in ETH Zurich, Switzerland by Mohanarajah Gajamohan, Michael Merz, Igor Thommen and Raffaello D'Andrea. [1] In the mentioned project they finally completed the whole cubic, but in our case we are only looking forward to build one side of it in the duration of this project.

We have taken up this project because it is a challenge for us to make this kind of a controller that can balance the system and maintain it. Similar kind of systems are used in self assemling robots and satellites. [2] [3] Since the work on using the reaction wheel and other ideas in these kind on projects is still ongoing we feel that this work is actual and worth the effort.

The principle of the device is simple. First we make the wheel go very fast and then bring it to a rapid stop by breaking the wheel with a servo motor. Energy what was stored on the wheel then transforms over to the whole body and the system should move to the desired position. If the modelling proves to be corret we should have the body in upright position.

Balancing of the body is done by using the torque of the reaction wheel dependent on the side that it is leaning towards to. To measure the offset of the body we use the Earth's gravity and an accelerometer. The angle of the offset and thus the required torque is calculated by the projection of the gravity vector what we get from the sensor.

Controlling of the system is done by a non-linear controller implemented by software on a microcontroller. Rapid breaking of the wheel is done by a servo motor mounted on the body. In next chapters we will give more in depth about what we did and how.

## 1.2 Hypothesis

In our project we are concentrating on the reaction wheel moving the devices body by changing the acceleration of the reaction wheel. We will be constructing on one side of the cube and will place a reaction wheel in the middle of it. Reaction wheel is there to make the "cube" jump up and then balance on one of its corners. Our hypothesis is that the wheel will be able to store enough energy to jump, when stopped and will be able to balance the system on one of its corners just by altering the torque produced by the wheel.

In order to change the torque and balance the system the controller devolped have to be as fast as possible. One of our hypothesis is that a linear controller would have more overshoot and it would be slower. Only way to find it out is to model both of them. Since we have a physical limit on the maximum speed and torque of the motor we must be awere that from certain angle the motor may not be sufficient enough to push the system in upright position again.

## 1.3 Aims of the project

The aim of this project is to build a prototype for a cube that jumps up and balances on one of its corners. More specifically the aims are listed below

- Build a simulation model for this specific system
- Build the real plant of this system
- Determination of system parameters
- Design a linear controller for the system
- Design a nonlinear controller for the system
- Impliment the designed controller on the real plant and evaluate the results

In the end of this paper we can hopefully give a positive answer to all of those points described above. We have fully understood that the amount of work and dedication needed to fulfil these goals is quite big. The biggest and most important goal is to learn and understand the controlling of the system and be able to implement that knowledge on fututre tasks.

## 1.4 Assumptions

In order to theoretically model the system we must make some assumptions which are listed below:

- The breaking system works nearly ideal
- The accelerometer works without huge error
- The base of the system is static with respect to the ground
- The system is able to rotate around the pivot point freely
- DC motor is able to change its speed and direction fairly quickly

While modelling the system these assumptions must be kept in mind in order to design the best possible solution.

# Chapter 2

# Development

## 2.1 Description

In this section we will looking the device more detailed and give the description of all the parts used in the system.

For the reaction wheel we are using a brushless DC motor from maxon EC 45 flat 50W. We chose a brushless DC motor because it has less friction and therefore we have less energy going to waste. The particular motor was chosen for its performance specifications. The main criteria was to find a motor with enough torque for our system. (All the values used onwards in this project are taken from the datasheets or are the assuming values of the physical details what would be used in the prototype.) As we can see from the datasheet the maximal continious torque on that motor is 83.4 nNm what is sufficient for our system. It also has high maximum rpm which is 6710. In our first model and culations we determined that the speed needed to jump the cube

$$\Delta E = mg\Delta h \tag{2.1}$$

$$\Delta E = \frac{1}{2}I\omega^2 \tag{2.2}$$

$$\frac{1}{2}I\omega^2 = mg\Delta h \tag{2.3}$$

$$I = m_w r^2 \tag{2.4}$$

$$mg\Delta h = \frac{1}{2}m_w r^2\omega^2 \tag{2.5}$$

$$\omega^2 = 2\frac{mg\Delta h}{m_w r^2} \tag{2.6}$$

Where $m$ is the mass of the whole system, $g$ is the gravitational acceleration, $\Delta h$ is the height differents of the two mass centres in diffetent stable positions. $m_w$ is the mass of the reaction wheel and $r$ is the radius of the wheel.

When we insert our initial values of the device, what we designed we get the rotational speed approximately 289 rpm. As we can see the motor is more than sufficient to make the cube jump.

The motor is controlled by a Maxon escon 36/3 EC servo controller driver. This driver was chosen because of the compatibility with the motor. On this patricular driver it is possible to control the motor with current control or speed control. In our system it is vital that we can control the motor by current control because $T_m(t) = K_m u(t)$ as we can see the torque, what is our main input to control, is what we need to manipulate.

To break the wheel we have built a breaking system with a servo motor Hitec HS-485HB. The breaking of the wheel is crucial in the jumping sequence to convert the wheels rotational energy into the potential energy of the whole body. Breaking is done by physically constricting the wheel which the servo.

While balancing the system we use the torque of the wheel to move its angle. For determining the position of the wheel we have mounted an accelerometer on the device. The accelerometer is MPU-6050. We are using the Earths gravitational accelerations projection on the vertical axes of the body to get the off-set angle.

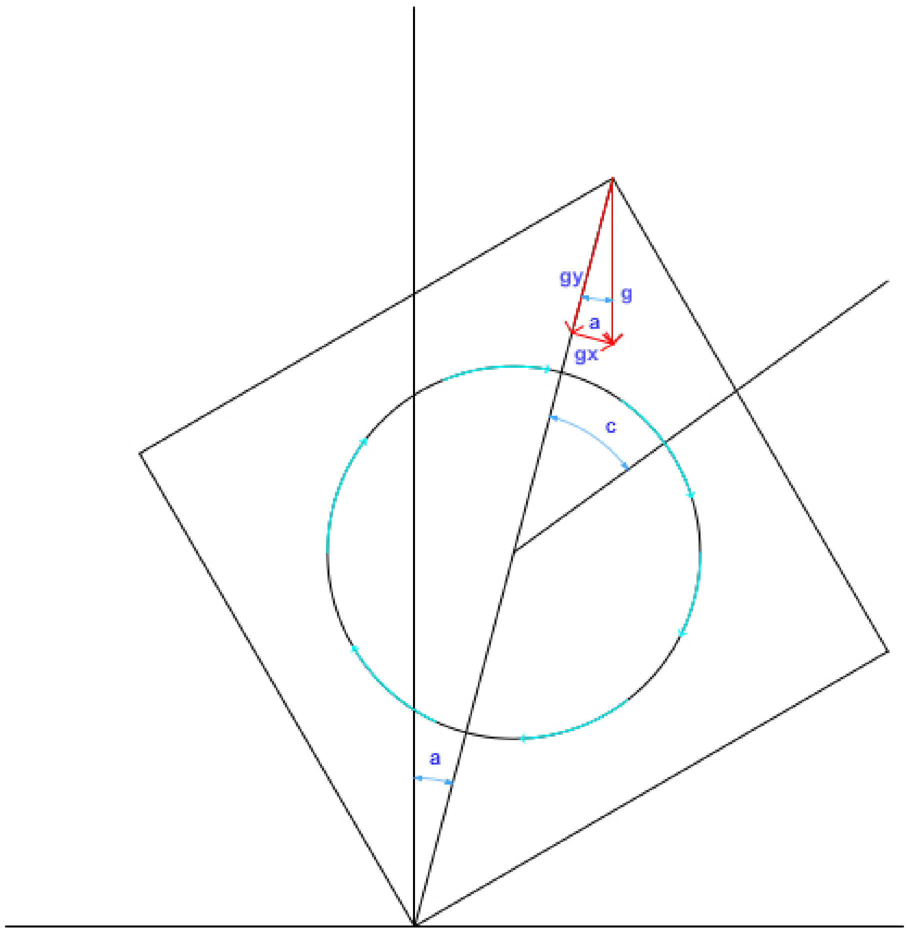



**Figure 2.1:** Cube offset angles and gravity vector

In the figure above c is later used as $\theta_w$ and a is used as $\theta_b$.

Body of the device is a square plate made out of aluminium. We have cavities in the plate because it makes the system less heavy and therefore we need less force to move it as we desire. On the square plate are mounted the reaction wheel made from the BLDC and a aluminium circle. Furthermore the breaking servo is mounted near the bottom bearing of the system. When the system is in the balancing point then on the top most corner is mounted the accelerometer.

## 2.2 Mechanical parts of the system

In order to implement the controller, what we have been designing in the duration of this project we must build an actual prototype for the system. Problablly the best way to do it is to take into consideration of the previously done projects. Since we have been studyng thoroughly the project made by the ETH students [1]. For the prototype we would be constructing only one side of the whole cube.

Base of the prototype will be a square shaped aluminium plate, in what we would have hole in the middle to mount the motor with the reaction wheel. Reaction wheel itself would be a round wheel with three spokes. The ideal solution for the prototype is shown on the figure below.

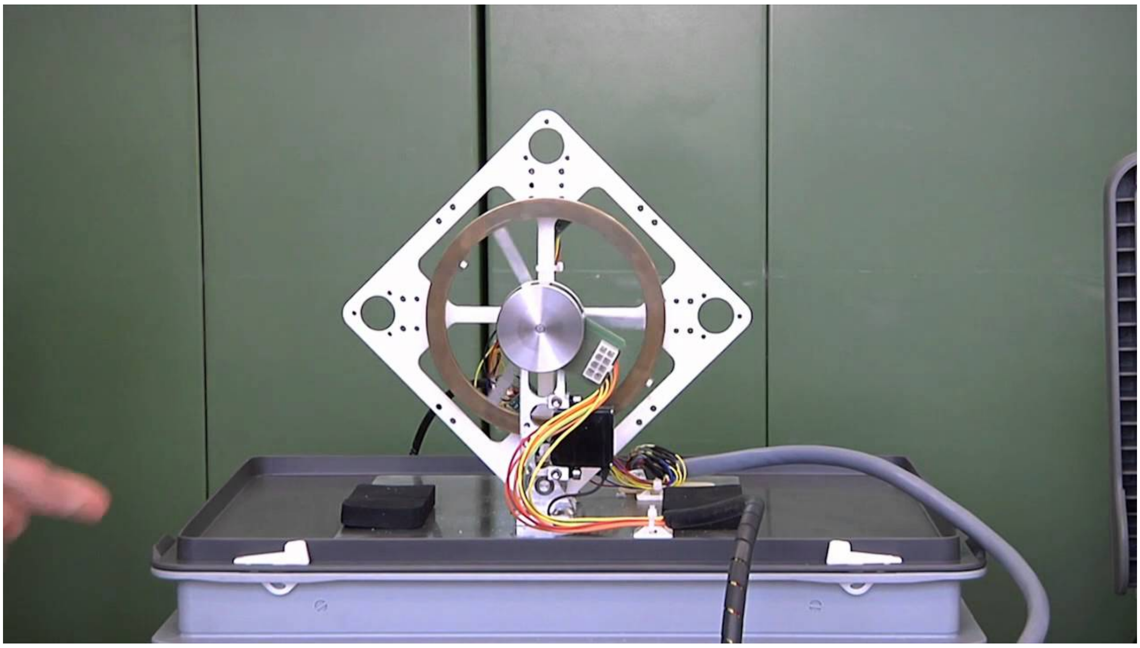


**Figure 2.2:** Possible mechanical setup for the project [4]

Mechanical pieces of the system:

- Aluminium square plate with dimensions of 15x15 cm and holes inside for the reaction wheel mount and to keep the weight low as possible.
- Reaction wheel made out of aluminium. Wheel itself is a circle with spokes and a hole in the middle to mount it on the brushless DC motor.
- Tabletop mount what can be fixed to one of the corners of the prototype by bearings to have the least amount of friction as possible.

This kind of design for the system should be sufficient enough for the aims of the project.

## 2.3 Electrical design

In previous section we looked at what could be the possible solution for the mechanical part of the system. In the current section we are considering the electronical parts that should be used inside the device inhand.

- Brushless DC motor EC 45 flat 50W from maxon
- Maxon escon 36/e EC servo controller driver
- Raspberry Pi 2B microcontroller for its fast processor and fast prototyping capability
- Accelerometer MPU 9150
- RC servo HSG-5084MG for the breaking system
- Embedded hall sensor on the motor for reading the wheel speed

Items listed above should be enough to implement the designed controller for the system. If there are any shortcomings of the electronics they will be dealt with while building the system.

## 2.4 Model of jump up

In our system it is crutial that we get the cube to jump up from the ground to the edge. In this section we are looking the theory behind the jumping motion.

First we look at the equations for rotational kinetic energy of the wheel and potential energy of the whole system.

$$E_{KR} = \frac{1}{2} I_\omega \dot{\Theta_\omega}^2 \tag{2.7}$$

Inertia of the wheel is described as

$$I_\omega = m_\omega r^2 \tag{2.8}$$

The potential and kinetic energy we are looking for is the difference between the two states(when the cube is on one side and when on edge).

$$E_{pot} = E_{pot}'' - E_{pot}' \tag{2.9}$$

Potential energy described on one side.

$$E_{pot}' = mgh = (m_w + m_b)gl \tag{2.10}$$

Potential energy described on the edge.

$$E_{pot}'' = mgh\sqrt{2} = (m_w + m_b)gl\sqrt{2} \tag{2.11}$$

Now we take into concideration that our system has ideal breaking system and the whole kinetic energy of the wheel is transformed into the potential energy of the whole system.

$$E_{KR} = E_{pot} \tag{2.12}$$

Now we combine the equations

$$\frac{1}{2} m_w r^2 \dot{\Theta_w}^2 = (m_w + m_b)g(l\sqrt{2} - l) \tag{2.13}$$

And finally whe take on to the one side of the equation all that we need.

$$\dot{\Theta_w}^2 = \frac{2(\sqrt{2}-1)(m_w + m_b)gl}{m_w r^2} \tag{2.14}$$

$E_{KR}$ is the kinetic energy of the wheel. $E_{pot}$, $E'_{pot}$ and $E''_{pot}$ are the potential energies used in the model. $m_w$ is the mass of the wheel. $m_b$ is the mass of the whole system without the wheel. $l$ is the lenght from the pivot point to the center of the mass. $g$ is the gravitational pull. $\dot{\Theta}_w$ is the rotational speed of the wheel. $r$ is the radius of the wheel.

From here we have the model for jumping up on the edge. In this part of the sequence we do not need to implement any control theory because we have no changing parameters. Jumping up is fine tuned with practical experiments because in real life we still have some unexpected variables effecting the system for example the data and powerlines going in to the section wheel and the sensor. In this case we are not concidering drag nor friction because we know that in reality we cannot make the breaking system ideal and therefore we have some error in calculation anyways. Furthermore in peliminary assesment the friction and drag losses are very low relative to the whole system energy changes.

Now when we enter the assuming parameters of the system into the equation what we determined above we can see what is the estimated wheel speed that we need to move the cube into the desired position. All the parameters used in this sequence are taken from the data.m file into which we got them from estimating the possible parameters of the system. After running through the calculations we get that the estimated wheel speed would be 30,247rad/s what is approximately 288 degrees/s.

## 2.5 Model of balancing in edge position

When the device has made the jump up sequence, it will start balancing on the corner of the cube. The model for balancing the cube is based on the Newton's laws of motion.

"Newton's laws of motion: The fundamental laws of mechanics that describe the way in which bodies in an internal frame move in response to the forces acting on them.

Newton's first law, also called Galileo's law, states that a body continues to move with a constant velocity or to main at rest unless acted on by an unbalanced external force.

Newton's second law states that the rate of change of momentum $p$ of a body equals the total force $F$ acting upon it:

$$F = dp/dt \tag{2.15}$$

If, as is normally the case, the mass of the body is constant, $F = \frac{d(mv)}{dt}$ reduces to $F = m\frac{dv}{dt}$ or

$$F = ma \tag{2.16}$$

where a is the acceleration of the body. Note that the force and acceleration are vectors. The first law is the null case of the second law (if $F = 0$ then $a = 0$).

Newton's third law states that if a body A exerts a force $F$ on body B, then body B exerts a force $-F$ on body A." [5]

From here we can see that if we apply a force on the wheel then it will be counteracted by the movement of the whole body. In our case we do it the opposite way. That means if the body is not on the balancing point it will be moved by the gravitational force towards the lower point of energy, which is the body on one of the sides in our case.

From the Newton's laws we will take that if the body is moved by an external force we have to couteract it with the force produced in the motor. Since torque is the moment of force [5] we can see that in the case of unbalanced system we have to use the wheels torque to get it back on the balancing point. From here we can state that $T_b$ the torque of the body must equal to $T_w$ the torque of the wheel.

$$T_b = T_w \tag{2.17}$$

In our case it is important to base the model that we can have so that the input to the motor is current. We need the input current because $T_m = K_m u$ where $T$ is the motor torque, $K_m$ is the torque constant and $u$ is the input current. Thus we get the equation:

$$T_b = K_m u \tag{2.18}$$

The offset angle is determined by the gravity vector. The accelerometer is set so that the x axis of the sensor is orthogonal to the vertical axis of the body when it is in upright position. The sensor is in the position that on the x axis of it it will allways show the projection of the gravity vector what is actually affecting the system. As we can see from the figure below the force what is affecting the body to tilt is $(m_b + m_w)gsin(\theta_b)$.

Now we must get the torque of the whole body to determine the relationship between the offset angle and required torque.

$$T_b = (m_b + m_w)l_b sin(\theta_b) \tag{2.19}$$

Where $m_b$ is the mass of the body, $m_w$ is the mass of the wheel, $l_b$ is the lenght from pivot point to the mass centre and $\theta_b$ is the offset angle. Furthermore we must take into concideration the friction between the bearings on the pivot point and the internal friction of the DC motor.

$$(m_b + m_w)l_b sin(\theta_b) - C_b\dot{\theta_b} = K_m u - C_w\dot{\theta_w} \tag{2.20}$$

Where $C_b$ is the friction coefficient of the body, $\dot{\theta}_b$ angular speed of the body, $C_w$ is the friction coefficient of the DC motor and $\dot{\theta}_w$ is the angular speed of the motor. From here we can get the equation for our system based on the current input.

$$u = \frac{(m_b + m_w)l_b gsin(\theta_b) + C_w\dot{\theta}_w - C_b\dot{\theta}_b}{K_m} \tag{2.21}$$

### 2.5.1 Nonlinear dynamics of the setup

In order to design the nonlinear state space controller for our system we must determine the nonlinear dynamic equation for the setup. The following equation is describing the system by the torques that are applied on the system while it is tilted and the reaction wheel tries to restabilize the body to the balancing point according to the body's acceleration.

$$\ddot{\Theta}_b(I_b + m_w l^2) = (m_b l_b + m_w l)gsin(\Theta_b) - T_m - C_b\dot{\Theta}_b + C_w\dot{\Theta}_w \tag{2.22}$$

Here we have the same equation made to correspond to the angular acceleration of the body. It is made to be according to the body because in our nonlinear control we have two inputs first one is according to the body.

$$\ddot{\Theta}_b = \frac{(m_b l_b + m_w l)gsin(\Theta_b) - T_m - C_b\dot{\Theta}_b + C_w\dot{\Theta}_w}{I_b + m_w l^2} \tag{2.23}$$

Second equation for the nonlinear dynamics of the system shows the dynamics of the body according to the wheels movement.

$$\ddot{\Theta}_w(I_b + m_w l^2) = \frac{(I_b + I_w + m_w l^2)(T_m - C_w\dot{\Theta}_w)}{I_w} - (m_b l_b + m_w l)gsin(\Theta_b) - C_b\dot{\Theta}_b \tag{2.24}$$

And now we make the equation according to the $\ddot{\Theta}_w$

$$\ddot{\Theta}_w = \frac{(I_b + I_w + m_w l^2)(T_m - C_w\dot{\Theta}_w)}{I_w(I_b + m_w l^2)} - \frac{(m_b l_b + m_w l)gsin(\Theta_b) - C_b\dot{\Theta}_b}{(I_b + m_w l^2)} \tag{2.25}$$

Both of the equations describe the dynamics of the system but the difference is that first one has the background system as the body. Second dynamic equation is where the wheel is taken as the dynamic background system. If we add up those equations we get the body's dynamic equation according to nonmoving background system.

### 2.5.2 State Space Model

Since we have determined our nonlinear dynamic equations we can compose our state space model.

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned}$$

The matrices A and B are properties of the system and are determined by the system structure and elements. The output equation matrices C and D are determined by the particular choice of output variables [6]. *dotx* and $y$ are the input and output matrices.

We assigne the state vector $x$ as follows

$$x(t) = \begin{bmatrix} \Theta_b \\ \dot{\Theta}_b \\ \dot{\Theta}_w \end{bmatrix}$$

where

$\Theta_b$ is the angular position of the body,$\dot{\Theta}_b$ is the angular velocity of the body and $\dot{\Theta}_w$ is the angular velocity of the reaction wheel.

Our systems output vector is:

$$y(t) = \begin{bmatrix} \Theta_b \end{bmatrix}$$

And the input vector is:

$$u(t) = \begin{bmatrix} i \end{bmatrix}$$

For calculating the matrices of the linearized system around the $x(0) = [0; 0; 0]$ point we are using the equations from the last section (2.23) and (2.24).

The property matrix A is given [6]

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{2,2} & a_{2,3} \end{pmatrix}$$

according to what we can give our system propery matrix A as:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ \frac{(m_b I_b + (m_w l))g}{I_b + m_w l^2} & -\frac{C_b}{(I_b + m_w * l^2)} & \frac{C_w}{I_b + m_w l^2} \\ -\frac{(mbI_b + m_w l)g}{I_b + m_w l^2} & \frac{C_b}{I_b + m_w l^2} & -\frac{C_w (I_b + I_w + m_w l^2)}{I_w (I_b + m_w l^2)} \end{bmatrix}.$$

Second property matrix B we get from $3 \times 1$ matrix

$$B = \begin{pmatrix} b_{1,1} \\ b_{2,1} \\ b_{3,1} \end{pmatrix}$$

is according to the formula above [6]

$$B = \begin{bmatrix} 0 \\ -\frac{K_m}{I_b+m_w l^2} \\ \frac{K_m(I_b+I_w+m_w l^2)}{I_w(I_b+m_w l^2)} \end{bmatrix}$$

The output matrix C is represented as follows

$$C = \begin{pmatrix} c_{1,1} & c_{1,2} & c_{1,3} \end{pmatrix}$$

is given by the formula [6]

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

And finally the second output property matrix is represented as:

$$D = (d_{1,1})$$

is given by the formula

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

“When we use the continuous time system given by $\dot{x}(t) = Ax(t) + Bu(t)$ and descretized using zero-order hold and the resulting discrete time model is given by

$$x[k+1] = A_d[k] + B_d u[k], k \in \mathbb{N} \tag{2.26}$$

where $A_d$ and $B_d$ are the discrete-time counterparts of the continous time state matrix $A$ and input matrix $B$. For the sake of simplicity we use the same notation to represent the continuous and discrete time versions of the state $x$ and input $u$.

Using the above discrete time model, a Linear Quadratic Regulator (LQR) feedback controller of the form

$$u[k] = -K_d(\hat{\theta}_b[k], \hat{\dot{\theta}}_b[kk], \hat{\dot{\theta}}_w[k]) \tag{2.27}$$

was designed, where $K_d = (K_{d1}, K_{d2}, K_{d3})$ is the LQR feedback gain ” [1]

## 2.6 Simulink model

Before we can implement the control system, what we have created for the cube, on the microcontroller we must simulate it. For the simulation we have designed a nonlinear and linear controller in simulink.

The highest level of the system will be shown on the figure below.

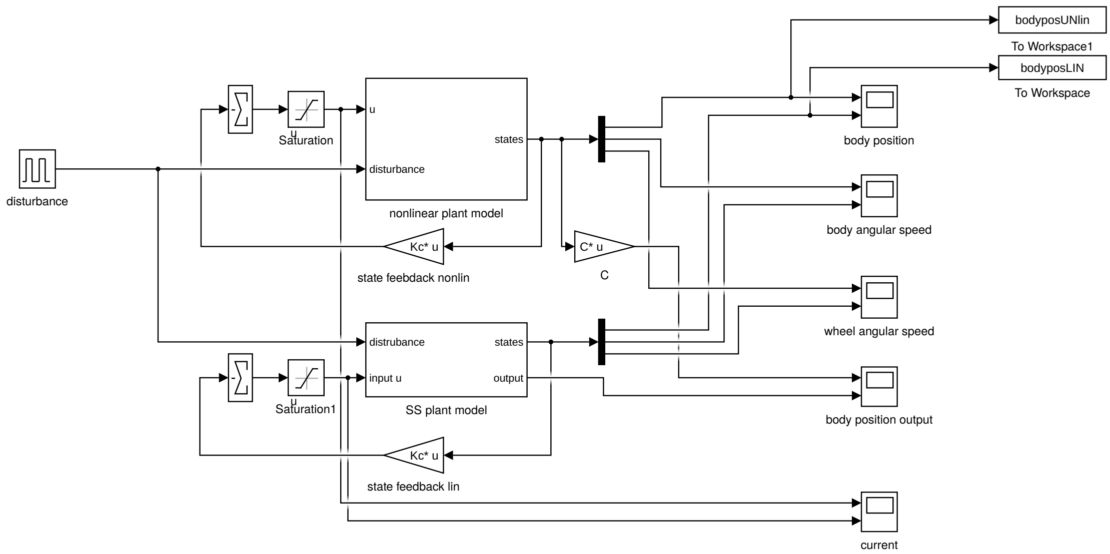



**Figure 2.3:** Highest level of the system

Here we can see the two systems what we created the top loop named nonlinear plant model is the nonlinear state space model of the system. The bottom loop what is named SS plant model what is the linear model of the system. The linearization was made by using the small angle approximation principle where $sin(x) = x$ [7].

In the following figure we can see the nonlinear plant model upclose.

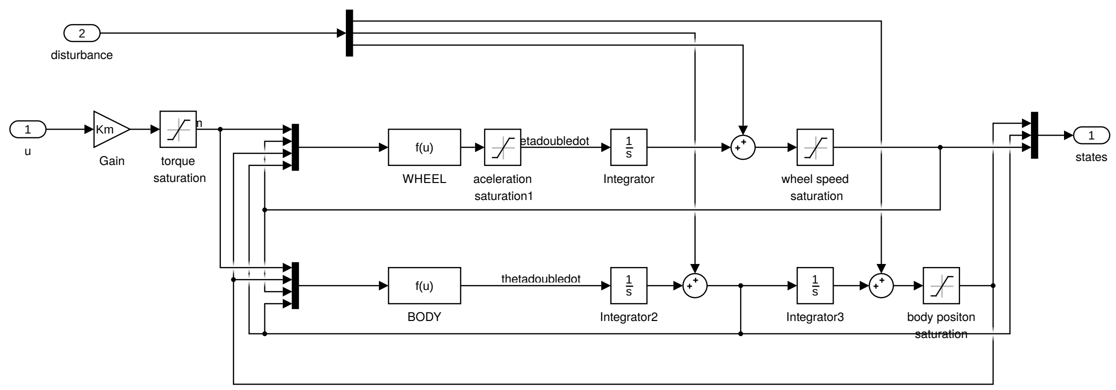



**Figure 2.4:** Nonlinear plant model

Following figure below shows the linear plant model where we used the small angle approximation.

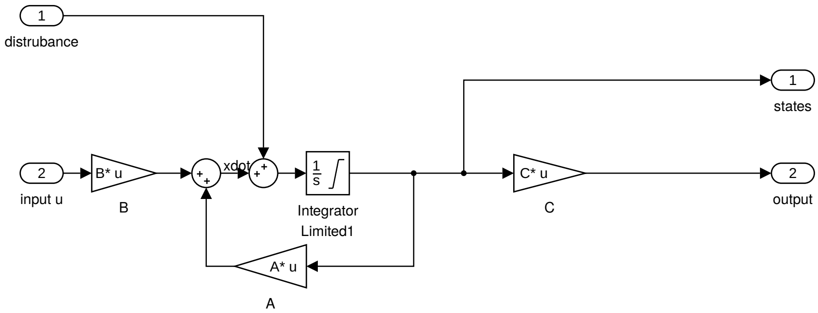



**Figure 2.5:** Linear plant model

As it is shown on all the models we are using the saturations to evaluate the maximal values of the inputs and outputs according to our set parameters. On the body position we have the max values of 1,6rad and -1,6rad what are approximately 90 degrees and -90 degrees. These both are the positions where the cube is on one or the other side. The current saturation has the maximum and minimum values 5 and -5 ampers what are the maximum currents that the motor is taking. In the wheel speed saturation we have the maximum values of the wheel speed what are give on the datasheet. Maximum and minimum wheel speeds are in our case 628rad/s and -628rad/s. In the torque saturation are the maximum and minimum values of the maximum constant torque that the motor can supply. According to the datasheet they are 0,0834 Nm and - 0,0834 Nm.

On all the following graphs the yellow line shows the nonlinear model and the blue line is the linear model.

We ran a simulation with a disturbance of 0,4rad and 0,1s and we got the following results:

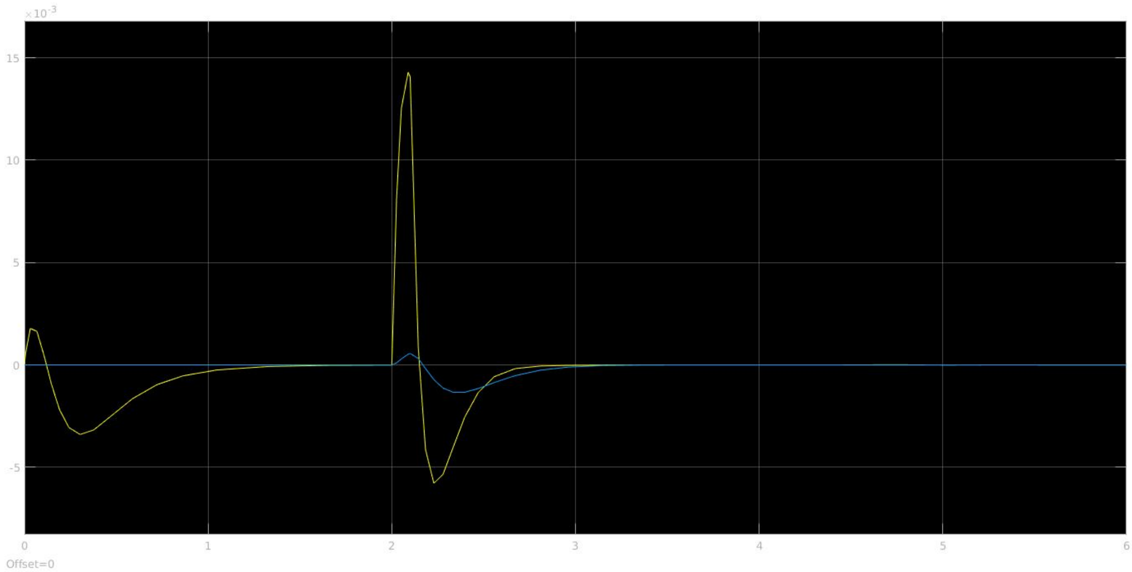



**Figure 2.6:** Body position while balancing

Following graph shows the bodys angular speed during the simulation.

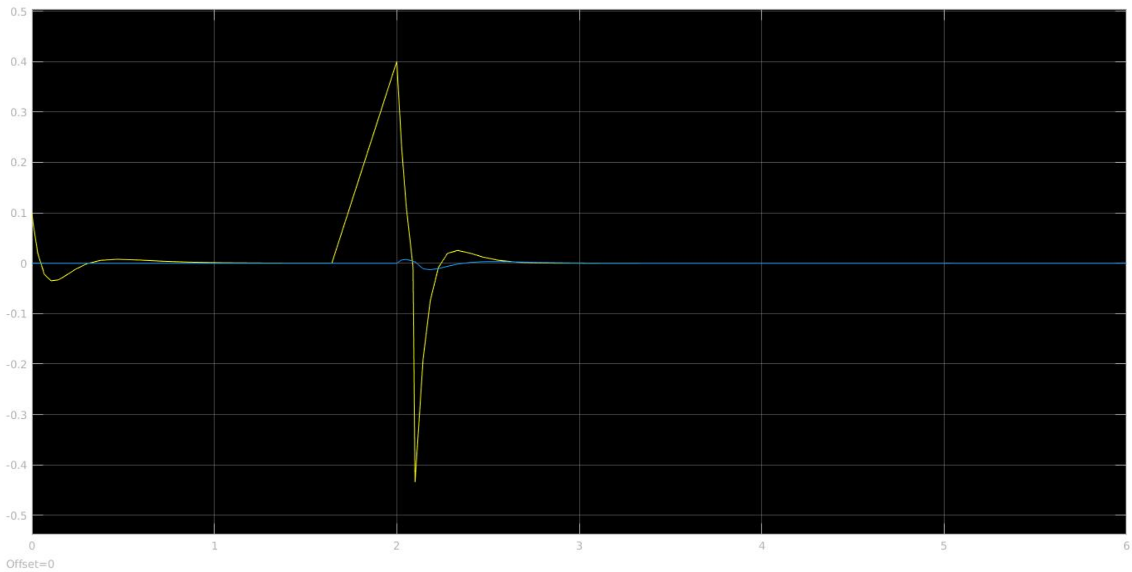



**Figure 2.7:** Body speed while balancing

Next one is the current input to the motor.

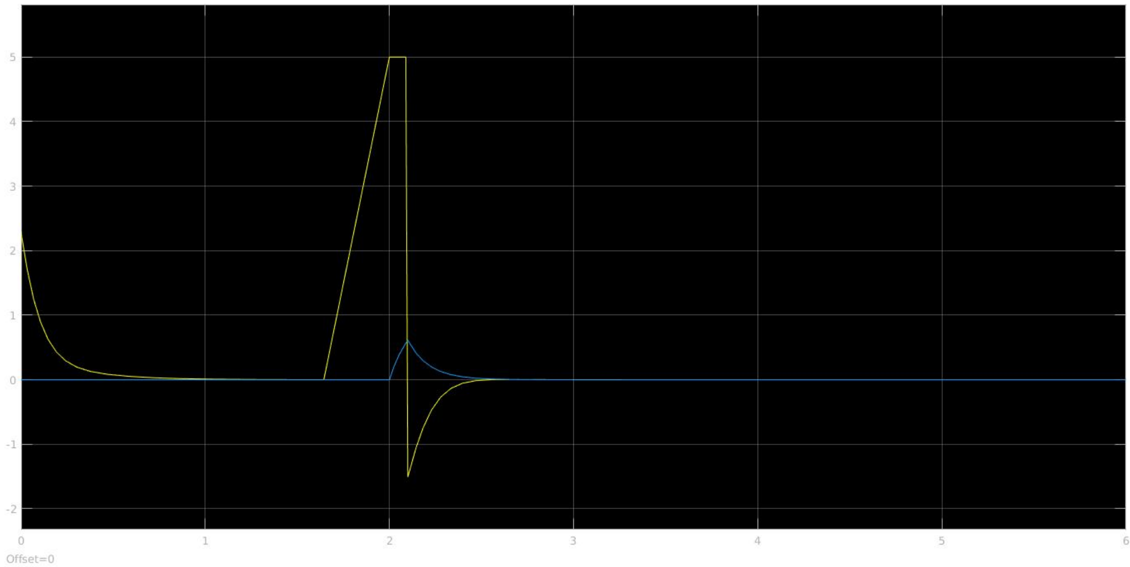



**Figure 2.8:** Current input while balancing

And the final graph is the angular speed of the reaction wheel.

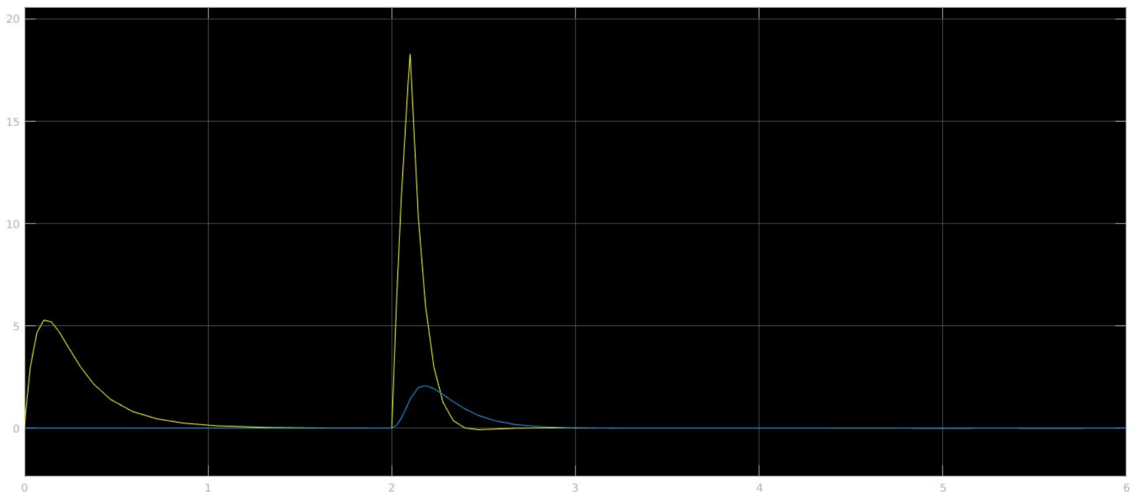



**Figure 2.9:** Reaction wheel angular speed while balancing

While simulating the balancing we tested through different amplitudes and found that the maximum disturbance that the system can balance is approximately 22 degrees. From that we can say that our system is stable and able to balance itself up to 22 degrees offset from the balanceing point. If the angle is bigger then the 22 degrees the system will tilt over because the motor is not strong enough to counteract the earths gravity.

## 2.7 Parameters identification

### 2.7.1 Mass centre

The simplest thing to determine from the parameters is the centre of mass what is crucial for calculating the torques of the system. To get the mass centre of the system we simply hang it freely from each of the corners. If the body is hanging from any of the corners you can evaluate the mass centre by drawing a vertical line from all of the hanging points and the cross section is the centre of mass.

From here we can measure the parameter $l_b$ what is the lenght from the centre of mass to the pivot point.

### 2.7.2 Motor friction coefficient

In this subsection we concentrate on measuring the friction coefficient of the DC motor we are using. We have come up with a experiment where we make the wheel rotate with a constant speed and then we measure the voltage on the motor and take the motor speed from the motor aswell.

$$T_m = K_m u \tag{2.28}$$

So if we look our experiment then we recon that the only force restricting the motor is friction. We must make a equation suitable for our experiment.

$$I_w \ddot{\Theta}_w(t) = K_m u(t) - C_w \dot{\Theta}_w \tag{2.29}$$

Where $I_w$ is the inertia of the wheel and $K_m$ is the torque constant of the brushless DC motor used and $u$ is the current input.

$$C_w = \frac{K_m u(t) - I_w \ddot{\Theta}_w(t)}{\dot{\Theta}_w(t)} \tag{2.30}$$

Here we can see that we have to take the reading of the current input and the wheel speed since we are spinning the wheel in constant speed we can have the acceleration as zero. The motor friction cofficient is measured on different speeds to get the most accurate readings. From the equation we can see if we spinn the wheel on constant speed we can lose the acceleration part and the formulae becomes:

$$C_w = \frac{K_m u(t)}{\dot{\Theta}_w(t)} \tag{2.31}$$

From here we can get the friction coefficient of the DC motor.

### 2.7.3 Friction coefficient of the body

Here we will look at the experiment what we are going to use to measure the friction coefficient of the whole body. First we are going to hang it upside down and the have the body in a certain angle and release it. From the dampening of the oscillation we can determine the friction coefficient of the body

$$(m_b + m_w)\ddot{\Theta}_b + C_b\dot{\Theta}_b + \frac{(m_b + m_w)g\Theta_b}{l_b} = 0 \tag{2.32}$$

From here we can get the friction coefficient for our system.

$$C_b = -\frac{(m_b + m_w)\ddot{\Theta}_b}{\dot{\Theta}_b} - \frac{(m_b + m_w)g\Theta_b}{\dot{\Theta}_b l_b} \tag{2.33}$$

In this experiment we must record the time trace of the $\theta$ so we have the position and the oscillation of the body. From the graph we can use the fitting of least squres method to get the coefficient for the body's friction.

## 2.8 Testing

Testing the system on the real plant has not been done by the time we are writing this report, because of the hardware needed for the device has just not arrived. Delays in the hardware delivery were mainly caused by the breaking down of the CNC bench on what the parts were supposed to be made.

# Chapter 3

# Discussion

## 3.1 Discussion

In this project we have looked at a cube that will jump up and balance itself. We have designed a nonlinear and linear controller for the system we have engineered. The project has been more challenging than expected. The first and the utmost problem we encountered was the hardware acquirement because the setup what we designed has been delayed more then 4 weeks at this point. To be more specific while completing the report in hand we still do not have the mechanical pieces for the system, but will hopefully have them by the presentation of the project.

In this paper you have mainly seen the theory behind the device. We now need to see if we completed any of our set aims.

- Build a simulation model for this specific system
  Done
- Build the real plant of this system
  Not done because of not having the hardware due to delays what were not inflicted by us.
- Determination of system parameters
  Theoretically achived but again not in practis because of harware delays
- Design a linear controller for the system
  Done
- Design a nonlinear controller for the system
  Done
- Impliment the designed controller on the real plant and evaluate the results
  Not done beacause lack of hardware

We must mention that the aims what were not achived will be hopefully completed by the time we present our project.

First we wanted to make simulation model for the system. As we can see from the simulink chapter we have designed a working model for that specific setup that we have described through out the paper in hand. By analyzing the graphs for the linear and nonlinear controller we can see that the linear controller designed seems to be working.

On the nonlinear version we can see nicely how the body gets out of the balancing position and then regains the upright position in about 1 second. In that case we must say that to achive the balancing point in that time is fairly quick if we take into concideration the complexity of the whole setup.

While designing the nonlinear controller we had difficulties at first making the state matrix $A$, because there were some mathematical errors concerning the nonlinear dynamic equation. Luckily the errors were corrected after consulting with a mathematician. Hereby we feel that we have to thank Päivo Simson from Tallinn Technical University, who helped us on crucial moments in the formulation of those equations.

Even that we can see that the linear version works in some degree we still have doubts about it. The doubts are mainly fuled by the very small fluxuation in the body displacement. As previously been acquainted with dynamical mechanics we could see that in real life that linear system would not work. Therefore we should be more critical towards ourselves and the work we do while taking on future projects of similar degree.

In the beginning of this project we concidered using PID controller. Thanks to our supervisors that idea was buried quikly due to the nature of the system. In the device at hand we have constantly changing factors and we quickly understood how the state space version is alot better. While designing the state space model we got very familiar with the method and will probably use that again when there are need to design a dynamic controller in the future.

The determination of the parameters was done theoretically. The suggested experiments for that will hopefully be done by the presentation of this project and we can then see if the theory what we have come up with holds it's ground.

# Chapter 4

# Conclusion

## 4.1 Conclusion and perspective

In this paper we have presented a cube shaped inverted pendulum prototype for one of the sides of the cube.

We have designed a possible solution for the cube shaped inverted pendulum control. The simulations what we have conducted during this project have proven that the control method used in the current designe works. For the control of the system we have implemented state space based control method.

In order to make the cube jump up we modelled physical model for the jump up sequence. In this section of the project we have made sure that it is possible to make the cube lift off the surface where it is laying on one of its sides. The lift off is made possible by storing energy on the momentum wheel and then rapidly bringing it to stop. Due to the enargy transfer from the wheel to the body we can make the system jump on to its corner. We have concluded that the speed necessary for the lift off to the point where the cube starts balancing is relativly small compared to the maximum speed of the motor proposed for the system.

The balancing simulations have shown us the maximum capabilities of the motor used to keep it balanced untill the certain angle of offset. Aswell we can see that the system reaction times are low what proves the efficiency of the proposed model.

Overall we can see that the system works and during this project we have learned alot about the modelling and controlling dynamic systems.

In future perspective of this project we see that it is good to build the whole cube. When the whole cube has been designed and built we see different uses for the system. For example reaction wheel based movement can be used for interplanetary exploration on data collecting robots. Because if the desired robot can move around different surfaces without having any external moving parts it is much more environment proof. It can be used in harsh conditions.

If the project would be carried on in upcoming semesters then we propose to have a bigger team to work on it because of the extensive possibilities of the future uses.

# Bibliography


[1] Igor Thommen Raffaello D'Andrea Mohanarajah Gajamohan, Michael Merz. *The Cubli: A Cube That Tan Jump Up and Balance*. Vilamoura, Algarve, Portugal, 2012.

[2] Daniela Rus John W. Romanishin, Kyle Gilpin. *M-Blocks: Momentum-driven, Magnetic Modular Robots*. Cambridge, Massachusetts.

[3] Youdan Kim Sungpil Kim. *Spin-Axis Stabilization of a Rigid Spacecraft Using Two Reaction Wheels*. Seoul, Republic of Korea, 2001.

[4] Swiss cubli takes the balance wheel to a new level. `http://www.ablogtowatch.com/swiss-cubli-takes-balance-wheel-new-level/`. Accessed: 2016-05-24.

[5] M. J. Clugston. *The Penguin Dictionary of Science*. London, England,3rd edition, 2009.

[6] State-space representation of lti systems. `http://web.mit.edu/2.14/www/Handouts/StateSpace.pdf`. Accessed: 2016-05-24.

[7] Spu 26 ref 6: Small angle approximation. `http://isites.harvard.edu/fs/docs/icb.topic1386948.files/Small%20angle%20approximation.pdf`. Accessed: 2016-05-20.

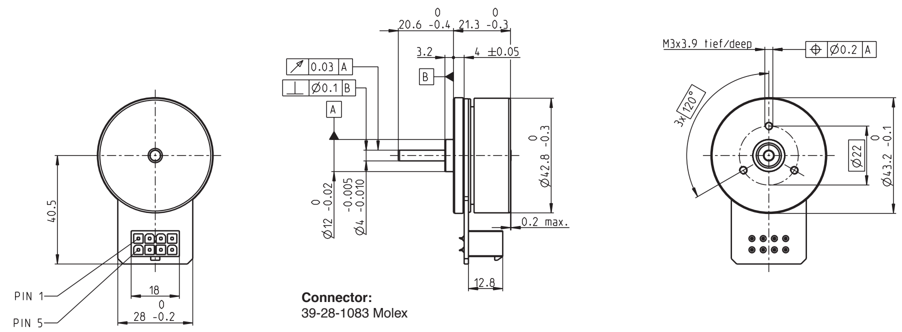
# Chapter 5

# Appendix

## 5.1 Datasheets

maxon flat motor

## EC 45 flat ∅42.8 mm, brushless, 50 Watt




**Connector:**
39-28-1083 Molex

**M 1:2**

- Stock program
- Standard program
- Special program (on request)

**Part Numbers**

| | | 339285 | 251601 | 339286 | 339287 |
|---|---|---|---|---|---|
| with Hall sensors | | 339285 | 251601 | 339286 | 339287 |
| **Motor Data** | | | | | |
| **Values at nominal voltage** | | | | | |
| 1 Nominal voltage | V | 18 | 24 | 24 | 36 |
| 2 No load speed | rpm | 6720 | 6710 | 4730 | 3360 |
| 3 No load current | mA | 247 | 185 | 106 | 42.3 |
| 4 Nominal speed | rpm | 5190 | 5240 | 3480 | 2360 |
| 5 Nominal torque (max. continuous torque) | mNm | 97.1 | 83.4 | 69.6 | 90.5 |
| 6 Nominal current (max. continuous current) | A | 3.52 | 2.33 | 1.41 | 0.828 |
| 7 Stall torque | mNm | 975 | 780 | 402 | 484 |
| 8 Stall current | A | 38.8 | 23.3 | 8.47 | 4.81 |
| 9 Max. efficiency | % | 85 | 83 | 79 | 82 |
| **Characteristics** | | | | | |
| 10 Terminal resistance phase to phase | Ω | 0.464 | 1.03 | 2.83 | 7.48 |
| 11 Terminal inductance phase to phase | mH | 0.322 | 0.572 | 1.15 | 5.15 |
| 12 Torque constant | mNm/A | 25.1 | 33.5 | 47.5 | 101 |
| 13 Speed constant | rpm/V | 380 | 285 | 201 | 95 |
| 14 Speed/torque gradient | rpm/mNm | 7.02 | 8.77 | 12 | 7.07 |
| 15 Mechanical time constant | ms | 9.92 | 12.4 | 17 | 10 |
| 16 Rotor inertia | $gcm^2$ | 135 | 135 | 135 | 135 |

**Specifications**

**Thermal data**

| | | |
|---|---|---|
| 17 | Thermal resistance housing-ambient | 4.53 K/W |
| 18 | Thermal resistance winding-housing | 4.75 K/W |
| 19 | Thermal time constant winding | 17.7 s |
| 20 | Thermal time constant motor | 227 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) | 53 N |
| | (static, shaft supported) | 1000 N |
| 28 | Max. radial load, 5 mm from flange | 20 N |

**Other specifications**

| | | |
|---|---|---|
| 29 | Number of pole pairs | 8 |
| 30 | Number of phases | 3 |
| 31 | Weight of motor | 110 g |

Values listed in the table are nominal.

**Connection**

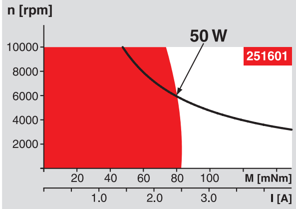
| | |
|---|---|
| Pin 1 | Hall sensor 1* |
| Pin 2 | Hall sensor 2* |
| Pin 3 | $V_{Hall}$ 4.5…18 VDC |
| Pin 4 | Motor winding 3 |
| Pin 5 | Hall sensor 3* |
| Pin 6 | GND |
| Pin 7 | Motor winding 1 |
| Pin 8 | Motor winding 2 |

*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**

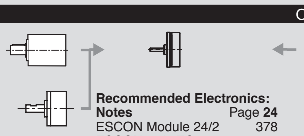
**Continuous operation**
In observation of above listed thermal resistance (lines 17 and 18) the maximum permissible winding temperature will be reached during continuous operation at 25°C ambient.
= Thermal limit.

**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 Module 24/2 | 378 |
| ESCON 36/3 EC | 379 |
| ESCON Mod. 50/4 EC-S | 379 |
| ESCON Module 50/5 | 379 |
| ESCON 50/5 | 380 |
| DEC Module 24/2 | 382 |
| DEC Module 50/5 | 382 |
| EPOS2 24/2 | 386 |
| 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)



April 2015 edition / subject to change

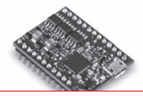
# 5.1.2 Motor driver



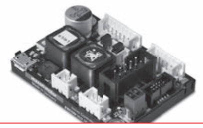
## ESCON Overview

The ESCON servo controllers are small-sized, powerful 4-quadrant PWM servo controller for the highly efficient control of permanent magnet-activated DC motors. The featured operating modes – speed control (closed loop), speed control (open loop), and current control – meet the highest requirements. The ESCON servo controllers are designed being commanded by an analog set value and features extensive analog and digital I/O functionality and are being configured via USB interface using the graphical user interface “ESCON Studio” for Windows PCs.

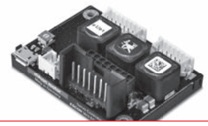
**NEW ESCON Module 24/2**
**466023**



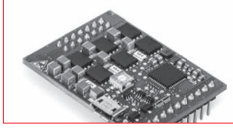
**ESCON 36/2 DC**
**403112**



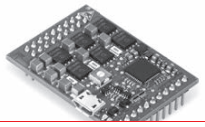
**ESCON 36/3 EC**
**414533**



**ESCON Module 50/4 EC-S**
**446925**

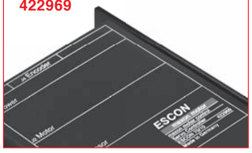


**ESCON Module 50/5**
**438725**



**ESCON 50/5**
**409510**

**ESCON 70/10**
**422969**



ESCON

Depending on the ESCON variant, the following **motor types** can be operated
- **DC motor:** Permanent-magnet DC motor
- **EC motor:** Brushless, electronically commutated permanent-magnet DC motor (BLDC) with and without Hall sensors.

Various **operating modes** allow an adaptable use in a wide range of drive systems
- **Current controller:** The current controller compares the actual motor current (torque) with the applied set value. In case of deviation, the motor current is dynamically readjusted.
- **Speed controller (closed loop):** The closed loop speed controller compares the actual speed signal with the applied set value. In case of deviation, the speed is dynamically readjusted.
- **Speed controller (open loop):** The open loop speed controller feeds the motor with a voltage proportional to the applied speed set value. Changes in load are compensated using the IxR methodology.

**Speed measurement** by
- **Digital incremental encoder:** The encoders deliver simple square signals for further processing. Their impulses are counted to determine the speed. Channels A and B are phase-shifted signals, which are being compared to determine the direction of rotation.
- **DC tacho:** The DC tacho delivers a speed-proportional analog voltage.
- **Available Hall sensors:** The Hall sensors deliver six different combinations of switching impulses per electrical turn which are counted to determine speed. They also deliver phase-shifted signals that are being compared to determine the direction of rotation.
- **Sensorless EC:** The speed is determined by the progression of the induced voltage. The electronics evaluates the zero crossing of the induced voltage (EMF).

To the numerous **inputs** and **outputs,** various functionalities can be assigned to.

**Set value** (speed or current), **current limitation,** as well as **offset** can be assigned as follows.
- **Analog value:** The value is defined by an analog voltage set via external or internal potentiometer.
- **PWM value:** The value is defined by fixed frequency and amplitude. The desired change is achieved by variation of the duty cycle of 10…90%.
- **RC Servo Value:** The value is set with a signal pulse with a duration of 1.0…2.0 ms.
- **Fixed value:** The value is defined by a fixed preset value.
- **2 fixed values:** Value 1 is defined by a fixed preset value 1. Value 2 is defined by a fixed preset value 2. A digital input is used to switch between the two preset values.

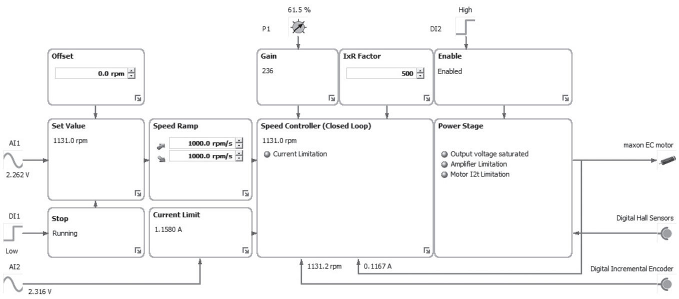
Various functionalities are available to **enable** the power stage.
- **Enable:** Enables or disables the power stage.
- **Enable & Direction:** Enables or disables the power stage and determines the motor shaft’s direction of rotation.
- **Enable CW:** Enables or disables the power stage in direction of rotation-dependent sense. The rotor can only turn clockwise (CW).
- **Enable CCW:** Enables or disables the power stage in direction of rotation-dependent sense. The rotor can only turn counterclockwise (CCW).
- **Enable CW & CCW:** Enables or disables the power stage in direction of rotation-dependent sense. The rotor can only turn in defined direction. The signals are interlocked against each other.

The **ramp function** permits controlled acceleration/deceleration of the motor shaft in both, open loop and closed loop speed controller mode.
- **Analog ramp:** The ramp is defined by a variable analog value.
- **Fixed ramp:** The ramp is defined by a fixed preset value.

**Stop:** The motor shaft decelerates with preset speed ramp until complete standstill.

**Ready:** The Ready signal can be used to transmit the operational status (respectively fault) to a superior control.

**Speed and Current Comparator:** The digital output is set depending on the actual value.
- **Limit:** The digital output is set as soon as the




ESCON Studio (Controller Monitor)

preset value is reached. It remains set as long as the value is exceeded.
- **Range:** The digital output is set as soon as the preset value range is reached. It remains set as long as the value remains in range.
- **Deviation:** The digital output is set as soon as the preset value deviation (based on the set value) is in range.

With the integrated **potentiometers** the additional following functions can be adjusted
- **Current Gain:** Adjustment of the current controller gain.
- **Speed Gain:** Adjustment of the speed controller gain.
- **IxR Factor:** The voltage drop caused by terminal resistance will be compensated in the range of [0…1000…2000].

Analog outputs allow monitoring of
- **Actual current:** Actually measured motor winding current.
- **Actual current averaged:** Actually measured motor winding current filtered by first order digital low-pass filter with a cut-off frequency of 5 Hz.
- **Actual speed:** Actually measured motor speed.
- **Actual speed averaged:** Actually measured motor speed filtered by 1st order digital low-pass filter with a cut-off frequency of 5 Hz.
- **Demand Current:** Demanded motor winding current.
- **Demand Speed:** Demanded motor speed.
- **Temperature Power Stage:** Actually measured power stage temperature.
- **Fixed value:** The output voltage is said fixed to the preset value.

**Easy startup**
Startup and parameterization are performed using the intuitive graphical user interface “ESCON Studio” with the help of simple to use, menu-guided wizards. The following wizards are available: Startup, Regulation Tuning, Firmware Update, Controller Monitor, Parameters, Data Recording, and Diagnostics.

**Protective equipment**
The servo controller has protective circuits against overcurrent, excess temperature, under- and overvoltage, against voltage transients, and against short-circuits in the motor cable. Furthermore it is equipped with protected digital inputs and outputs and an adjustable current limitation for protecting the motor and the load. The motor current and the actual speed of the motor shaft can be monitored by means of the analog output voltage.

**Comprehensive documentation**
Using the “Feature Comparison Chart”, the suitable ESCON servo controller can easily be determined. The “Hardware Reference” comprises the specifications of the hardware in detail. The documents “Firmware Version” and “Release Notes” describe changes and improvements of firmware and software. In addition, the graphical user interface “ESCON Studio” features a comprehensive online help.


| Software |
|---|
| Installation Program: ESCON Setup |
| Graphical User Interface: ESCON Studio |
| ✓ Startup Wizard |
| ✓ Regulation Tuning |
| ✓ Diagnostic |
| ✓ Firmware Update |
| ✓ Controller Monitor |
| ✓ Parameters |
| ✓ Data Recording |
| ✓ Online Help |
| Language: German, English, French, Italian, Spanish, Japanese, Chinese |
| Operating System: Windows 8, Windows 7, Windows XP SP3 |
| Communication interface: USB 2.0/3.0 (full speed) |

| Accessories ESCON* | M 24/2 | 36/2 DC | 36/3 EC | M 50/4 EC-S | M 50/5 | 50/5 | 70/10 |
|---|---|---|---|---|---|---|---|
| 404404 ESCON 36/2 DC Connector Set | – | ✓ | – | – | – | – | – |
| 425255 ESCON 36/3 EC Connector Set | – | – | ✓ | – | – | – | – |
| 403962 DC Motor Cable | – | ✓ | – | – | – | – | – |
| 403964 I/O Cable 7core (analog I/O’s) | – | ✓ | ✓ | – | – | – | – |
| 403965 I/O Cable 6core (digital I/O’s) | – | ✓ | ✓ | – | – | – | – |
| 275934 Encoder Cable | – | ✓ | – | – | – | ✓ | ✓ |
| 403957 Power Cable | – | ✓ | ✓ | – | – | – | – |
| 403968 USB Type A - micro B Cable | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 418719 Adapter BLACK FPC11poles | – | – | ✓ | – | – | – | – |
| 418723 Adapter BLUE FPC8poles | – | – | ✓ | – | – | – | – |
| 418721 Adapter GREEN FPC8poles | – | – | ✓ | – | – | – | – |
| 486400 ESCON Module 24/2 Motherboard | ✓ | – | – | – | – | – | – |
| 438779 ESCON Module Motherboard | – | – | – | – | ✓ | – | – |
| 450237 ESCON Module Motherboard Sensorless | – | – | – | ✓ | – | – | – |
| 409286 ESCON USB Stick | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

*not included in delivery

# **ESCON** Feature Comparison Chart

NEW

| | ESCON Module 24/2 | ESCON 36/2 DC |
|---|---|---|
| DC motors up to | 48 W | 72 W |
| EC motors up to | 48 W | – |
| **Sensors** | | |
| | Digital Incremental Encoder (2 channel with or without Line Driver) | Digital Incremental Encoder (2 channel with or without Line Driver) |
| | DC Tacho | DC Tacho |
| | Without sensor (DC motors) | Without sensor (DC motors) |
| | Digital Hall Sensors (EC motors) | – |
| **Operating Mode** | | |
| | Current controller (torque control), Speed controller (closed and open loop) | Current controller (torque control), Speed controller (closed and open loop) |
| **Electrical Data** | | |
| Nominal operating voltage $V_{CC}$ | 10 - 24 VDC | 10 - 36 VDC |
| Max. output voltage | 0.98 x $V_{CC}$ | 0.98 x $V_{CC}$ |
| Max. output current | 6 A (<4 s) | 4 A (<60 s) |
| Continuous output current | 2 A | 2 A |
| Pulse width modulation frequency | 53.6 kHz | 53.6 kHz |
| Sampling rate PI current controller | 53.6 kHz | 53.6 kHz |
| Sampling rate PI speed controller | 5.36 kHz | 5.36 kHz |
| Max. efficiency | 92% | 95% |
| Max. speed (DC) | limited by Max. speed (motor) and max. output voltage (controller) | limited by Max. speed (motor) and max. output voltage (controller) |
| Max. speed (EC; 1 pole pair) | 150 000 rpm | – |
| Built-in motor choke | – | 300 µH / 2 A |
| **Inputs/Outputs** | | |
| Hall sensor signals | H1, H2, H3 | – |
| Encoder signals | A, A\, B, B\ | A, A\, B, B\ |
| Max. encoder input frequency differential (single-ended) | 1 MHz (100 kHz) | 1 MHz (100 kHz) |
| Potentiometers | – | 1 |
| Digital inputs | 2 | 2 |
| Digital inputs/outputs | 2 | 2 |
| Analog inputs<br>Resolution, Range, Circuit | 2<br>12-bit, -10…+10 V, differential | 2<br>12-bit, -10…+10 V, differential |
| Analog outputs<br>Resolution, Range | 2<br>12-bit, -4…+4 V | 2<br>12-bit, -4…+4 V |
| Auxiliary voltage output | +5 VDC (IL ≤10 mA) | +5 VDC (IL ≤10 mA) |
| Hall sensor supply voltage | +5 VDC (IL ≤30 mA) | – |
| Encoder supply voltage | +5 VDC (IL ≤70 mA) | +5 VDC (IL ≤70 mA) |
| Status Indicators | Operation: green LED / Error: red LED | Operation: green LED / Error: red LED |
| **Environmental Conditions** | | |
| Temperature – Operation | -30…+60°C | -30…+45°C |
| Temperature – Extended range | +60…+80°C; Derating: -0.100 A/°C | +45…+81°C; Derating: -0.056 A/°C |
| Temperature – Storage | -40…+85°C | -40…+85°C |
| Humidity (condensation not permitted) | 20…80% | 20…80% |
| **Mechanical Data** | | |
| Weight | Approx. 7 g | Approx. 30 g |
| Dimensions (L x W x H) | 35.6 x 26.7 x 12.7 mm | 55.0 x 40.0 x 16.1 mm |
| Mounting holes | Plugable (socket headers with 2.54 mm pitch) | for screws M2.5 |
| **Part Numbers** | | |
| | **466023** ESCON Module 24/2 | **403112** ESCON 36/2 DC |
| | Order accessories separately, from page 398 | Order accessories separately, from page 398 |


April 2015 edition / subject to change

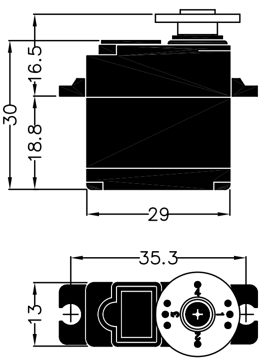
### 5.1.3 Servo motor for the breaking system

| REVISIONS | | | |
|---|---|---|---|
| SYM. | DATE | APPROVED BY | DESCRIPTION |
| △ | | | |

GENERAL SPECIFICATION OF
HSG-5084MG DIGITAL MICRO GYRO SERVO

1.TECHNICAL VALUE

| | | |
|---|---|---|
| CONTROL SYSTEM | :+PULSE WIDTH CONTROL 1500usec NEUTRAL 1100~1900usec | |
| OPERATING VOLTAGE RANGE | :4.8V ONLY | |
| OPERATING TEMPERATURE RANGE | :-20°C TO +60°C(-4°F TO +140°F) | |
| TEST VOLTAGE | :AT 4.8V | AT 6.0V |
| OPERATING SPEED | :0.07sec/60° AT NO LOAD | 0.05sec/60° AT NO LOAD |
| STALL TORQUE | :1.5kg.cm(20.83oz.in) | 1.9kg.cm(26.39oz.in) |
| STANDING TORQUE | :1.4kg.cm(19.44oz.in)/5° HOLDOUT | 1.7kg.cm(23.61oz.in)/5° HOLDOUT |
| IDLE CURRENT | :20mA @ 5V ON DATA SHEET | 20mA AT STOPPED |
| RUNNING CURRENT | :180mA @ 4.5V ON DATA SHEET | 360mA/60° AT NO LOAD RUNNING |
| STALL CURRENT | :1.87A @ 4.5V ON DATA SHEET | 2A |
| DEAD BAND WIDTH | :1us | 2usec |
| OPERATING TRAVEL | :35°/ 400us | |
| DIRECTION | :CLOCK WISE/PULSE TRAVELING 1100 TO 1900usec | |
| MOTOR TYPE | :CORBON BRUSH | |
| POTENTIOMETER TYPE | :2 SLIDER/DIRECT DRIVE | |
| AMPLIFIER TYPE | :DIGITAL AMPLIFIER & MOSFET DRIVER | |
| DIMENSIONS | :29x13x30mm(1.14x0.51x1.18in) | |
| WEIGHT | :21.7g(0.76oz) | |
| BALL BEARING | :SINGLE/MR106 | |
| GEAR MATERIAL | :1 METAL-PLASTIC & 4 METAL | |
| HORN GEAR SPLINE | :24 SEGMENTS/⌀5.76 | |
| SPLINED HORNS | :MICRO/M-I, M-O, M-X | |
| CONNECTOR WIRE LENGTH | :160mm(6.29in) | |
| CONNECTOR WIRE STRAND COUNTER | :20EA | |
| CONNECTOR WIRE GAUGE | :28AWG | |




2.FEATURES
HIGH PERFORMANCE DIGITAL AMPLIFIER
LIGHTWEIGHT METAL GEARS WITH BALL BEARING

3.APPLICATIONS
FOR GYRO

| PART NO. | PART NAME | PART SPECIFICATION | MATERIAL | FINISH | TREATMENT | QUANTITY |
|---|---|---|---|---|---|---|

| TOLERANCES NOT OTHERWISE SPECIFIED | | APPROVALS | DATE | DWG.SIZE A4 | HITEC RCD KOREA,INC. | | | |
|---|---|---|---|---|---|---|---|---|
| FRACTIONAL | | APPROVED BY | | SCALE 1/1 | TITLE GENERAL SPECIFICATION | | | |
| DECIMAL | | CHECKED BY | | | | | | |
| ANGLES | | | | | | | | |
| CONCENTRICTY | | DESIGNED BY H,RYU | MAR 21, 2007 | UNIT mm | MODEL NO. HSG-5084MG | DWG.NO. | REF.NO. | SHEET OF |

### 5.1.4 Matlab data file

```

clc
clear

motormas = 0.110 % kg
platemas = 0.115 % kg
servomass = 0,0217 % kg

Tm = 0.0834 % BLDC max continous torque
Km = 0.00335 % torque constant of the BLDC
g = 9.81 % m/s^2

l = 0.085 % m lenght to the center of the plate
rw = 0.06 % m reaction wheel radius
mw = 0.018 % m reaction wheel mass
b = 0.075 %m lenght to the mass centre of the body

mb = motormas+platemas+servomass % body mass

Iw = mw*(rw^2) %reaction wheel inertia


Ib = (mb*(2*l)^2)/3 %whole body inertia

%Tm = Km*u

maxthetaddot=Tm/Iw

%parameters what cannot be measured directly
Cb = 0.00102 %kg*m2 *s-1
Cw = 0.0005 %kg*m2 *s-1
lb = 0.075 %m
%since we were unable to aquire the hardware and the setup is similar to
%the ETH on we are going to use theyr values



% dx(t)/dt = Ax(t) + Bu(t)

% x:=(?b, d?b/dt,d?w/dt) % angular position of the body, angular speed of
% the body, angular speed ot the wheel


A = [0 1 0;
     ((((mb*lb)+(mw*l))*g)/(Ib+(mw*l^2))) (-Cb/(Ib+(mw*l^2))) (Cw/(Ib+mw*l^2));
     (-((mb*lb+mw*l)*g)/(Ib+(mw*l^2))) Cb/(Ib+(mw*l^2)) (-(Cw*(Ib+Iw+(mw*l^2)))/(Iw*(Ib+mw*(l^2))))) ]

B = [0;
    (-(Km)/(Ib+(mw*l^2)));
    ((Km*(Ib+Iw+mw*l^2))/(Iw*(Ib+mw*l^2))) ]
```

```
C = [1   0   0]

D = 0
```

model data script

### 5.1.5 Matlab state space model

```
ssmodel = ss(A,B,C,D)

sys_order = order(ssmodel)
determinant = det(ctrb(A,B))

Q = [0.005 0 0
     0 0.001 0
     0 0 0.001]%
R = 1;
Kc = lqr(A,B,Q,R)

Ac = [(A-B*Kc)]
Bc = [B]
Cc = [C]
Dc = [D]

sscontroled = ss(Ac,Bc,Cc,Dc)

controlled_poles=pole(sscontroled)
```

model data script