
Synchronised Cubli operation

It is possible to synchronise motion between two Cublis.

7th semester project



**AALBORG UNIVERSITY
WORKSHEETS**

Aalborg University
Control and Automation



Department of Automation and Control
Aalborg University
<http://www.aau.dk/>

AALBORG UNIVERSITY STUDENT REPORT

Title:

Synchronised Cubli operation

Theme:

Control and Automation technologies

Project Period:

Fall 2020

Project Group:

CA7-733

Participant(s):

Albert Smed Sandager
Christian Mølgaard Nielsen
Ivelin Krasimirov Penchev
Jaan Moos
Kristian Helmer Kjær Larsen
Yuheng Wang

Supervisor(s):

Jan Dimon Bendtsen
John-Josef Leth

Copies:

Page Numbers: 127

Date of Completion:

February 3, 2021

The content of this report is freely available, but publication (with reference) may only be pursued due to agreement with the author.

Abstract:

These worksheets is a continuation of the work done by previous AAU student projects and verifies the possibility of synchronising start up and shut down between two Cublis. A tuned linear-quadratic-regulator (LQR)-based controller for balancing is presented, along with a start up speed controller to facilitate the synchronisation. To reduce the impact with the ground when a Cubli is turned off, two shut down procedures are considered: one braking a spinning wheel before impact, and a second applying torque to the reaction wheel while falling. The 802.11/b/g/n Wi-Fi protocol is used for communication. The network experiences periodical communication dropouts and delay variations. The communication architecture is designed as a state machine to facilitate the network implementation. Networked control is also tested on the systems, but communication delays cause instability. However, for synchronising motion between two Cublis, it has been observed that the delay is insignificant, but the synchronisation is affected by the communication dropouts. Test results verify that the two Cublis can synchronise start up without any notable delay, as well as the beginning of the shut down procedures.

Contents

1	Introduction	2
1.1	Original ETH designs	3
1.1.1	The 2D Cubli	3
1.1.2	The 3D Cubli	4
1.2	Cublis at Aalborg University	4
2	Model of the system	7
2.1	Kinetic energy	9
2.2	Potential energy	10
2.3	Non-conservative forces	11
2.4	Lagrange's equation	12
3	Parameter estimations	15
3.1	Reaction wheel parameters	15
3.1.1	Mass and inertia	15
3.1.2	Friction	15
3.1.3	Distance to the center of mass	17
3.2	Frame parameters	17
3.2.1	Distance to the center of mass	17
3.2.2	Mass	17
3.2.3	Inertia and friction	18
3.3	Conclusion	21
4	State space representation	23
4.1	Conversion to a state space model	23
4.2	Analysis of the state space model	25
4.2.1	Pole location	26
4.2.2	Zero location	26
4.2.3	Controllability	26
4.2.4	Observability	27
4.3	The condition numbers and different approaches	27
4.3.1	Alternative approach for pole and zero location	29
4.3.2	Alternative approach for controllability and observability	30

5 Controller design	31
5.1 Linear-Quadratic Regulator	31
5.1.1 Optimization problem definition	32
5.1.2 Controller scheme	32
5.1.3 Choosing the Q and R matrices	33
5.1.4 Optimal gain definition	33
5.1.5 Optimal gain computation	34
5.1.6 LQR example	34
5.2 Linear Quadratic Regulator for the Cubli	36
5.2.1 Initial guess	37
5.2.2 Tuning	38
5.3 Reference angle adjustment	43
5.4 Start up	44
5.4.1 Start up maneuver by [5]	44
5.4.2 New start up maneuver	44
5.4.3 Start up model	45
5.5 Shut down	47
5.5.1 Brake before impact	47
5.5.2 Decelerating by applying torque	48
6 Filters	51
6.1 Complementary filter	51
6.2 Kalman filter	52
6.2.1 Kalman filter theory	52
6.2.2 Kalman filter design	53
6.2.3 Kalman filter simulation	54
6.2.4 Kalman filter implementation	55
7 Communication	57
7.1 Analysis of Cubli communication	57
7.2 Technology choice	60
7.3 XBEE implementation	61
7.3.1 Transmitting Arduino function	62
7.3.2 Receiving Arduino function	63
7.3.3 Buffer implementation	64
7.4 Communication implementation strategy	66
7.4.1 State machine motivation	66
7.4.2 State machine implementation	67
7.5 Testing Synchronised Motion	70
7.6 Networked Control	71
7.7 Consideration of other communication modules	73
7.7.1 HM-10 BLE modules	73

7.7.2 nRF24L01+ modules	73
Glossary	75
Bibliography	76
Appendices	78
A Hardware description	79
B Estimation of friction coefficient for the reaction wheel	85
C Center of mass of the frame	88
D Estimation of friction coefficient and inertia of the frame	90
E LQR controller tuning	92
F Required velocities for the start up procedure using speed controller	94
G Shut down procedures	96
H Synchronised Start Up	101
I Estimation of delay between input and output of a Cubli	103
J Requirement for synchronisation	105
K Configuration of XBEE S6B communication modules	107
L Communication delay mean and standard deviation.	108
M Communication bandwidth	111
N Networked control	114
O Kalman filter data gathering	117
P Review from group 730	119
Q Review from group 731	125
R Review from group 732	126

1 Introduction

The following worksheets concern the work on Aalborg University's 2D Cubli, a reaction wheel-based inverted pendulum. In addition to these worksheets, a conference paper for the 2020 SEventh seMester Conference (SEMCON) at Aalborg University is created, which can be found at: https://github.com/KristianHLarsen/AAU-Cubli-2D-2020/blob/main/Documentation/CA7_Paper.pdf

The 2D Cubli is a simplified version of the 3D Cubli, a self-balancing cube. Sketches of these can be seen on Figure 1.1.

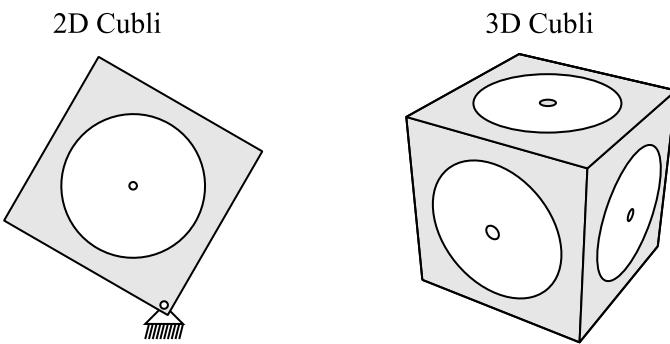


Figure 1.1: Sketch of the 2D and 3D Cubli.

On Figure 1.2 the current version of the Aalborg University 2D Cubli can be seen.

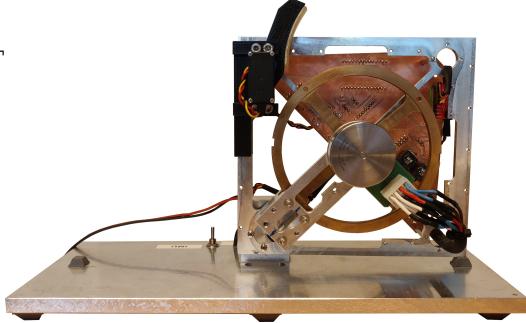
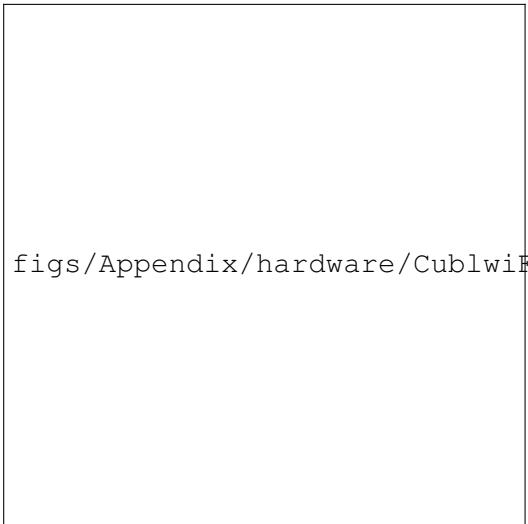


Figure 1.2: Front and back view of the current Cubli used in this project.

1.1 Original ETH designs

The Cubli is based on the now completed work done at ETH Zürich (Eidgenössische Technische Hochschule Zürich) [1, 2, 3, 4].

The name Cubli comes from the Swiss German diminutive for cube, referring to the shape of the 3D Cubli.

The design of the 3D Cubli started out with the question:

"How to build a 15 cm sided cube that can jump up and balance on its corner, using off-the-shelf motors, batteries, and electronic components?" [1, p. 1]

1.1.1 The 2D Cubli

To examine the feasibility of the project, a simplified prototype was developed [1]. This prototype was a 2D Cubli, having one reaction wheel controlled by a DC motor, and allowing rotational motion around a fixed axis at one of its corners. A servomotor-controlled brake allows for the necessary force to make the Cubli jump up from a resting position on its side to its balancing point with its vertex.

A simple model for the jump up was derived to allow for approximation of the necessary speed for a jump up.

A state space model for the 2D Cubli balancing at its vertex was developed and parameter estimation was undertaken.

This then allowed a LQR based feedback controller using the torque from the motor to be developed, implemented and tested.

1.1.2 The 3D Cubli

Following the prototype 2D Cubli, the full three dimensional Cubli was developed as described in [2]. This Cubli is a self-contained unit, without need for external power supply or support. It uses three motor controlled reaction wheels placed perpendicular to each other to allow movement of the cube.

Its jump up procedure consists of two phases: first it has to jump up so it balances on an edge, and then jump up so it balances on a vertex.

The system dynamics were modelled using Kane's equations, and the system parameters were estimated [2].

Multiple control approaches were developed. A linear feedback controlled from a LQR approach[2], as well as multiple different non-linear controllers based on Lyapunov equations were designed [3, 4]. All these controllers were implemented and tested. Control schemes for better jump up maneuvers and controlled falling was also designed.

1.2 Cublis at Aalborg University

Semester projects concerning Cubli-based inverted pendulums have been going on at the section of Automation and Control at Aalborg University since 2014. The main body of work has been concerning a 2D Cubli, but a 3D Cubli is also currently being developed.

The 2D Cubli, from now on referred to as just Cubli, has gone through multiple design iterations.

The Cubli currently is capable of both using a potentiometer at the rotational axis of the frame, as well as an IMU attached to the frame, for measuring the angle of the frame.

The last semester project concerning the Cubli completed the following tasks:

1. A standard DC-motor model was presented.
2. A linear model for the system was developed using Newtonian mechanics in accordance with the model from [1].
3. Parameters for this model was estimated through direct measurements and using the MATLAB toolkit SENSTOOLS.
4. A state space model for the system was developed from the model of the system's dynamics.
5. Two approaches for designing a full state feedback controller were attempted. One using pole placement, and one using LQR. The LQR based controller was tested and tuned.

6. A complementary filter for the IMU was developed.
7. A description of the network communication between different IMUs was completed.
8. An Arduino program was developed and implemented, capable of starting up the Cubli and balancing it on its side.

The following worksheets are concerned with the further development of the Cubli as well as understanding of the work done on previous semester projects. A description of the hardware of the Cubli can be found in Appendix A.

2 Model of the system

Since linear continuous control theory is used to design a controller for the system, a linear model of the system is required. The model presented in this chapter is in accordance with models from [1], as well as from previous work done on the Cubli [5] [6]. However, a different approach to the derivation of the model has been taken.

A simplified sketch of the system with the relevant variables can be found in Figure 2.1.

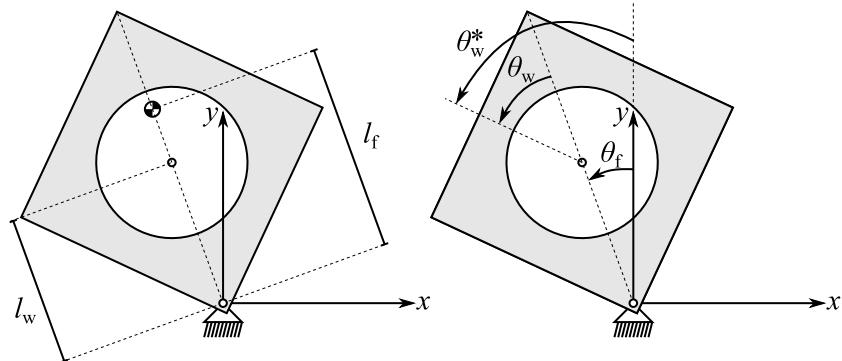


Figure 2.1: A sketch of the system illustrating the angles and the two lengths l_f and l_w . The CoM of the frame is denoted by the symbol.

Additionally a description of all the variables and parameters of the system can be found in Table 2.1.

Table 2.1: System variables and parameters

Symbol	Description	Unit
θ_f	The angle of the frame with respect to the vertical axis.	rad
m_f	The mass of the frame.	kg
l_f	The distance from the CoM of the frame to the origin of the coordinate system.	m
J_f	The moment of inertia of the frame.	$\text{kg} \cdot \text{m}^2$
b_f	The viscous damping coefficient of the bearing for the frame.	$\text{N} \cdot \text{m} \cdot \text{s} \cdot \text{rad}^{-1}$
θ_w	The angle of the reaction wheel with respect to the frame.	rad
θ_w^*	The angle of the reaction wheel with respect to vertical.	rad
m_w	The mass of the reaction wheel.	kg
l_w	The distance from the CoM of the reaction wheel to the origin of the coordinate system.	m
J_w	The moment of inertia of the reaction wheel.	$\text{kg} \cdot \text{m}^2$
b_w	The viscous damping coefficient of the bearing between the frame and the wheel.	$\text{N} \cdot \text{m} \cdot \text{s} \cdot \text{rad}^{-1}$
g	Gravitational acceleration.	$\text{m} \cdot \text{s}^{-2}$

The model is derived using Lagrangian mechanics [7], and thus the system is described using generalised coordinates. The system has two degrees of freedom and therefore two generalised coordinates are needed. For this, the angle of the frame and the angle of the reaction wheel, both with respect to the vertical axis, have been chosen. Thus:

$$\mathbf{q} = \begin{bmatrix} \theta_f \\ \theta_w^* \end{bmatrix} \quad (2.1)$$

where:

- \mathbf{q} : Vector containing the generalised coordinates.
- θ_f : The angle of the frame with respect to the vertical axis. [rad]
- θ_w^* : The angle with respect to the vertical axis, Note the * denotes it is with respect to the vertical axis. θ_w refers to the angle with respect to the frame. [rad]

The Lagrangian of the system is defined as:

$$L \equiv T - V \quad (2.2)$$

where:

L	: The Lagrangian of the system.	[J]
T	: The total kinetic energy of the system.	[J]
V	: The total potential energy of the system.	[J]

First, the kinetic energy of the system is found.

2.1 Kinetic energy

For a rigid body, the kinetic energy can be described as:

$$T = T_{\text{com}} + T_{\text{rot}} = \frac{1}{2} \cdot m \cdot v^2 + \frac{1}{2} \cdot J \cdot \dot{\theta}^2 \quad (2.3)$$

where:

T_{com}	: The kinetic energy from the movement of the CoM.	[J]
T_{rot}	: The rotational kinetic energy.	[J]
m	: The mass of the body.	[kg]
v	: The velocity of the CoM of the body.	[m · s ⁻¹]
J	: The moment of inertia of the body.	[kg · m ²]
$\dot{\theta}$: The angular velocity of the body.	[rad · s ⁻¹]

The kinetic energy of the system can be divided into three parts. The kinetic energy due to the rotation of the frame, the movement of the CoM of the reaction wheel, and the rotation of the reaction wheel.

$$T = T_{\text{rot,f}} + T_{\text{com,w}} + T_{\text{rot,w}} \quad (2.4)$$

where:

$T_{\text{rot,f}}$: The rotational kinetic energy of the frame.	[J]
$T_{\text{com,w}}$: The kinetic energy from the movement of the CoM of the reaction wheel.	[J]
$T_{\text{rot,w}}$: The rotational kinetic energy of the the reaction wheel.	[J]

The kinetic energy of the system can then be described using the parameters and variables of the system:

$$T = \frac{1}{2} \cdot J_f \cdot \dot{\theta}_f^2 + \frac{1}{2} \cdot m_w \cdot (\dot{x}_w^2 + \dot{y}_w^2) + \frac{1}{2} \cdot J_w \cdot \dot{\theta}_w^{*2} \quad (2.5)$$

where:

J_f	: The moment of inertia of the frame.	$[\text{kg} \cdot \text{m}^2]$
m_w	: The mass of the reaction wheel.	$[\text{kg}]$
x_w	: The horizontal position of the CoM of the reaction wheel.	$[\text{m}]$
y_w	: The vertical position of the CoM of the reaction wheel.	$[\text{m}]$
J_w	: The moment of inertia of the reaction wheel.	$[\text{kg} \cdot \text{m}^2]$

The translational kinetic energy of the reaction wheel can be simplified using the Pythagorean trigonometric identity as follows:

$$\dot{x}_w^2 + \dot{y}_w^2 = (l_w \cdot \cos(\theta_f) \cdot \dot{\theta}_f)^2 + (-l_w \cdot \sin(\theta_f) \cdot \dot{\theta}_f)^2 = l_w^2 \cdot \dot{\theta}_f^2 \quad (2.6)$$

where:

l_w	: The distance from the CoM of the reaction wheel to the origin of the coordinate system	$[\text{m}]$
-------	--	--------------

Thus, Equation 2.5 simplifies to:

$$T = \frac{1}{2} \cdot J_f \cdot \dot{\theta}_f^2 + \frac{1}{2} \cdot m_w \cdot l_w^2 \cdot \dot{\theta}_f^2 + \frac{1}{2} \cdot J_w \cdot \dot{\theta}_w^{*2} \quad (2.7)$$

Next, the potential energy of the system is considered.

2.2 Potential energy

The only conservative force acting on the system is the gravitational pull on frame and the wheel. Potential energy due to gravity can be modelled as:

$$V = g \cdot m \cdot y \quad (2.8)$$

where:

g	: Gravitational acceleration.	$[\text{m} \cdot \text{s}^{-2}]$
y	: The height.	$[\text{m}]$

Using trigonometric relations, the potential energy of the system is found to be:

$$V = g \cdot (m_w \cdot l_w + m_f \cdot l_f) \cdot \cos(\theta_f) \quad (2.9)$$

where:

m_f	: Mass of the frame.	$[\text{kg}]$
l_f	: The distance from the CoM of the frame to the origin of the coordinate system.	$[\text{m}]$

Thus, the Lagrangian is found for the system using Equation 2.2, 2.7, and 2.9:

$$L = \frac{1}{2} \cdot J_f \cdot \dot{\theta}_f^2 + \frac{1}{2} \cdot m_w \cdot l_w^2 \cdot \dot{\theta}_f^2 + \frac{1}{2} \cdot J_w \cdot \dot{\theta}_w^{*2} - g \cdot (m_w \cdot l_w + m_f \cdot l_f) \cdot \cos(\theta_f) \quad (2.10)$$

2.3 Non-conservative forces

The non-conservative forces acting on the system are divided into two groups:

- Friction in the bearings.
- Torque from the motor.

Both these non-conservative forces can be included in Lagrange mechanics by deriving the equations from d'Alembert's principle instead of Hamilton's principle. This derivation is not done in these worksheets.

First, the friction in the bearings is considered. Since linear continuous control theory is used, a linear model is needed. Thus, the friction in the bearings is modelled using Rayleigh's dissipation function [8]:

$$D = \frac{1}{2} \cdot \dot{\boldsymbol{q}}^T \mathbf{c} \dot{\boldsymbol{q}} \quad (2.11)$$

where:

- D : Rayleigh's dissipation function.
 \mathbf{c} : The damping matrix.

The frictional force from the bearing of the frame is dependent on the angular velocity of the frame $\dot{\theta}_f$. The bearing for the wheel on the other hand, is dependent on the angular velocity of the wheel with respect to the frame, i.e. $\dot{\theta}_w^* - \dot{\theta}_f$. Thus the damping matrix becomes:

$$\mathbf{c} = \begin{bmatrix} b_f + b_w & -b_w \\ -b_w & b_w \end{bmatrix} \quad (2.12)$$

where:

- b_f : The viscous damping coefficient of the bearing for the frame. $[N \cdot m \cdot s \cdot rad^{-1}]$
 b_w : The viscous damping coefficient of the bearing between the frame and the wheel. $[N \cdot m \cdot s \cdot rad^{-1}]$

Rayleigh's dissipation function for the system becomes:

$$D = \frac{1}{2} \cdot b_f \cdot \dot{\theta}_f^2 + \frac{1}{2} \cdot b_w \cdot (\dot{\theta}_w^* - \dot{\theta}_f)^2 \quad (2.13)$$

Next, the torque from the motor is considered. It can be modelled as two generalised forces action on the frame and wheel. The wheel experiences a positive torque, while the frame experiences an equal and opposite torque as explained by Newton's third law. Thus, the generalised forces become:

$$\mathbf{Q} = \begin{bmatrix} -\tau_m \\ \tau_m \end{bmatrix} \quad (2.14)$$

where:

- \mathbf{Q} : Vector containing the generalised forces.
 τ_m : The torque from the motor. [N m]

2.4 Lagrange's equation

The system dynamics can now be presented as two differential equations through Lagrange's equation on the form:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\mathbf{q}}} \right) - \frac{\partial L}{\partial \mathbf{q}} = \mathbf{Q} - \frac{\partial D}{\partial \dot{\mathbf{q}}} \quad (2.15)$$

where:

- t : Time. [s]

Inserting the equations for L , \mathbf{q} , D , and \mathbf{Q} yields the following:

$$\begin{aligned} & \left[J_f \cdot \ddot{\theta}_f + m_w \cdot l_w^2 \cdot \ddot{\theta}_f - g \cdot (m_w \cdot l_w + m_f \cdot l_f) \cdot \sin(\theta_f) \right] \\ & \quad J_w \cdot \ddot{\theta}_w^* \\ &= \left[-\tau_m - (b_f \cdot \dot{\theta}_f - b_w \cdot (\dot{\theta}_w^* - \dot{\theta}_f)) \right] \\ & \quad \tau_m - b_w \cdot (\dot{\theta}_w^* - \dot{\theta}_f) \end{aligned} \quad (2.16)$$

Next, since the wheel's position is measured with respect to the motor connected to the frame, a coordinate transformation is needed:

$$\begin{aligned} \theta_w &= \theta_w^* - \theta_f \\ \dot{\theta}_w &= \dot{\theta}_w^* - \dot{\theta}_f \\ \ddot{\theta}_w^* &= \ddot{\theta}_w + \ddot{\theta}_f \end{aligned} \quad (2.17)$$

where:

- θ_w : The angle of the wheel with respect to the frame. [rad]

Next, Equation 2.16 is combined with Equation 2.17:

$$\begin{aligned} & \left[J_f \cdot \ddot{\theta}_f + m_w \cdot l_w^2 \cdot \ddot{\theta}_f - g \cdot (m_w \cdot l_w + m_f \cdot l_f) \cdot \sin(\theta_f) \right] \\ & \quad \frac{J_w \cdot (\ddot{\theta}_w + \ddot{\theta}_f)}{= \left[-\tau_m - (b_f \cdot \dot{\theta}_f - b_w \cdot \dot{\theta}_w) \right]} \end{aligned} \quad (2.18)$$

Rewriting with respect to the accelerations:

$$\begin{aligned} \ddot{\theta}_f &= \frac{g \cdot (m_w \cdot l_w + m_f \cdot l_f)}{J_f + m_w \cdot l_w^2} \cdot \sin(\theta_f) + \frac{-b_f}{J_f + m_w \cdot l_w^2} \cdot \dot{\theta}_f + \frac{b_w}{J_f + m_w \cdot l_w^2} \cdot \dot{\theta}_w \\ &+ \frac{-1}{J_f + m_w \cdot l_w^2} \cdot \tau_m \end{aligned} \quad (2.19)$$

$$\ddot{\theta}_w = \frac{-b_w}{J_w} \cdot \dot{\theta}_w + \frac{1}{J_w} \cdot \tau_m - \ddot{\theta}_f$$

The $\sin(\theta_f)$ is approximated with θ_f to linearise the system around the balancing point where $\theta_f = 0$.

$$\begin{aligned} \ddot{\theta}_f &= \frac{g \cdot (m_w \cdot l_w + m_f \cdot l_f)}{J_f + m_w \cdot l_w^2} \cdot \theta_f + \frac{-b_f}{J_f + m_w \cdot l_w^2} \cdot \dot{\theta}_f + \frac{b_w}{J_f + m_w \cdot l_w^2} \cdot \dot{\theta}_w \\ &+ \frac{-1}{J_f + m_w \cdot l_w^2} \cdot \tau_m \end{aligned} \quad (2.20)$$

$$\ddot{\theta}_w = \frac{-b_w}{J_w} \cdot \dot{\theta}_w + \frac{1}{J_w} \cdot \tau_m - \ddot{\theta}_f$$

And inserting the equation for $\ddot{\theta}_f$ into the second equation:

$$\begin{aligned} \ddot{\theta}_f &= \frac{g \cdot (m_w \cdot l_w + m_f \cdot l_f)}{J_f + m_w \cdot l_w^2} \cdot \theta_f + \frac{-b_f}{J_f + m_w \cdot l_w^2} \cdot \dot{\theta}_f + \frac{b_w}{J_f + m_w \cdot l_w^2} \cdot \dot{\theta}_w \\ &+ \frac{-1}{J_f + m_w \cdot l_w^2} \cdot \tau_m \\ \ddot{\theta}_w &= \frac{-b_w}{J_w} \cdot \dot{\theta}_w + \frac{1}{J_w} \cdot \tau_m \\ &- \left(\frac{g \cdot (m_w \cdot l_w + m_f \cdot l_f)}{J_f + m_w \cdot l_w^2} \cdot \theta_f + \frac{-b_f}{J_f + m_w \cdot l_w^2} \cdot \dot{\theta}_f + \frac{b_w}{J_f + m_w \cdot l_w^2} \cdot \dot{\theta}_w + \frac{-1}{J_f + m_w \cdot l_w^2} \cdot \tau_m \right) \end{aligned} \quad (2.21)$$

Simplifying yields:

$$\begin{aligned}
 \ddot{\theta}_f &= \frac{g \cdot (m_w \cdot l_w + m_f \cdot l_f)}{J_f + m_w \cdot l_w^2} \cdot \theta_f + \frac{-b_f}{J_f + m_w \cdot l_w^2} \cdot \dot{\theta}_f \\
 &\quad + \frac{b_w}{J_f + m_w \cdot l_w^2} \cdot \dot{\theta}_w + \frac{-1}{J_f + m_w \cdot l_w^2} \cdot \tau_m \\
 \ddot{\theta}_w &= \frac{-g \cdot (m_w \cdot l_w + m_f \cdot l_f)}{J_f + m_w \cdot l_w^2} \cdot \theta_f + \frac{b_f}{J_f + m_w \cdot l_w^2} \cdot \dot{\theta}_f \\
 &\quad + \frac{-b_w \cdot (J_w + J_f + m_w \cdot l_w^2)}{J_w \cdot (J_f + m_w \cdot l_w^2)} \cdot \dot{\theta}_w + \frac{J_w + J_f + m_w \cdot l_w^2}{J_w \cdot (J_f + m_w \cdot l_w^2)} \cdot \tau_m
 \end{aligned} \tag{2.22}$$

3 Parameter estimations

In this chapter the parameters from the model derived in chapter 2 are estimated through various methods as described in the following sections. Most of these methods were also used by [1], [5], and [6].

The parameters are parted into two groups: those concerning the reaction wheel in section 3.1 and those concerning the frame in section 3.2.

3.1 Reaction wheel parameters

For the wheel, the following parameters need to be determined: J_w , m_w , b_w , and l_w .

3.1.1 Mass and inertia

To avoid unbalancing the wheel by detaching it from the frame, the mass and inertia of the wheel was not determined. Instead, the results found by [5] are used:

$$\begin{aligned} m_w &= 0.220 \text{ kg} \\ J_w &= 0.699 \cdot 10^{-3} \text{ kg} \cdot \text{m}^2 \end{aligned} \tag{3.1}$$

3.1.2 Friction

To find the viscous friction coefficient for the reaction wheel, the following equation is used [9, p. 73]:

$$J_w \cdot \ddot{\theta}_w = \tau_m - b_w \cdot \omega_w \tag{3.2}$$

Allowing the motor to achieve a steady state velocity by applying a constant armature voltage, the acceleration becomes 0, yielding:

$$0 = \tau_m - b_m \cdot \omega_m \tag{3.3}$$

Isolating the viscous friction coefficient:

$$b_m = \frac{\tau_m}{\omega_m} \tag{3.4}$$

A number of experiments are conducted where the velocity has been allowed to settle at different steady state values and then measuring the armature current. The motor torque can be calculated from the armature current via:

$$\tau_m = K_t \cdot i_a \quad (3.5)$$

where:

K_t	: Motor torque constant	$[N \cdot m \cdot A^{-1}]$
i_a	: Armature current	$[A]$

A more detailed report of the test can be seen in Appendix B.

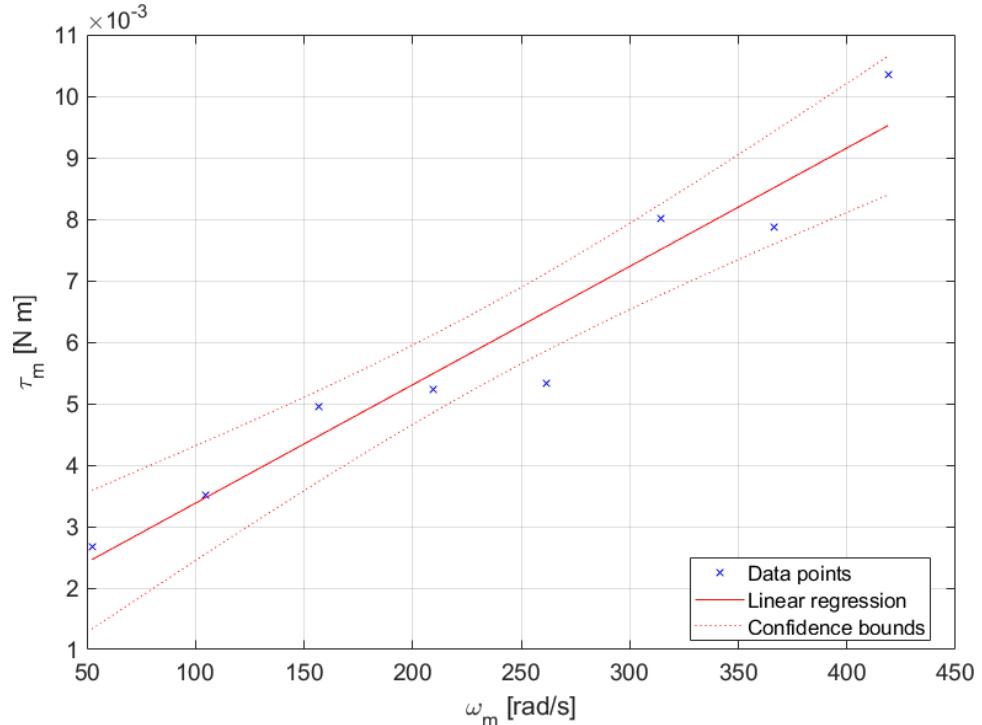


Figure 3.1: Linear regression of the motor torque and the angular velocity of the reaction wheel.

A linear regression for all the data points have then been conducted as seen on Figure 3.1. The slope of this regression then equals the viscous friction coefficient as seen in Equation 3.4. The friction b_m was calculated to be $1.93 \cdot 10^{-5} N \cdot m \cdot s \cdot rad^{-1}$.

The non-zero intercept is due to Coulomb friction. However, it is ignored for the purpose of maintaining a linear model.

3.1.3 Distance to the center of mass

The distance to the center of mass for the reaction wheel from the center of the bearing of the frame is calculated using the Pythagorean theorem, see Figure 3.2.

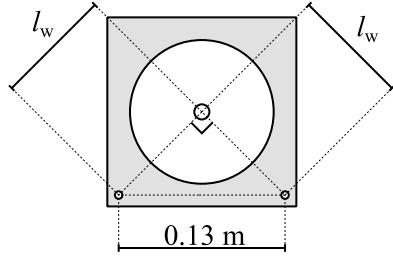


Figure 3.2: Sketch of the geometry for the calculating the distance to center of mass for the reaction wheel.

Solving the Pythagorean theorem for l_w :

$$(0.13 \text{ m})^2 = l_w^2 + l_w^2$$

$$\Downarrow$$

$$l_w = 0.092 \text{ m} \quad (3.6)$$

3.2 Frame parameters

Next, the different parameters of the frame are estimated, these being: l_f , m_f , J_w , and b_w .

3.2.1 Distance to the center of mass

The distance from the center of mass of the frame to the origin was estimated by hanging the Cubli from two different corners of the frame and then measuring the angle with respect to vertical. Then by using the distance between the two corners, 0.130 m, and trigonometry the distance was estimated to be 0.0958 m. A more detailed description of the test can be seen in Appendix C.

3.2.2 Mass

The mass of the frame was found by detaching the frame from the stand and measuring its weight. The weight of the wheel was then subtracted. The mass of the frame was determined to be 0.546 kg.

3.2.3 Inertia and friction

An experiment has been designed to determine both the inertia of the frame and the friction coefficient for the bearing. The journal can be found in Appendix D.

The Cubli is hung upside down and allowed to swing freely as a pendulum. The Cubli is initially held at an angle before being released. During the swinging motion the time and the position of the frame are measured. These measurements are then analysed using SENSTOOLS, a toolbox for MATLAB created by Morten Knudsen.

To estimate the inertia of the frame and the friction coefficient, SENSTOOLS requires a model of the system.

Equation 2.22 is used as a starting point:

$$\begin{aligned}\ddot{\theta}_f &= \frac{g \cdot (m_w \cdot l_w + m_f \cdot l_f)}{J_f + m_w \cdot l_w^2} \cdot \theta_f + \frac{-b_f}{J_f + m_w \cdot l_w^2} \cdot \dot{\theta}_f \\ &\quad + \frac{b_w}{J_f + m_w \cdot l_w^2} \cdot \dot{\theta}_w + \frac{1}{J_f + m_w \cdot l_w^2} \cdot \tau_m \\ \ddot{\theta}_w &= \frac{-g \cdot (m_w \cdot l_w + m_f \cdot l_f)}{J_f + m_w \cdot l_w^2} \cdot \theta_f + \frac{b_f}{J_f + m_w \cdot l_w^2} \cdot \dot{\theta}_f \\ &\quad + \frac{-b_w \cdot (J_w + J_f + m_w \cdot l_w^2)}{J_w \cdot (J_f + m_w \cdot l_w^2)} \cdot \dot{\theta}_w + \frac{J_w + J_f + m_w \cdot l_w^2}{J_w \cdot (J_f + m_w \cdot l_w^2)} \cdot \tau_m\end{aligned}\tag{3.7}$$

Ignoring the dynamics of the wheel and setting $\tau_m = 0$, since no torque is applied by the motor during the experiment:

$$\ddot{\theta}_f = \frac{g \cdot (m_w \cdot l_w + m_f \cdot l_f)}{J_f + m_w \cdot l_w^2} \cdot \theta_f + \frac{-b_f}{J_f + m_w \cdot l_w^2} \cdot \dot{\theta}_f\tag{3.8}$$

Since the Cubli is upside down during these experiments the sign of the term: $\frac{g \cdot (m_w \cdot l_w + m_f \cdot l_f)}{J_f + m_w \cdot l_w^2} \cdot \theta_f$ is switched:

$$\ddot{\theta}_f = \frac{-g \cdot (m_w \cdot l_w + m_f \cdot l_f)}{J_f + m_w \cdot l_w^2} \cdot \theta_f + \frac{-b_f}{J_f + m_w \cdot l_w^2} \cdot \dot{\theta}_f\tag{3.9}$$

Laplace transforming the equation and using the initial value theorem, where the initial velocity of the frame is set to zero yields the following equation:

$$\Theta_f \cdot s^2 - \theta_f(0) \cdot s = \frac{-g \cdot (m_w \cdot l_w + m_f \cdot l_f)}{J_f + m_w \cdot l_w^2} \cdot \Theta_f + \frac{-b_f}{J_f + m_w \cdot l_w^2} \cdot \Theta_f \cdot s\tag{3.10}$$

where:

$$\begin{aligned}
 \Theta_f &: \text{The Laplace transform of } \theta_f \\
 s &: \text{The complex frequency parameter} \\
 \theta_f(0) &: \text{The initial position of the frame} & [\text{rad}]
 \end{aligned}$$

Then solving for Θ_f :

$$\Theta_f = \frac{\theta_f(0) \cdot s}{s^2 + \frac{b_f \cdot s + g \cdot (m_w \cdot l_w + m_f \cdot l_f)}{J_f + m_w \cdot l_w^2}} \quad (3.11)$$

By then taking the impulse response of this transfer function the response of the system is found.

When implementing this model in SENSTOOLS, J_f and b_f are used as the two unknown parameters "par(1)" and "par(2)" respectively.

The input to the system, u , is specified as a zero vector, as no external, non-conservative forces are acting on the system.

The measurements need to be converted such that the time step between each pair of angle and time measurements are equal. To achieve this, linear interpolation via MATLAB's "interp1" command is used. The time vector is given in seconds and the angle vector in radians, where zero radians is equal to the equilibrium angle.

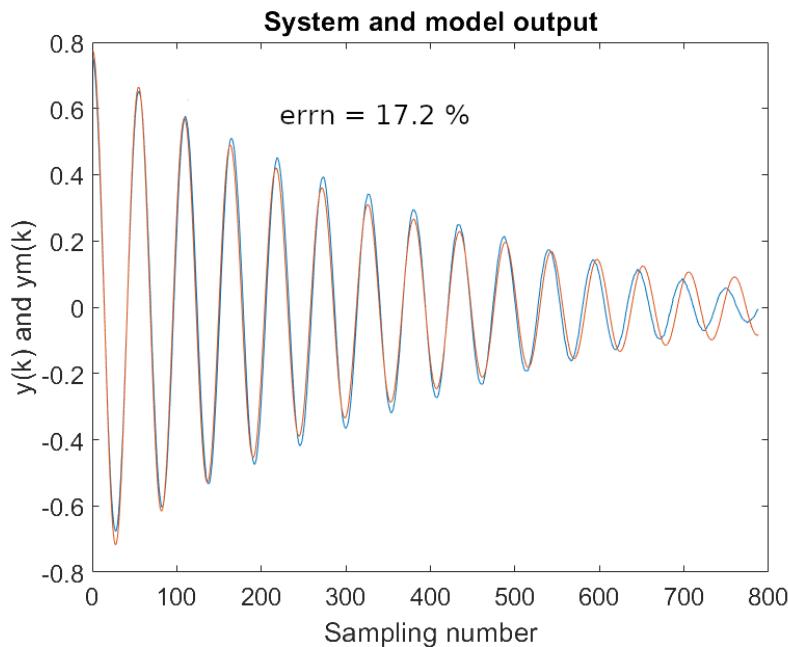


Figure 3.3: Results from test run 3. The blue line is the interpolated measured values and the red line is the fitted response by SENSTOOLS.

Three different runs of the experiment are conducted, resulting in comparable

estimates for the two coefficients as seen in Table 3.1. Test run 2 has been done from the opposite starting position of the other two.

Table 3.1: Estimates of the inertia and friction coefficient. errn is the normed root mean square output error.

Run	J_f [kg · m ²]	b_f [N · m · s · rad ⁻¹]	errn
1	0.0067	0.0036	16.2%
2	0.0067	0.0033	15.7%
3	0.0067	0.0038	17.2%
Average	0.0067	0.0036	

The errn values are higher than what can be expected by electro-mechanical systems as described by [10]. This is probably explained by the linearisation of the model causing large deviations due to the large initial angles used in the experiments.

3.3 Conclusion

Symbol	Description	New value	Value found by [6]
m_f	The mass of the frame. [kg]	0.546	0.551
l_f	The distance from the center of mass of the frame to the origin of the coordinate system. [m]	0.0958	0.1297425589
J_f	The moment of inertia of the frame. [$\text{kg} \cdot \text{m}^2$]	0.0067	0.009
b_f	The viscous damping coefficient of the bearing for the frame. [$\text{N} \cdot \text{m} \cdot \text{s} \cdot \text{rad}^{-1}$]	0.0036	0.0083
m_w	The mass of the reaction wheel. [kg]	0.220	0.220
l_w	The distance from the center of mass of the reaction wheel to the origin of the coordinate system. [m]	0.092	0.092
J_w	The moment of inertia of the reaction wheel. [$\text{kg} \cdot \text{m}^2$]	$0.699 \cdot 10^{-3}$	$0.699 \cdot 10^{-3}$
b_w	The viscous damping coefficient of the bearing between the frame and the wheel. [$\text{N} \cdot \text{m} \cdot \text{s} \cdot \text{rad}^{-1}$]	$1.93 \cdot 10^{-5}$	$1.83 \cdot 10^{-6}$
g	Gravitational acceleration. [m · s ⁻²]	9.816	9.81

In general, most parameters are of comparable size to the ones found in [6]. The discrepancy between the two values for b_w is assumed to be caused by a lack of conversion from RPM to rad · s⁻¹ in [6].

The new values will be used in the following worksheets.

4 State space representation

In this chapter, the model derived in chapter 2 using the parameters determined in chapter 3 is converted into a SISO state space model in section 4.1 and then analysed in section 4.2.

4.1 Conversion to a state space model

This section derives the state space representation of the Cubli, which is a prerequisite for using the LQR method for the controller design.

State space representation is used when a dynamic system with multiple inputs and/or outputs is described by second (or higher)-order differential equations. In that case, the second-order differential equations are reorganized as a set of first-order differential equations in a vector form. [9, ch. 7].

The system state equations can be represented in the state variable form as the matrix-vector-equation:

$$\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu} \quad (4.1)$$

where:

\mathbf{x} : The system states vector

u : The system input

\mathbf{A} : The system matrix

\mathbf{B} : The input matrix

And the system's output can be represented as the matrix-vector-equation:

$$y = \mathbf{Cx} + \mathbf{Du} \quad (4.2)$$

where:

y : The system output

\mathbf{C} : The output matrix

\mathbf{D} : The direct transmission term matrix

The state variables are chosen to be the variables describing the mechanical energy of the system. The potential energy is described through the angular position

of the frame. The kinetic energy is described through the angular velocity of the frame as well as the relative angular velocity of the wheel with respect to the frame.

The system states can be combined into a column vector.

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} \theta_f \\ \dot{\theta}_f \\ \dot{\theta}_w \end{bmatrix} \quad (4.3)$$

The system input is the torque from the motor:

$$u = \tau_m \quad (4.4)$$

In order to represent the system state equations in the state variable form, the first derivatives of the system states \mathbf{x} need to be found. Using the second-order differential equations in Equation 2.22 and $\dot{\theta}_f = \frac{d\theta_f}{dt}$, the Cubli's dynamics can be expressed as a set of first-order differential equations:

$$\dot{x}_1 = x_2$$

$$\begin{aligned} \dot{x}_2 &= \frac{g \cdot (m_w \cdot l_w + m_f \cdot l_f)}{J_f + m_w \cdot l_w^2} \cdot x_1 + \frac{-b_f}{J_f + m_w \cdot l_w^2} \cdot x_2 \\ &\quad + \frac{b_w}{J_f + m_w \cdot l_w^2} \cdot x_3 - \frac{1}{J_f + m_w \cdot l_w^2} \cdot u_1 \end{aligned} \quad (4.5)$$

$$\begin{aligned} \dot{x}_3 &= \frac{-g \cdot (m_w \cdot l_w + m_f \cdot l_f)}{J_f + m_w \cdot l_w^2} \cdot x_1 + \frac{b_f}{J_f + m_w \cdot l_w^2} \cdot x_2 \\ &\quad + \frac{-b_w \cdot (J_w + J_f + m_w \cdot l_w^2)}{J_w \cdot (J_f + m_w \cdot l_w^2)} \cdot x_3 + \frac{J_w + J_f + m_w \cdot l_w^2}{J_w \cdot (J_f + m_w \cdot l_w^2)} \cdot u_1 \end{aligned}$$

Now the Cubli's state equations in matrix form can be defined by:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ \frac{g \cdot (m_w \cdot l_w + m_f \cdot l_f)}{J_f + m_w \cdot l_w^2} & \frac{-b_f}{J_f + m_w \cdot l_w^2} & \frac{b_w}{J_f + m_w \cdot l_w^2} \\ \frac{-g \cdot (m_w \cdot l_w + m_f \cdot l_f)}{J_f + m_w \cdot l_w^2} & \frac{b_f}{J_f + m_w \cdot l_w^2} & \frac{-b_w \cdot (J_w + J_f + m_w \cdot l_w^2)}{J_w \cdot (J_f + m_w \cdot l_w^2)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{1}{J_f + m_w \cdot l_w^2} \\ \frac{J_w + J_f + m_w \cdot l_w^2}{J_w \cdot (J_f + m_w \cdot l_w^2)} \end{bmatrix} u_1 \quad (4.6)$$

The system output is chosen to be θ_f :

$$y = \theta_f \quad (4.7)$$

The Cubli's output equation in matrix form can be expressed as:

$$y = [1 \ 0 \ 0] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + 0 \cdot u_1 = [1 \ 0 \ 0] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad (4.8)$$

Equation 4.6 and Equation 4.8 represent the final state space model of the system and the output:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ \frac{g \cdot (m_w \cdot l_w + m_f \cdot l_f)}{J_f + m_w \cdot l_w^2} & \frac{-b_f}{J_f + m_w \cdot l_w^2} & \frac{b_w}{J_f + m_w \cdot l_w^2} \\ \frac{-g \cdot (m_w \cdot l_w + m_f \cdot l_f)}{J_f + m_w \cdot l_w^2} & \frac{b_f}{J_f + m_w \cdot l_w^2} & \frac{-b_w \cdot (J_w + J_f + m_w \cdot l_w^2)}{J_w \cdot (J_f + m_w \cdot l_w^2)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{1}{J_f + m_w \cdot l_w^2} \\ \frac{J_w + J_f + m_w \cdot l_w^2}{J_w \cdot (J_f + m_w \cdot l_w^2)} \end{bmatrix} u_1$$

$$y_1 = [1 \ 0 \ 0] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad (4.9)$$

Inserting the values found in chapter 3 results in:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 83 & -0.42 & 0.0023 \\ -83 & 0.42 & -0.030 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ -1.2 \cdot 10^2 \\ 1.5 \cdot 10^3 \end{bmatrix} u_1 \quad (4.10)$$

$$y_1 = [1 \ 0 \ 0] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

Note, that the values are up to 5 orders of magnitude different. This might cause problems with the numerical computation in the following sections. A discussion and analysis of this can be found in section 4.3.

4.2 Analysis of the state space model

In the following section, the state space model will be analysed. The following is of interest:

1. Pole location
2. Zero location

3. Controllability
4. Observability

4.2.1 Pole location

The poles of a state space model can be calculated by finding the eigenvalues of the \mathbf{A} matrix.

$$p = \text{eig}(\mathbf{A}) = \{-9.3, 8.9, -0.028\} \quad (4.11)$$

where:

p : The poles of the system.

Since one of the poles is in the right half-plane, the system is unstable in its open loop configuration.

4.2.2 Zero location

The zeros of a Single Input, Single Output state space model can be found by solving the equation:

$$\det \begin{pmatrix} [z \cdot \mathbf{I} - \mathbf{A}] & -\mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{pmatrix} = 0 \quad (4.12)$$

\Downarrow

$$z = 0$$

z : The zeros of the system.

\mathbf{I} : The identity matrix.

Since the zero is located in the origin of the complex plane, the system has 0 DC gain.

4.2.3 Controllability

A system is controllable if the controllability matrix has full rank.

$$\begin{aligned} \mathcal{C} &= [\mathbf{B} \ \mathbf{AB} \ \dots \ \mathbf{A}^{n-1}\mathbf{B}] \\ &= \begin{bmatrix} 0 & -1.2 \cdot 10^2 & 52 \\ -1.2 \cdot 10^2 & 52 & -9.7 \cdot 10^3 \\ 1.5 \cdot 10^3 & -95 & 9.7 \cdot 10^3 \end{bmatrix} \end{aligned} \quad (4.13)$$

where:

- \mathcal{C} : The controllability matrix.
- n : The number of states of the system.

For a SISO system, the controllability matrix is a square matrix. Thus, the determinant of the controllability matrix must be non-zero for the matrix to have full rank.

$$\det(\mathcal{C}) = 1.6 \cdot 10^9 \neq 0 \quad (4.14)$$

Consequently, the system is controllable.

4.2.4 Observability

A system is observable if the observability matrix has full rank.

$$\begin{aligned} \mathcal{O} &= \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \vdots \\ \mathbf{CA}^{n-1} \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 83 & -0.42 & 0.0023 \end{bmatrix} \end{aligned} \quad (4.15)$$

For a SISO system, the observability matrix is a square matrix. Thus, the determinant of the observability matrix must be non-zero for the matrix to have full rank.

$$\det(\mathcal{O}) = 0.0023 \neq 0 \quad (4.16)$$

Consequently, the system is observable.

4.3 The condition numbers and different approaches

The calculations in subsection 4.2.1 to 4.2.4 are based on numerical matrix computations. Therefore, the results may be influenced by instability due to ill-conditioned matrices. To determine if this is the case, the condition numbers for several matrices are examined.

$$\text{cond}(\mathbf{X}) = \|\mathbf{X}\|_2 \cdot \|\mathbf{X}^{-1}\|_2 \quad (4.17)$$

where:

- \mathbf{X} : An arbitrary matrix.
- $\|\cdot\|_2$: Spectral norm.

Generally, a low condition number indicates a well-conditioned system of equations, and a high condition number indicates an ill-conditioned one. However, there is no clear divide between well- and ill-conditioned systems.

The condition numbers for the matrices used in subsection 4.2.1 to 4.2.4 is calculated using MATLAB's "cond" command:

$$\begin{aligned} \text{cond}(\mathbf{A}) &= 6.0 \cdot 10^3 \\ \text{cond}\left(\begin{bmatrix} \mathbf{A} & -\mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix}\right) &= 1 \\ \text{cond}(\mathcal{C}) &= 1.2 \cdot 10^2 \\ \text{cond}(\mathcal{O}) &= 3.1 \cdot 10^6 \end{aligned} \tag{4.18}$$

The matrix used for finding the zeros of the system is well-conditioned, indicated by the condition number of 1. However, the rest are difficult to judge.

Thus, different mathematical approaches for finding the pole locations, the controllability, and the observability of the system has been found.

4.3.1 Alternative approach for pole and zero location

First, the transfer function for the system is found by taking the Laplace transform of Equation 2.20:

$$\begin{aligned}\ddot{\theta}_f &= \frac{g \cdot (m_w \cdot l_w + m_f \cdot l_f)}{J_f + m_w \cdot l_w^2} \cdot \theta_f + \frac{-b_f}{J_f + m_w \cdot l_w^2} \cdot \dot{\theta}_f + \frac{b_w}{J_f + m_w \cdot l_w^2} \cdot \dot{\theta}_w \\ &\quad + \frac{-1}{J_f + m_w \cdot l_w^2} \cdot \tau_m \\ \ddot{\theta}_w &= \frac{-b_w}{J_w} \cdot \dot{\theta}_w + \frac{1}{J_w} \cdot \tau_m - \ddot{\theta}_f \\ &\Downarrow \mathcal{L} \end{aligned} \tag{4.19}$$

$$\begin{aligned}\Theta_f \cdot s^2 &= \frac{g \cdot (m_w \cdot l_w + m_f \cdot l_f)}{J_f + m_w \cdot l_w^2} \cdot \Theta_f + \frac{-b_f \cdot s}{J_f + m_w \cdot l_w^2} \cdot \Theta_f + \frac{b_w \cdot s}{J_f + m_w \cdot l_w^2} \cdot \Theta_w \\ &\quad + \frac{-1}{J_f + m_w \cdot l_w^2} \cdot T_m \\ \Theta_w \cdot s^2 &= \frac{-b_w \cdot s}{J_w} \cdot \Theta_w + \frac{1}{J_w} \cdot T_m - \Theta_f \cdot s^2\end{aligned}$$

where:

- Θ_f : The Laplace transform of θ_f .
- Θ_w : The Laplace transform of θ_w .
- T_m : The Laplace transform of τ_m .

Then isolating $\frac{\Theta_f}{T_m}$:

$$\frac{\Theta_f}{T_m} = \frac{b(s)}{a(s)} \tag{4.20}$$

where:

$$b(s) = J_w \cdot s$$

$$\begin{aligned}a(s) &= -J_w \cdot (J_f + m_w \cdot l_w^2) \cdot s^3 \\ &\quad + ((-b_f - b_w) \cdot J_w - b_w \cdot (J_f + m_w \cdot l_w^2)) \cdot s^2 \\ &\quad + (g \cdot (m_f \cdot l_f + m_w \cdot l_w) \cdot J_w - b_w \cdot b_f) \cdot s \\ &\quad + b_w \cdot g \cdot (m_w \cdot l_w + m_f \cdot l_f)\end{aligned} \tag{4.21}$$

From this the poles and zeros can be found by setting $a(s) = 0$ and $b(s) = 0$ and solving for s :

$$\begin{aligned} z &= 0 \\ p &= \{-9.3, -0.028, 8.9\} \end{aligned} \tag{4.22}$$

For finding the roots of the 3rd-degree polynomial $a(s)$, MATLAB's "vpasolve" command is used. These results are the same as Equation 4.11 and 4.12.

4.3.2 Alternative approach for controllability and observability

For the controllability and observability the determinants are calculated algebraically:

$$\begin{aligned} \det(\mathcal{C}) &= \frac{g \cdot (m_w \cdot l_w + m_f \cdot l_f)}{J_w \cdot (J_f + m_w \cdot l_w^2)^3} \approx 1.6 \cdot 10^9 > 0 \\ \det(\mathcal{O}) &= \frac{b_w}{J_f + m_w \cdot l_w^2} \approx 0.0023 > 0 \end{aligned} \tag{4.23}$$

Thus, the determinants are non-zero and both matrices have full rank, resulting in once again the system being controllable and observable.

5 Controller design

This chapter contains an explanation of the chosen control strategy for balancing the Cubli: full state feedback using a Linear–Quadratic Regulator. This method is used for the ETH Zürich prototype Cubli[1], and was also previously implemented on AAU’s Cubli [6].

In addition, the modeling as well as implementation and testing of a new start up procedure, which is a prerequisite for synchronising start up between two Cublis, is discussed in this chapter. The model is in accordance with the model described in [1].

Finally, designing of two new shut down procedures, namely braking before impact and deceleration, are considered in this chapter. These methods are then tested and analyzed in order to evaluate their performance together with disturbances in order to choose the more robust method of the two.

Section 5.1 contains a general description of LQR with a simple example. Section 5.2 presents designing of the Cubli’s LQR controller, along with its initial guess and tuning. Next, section 5.3 provides a description for the automatic center of mass adjustment. Section 5.4 describes and models the new start up method for the Cubli and section 5.5 specifies two approaches, that are implemented for shut down.

5.1 Linear-Quadratic Regulator

This section provides understanding of the Linear Quadratic Regulator approach for designing an optimal controller for the Cubli. Furthermore, a simple step-by-step example using the method is presented, in order to provide context and illustrate understanding of the topic.

Designing an optimal controller for the Cubli can be achieved using Optimal Control theory, which deals with optimizing the performance of a controller. Optimal control theory provides methods for deriving optimal location of the closed-loop system poles on the complex plane, which is crucial for designing a stable system with given performance characteristics. [11, p.399]

One popular method to achieve optimal control is using a Linear Quadratic Regulator (LQR) approach. Intuitively, LQR’s principle is to balance the actuator input and the system’s aggressiveness by weighing the priorities for each input u_n and state x_n . While similar techniques for controller design, such as pole placement, require knowledge about desired pole locations on the complex plane, the

LQR method derives the optimal pole locations based on the above-mentioned weights. This facilitates the controller design and provides a single solution to the optimization problem.[12] [11, p.xvi]

5.1.1 Optimization problem definition

As mentioned, finding the optimal controller using the LQR approach is an optimization problem with a solution that maximizes performance based on desired system characteristics. This problem can be inverted and defined as the solution that instead minimizes a cost function. For LQR, the cost function is given by: [13, p.359, 390].

$$J = \int_0^\infty \mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{u}^T \mathbf{R} \mathbf{u} \, dx \quad (5.1)$$

where:

- J : The cost function
- \mathbf{x} : The state vector
- \mathbf{u} : The input vector
- \mathbf{Q} : Positive semi-definite matrix containing state weights
- \mathbf{R} : Positive definite matrix containing input weights

Note that the upper bound of the integral in the cost function is infinite. This type of cost function is known to have infinite planning horizon, which simplifies the mathematics for determining the LQR controller [11, p.419].

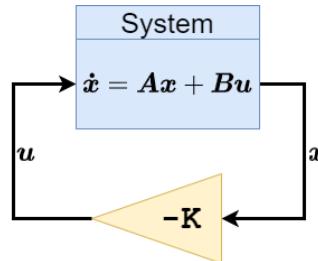


Figure 5.1: Full-state feedback controller

5.1.2 Controller scheme

The solution to the above mentioned optimization problem can be achieved using a full-state feedback control law, which is illustrated in Figure 5.1, meaning that all system states are fed back to the controller. This implies that there exists a gain matrix \mathbf{K} which provides the optimal control to minimize the cost function J .

In addition, using a full-state feedback controller, the input vector \mathbf{u} can now be expressed as a function of \mathbf{K} as follows: [13, p.235]

$$\mathbf{u} = -\mathbf{K}\mathbf{x} \quad (5.2)$$

Substituting the input vector from Equation 5.2 in the state space representation of the system yields the following equation:

$$\dot{\mathbf{x}} = (\mathbf{A} - \mathbf{B}\mathbf{K})\mathbf{x} \quad (5.3)$$

where:

- \mathbf{A} : The system variable matrix from the state-space model
- \mathbf{B} : The input matrix from the state-space model
- \mathbf{K} : The controller gain matrix

5.1.3 Choosing the Q and R matrices

Now that the state-space representation is expressed as a function of the gain matrix \mathbf{K} , the Q and R matrices must be chosen to represent the desired performance of the system. Generally, large Q entries, compared to corresponding R entries would prioritize faster responses when reaching a desired state (for each of the system states). The opposite would minimize the input force/torque, aiming to e.g. minimize power consumption. There exist methods for deciding the Q and R entries, which are used as a starting point for the tuning. However, the two matrices would be further tuned based on simulation and experiments, observing what Q and R entries yield satisfactory results. [14, p.196]

5.1.4 Optimal gain definition

Following the LQR approach, the optimal gain \mathbf{K} is given by the following equation: [13, p.391]

$$\mathbf{K} = \mathbf{R}^{-1}\mathbf{B}^T\mathbf{S} \quad (5.4)$$

where \mathbf{S} is a symmetric matrix which solves the algebraic Riccati equation (ARE) given by: [13, p.390]

$$\mathbf{A}^T\mathbf{S} + \mathbf{S}\mathbf{A} - \mathbf{S}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{S} + \mathbf{Q} = 0 \quad (5.5)$$

Please note that deriving Equation 5.4 and Equation 5.5 is not in the scope of this project. The key point is understanding that those equations result in the optimal gain \mathbf{K} , which minimizes the cost function J .

5.1.5 Optimal gain computation

It is important to note that there exist more than one \mathbf{S} which satisfies the Riccati equation. This implies several optimal gain candidates $\mathbf{K}_1, \dots, \mathbf{K}_n$. However, it is known that only one of the possible \mathbf{K} matrices would yield a stable system. This is evaluated by computing the eigenvalues of $(\mathbf{A} - \mathbf{B}\mathbf{K})$ for each $\mathbf{K}_1, \dots, \mathbf{K}_n$, which represent the close-loop system from Equation 5.3. A stable system is characterized by eigenvalues on the left half of the complex plane. Therefore, the optimal gain \mathbf{K} which minimizes the cost function J is the one yielding eigenvalues of $(\mathbf{A} - \mathbf{B}\mathbf{K})$ with negative real part. [13, p.395]

5.1.6 LQR example

This subsection provides a simple example of LQR implementation in a mass damper system. The example is inspired by [15]. Let the mass $m = 2.00$, the friction coefficient $b = 0.300$ and the position of the system p . Figure 5.2 shows all forces exerted on the mass, i.e. friction $b \cdot \dot{p}$ and input force $F(t)$. In order to derive optimal control gain, the following step-by-step procedure can be derived from the LQR description provided earlier in this section:

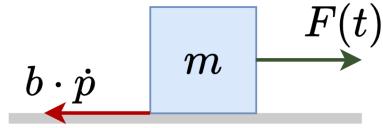


Figure 5.2: Example mass damper system with mass m , friction coefficient b , position p and input force $F(t)$

1. Derive the state space model.
2. Choose \mathbf{Q} and \mathbf{R} .
3. Solve the algebraic Riccati Equation to get all solutions $\mathbf{S}_1, \dots, \mathbf{S}_n$.
4. Compute $\mathbf{K}_1, \dots, \mathbf{K}_n = \mathbf{R}^{-1}\mathbf{B}^T\mathbf{S}$ for all $\mathbf{S}_1, \dots, \mathbf{S}_n$.
5. Choose the \mathbf{K}_i which yields a stable system, i.e. $\mathbf{A} - \mathbf{B}\mathbf{K}_i$ has negative eigenvalues.

The Equation of motion for the system can be expressed as:

$$m \cdot \ddot{p} = F(t) - b \cdot \dot{p} \quad (5.6)$$

Let the state vector comprise the position p and velocity v of the mass. Then the state-space representation of the system in the form $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$ can be expressed as follows:

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{p} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{b}{m} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} \cdot F(t) \quad (5.7)$$

Equation 5.7 completed step one from the LQR methodology. Now the \mathbf{Q} and \mathbf{R} matrices need to be chosen. Note that their dimensions have to comply with the dimensions of the \mathbf{x} and \mathbf{u} respectively, so that the terms $\mathbf{x}^T \mathbf{Q} \mathbf{x}$ and $\mathbf{u}^T \mathbf{R} \mathbf{u}$ from Equation 5.1 are legal operations. Let us define \mathbf{Q} as the 2×2 identity matrix, and R be a 1×1 matrix $\mathbf{R} = [1.00 \cdot 10^{-2}]$. Such a choice would prioritize the correction of the state variables and would not try to minimize the input forces. The next step is to solve the algebraic Riccati equation (ARE) to obtain all $\mathbf{S}_1, \dots, \mathbf{S}_n$. To do that, we first simplify Equation 5.5 by plugging in the known matrices. This is done in MATLAB, which results in the following expression of the Riccati equation:

$$\begin{bmatrix} -25.0 \cdot s_{21}^2 + 1.00 & -25.0 \cdot s_{21} \cdot s_2 + s_1 - 0.150 \cdot s_{21} \\ -25.0 \cdot s_{21} \cdot s_2 + s_1 - 0.150 \cdot s_{21} & -25.0 \cdot s_2^2 - 0.300 \cdot s_2 + 2 \cdot s_{21} + 1.00 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (5.8)$$

where s_1 , s_2 and s_{21} are the coefficients in the symmetric matrix \mathbf{S} from Equation 5.5 given by:

$$\mathbf{S} = \begin{bmatrix} s_1 & s_{21} \\ s_{21} & s_2 \end{bmatrix} \quad (5.9)$$

Equation 5.8 has nonlinear terms, and thus there are finite many solutions for \mathbf{S} . Using MATLAB, the following solutions of the Riccati equation can be computed:

$$\begin{aligned} \mathbf{S}_1 &= \begin{bmatrix} 0.775 & -0.200 \\ -0.200 & -0.161 \end{bmatrix} \\ \mathbf{S}_2 &= \begin{bmatrix} -0.775 & -0.200 \\ -0.200 & 0.149 \end{bmatrix} \\ \mathbf{S}_3 &= \begin{bmatrix} -1.14 & 0.200 \\ 0.200 & -0.243 \end{bmatrix} \\ \mathbf{S}_4 &= \begin{bmatrix} 1.18 & 0.200 \\ 0.200 & 0.231 \end{bmatrix} \end{aligned} \quad (5.10)$$

Using these solutions, we can now compute the candidates for the optimal gain $\mathbf{K}_1, \dots, \mathbf{K}_4$. This is done by substituting each of the above-stated \mathbf{S}_{1-4} into

Equation 5.4, i.e. $\mathbf{K}_i = \mathbf{R}^{-1}\mathbf{B}^T\mathbf{S}_i$:

$$\begin{aligned}\mathbf{K}_1 &= [-10.0 \quad -8.05] \\ \mathbf{K}_2 &= [-10.0 \quad 7.45] \\ \mathbf{K}_3 &= [10.0 \quad -12.1] \\ \mathbf{K}_4 &= [10.0 \quad 11.5]\end{aligned}\tag{5.11}$$

The last step in the procedure is to compute the eigenvalues of $(\mathbf{A} - \mathbf{B}\mathbf{K})$ for all \mathbf{K}_i , and observe which of the \mathbf{K} matrices yields negative eigenvalues. The "eig" MATLAB command is used to obtain the following results:

$$\begin{aligned}\text{eig}(\mathbf{A} - \mathbf{B}\mathbf{K}_1) &= \{-1.02, 4.90\} \\ \text{eig}(\mathbf{A} - \mathbf{B}\mathbf{K}_2) &= \{1.02, -4.90\} \\ \text{eig}(\mathbf{A} - \mathbf{B}\mathbf{K}_3) &= \{1.02, 4.90\} \\ \text{eig}(\mathbf{A} - \mathbf{B}\mathbf{K}_4) &= \{-1.02, -4.90\}\end{aligned}\tag{5.12}$$

It can be concluded that $\mathbf{K}_4 = [10 \quad 11.536]$ is the optimal gain matrix for the mass damper example for the chosen \mathbf{Q} and \mathbf{R} . It is a solution to the Algebraic Riccati equation and it results in a stable system, i.e. places the poles in the left half-plane. Therefore, it solves the optimization problem and minimizes the cost function J . Using the same method, it is possible to obtain the optimal gain for the Cubli controller using the data from chapter 3.

5.2 Linear Quadratic Regulator for the Cubli

In the following section, the LQR based controller is designed using the state space model found in chapter 4.

Due to the three states of the system and the one input, the \mathbf{Q} is a 3×3 matrix and R is a scalar value. The state feedback controller will then be a 1×3 matrix.

5.2.1 Initial guess

For initial design of the controller, Bryson's Rule is employed [14, p. 196]:

$$Q = \begin{bmatrix} \frac{1}{(\max(x_1))^2} & 0 & 0 \\ 0 & \frac{1}{(\max(x_2))^2} & 0 \\ 0 & 0 & \frac{1}{(\max(x_3))^2} \end{bmatrix} \quad (5.13)$$

$$R = \left[\frac{1}{(\max(u_1))^2} \right]$$

The maximum value for the different states and the input can be found in Table 5.1.

Table 5.1: The maximum value for the different states and the input of the system.

States and inputs	Variable	Maximum value
x_1	θ_f	$\frac{\pi}{4}$ rad
x_2	$\dot{\theta}_f$	N/A
x_3	$\dot{\theta}_w$	6000 RPM $\approx 628 \text{ rad} \cdot \text{s}^{-1}$
u_1	τ_m	7.5 A $\approx 0.25 \text{ N} \cdot \text{m}$

Inserting these into Equation 5.13:

$$Q = \begin{bmatrix} 1.62 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 2.53 \cdot 10^{-6} \end{bmatrix} \quad (5.14)$$

$$R = 15.8$$

Note, $Q_{2,2}$ has been set to 0, as there is no upper allowable speed for the frame.

Using these and the model from chapter 4 in MATLAB's "lqr" command results in the following controller:

$$K = [-1.58 \quad -1.66 \cdot 10^{-1} \quad -4.20 \cdot 10^{-4}] \quad (5.15)$$

On Figure 5.3 the simulated response of the system can be seen using the state space model. Only the initial position of the frame is non-zero.

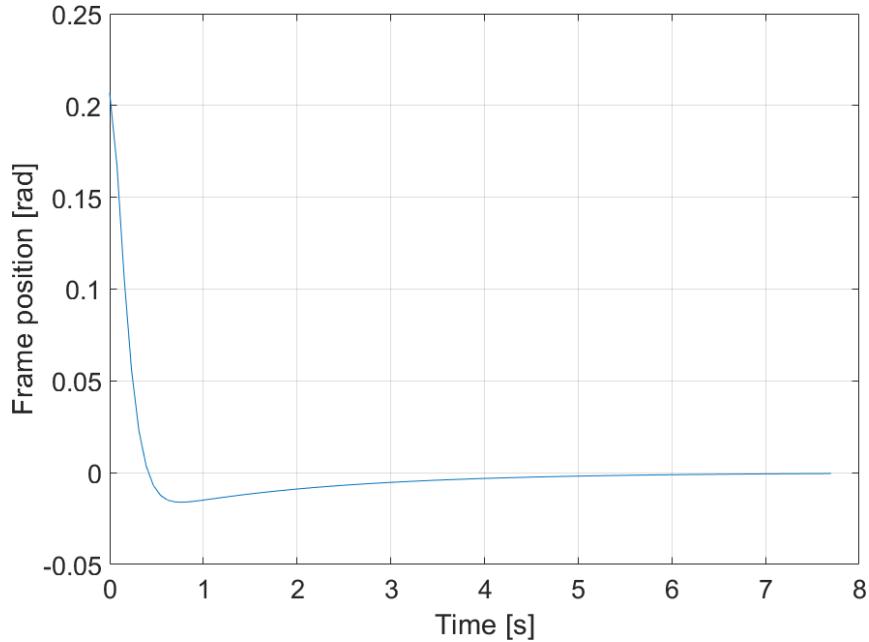


Figure 5.3: System response to non-zero initial condition for the position of the frame, with LQR using Bryson's rule.

However, the transient response of the system can be improved by having a decreased overshoot and increased settling time.

5.2.2 Tuning

To improve the transient response of the system the \mathbf{Q} and \mathbf{R} matrices found by Bryson's rule are tuned:

1. $\mathbf{Q}_{1,1}$ is increased as the position of the frame is of great importance.
2. \mathbf{R} is decreased as the power consumption of the DC-motor is of little importance.
3. $\mathbf{Q}_{2,2}$ and $\mathbf{Q}_{3,3}$ are not changed.

Through trial and error, the following \mathbf{Q} and \mathbf{R} matrices have been found by multiplying $\mathbf{Q}_{1,1}$ with 100 and \mathbf{R} by 0.05:

$$\mathbf{Q} = \begin{bmatrix} 162 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 2.53 \cdot 10^{-6} \end{bmatrix} \quad (5.16)$$

$$\mathbf{R} = 0.792$$

These values results in the following feed back controller:

$$\mathbf{K} = [-15.1 \quad -0.529 \quad -1.81 \cdot 10^{-3}] \quad (5.17)$$

The transient response of the system with the tuned controller can be found on Figure 5.4.

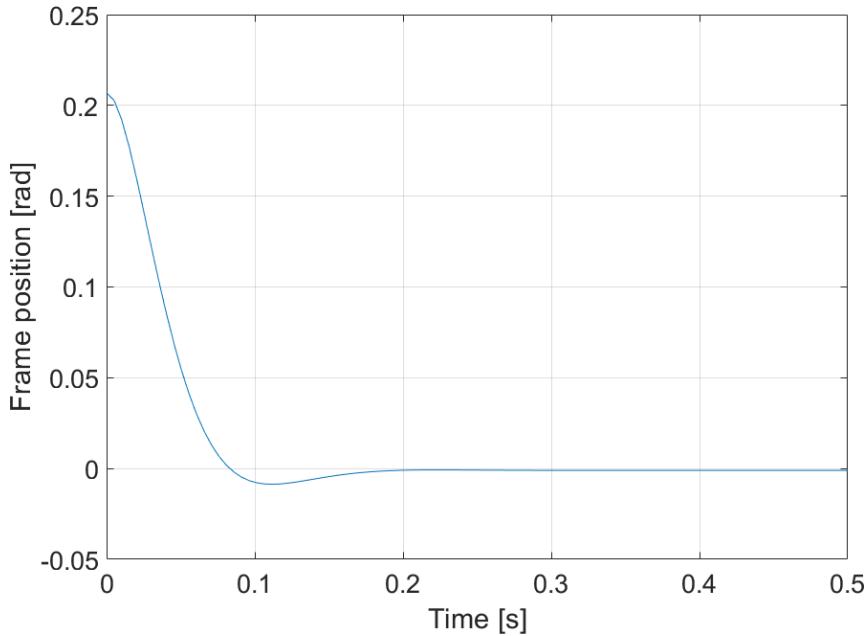


Figure 5.4: System response to non-zero initial condition for the position of the frame, with LQR using the tuned \mathbf{Q} and \mathbf{R} .

A comparison to the transient response of the two feedback controllers can be found on Figure 5.5.

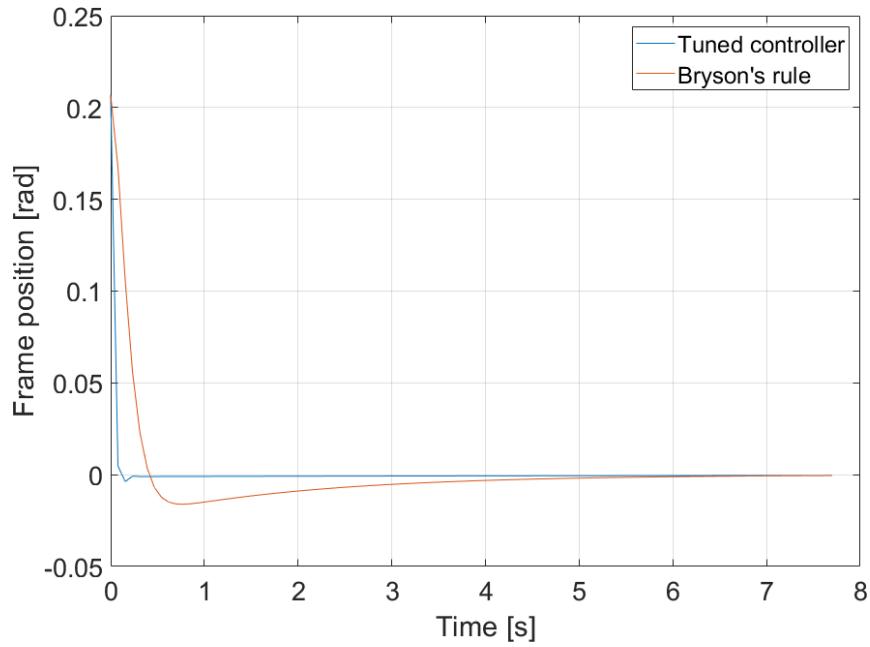


Figure 5.5: System response to non-zero initial condition for the position of the frame, with LQR using the tuned Q and R compared to Bryson's rule.

Lastly, a simple experiment is conducted to compare the two controllers, as well as the controller using the Q and R matrices from [6].

The Cubli was placed at an angle of $\approx 10^\circ$ using an angled block of wood. The controller was then turned on, and the angle of the frame with respect to time was then measured using the IMU. The results can be seen on Figure 5.6,

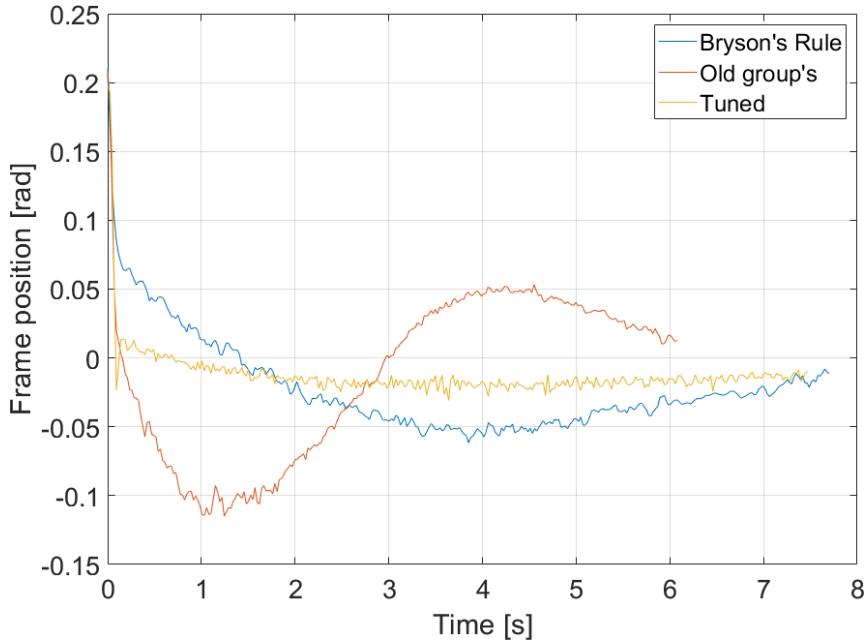


Figure 5.6: Test results for the three different controllers.

As can be seen from the figures, the tuned controller has a faster response and a smaller overshoot than the other two controllers. A more detailed report of the test can be seen in Appendix E.

Additionally, non-linearities are added to the model in Simulink so the test results can be compared better to the model's response. The non-linearities are:

1. The sine function from Equation 2.19.
2. The angle of the frame is limited to $\frac{\pi}{4}$ rad.
3. Coloumb friction is added to the reaction wheel, using the y -intercept found in subsection 3.1.2.
4. The input current is limited to 7.5 A.
5. The reaction wheel velocity is limited to 6000 RPM.
6. The controller is set to sample every 5.2 ms.
7. A pure delay of 10.4 ms is added between the input and output of the controller, see test results in Appendix I.

On Figure 5.7 the linear block diagram can be seen. On Figure 5.8 the non-linearities 1 - 5 have been added.

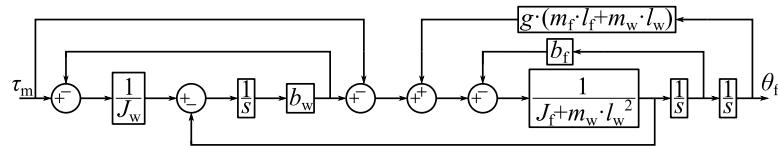


Figure 5.7: Linear block diagram of the system with no controller.

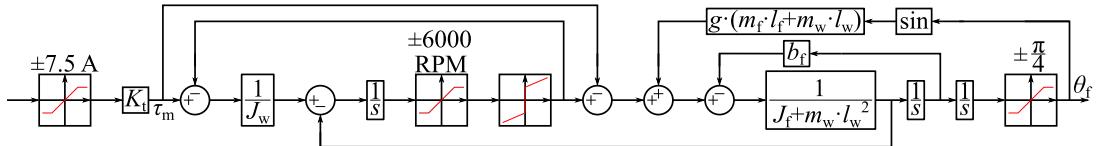


Figure 5.8: Non-linear block diagram of the system with no controller. Saturation limits are denoted by $\boxed{\text{---}}$ and the combination of Coulomb and viscous friction is denoted by $\boxed{\text{---}}$.

On Figure 5.9, the response of the linear and non-linear model is compared to the test results using the initial frame position from the test results.

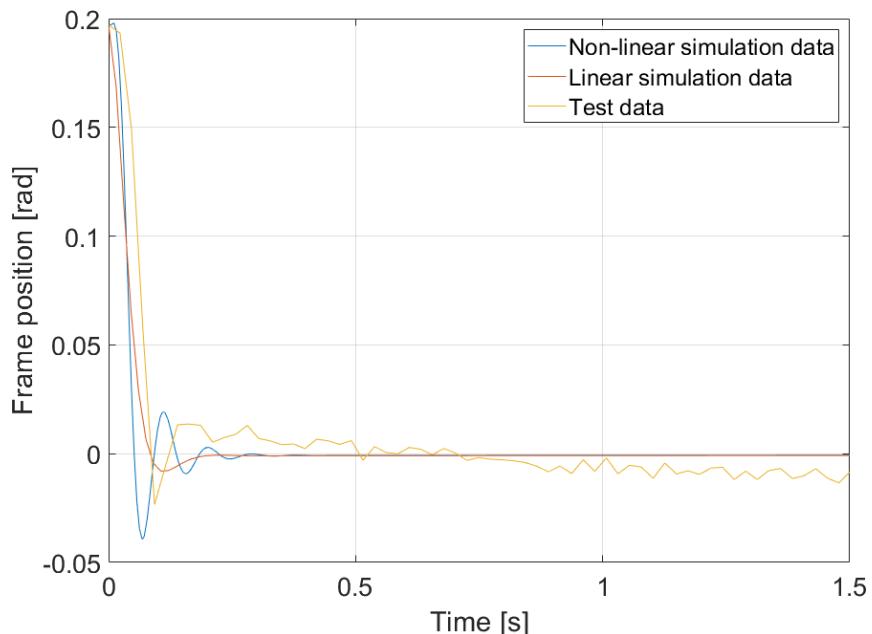


Figure 5.9: The test results for the tuned controller compared to the simulated response using both the linear and non-linear model.

The overshoot of the system is comparable to the non-linear model, but the real system appears slower. This might be due to discrepancies in when the data transmitting is started, as it might not have happened exactly when the controller turned on.

Note, the reason for the non-zero final value of the test results is presumably due to the centre of mass of the Cubli not being exactly along the diagonal. For dealing with this in the control loop, [5] implemented a dynamic reference angle adjustment.

5.3 Reference angle adjustment

In this section the method for continually adjusting the reference angle is described. The method for adjusting the reference angle is exactly the same as [5]. This is to compensate for the centre of mass of the Cubli not being exactly along the diagonal of the frame as seen Figure 5.10.

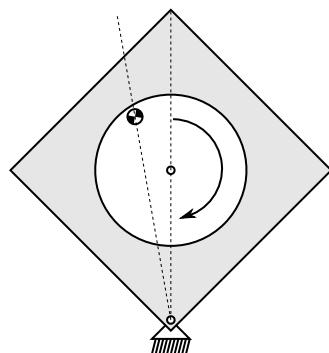


Figure 5.10: Sketch of the discrepancy between the reference angle and the equilibrium angle. The CoM of the Cubli is denoted by \oplus symbol. The curved arrow denotes the rotational direction of the reaction wheel.

The principle of this method is to check the average rotational velocity of the reaction wheel, when the Cubli is attempting to balance at the original reference angle. If the average rotational velocity is positive, then the system changes the angle reference clockwise, and vice versa. The average is taken every 50 loops and the correction is done dynamically as follows: the reference angle is changed by a big step of 0.01 rad if the average rotation speed is bigger than 100 RPM, which means the Cubli is further away from equilibrium; otherwise, it will be changed by a small step of 0.001 rad.

However, the reaction wheel's speed may also be influenced by disturbances. This is dealt with by comparing the 50 measurements of the velocity with the average of those 50 measurements. If 10 or more of the measurements are more than 20% different from the average value, the reference angle is not updated, as it is assumed the system is being disturbed or not near an equilibrium.

5.4 Start up

Initially, the Cubli needs to get from a resting position to near the balancing position, which requires a jump. This section explains the previous way from [5], and also describes the development of a new method.

5.4.1 Start up maneuver by [5]

First, the resting position is defined in the code as being more than $\pm 40^\circ$ away from the balancing point. This is measured using the potentiometer.

The start up maneuver was previously done by applying a current of $\approx 0.94\text{ A}$ to the motor for a set amount of time and then braking.

The time duration before applying the brake was 3 s if the Cubli is lying to one side, and 3.5 s if the Cubli is lying down to the other side. The difference in the time between the two sides is due to the Cubli being imbalanced. After waiting for the set amount of time, the brake is applied, thus launching the Cubli up towards the balancing position. When it gets within $\pm 40^\circ$ of the balancing point, the feedback controller and reference angle adjustment start attempting to balance the Cubli.

However, this method for start up requires the initial speed of the reaction wheel to be 0, and the different timings give rise to problems concerning the synchronisation between multiple Cublis.

Thus, a different approach has been developed.

5.4.2 New start up maneuver

The idea is to spin the reaction wheel to a predefined angular velocity before applying the brake, see Figure 5.11.

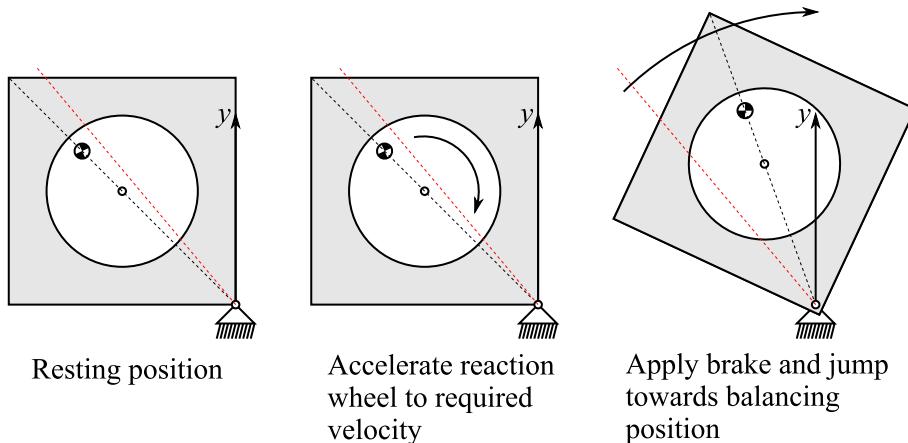


Figure 5.11: Sketch of the start up maneuver. The red stippled line shows the $\pm 40^\circ$ limit.

For this a feedback loop is used, as seen on Figure 5.12.

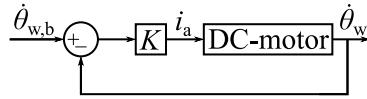


Figure 5.12: Feedback loop used for the speed controller.

The gain $K = 0.03$ is found through trial and error. This new method allows non-zero initial velocities of the reaction wheel, as well as easier synchronisation, as now only the brake has to be actuated at approximately the same time.

5.4.3 Start up model

A simple model is developed to approximate the necessary angular velocity of the wheel for the start up maneuver. The model is based on the model presented in [1].

The braking of the reaction wheel is modelled as a perfect inelastic collision between the frame and the reaction wheel. Thus, momentum is conserved during the collision. The conservation of energy becomes:

$$\sum L_b = \sum L_a \quad (5.18)$$

$$J_w \cdot \dot{\theta}_{w,b} + (J_f + J_w + m_w \cdot l_w^2) \cdot \dot{\theta}_{f,b} = J_w \cdot \dot{\theta}_{w,a} + (J_f + J_w + m_w \cdot l_w^2) \cdot \dot{\theta}_{f,a}$$

where:

L	: Angular momentum	$\text{kg} \cdot \text{m}^2 \cdot \text{s}^{-1}$
b	: Before the collision	
a	: After the collision	

Before the collision, the frame is at rest, $\dot{\theta}_{f,b} = 0$, and after the collision the reaction wheel is assumed to be at rest, so $\dot{\theta}_{w,a} = 0$. Equation 5.18 then becomes:

$$J_w \cdot \dot{\theta}_{w,b} = (J_f + J_w + m_w \cdot l_w^2) \cdot \dot{\theta}_{f,a} \quad (5.19)$$

Which then simplifies to:

$$\dot{\theta}_{w,b} = \frac{J_f + J_w + m_w \cdot l_w^2}{J_w} \cdot \dot{\theta}_{f,a} \quad (5.20)$$

From the brake is applied, to the frame moves to a stand still at its vertical position, the change in potential energy must equal the change in the kinetic energy

of the frame, if conservation of energy is assumed.

$$\Delta T = \Delta V$$

$$\frac{1}{2} \cdot (J_f + J_w + m_w \cdot l_w^2) \cdot \dot{\theta}_f^2 = g \cdot (m_f \cdot l_f + m_w \cdot l_w) \cdot (\cos(0) - \cos(\theta_{f,i})) \quad (5.21)$$

where:

$\theta_{f,i}$: Initial angular position of the frame. rad

Then isolating $\dot{\theta}_f$:

$$\dot{\theta}_f = \sqrt{\frac{2 \cdot g \cdot (m_f \cdot l_f + m_w \cdot l_w)}{J_f + J_w + m_w \cdot l_w^2} \cdot (\cos(0) - \cos(\theta_{f,i}))} \quad (5.22)$$

And combining with Equation 5.20, when $\dot{\theta}_{f,a} = \dot{\theta}_f$:

$$\dot{\theta}_{w,b} = \frac{J_f + J_w + m_w \cdot l_w^2}{J_w} \cdot \sqrt{\frac{2 \cdot g \cdot (m_f \cdot l_f + m_w \cdot l_w)}{J_f + J_w + m_w \cdot l_w^2} \cdot (\cos(0) - \cos(\theta_{f,i}))} \quad (5.23)$$

From this, the required angular velocity of the reaction wheel before braking can be calculated from an initial position of the frame.

Setting $\theta_{f,i} = \frac{\pi}{4}$ yields:

$$\dot{\theta}_{w,b} = 143 \text{ rad} \cdot \text{s}^{-1} \approx 1366 \text{ RPM} \quad (5.24)$$

The determined values for start up for the modified, old Cubli and the new Cubli can be found in Table 5.2:

Table 5.2: The old, modified Cubli and new Cubli's start up velocities

Cubli	Left side	Right side
Old, modified	1400 RPM	1540 RPM
New	1320 RPM	1597 RPM

The small differences in velocities for start up are presumed to be due to the Cubli being unbalanced. On average the values found by the bisection-based approach are higher than the estimate. This is due to the lack of friction in the model and the collision not being perfectly inelastic in reality. Figure 5.13 presents the final test result of the start up procedure for the old, modified Cubli for the left side. The reference value for the speed controller is set to 1400 RPM.

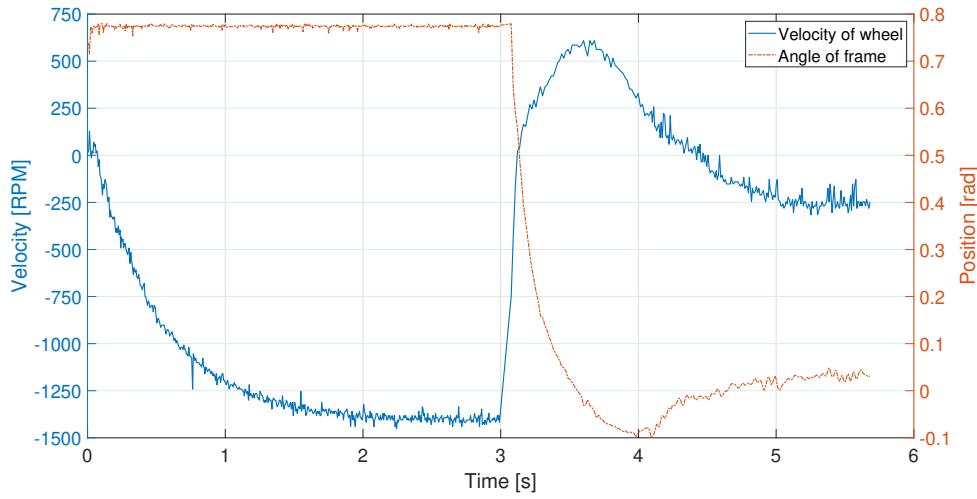


Figure 5.13: The test results for the implemented start up procedure for the old, modified Cubli for the left side. Graph depicts the angular velocity of the reaction wheel $\dot{\theta}_w$ as well as the angular position of the frame θ_f .

The overshoot of 0.1 rad is considered to be a satisfactory result for the start up procedure, as the LQR controller takes over after the brake has been released and attempts to balance the Cubli.

A more detailed report of the test can be seen in Appendix F.

5.5 Shut down

Previously, when the Cubli was set to stop balancing, the torque from the motor was set to 0 and the Cubli fell freely. To lessen the impact, two different approaches have been attempted. This section explains and compares the two methods.

5.5.1 Brake before impact

One way of lessening the impact with the ground is to accelerate the reaction wheel during the fall in the opposite direction of the fall and then applying the brake just before impact. Although spinning the wheel in that direction applies torque in the direction of fall, actuating the brake momentarily applies a force in the opposite direction of the fall. Therefore the acceleration of the frame just before the impact decreases.

The maximum velocity of the reaction wheel before applying the brake is limited to 1400 RPM. The purpose of this is to avoid producing a jumping effect for the Cubli. A speed controller, similar to the one described in subsection 5.4.2, is used with the gain being $K = 0.1$. The brake is actuated after the angular position of the frame has surpassed 0.33 rad which has been found empirically. The actuation of

the brake is delayed, which is due to the time it takes the servo to travel from one defined position to another.

5.5.2 Decelerating by applying torque

In the second method, the wheel is accelerated in the direction of the fall, thus reducing the acceleration of the frame and lessening the impact. A model for the necessary applied torque with respect to the angle of the frame is developed.

First, Newton's second law is applied to the free body diagram on Figure 5.14.

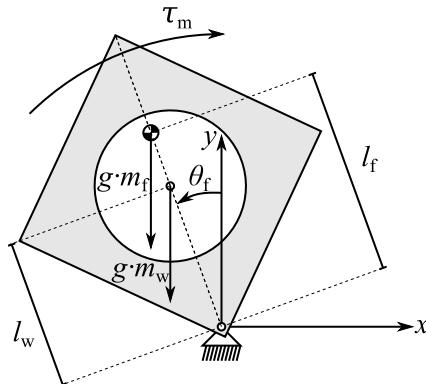


Figure 5.14: Sketch of the free body diagram used for developing the shutdown procedure. The CoM of the frame is denoted by the \bullet symbol. Note the two forces $g \cdot m_w$ and $g \cdot m_f$ together apply a positive torque called τ_g not present on the sketch.

$$\begin{aligned} J \cdot \ddot{\theta} &= \sum \tau \\ (J_f + J_w + m_w \cdot l_w^2) \cdot \ddot{\theta}_f &= -\tau_m + \tau_g \end{aligned} \quad (5.25)$$

where:

$$\tau_g : \text{Torque due to gravity.} \quad [\text{N} \cdot \text{m}]$$

Note, the movement of the wheel is ignored as well as any friction. If the acceleration should be reduced by a factor α compared to a free fall, the torque applied by the motor should be a factor α of the torque due to gravity.

$$\tau_m = \alpha \cdot \tau_g \quad (5.26)$$

The torque due to gravity can be described as:

$$\tau_g = g \cdot (m_f \cdot l_f + m_w \cdot l_w) \cdot \sin(\theta_f) \quad (5.27)$$

Inserting and simplifying

$$\tau_m = \alpha \cdot g \cdot (m_f \cdot l_f + m_w \cdot l_w) \cdot \sin(\theta_f) \quad (5.28)$$

The torque from the motor can be described by $\tau_m = K_t \cdot i_a$, and substituting in this expression:

$$K_t \cdot i_a = \alpha \cdot g \cdot (m_f \cdot l_f + m_w \cdot l_w) \cdot \sin(\theta_f) \quad (5.29)$$

$$i_a = \frac{\alpha \cdot g \cdot (m_f \cdot l_f + m_w \cdot l_w)}{K_t} \cdot \sin(\theta_f)$$

Through trial and error acceptable results for the descent were found when $\alpha = 1.0$. Note that according to the model, the Cubli would not move with this value and would maintain the angle until the reaction wheel reaches the maximum limit for velocity. In other words, if $\alpha = 1.0$, this would make $\tau_m = \tau_g$ according to Equation 5.26. However, the model is imperfect.

On Figure 5.15, test results from the two new methods, as well as the free fall can be seen.

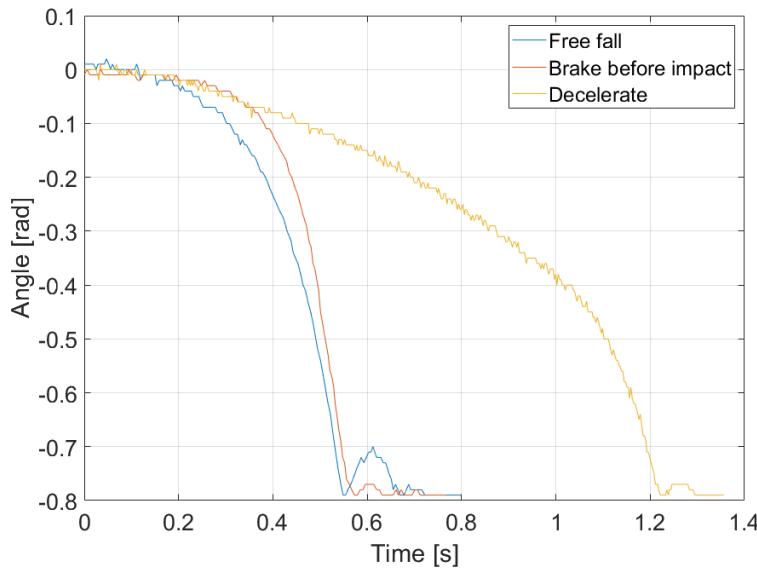


Figure 5.15: The test results for the two new methods and free fall.

It can be seen that in the case of free fall, the bounce up after impact is approximately 0.1 rad, while in the cases of braking before impact and decelerating, it is 0.03 rad. Braking before impact method is more unreliable as its performance depends notably on the initial conditions of the system. Meanwhile, the deceleration method for shut down has proven to be more robust when disturbances are

acting on the system. From these results, it has been chosen to use the deceleration method, as described in subsection 5.5.2. A more detailed description of the tests performed as well as the reasoning for choosing acceleration method over braking before impact can be seen in Appendix G.

A comparison with simulations from the non-linear model described in subsection 5.2.2 can be seen on Figure 5.16.

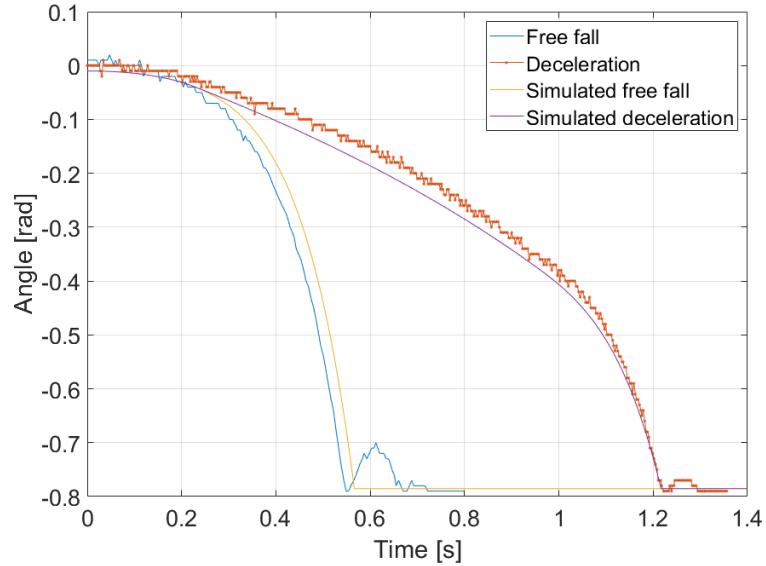


Figure 5.16: Simulations using the non-linear model comparing free fall and deceleration with the simulated responses.

As can be seen from the figure, there are minor differences between the model and experimental data. It should be noted that the simulated deceleration is started at -0.01 rad, since the 0 rad initial condition would not make the Cubli fall over in the simulation. The minor discrepancies between the simulation and experimental data might also be caused by other minor variations in the initial condition of the system.

6 Filters

Since noise from the sensors has an impact on the controller of the Cubli a filter needs to be implemented. This chapter contains the description of two different filters: a complementary filter and a Kalman filter, followed by a discussion and conclusion of which is more suitable for this project.

6.1 Complementary filter

The complementary filter in this system is the same as in [6]. In the Cubli, the position of the frame is calculated from two sensors: the accelerometer and the gyroscope. The complementary filter is able to filter the low frequencies of the gyroscope measurements and the high frequencies of accelerometer measurements, which makes the data more accurate and with less noise.

In this filter, the relation between the data measured by the accelerometer and the gyroscope with the position of the frame is:

$$\theta_f = \frac{1}{1 + \tau \cdot s} \cdot \theta_{\text{acc}} + \frac{\tau \cdot s}{1 + \tau \cdot s} \cdot \frac{1}{s} \cdot \dot{\theta}_{\text{gyro}} \quad (6.1)$$

where:

- τ : The time constant of the filter
- θ_{acc} : The position of the frame measured by the accelerometer
- $\dot{\theta}_{\text{gyro}}$: The speed of the frame measured by the gyroscope

The time domain discrete version of it is:

$$\begin{aligned} \theta_f[n] &= \frac{2 \cdot \tau - \Delta T}{2 \cdot \tau + \Delta T} \cdot \theta_f[n-1] \\ &+ \frac{\Delta T}{2 \cdot \tau + \Delta T} \cdot (\tau \cdot \dot{\theta}_{\text{gyro}}[n-1] + \dot{\theta}_{\text{gyro}}[n] + \theta_{\text{acc}}[n-1] + \theta_{\text{acc}}[n]) \end{aligned} \quad (6.2)$$

where:

- ΔT : The sample period

According to [6], the value of τ is 0.3471.

6.2 Kalman filter

This section describes the theory and design of a Kalman filter, as well as a simulation performed on data from the Cubli as a verification of the filter. Finally, the feasibility of the filter is discussed, as well as the reasoning behind why it has not been implemented.

6.2.1 Kalman filter theory

A Kalman filter is a recursive filter, operating in two steps: prediction and estimation. The principle behind the filter is to predict the system states based on the previous sample's estimated state and the applied control input. This prediction is then compared to the sensor measurements to produce the current sample's estimate. After several recursive steps, the uncertainty of the system becomes smaller, and the estimated state becomes more accurate, subject to the specifications of the Kalman filter.

The two steps of the Kalman filter are given by the following equations:

Prediction step:

$$\begin{aligned}\hat{\mathbf{x}}_k &= \mathbf{A}_d \hat{\mathbf{x}}'_{k-1} + \mathbf{B}_d \mathbf{u}_k \\ \mathbf{P}_k &= \mathbf{A}_d \mathbf{P}_{k-1} \mathbf{A}_d^T + \mathbf{Q}_k\end{aligned}\tag{6.3}$$

Estimation step:

$$\begin{aligned}\hat{\mathbf{x}}'_k &= \hat{\mathbf{x}}_k + \mathbf{K} (\mathbf{z}_k - \mathbf{C}_d \hat{\mathbf{x}}_k) \\ \mathbf{P}'_k &= \mathbf{P}_k - \mathbf{K} \mathbf{C}_d \mathbf{P}_k \\ \mathbf{K} &= \mathbf{P}_k \mathbf{C}_d^T \left(\mathbf{C}_d \mathbf{P}_k \mathbf{C}_d^T + \mathbf{R}_k \right)^{-1}\end{aligned}\tag{6.4}$$

where:

\hat{x}_k	:	The predicted state
\hat{x}'_k	:	The estimated state
\hat{x}'_{k-1}	:	The previously estimated state
z_k	:	The measurement of the sensors
u_k	:	The control signal of the system
A_d	:	The discrete system matrix
B_d	:	The discrete input matrix
C_d	:	The discrete output matrix
P_k	:	The uncertainty (covariance matrix) of the predicted state
P_{k-1}	:	The uncertainty (covariance matrix) of the previously estimated state
P'_k	:	The uncertainty (covariance matrix) of the new estimated state
Q_k	:	The covariance matrix of the process noise
K	:	The Kalman gain matrix
R_k	:	The covariance matrix of the measurement noise

In the next iteration $k + 1$, \hat{x}'_k and P'_k are fed back as \hat{x}_{k-1} and P_{k-1} in Equation 6.3. With more iterations, some of the noise from the environment and the measurements is filtered. Therefore, a more accurate state estimation is provided to the controller.

6.2.2 Kalman filter design

As seen in Equation 6.3 and 6.4 a discrete state space model is required. The continuous state space model of the Cubli is described in Equation 4.10, and the sample time of the system is 5.2 ms. According to that, the discrete system matrix A_d , the discrete input matrix B_d and the discrete output matrix C_d can be calculated by the "c2d" command in MATLAB. The result is:

$$\begin{aligned} A_d &= \begin{bmatrix} 1.0011 & 0.0052 & 0.0 \\ 0.4322 & 0.9990 & 0.0 \\ -0.4322 & 0.0010 & 0.9998 \end{bmatrix} \\ B_d &= \begin{bmatrix} -0.0016 \\ -0.6069 \\ 8.0455 \end{bmatrix} \\ C_d &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{6.5}$$

The initial conditions of the Kalman filter of this system are:

$$\begin{aligned}\hat{\mathbf{x}}_0 &= [0 \ 0 \ 0]^T \\ P_0 &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\end{aligned}\quad (6.6)$$

The covariance matrices \mathbf{Q}_k and \mathbf{R}_k are tuned with respect to observations of the system and the measurements using a MATLAB simulation and data gathered in Appendix O. It must be noted that \mathbf{Q}_k and \mathbf{R}_k are modelled as constant matrices, since the covariances of the noises are assumed to be constant. The tuned \mathbf{Q}_k and \mathbf{R}_k are presented in Equation 6.7.

$$\begin{aligned}\mathbf{Q}_k &= \begin{bmatrix} 0.00001 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 50 \end{bmatrix} \\ \mathbf{R}_k &= \begin{bmatrix} 0.01 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1000 \end{bmatrix}\end{aligned}\quad (6.7)$$

It must be noted that further tuning has to be done when implementing the Kalman filter, as the current tuning has been only performed using the MATLAB simulation. However, with this setting, the poles of $(\mathbf{A}_d - \mathbf{C}_d \mathbf{K})$ are 0.9434, 0.8000 and 0.3799 indicating the stability of the system, since all eigenvalues of the discrete system lie within the unit circle.

6.2.3 Kalman filter simulation

Figure 6.1 shows the MATLAB simulated Kalman filter. The data is obtained as described in Appendix O.

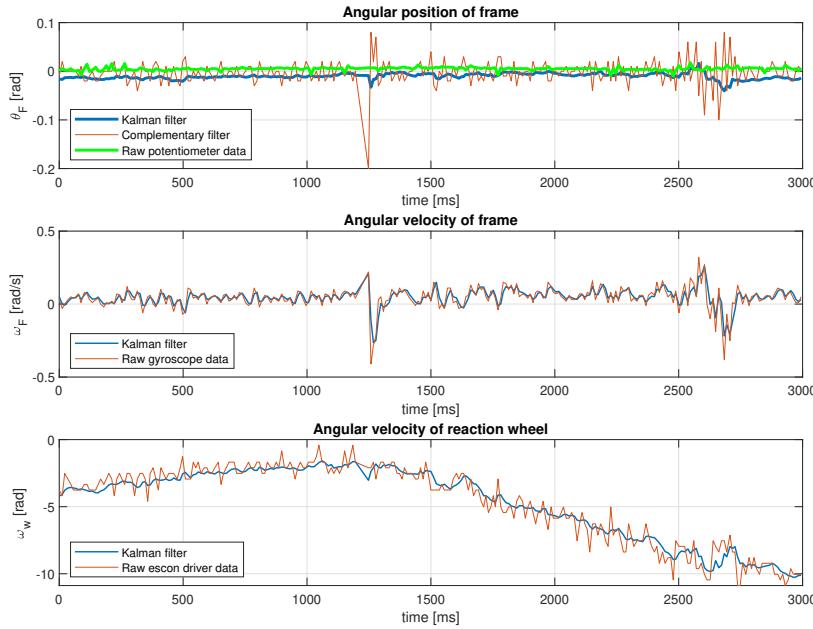


Figure 6.1: Kalman filter simulation

The top of the figure shows the angular position output from the Kalman filter, the complementary filter and the potentiometer data. The Kalman filter output is comparable to the raw potentiometer data, as opposed to the more noisy complementary filter. The bottom two graphs from Figure 6.1 show that the Kalman filter acts as a low-pass filter on the raw data for the angular velocities of the frame and the reaction wheel.

6.2.4 Kalman filter implementation

A Kalman filter Arduino function has been developed using the "BasicLinearAlgebra" library[16]. To estimate whether the computation time of the function hinders the controller, tests have been performed where the data is filtered with the complementary filter, but the Kalman filter function is also called. The controller was observed to become less responsive, although the Kalman filter estimate is not used in the controller. For this reason, the Kalman filter has not been implemented on the Cubli, since its calculation time is too long. Instead, the complementary filter implemented by [6] is used.

However, an implementation error has been discovered in the Kalman filter test program later on. It must be noted that in future development, the Kalman filter might be a suitable filter selection for the Cublis. However, the complementary filter is used in this project.

7 Communication

This chapter contains an analysis of the communication between the two Cublis to set requirements for a wireless communication system. This is done to be able to choose a suitable technology for the system, which is described in section 7.2. Moreover, it describes how the communication is implemented as a state machine. This is followed by the main results regarding synchronising the two Cublis, and finally, method and results for networked control between the two Cublis are presented.

7.1 Analysis of Cubli communication

This section presents an analysis of the communication between the Cublis. To provide room for further development, a two-way communication strategy is chosen for the project.

The sample frequency for the system in the current Cubli model is set to 192.31 Hz (5.2ms period) when utilizing the IMU for sensing. In the code, the different variables used for control are defined as floats, which have the size of 4 bytes on the Arduino Vidor micro controller [17]. Furthermore, it has been decided that a command field of 8 bit is used to send commands between the two Cublis. It is decided to transmit 4 variables in addition to the command field for control purposes. This means that the packet size for the transmitted data is 17 bytes. In Figure 7.1 the structure of the packet used is illustrated.

1 Byte	4 Byte	4 Byte	4 Byte	4 Byte
CMD	Angular Error	Speed of Frame	Speed of Wheel	Current

Figure 7.1: The first CMD byte is used to send commands between Cubli's. The next 4 fields are used for transmitting different values, but these can be used for different purposes for example control over network.

With these decisions it is possible to create a mathematical model of the communication between the two Cublis.

First, the maximum amount of data that has to be transmitted every second can be expressed by the affine linear function [18].

$$\gamma(t) = r \cdot t + b \quad (7.1)$$

$$\Downarrow \quad (7.2)$$

$$\gamma_{r,b}(t) = \frac{P \cdot t}{T} + P \cdot \left(\frac{\tau}{T} + 1 \right) \quad (7.3)$$

where:

γ	: Linear model of arrival curve of the transmitted data	[bit]
P	: Packet size	[bit]
r	: Arrival rate	[bit · s ⁻¹]
T	: Data arrival interval	[s]
b	: Burst parameter	[bit]
τ	: Delay jitter	[s]
t	: Time	[s]

The system has a data arrival interval of 5.2 ms and a packet size of 136 bit. The delay jitter is unknown, however, it is normally found experimentally because it depends on the implementation of the communication system. Substituting the known values yields:

$$\gamma_{r,b}(t) = \frac{136 \cdot t}{5.2 \cdot 10^{-3}} + 136 \cdot \left(\frac{\tau}{5.2 \cdot 10^{-3}} + 1 \right) \quad (7.4)$$

A sketch of the relationship between the affine linear curve $\gamma_{r,b}(t)$ and the data sent is illustrated in Figure 7.2.

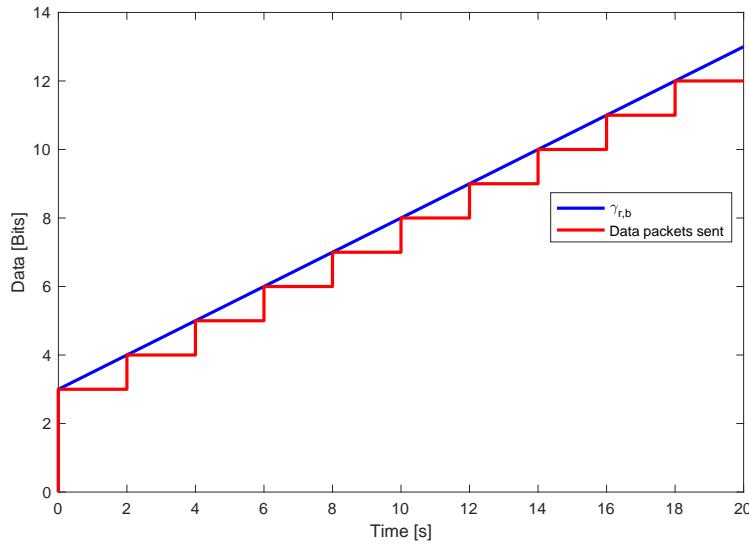


Figure 7.2: Sketch of the relation between the affine linear curve $\gamma_{r,b}(t)$ and the data sent.

Furthermore, an upperbound requirement for the network bandwidth can be defined as

$$R > r \quad (7.5)$$

Where

$$R : \text{Network bandwidth} \quad [\text{bit} \cdot \text{s}^{-1}]$$

This means that the network bandwidth required for the system, without taking headers into account, needs to be $R > 26.16 \cdot 10^3$ bit/s and burst parameter is $b > 136 \cdot (\frac{\tau}{5.2 \cdot 10^{-3}} + 1)$ for the communication modules [18, p. 13].

The delay through the network can be described by the following [18, p. 24].

$$\Delta = \frac{b}{R} + T_{\text{delay}} \quad (7.6)$$

where

$$\begin{aligned} \Delta &: \text{Network delay} & [\text{s}] \\ T_{\text{delay}} &: \text{Other delays in the system} & [\text{s}] \end{aligned}$$

T_{delay} can be caused for example by the serialisation in the system.

The requirement for the network delay is that it has to be as small as possible as this has an effect on the controller for the Cubli but also that the delay jitter has an effect on the controller. Therefore it has been decided to investigate the use of a layout buffer to reduce the effect of delay jitter in the system [18, p. 7].

Lastly the buffer requirement for the implementation is set to [18, p. 7]

$$B = 2 \cdot \Delta \cdot \frac{r}{8} \quad (7.7)$$

where

$$B : \text{Buffer size} \quad [\text{Byte}]$$

Furthermore, the rate r is divided by 8 to convert the buffer requirement unit from bits to bytes.

It is assumed that the buffer never starves and produces a constant output at rate r . Moreover, it is assumed that the network imposes variable delay and has no packet losses.

However, in reality there will be packet loss present to some degree on the wireless communication network.

A sketch of a network model with respect to Equation 7.4, Equation 7.6 and the network bandwidth can be seen on Figure 7.3.

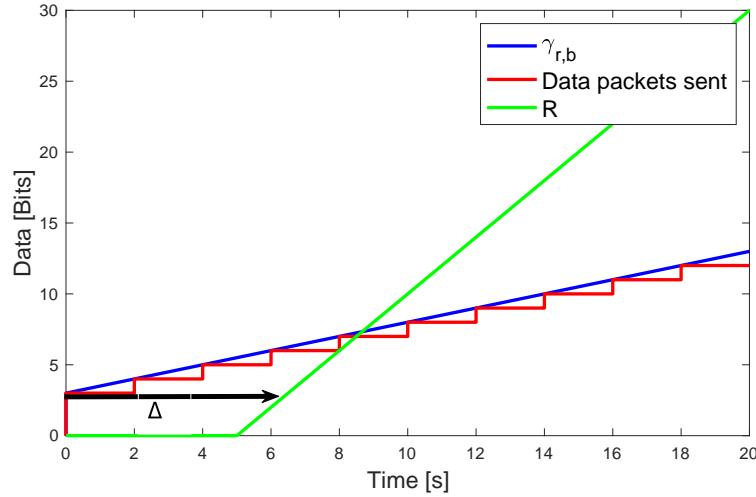


Figure 7.3: Sketch of a network model with respect to Equation 7.4, Equation 7.6 and the network bandwidth.

7.2 Technology choice

Table 7.1 shows different wireless technologies available, which can fulfill the requirements set in section 7.1.

The choice of communication technology is based on the following parameters in prioritized order:

1. Bandwidth
2. Range
3. Power usage
4. Cost

The communication technologies considered in Table 7.1 are technologies that are widely available and have a relatively low power consumption.

	Bluetooth 4.0 [19]	WiFi (802.11b/g/n) [20]	Zigbee [21]	802.15.4 [22]
Bandwidth	100 kbit/s	1 Mbit/s	250 kbit/s	250 kbit/s
Range	20 m - 30 m	<50 m	3.2 km (LoS)	1.6 km (LoS)
Transmit current	15 mA	309 mA	215 mA	120 mA
Receive current	8.5 mA	100 mA	55 mA	45 mA
Market price	13 USD	31.90 USD	24.70 USD	19.99 USD

Table 7.1: Table showing different suitable solutions for the wireless communication between Cubli's.

Even though it has a higher market price, compared to e.g. Bluetooth, Zigbee is the initial choice of communication technology. This is because the required good-put is 26.16 kBit/s, which means that the bandwidth for the chosen communication technology needs extra space for the protocol header. Furthermore, it consumes less power than other technologies, such as WiFi, making it a suitable choice for the Cubli.

XBEE S1 modules have been used for initial testing, since they are the only readily available Zigbee modules at AAU. Testing and inspection of the XBEE S1 modules and the whole range of XBEE Zigbee modules have showed that good-put of 26.16 kBit/s is not possible on the devices. Therefore, the communication technology choice is switched from Zigbee to WiFi and an XBEE S6B WiFi module has been selected instead. This is because it has the same pinout as other XBEE devices, which eliminates the need of adding extra hardware on the Cubli. Furthermore, the XBEE S6B can operate in Ad Hoc mode, which enables peer to peer communication. The XBEE S6B modules have a bandwidth of up to 320 kBit/s, which is sufficient for the 26.16 kBit/s requirement [23].

7.3 XBEE implementation

This section describes the implementation and configuration of the Zigbee modules on the Cublis. Figure 7.4 presents an implementation diagram of the XBEE modules on the Cublis.

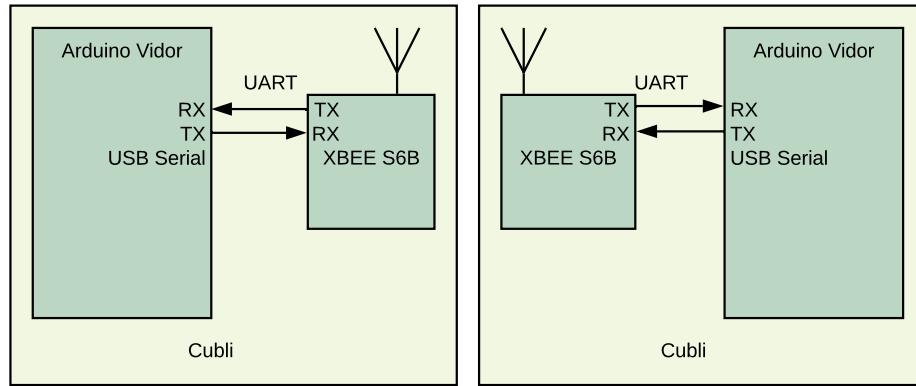


Figure 7.4: Illustration implementation of the XBEE modules for communication.

To transmit packets via the XBEE S6B modules, they have to be set up as end devices in the configuration software XCTU [24] in transparent AT mode. This means the modules work as a wireless Universal Asynchronous Receiver/Transmitter (UART) connection. The setup of the configuration of the XBEE S6B modules is described in Appendix K.

The modules have been tested with the packet size in Appendix L and Appendix M where it has been concluded that the achieved bandwidth of the implementation is 88.918 kBit/s and the average delay is 3.8 ms. This means that the modules are able to meet the bandwidth requirement set in section 7.1

7.3.1 Transmitting Arduino function

To send packets the following Arduino code has been implemented:

```

1 struct controlData_t {
2     float val1;
3     float val2;
4     float val3;
5     float val4;
6 };
7
8 typedef union ZIGBEE_Packet_t {
9     controlData_t packet;
10    uint8_t ZBPacket[sizeof(controlData_t)];
11 };
12 ZIGBEE_Packet_t txdata;
13 ZIGBEE_Packet_t rxdata;
14 #define PACKET_SIZE sizeof(txdata.ZBPacket)
15
16 unsigned long transmit_timer = 0;

```

```

17 unsigned long timer_threshold = 3000; // time after which we can ↪
18   transfer again
19
20 void transmit(uint8_t cmd, bool timer_enable) {
21   if(timer_enable)
22   {
23     if (micros()-transmit_timer > timer_threshold) {
24       transmit_timer = micros();
25       Serial1.write(cmd);
26       for (int k = 0; k < PACKET_SIZE; k++) {
27         Serial1.write(txdata.ZBPacket[k]);
28       }
29     }
30   }
31 else{
32   transmit_timer = micros();
33   Serial1.write(cmd);
34   for (int k = 0; k < PACKET_SIZE; k++) {
35     Serial1.write(txdata.ZBPacket[k]);
36   }
37 }
```

The packet is defined as a struct, which is then defined as a union together with a byte field used for transmission. The function transmit() then transmits the data over UART with a start character used for the receiver function, namely the cmd variable.

7.3.2 Receiving Arduino function

```

1 void receive() {
2   while (Serial1.available() > PACKET_SIZE ) {
3     uint8_t cmdtemp = Serial1.read();
4     if (cmdtemp == 'L' || cmdtemp == 'R' || cmdtemp == 'S' || cmdtemp ==←
5       'V' || cmdtemp == 'B' || cmdtemp == 'C' || cmdtemp == 'D') {
6       for (int k = 0; k < PACKET_SIZE; k++) {
7         rxdata.ZBPacket[k] = Serial1.read();
8       }
9       rxdatafull.cmd = cmdtemp;
10      rxdatafull.val1 = rxdata.packet.val1;
11      rxdatafull.val2 = rxdata.packet.val2;
12      rxdatafull.val3 = rxdata.packet.val3;
13      rxdatafull.val4 = rxdata.packet.val4;
14      rxbuffer.push(rxdatafull);
15      cmdtemp = 0;
16    }
17  }
```

The receive function checks the UART buffer on the Arduino MKR Vidor 4000 and if a new packet is available it stores that into the byte array rxdata.ZBPacket. Furthermore the receive() function checks for which type of packet is used with the cmdtemp variable. Then, the rxdata.packet and the command variable are stored in a buffer.

When the received packet needs to be used by the control algorithm, the following function is used:

```

1 void get_rx_data() {
2     if (rxbuffer.isEmpty() != true) { // Print 1 element from buffer.
3         tempdata = rxbuffer.shift(); // get packet from buffer.
4     }
5     while (rxbuffer.size() >= 6) {
6         tempdata = rxbuffer.shift(); // Throw packets away if buffer starts ←
7             to overflow.
8     }

```

7.3.3 Buffer implementation

The following parameters for the communication system have been found in Appendix M and Appendix L.

Parameter	Value
Delay, Δ	3.8 ms
Bandwidth, R	88.91 kBit/s
Bandwidth, r	26.16 kBit/s

Table 7.2: Parameters for communication

Since the delay is larger than 5.2 ms for 2 way communication and the buffer introduces even more delay in the system it has been decided not to use the following implementation of the buffer in the final Cubli. However, this section continues with a description of the buffer implementation, since it may be used in a future development of the project.

The required buffer size can be computed for the implementation using Equation 7.7:

$$\begin{aligned}
 B &= 2 \cdot \Delta \cdot \frac{r}{8} \\
 &= 2 \cdot 3.8 \cdot 10^{-3} \cdot \frac{68 \cdot 10^3}{8} \\
 &= 10.8
 \end{aligned} \tag{7.8}$$

This means that the required buffer size for the implementation has to be larger than 10.8 bytes. Therefore, implemented buffer has to have a size of at least 1 packet. The implementation of the buffer is shown in the following code:

```

1 // struct for buffering and accessing data:
2
3 struct controlData_full {
4     uint8_t cmd; // state of other cubli
5     float val1;
6     float val2;
7     float val3;
8     float val4;
9 };
10
11 #include <CircularBuffer.h>
12 #define BUFFER_SIZE 4
13 controlData_full tempdata;
14 controlData_full rxdatafull;
15 CircularBuffer<controlData_full, BUFFER_SIZE> rxbuffer;
```

In section 7.1 it is mentioned that the playout buffer introduces a constant delay to eliminate delay jitter. This delay can be computed by the buffer size and the sample time:

$$\begin{aligned}
 P_{\text{const}} &= B_{\text{impl}} \cdot T_{\text{samp}} \\
 &= 2 \cdot 5.2 \cdot 10^{-3} \\
 &= 10.4 \text{ ms}
 \end{aligned} \tag{7.9}$$

where:

P_{const}	: Constant delay introduced by playout buffer	[s]
B_{impl}	: Implemented buffer size	[Packet]
T_{samp}	: Implemented sample frequency	[s]

This means that the buffer introduces a delay of 3.8 ms every time the buffer size is incremented. As mentioned above, the buffer is not used in the final Arduino implementation.

7.4 Communication implementation strategy

So far this chapter described the technology choice and the implementation of a communication channel between the two Cublis. This section describes what that channel is used for in this project. In particular, insight into the choice of communication strategy is provided, as well as a description of its implementation. Lastly, examples of the start up and shut down procedures are given with respect to interaction between the two Cublis.

7.4.1 State machine motivation

For the implementation of the main objective for this project, i.e. synchronizing the start up procedure between the two Cublis, a state machine architecture has been selected. A state machine is a design structure, which organizes the output from a system based on a finite set of states. An example state machine design for the Cubli can be a simple two-state structure with states "lying down" and "balancing". The Cubli is expected to switch between the stand up procedure and the feedback controller based on which state it is.

For this project, a state machine brings benefits in structuring the transfer of information. Exchanging the Cubli states between the two Cublis facilitates the prevention of deadlocks and organizes the implementation of different scenarios, such as startup and shut down. Since the main objective of this project is to achieve synchronised start up, it is important to outline the desired procedure for that operation:

1. Motors must be off until both systems are switched on.
2. Both reaction wheels must have reached their start up velocity before applying the brake.
3. Braking must be synchronized, subject to communication delays.
4. If a Cubli falls because of disturbances (while both systems are still on), the fallen Cubli must get up.
5. If both Cublis fall due to disturbances, they must perform a synchronised start up.
6. If at least one system is turned off, both Cublis must begin their shut down procedures.

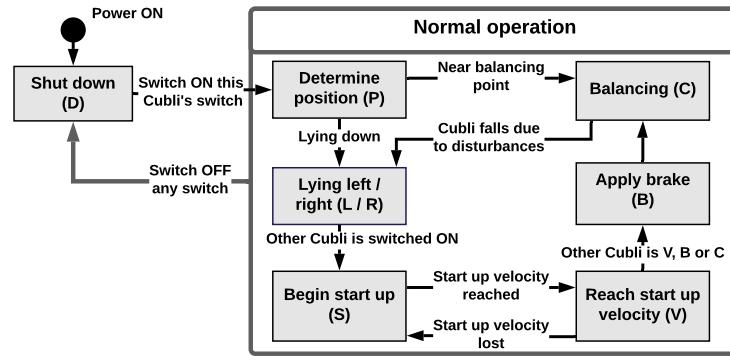


Figure 7.5: State machine structure

7.4.2 State machine implementation

In order to achieve the desired behaviour listed in subsection 7.4.1, the state machine illustrated on Figure 7.5 is implemented (link to implementation). It consists of 8 states, the initial one being 'D', meaning "Shut down", going through 'P' where the current position is determined, leading to either 'C' if the Cubli is up and balancing, or 'L' and 'R', based on the direction it is lying to. In the latter case, the state is then changed to 'S' when the motor is started and reaches state 'V' when the velocity of the reaction wheel is sufficient for braking. Next is the state 'B' in which the Cubli brakes and stabilises, followed by the last state - 'C', which indicates that the Cubli is up and balancing. At any point, if any of the systems is turned off, the state escapes to the shut down state 'D'. However, if the cubli falls down due to outside disturbances, the state is changed to 'L' or 'R'.

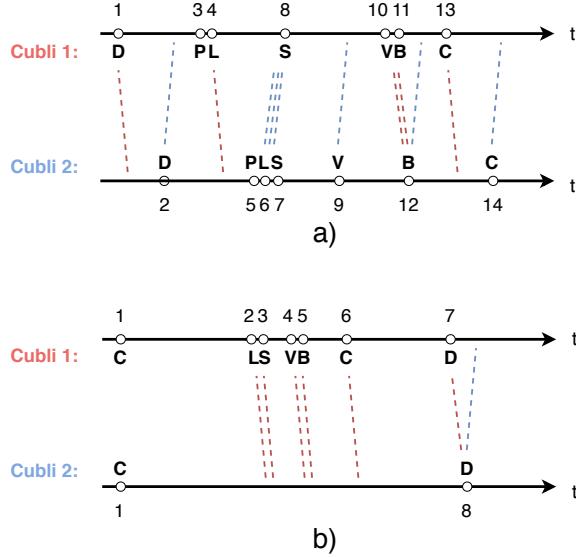


Figure 7.6: Example timelines for a) Start up and b) Shut down

Start up: Figure 7.6a shows a possible scenario during start up, illustrated by the development of the two Cubli states through time. At times 1 and 2 the two Cublis are powered on. At time 3 and 5, the switches respectively on Cubli 1 and 2 are switched ON. At time 5, Cubli 2 is already aware that Cubli 1 is not in state 'D', so it switches to state 'S' directly after achieving states 'L' and 'P'. At time 8, Cubli 1 receives information that the other Cubli is not in state 'D' anymore, so it reaches state 'S' too, resulting in both motors being started. At time 9, Cubli 2 has reached the required velocity but does not brake until time 12, when it has found out that Cubli 1 has also reached its start up velocity. Just before that, subject to communication delays, Cubli 1 has applied the brake since it has already received the Cubli 2 state 'V'. This results in a synchronised start up model with a delay of ≈ 3.8 ms as described in section 7.3.

Shut down: Figure 7.6b represents a Cubli falling due to outside disturbances (Cubli 1), then getting up and balancing again, followed by a synchronised shut down. In particular, at time 2, Cubli 1 reaches state "L", meaning that it has fallen due to some disturbances. The graph shows that Cubli 1 goes through the start up procedure, while Cubli 2 continues to balance, since both switches are still turned on, as described in subsection 7.4.1. After time 6, both Cublis are balancing again, indicated by both their states being "C". At time 7, the switch on Cubli 1 is switched off, leading to a change in state to "D", which is then received at time 8 on the second Cubli. This means that both Cublis begin their shut down procedures with a delay, equal to the start up delay, i.e. ≈ 3.8 ms. It is important to note that synchronising the instance of time at which both Cublis contact the ground is not

in the scope of this project.

7.5 Testing Synchronised Motion

The synchronisation of the motion of the two Cublis is examined by testing the synchronised start up procedure.

To allow easier measurement of the delay between the two Cublis, an LED is enabled on the Cubli when it goes into state 'B', i.e. it applies the brake.

Videos are then recorded at 480 FPS of the two Cublis during the synchronised start up procedure. These videos are then analysed by finding the number of frames between the two LEDs turning on. This can then be converted to a maximum delay using the frame rate of the video. In Table 7.3 the test results can be seen.

Table 7.3: Measurement of the delay between two Cublis.

Test #	1	2	3	4	5	6	7	8	9	10
Frames	6	12	84	2	4	4	6	5	5	5
Max. delay [ms]	12	24	175	4	8	8	12	10	10	10

These values are then compared to a lower bound for a noticeable delay as described in Appendix J, which can be seen on Figure 7.7.

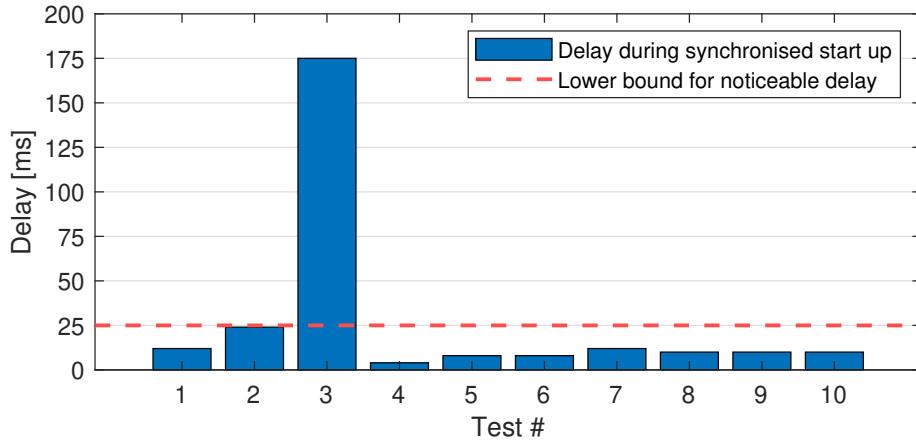


Figure 7.7: Comparison of the maximum delay

As can be seen, only the 3rd test has a higher delay than the lower bound for noticeable delay. The large delay for this is assumed to be caused by the delay spikes on the communication network as described in Appendix L.

7.6 Networked Control

To further test the communication setup, networked control is attempted to be implemented between the two Cublis. During network control, the communication network is used in the control loop.

After start up, the Cublis stabilise using the regular control scheme. Then the Cublis begin calculating each other's motor torque as seen in Figure 7.8. One Cubli is transmitting its measurement data to the other Cubli, which is then computing the control input of the first Cubli. It transmits it back so it can be applied to the first Cubli.

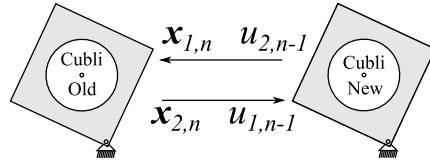


Figure 7.8: Sketch of the networked control. The old Cubli has subscript 2 and the new Cubli has the subscript 1.

However, when this is implemented on the two Cublis the delay between input and output becomes too large and the system becomes unstable (test results in Appendix N). In Figure 7.9 results from one of the Cublis during networked control can be seen.

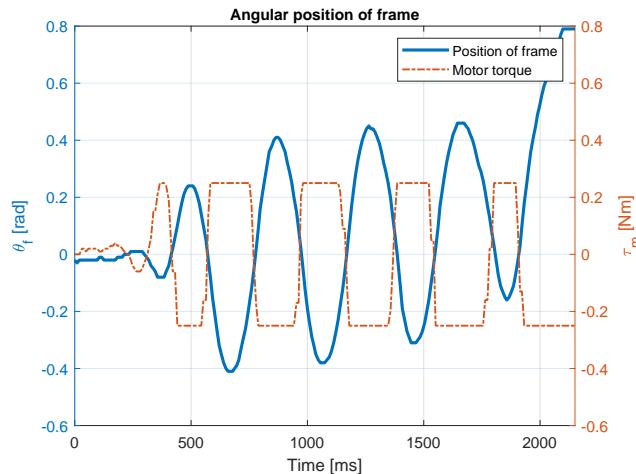


Figure 7.9: Unstable response from a Cubli during networked control.

Usually a Smith predictor can be added to a control loop to counteract the effects of delays on the system. However, the Cubli is unstable in its open-loop configuration, as seen in Equation 4.22, and thus a Smith predictor can not be used [25, p. 273].

7.7 Consideration of other communication modules

The previous section describes the implemented communication, and the process of designing it. However since the XBEE S6B modules introduced delay spikes (see Appendix L), it was considered to attempt using other modules for the communication. The modules were tested using the same code as for the XBEE S6B, to verify if they would be able to pass the requirements of being fast enough.

7.7.1 HM-10 BLE modules

The HM-10 is a low cost Bluetooth Low-Energy module that works through a UART connection to the microcontroller, seen in Figure 7.10. This module is setup using AT commands, which is the most common command set for configuring communication devices. The HM-10 module has a bandwidth of 6 KB/s. However, when tested for a roundtrip delay it showed that the delay is approximately 56 ms. This is more than a factor of 10 from the roundtrip requirement of 5.2 ms determined in section 5.2. This module was therefore not considered suitable for this project. However, the delay spikes that can be seen on Figure L.2 from Appendix L, are not present when using this module. This suggests that it is the XBEE S6B module itself that is the reason for the occurring delays.

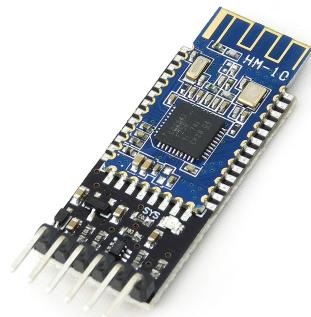


Figure 7.10: HM-10 module [26].

7.7.2 nRF24L01+ modules

Since the HM-10 modules were too slow, the nRF24L01+ RF modules were considered instead. These modules were tested with two Arduino UNOs, and a code for delay approximation. It showed that the roundtrip delay was approximately 3 ms. This delay is below the requirement of 5.2 ms, and therefore these modules would be sufficient for this project. After this test was performed, an implementation of the modules were performed on the Arduino MKR Vidor 4000. This seemed to be a problem since the modules communicate by Serial Peripheral Interface (SPI). The SPI communication of the Vidor 4000 would not allow the Master in Slave out

(MISO) connection to be opened, and the connection to the module could not be established. Therefore, the modules were not considered suitable for this project. The nRF24L01+ module itself can be seen in Figure 7.11.

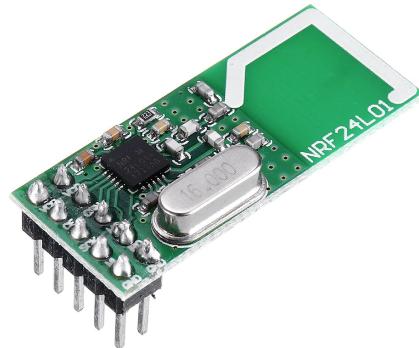


Figure 7.11: nRF24L01+ module [27].

Glossary

CoM Centre of Mass. 7, 8, 9, 10, 43, 48, 89

IMU Inertial Measurement Unit. 4, 5, 40, 57

LoS Line of Sight. 61

LQR Linear–Quadratic Regulator. 3, 4, 23, 31, 36, 38, 39, 40

SISO Single Input, Single Output. 23, 26, 27

UART Universal Asynchronous Receiver/Transmitter. 62, 63

Bibliography

- [1] M. Gajamohan et al. "The Cubli: A cube that can jump up and balance". In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2012, pp. 3722–3727. doi: 10.1109/IROS.2012.6385896.
- [2] M. Gajamohan et al. "The Cubli: A reaction wheel based 3D inverted pendulum". In: *2013 European Control Conference (ECC)*. 2013, pp. 268–274. doi: 10.23919/ECC.2013.6669562.
- [3] M. Muehlebach, G. Mohanarajah, and R. D'Andrea. "Nonlinear analysis and control of a reaction wheel-based 3D inverted pendulum". In: *52nd IEEE Conference on Decision and Control*. 2013, pp. 1283–1288. doi: 10.1109/CDC.2013.6760059.
- [4] M. Muehlebach and R. D'Andrea. "Nonlinear Analysis and Control of a Reaction-Wheel-Based 3-D Inverted Pendulum". In: *IEEE Transactions on Control Systems Technology* 25.1 (2017), pp. 235–246. doi: 10.1109/TCST.2016.2549266.
- [5] John Skøtte Knudsen, Lukas Røjkjær-Jensen, and Mads Jacob Riisager. *Cubli - self balancing cube*. 2019.
- [6] John Skøtte Knudsen et al. *AAU3 - self balancing cube*. 2019.
- [7] Herbert Goldstein, Charles Poole, and John Safko. *Classical Mechanics*. eng. Third edition. Addison Wesley, 2002. ISBN: 9780201657029.
- [8] Singiresu S. Rao. *Mechanical Vibrations*. eng. Fifth edition. Prentice Hall, 2011. ISBN: 9780132128193.
- [9] Gene F. Franklin, J. David Powell, and Abbas Emami-Naeini. *Feedback control of dynamic systems*. 7th ed. Pearson, 2015. ISBN: 9780133496598.
- [10] Morten Haack Knudsen. *Experimental modelling of dynamic systems*. 2004.
- [11] Sami Fadali and Antonio Visioli. *Digital Control Engineering*. Jan. 2012. doi: 10.1016/B978-0-12-374498-2.X0001-X.
- [12] Mohamed Essahafi. "Evaluation Performance of PID, LQR, Pole Placement Controllers for Heat Exchanger". In: *International Journal of Electrical and Computer Engineering* 08 (July 2014).
- [13] *Linear state-space control systems*. eng. Hoboken, N.J: John Wiley and Sons. ISBN: 9780470117873.

- [14] João P. Hespanha. *Linear Systems Theory: Second Edition*. NED - New edition, 2. Princeton University Press, 2018. ISBN: 9780691179575.
- [15] Christopher Lum. *Introduction to Linear Quadratic Regulator (LQR) Control*. <https://youtu.be/wEevt2a4SKI>. [Visited 2020-09-20]. Video lecture, Dec. 2018.
- [16] tomstewart89. *tomstewart89/BasicLinearAlgebra*. <https://github.com/tomstewart89/BasicLinearAlgebra>. [Visited 21-11-2020].
- [17] Arduino. *Arduino MKR VIDOR*. <https://store.arduino.cc/arduino-mkr-vidor-4000>. [Visited 9-10-2020]. 2020.
- [18] Patrick Thiran Jean-Yves le Boudec. *NETWORK CALCULUS A Theory of Deterministic Queuing Systems for the Internet*. 2020.
- [19] seeedstudio. *CSR HM-11 Module*. <https://www.seeedstudio.com/Bluetooth-V4-0-HM-11-BLE-Module-p-1803.html>. [Visited 28-09-2020]. 2017.
- [20] Digi. *Digi XBee Wi-Fi (S6B)*. <https://www.digi.com/products/embedded-systems/digi-xbee/rf-modules/2-4-ghz-rf-modules/xbee-wi-fi#specifications>. [Visited 28-09-2020]. 2017.
- [21] Digi. *Digi XBee-PRO S2C Zigbee*. <https://www.digi.com/products/embedded-systems/digi-xbee/rf-modules/2-4-ghz-rf-modules/xbee-zigbee#specifications>. [Visited 28-09-2020]. 2017.
- [22] Digi. *Digi XBee-PRO S2C 802.15.4*. <https://www.digi.com/products/embedded-systems/digi-xbee/rf-modules/2-4-ghz-rf-modules/xbee-802-15-4#specifications>. [Visited 28-09-2020]. 2020.
- [23] DIGI. *XBee S6B WIFI*. <https://www.digi.com/resources/documentation/digidocs/PDFs/90002180.pdf>. [Visited 25-10-2020]. 2020.
- [24] DIGI. *XCTU*. <https://www.digi.com/products/embedded-systems/digi-xbee/digi-xbee-tools/xctu>. [Visited 25-10-2020]. 2020.
- [25] K.J. Åström and T. Hägglund. *Advanced PID Control*. ISA-The Instrumentation, Systems, and Automation Society, 2006. ISBN: 9781556179426.
- [26] Nettigo. *HM-10 BLE*. <https://nettigo.eu/products/bluetooth-module-hm-10-4-0-ble>. [Visited 13-12-2020]. 2020.
- [27] Gearbest. *nRF24L01+*. https://www.gearbest.com/other-consumer-electronics/pp_3006145457952474.html. [Visited 13-12-2020]. 2020.
- [28] Arduino. *Arduino MKR Vidor 4000*. <https://store.arduino.cc/arduino-mkr-vidor-4000>. [Visited 27-10-2020]. 2020.

- [29] elfadistrelec. *Arduino MKR Vidor 4000*. https://www.elfadistrelec.dk/Web/WebShopImages/landscape_large/1-/01/Arduino_MKR_Vidor4000_30117101-01.jpg. [Visited 27-10-2020]. 2020.
- [30] Maxon. *Escon 50/5*. https://www.maxongroup.com/medias/sys_master/8830273519646.pdf. [Visited 27-10-2020]. 2020.
- [31] RS. *Maxon 251601 EC motor*. https://media.rs-online.com/t_large/F0495063-03.jpg. [Visited 27-10-2020]. 2020.
- [32] Digikey. *6538S-1-103*. <https://docs.rs-online.com/9cd4/A70000006422144.pdf>. [Visited 03-11-2020]. 2020.
- [33] invensense. *MPU-6050*. <https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf>. [Visited 03-11-2020]. 2020.
- [34] Sparkfun. *MPU-6050*. <https://www.sparkfun.com/products/11028>. [Visited 03-11-2020]. 2020.
- [35] DIGI. *XBEE S6B*. https://www.digi.com/resources/library/data-sheets/ds_xbeewifis6b. [Visited 03-11-2020]. 2020.
- [36] cutedigi. *XBEE S6B*. https://cdn10.bigcommerce.com/s-rs1s2e/products/429/images/874/XB2B_WFPS_001__09668.1399527428.1280.1280.jpg?c=2. [Visited 03-11-2020]. 2020.
- [37] horizonhobby. *Hitec HS-5245MG*. https://www.horizonhobby.com/on/demandware.static/Sites-horizon-us-Site/Sites-horizon-master/default/Manuals/HRC35245S-Spec_Sheet.pdf. [Visited 03-11-2020]. 2020.
- [38] t9hobbysport. *Hitec HS-5245MG*. https://www.t9hobbysport.com/_app_/resources/images/www.t9hobbysport.com/main/hitec/5245MG.jpg. [Visited 03-11-2020]. 2020.
- [39] Maxon Motors. *Maxon Motor 251601 Datasheet*. https://www.maxongroup.com/medias/sys_master/root/8833813119006/19-EN-263.pdf. [Visited 21-09-2020]. 2019.

A Hardware description

The Cubli itself consists of a aluminium frame and a aluminium plate in the bottom. The frame and the bottom plate are connected by a bearing so the frame can rotate on its corner. To make the frame balance on it's corner some different hardware must be applied. This hardware can be seen on Fig. A.1 and A.2. The hardware will be described in depth in this appendix as well.

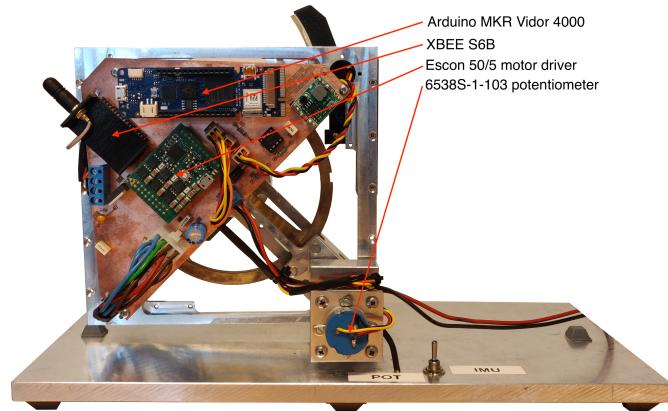


Figure A.1: Cubli front

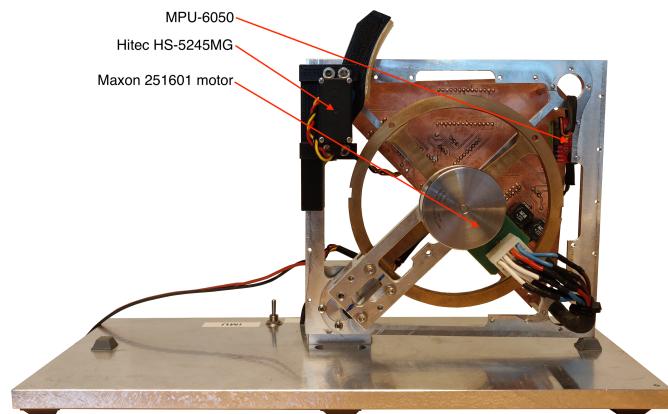


Figure A.2: Cubli back

A.1 Arduino MKR Vidor 4000

The computational unit for this project is a Arduino MKR Vidor 4000. The Vidor is chosen because it consists of a microprocessor (MCU) in the form of an Arm® Cortex®-M0 32-bit SAMD21 chip. This chip consists of 256 KB internal memory, 32 KB of SRAM and a CPU clock frequency of maximum 48 MHz.

The Vidor also consists of an FPGA chip which in this project functions as a PWM generator. The PWM frequency used for the motor driver is a 5 KHz signal. The reason for using an FPGA for generating this, is that the PWM signals produced by most other microcontrollers at this frequency is not as precise. The FPGA can do this due to the fact that the FPGA can execute more processes parallel where as most other microcontrollers are doing everything sequentially. By using the parallel executing process of the FPGA it is possible to make a clean PWM signal. Because the ARM microcontroller and the FPGA is using the internal clock of 48 MHz, the computation time is very small which makes it suitable for both delivering a PWM signal from the FPGA while also keeping up with the calculations from the ARM microcontroller.[28] A picture of the Arduino MKR Vidor 4000 can be seen in Fig. A.3.



Figure A.3: Arduino MKR Vidor 4000 [29]

A.2 Escon 50/5 and Maxon 251601 motor

For moving the reaction wheel, the need for a powerful and precise motor is needed. For this, an EC motor and a motor driver from the Swiss electronic manufacturer Maxon was chosen. This is mainly because the original Cubli project from ETH Zürich use this motor and motor driver for their setup. The motor itself is a brushless Maxon 251601 motor (see Fig. A.4), that can run with a maximum power of 50 W. This is when it is running at it's maximum speed of 6000 RPM at 24 V with a continuous current of 2.33 A. For controlling this motor, the motor driver Escon 50/5 is used (see Fig. A.5). This motor driver is programmable, to

make it possible for use with great variety of EC and DC motors. This also makes it possible to define exactly, what the maximum ratings of the EC motor for this project should be. To control this driver a PWM signal needs to be defined. When defining the duty cycle of the PWM signal, the duty cycle in the interval between 0% and 50% can control the speed of the motor in one direction, while the interval between 50% and 100% can control the speed in the opposite direction. The motor driver can handle a maximum current of 15 A, and a maximum voltage of 56 V. By those limitations it is more than enough to drive the EC motor for this project.[30]



Figure A.4: Maxon 251601 EC motor [31]

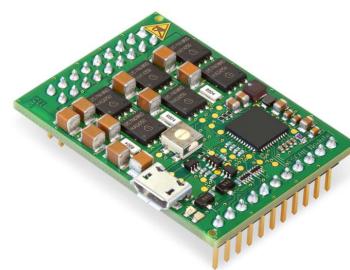


Figure A.5: Escon 50/5 motor driver [30]

A.3 6538S-1-103 potentiometer

For the Cubli a potentiometer is used for measuring the angle of the frame. Because this is the case the potentiometer only have to use 90 degrees of it's resolution. Since it's such a small area, a 10K 6538S-1-103 precision potentiometer is used. This has a output smoothness of 0.1%. The output smoothness is a percentage of spurious variation on the output, in comparison to the voltage input. Since only 90 degrees of the potentiometer is used, the output voltage needs to be scaled so that the 90 degrees go from 0 V to 3.3 V. This scale is needed for the Analog to Digital Converter (ADC) of the Arduino, so it will be able to scale it from 0 to 1024. To do so a LM358 operational amplifier is used. The potentiometer is mainly used for making precision tests, and to find out whether the Cubli's frame is lying down or standing up.[32] A picture of the 6538S-1-103 potentiometer can be seen in Fig. A.6.



Figure A.6: 6538S-1-103 potentiometer [32]

A.4 MPU-6050

An Inertial Measurement Unit (IMU) is attached to the frame of the Cubli. This device consists of an accelerometer to measure the acceleration of the frame and a gyroscope to measure the rotation of the frame. In this case the MPU-6050 IMU has been chosen, as it is a high precision IMU that is easy to implement. The MPU-6050 is the world's first IMU that includes a 3-axis accelerometer and a 3-axis gyroscope all in one, which makes it compact so it doesn't take up much space on the Cubli. The IMU is connected by a I²C connection to the Arduino, and with this connection the maximum bus speed is 400 kHz. This speed is suitable enough for the system to get enough readings, and keep it stable.[33] A picture of the MPU-6050 can be seen in Fig. A.7.



Figure A.7: MPU-6050 module [34]

A.5 XBEE S6B

To make it possible for the two Cublis to communicate wirelessly, a XBEE S6B module is applied to each of the two Cublis. Those module can transfer between 1 Mb/s to 72 Mb/s over the air. The modules works by using the 802.11b/g/n wireless standard at a 2.4 GHz band. This means that the modules works as Wi-Fi connections. It is therefore possible to configure the modules to connect to Wi-Fi routers, be a router themselves and even connect to each other as a Ad Hoc network. In this project the modules will use the Ad Hoc approach. The communication between the Arduinos and the modules are done over a serial UART connection which limits the maximum throughput of data to 320 Kb/s. In App. M the bandwidth for the system was found to be 88.918 Kb/s, which makes the connection of 320 Kb/s more than suitable for this project. [35] A picture of this XBEE module can be seen in Fig. A.8.



Figure A.8: XBEE S6B SMT module [36]

A.6 Hitec HS-5245MG

To make it possible for the Cubli's reaction wheel to brake, a brake in the form of a servo motor has been applied. This servo motor has been chosen as the Hitec HS-5245MG, because it is a high speed precision servo with metal gears. Those specifications are necessary to be able to break the reaction wheel rapidly, without breaking the servo itself. The servo operates at 5 V where it can deliver approximately 0.49 Nm. This has been tested to be suitable for the servo to brake the reaction wheel fast enough, for the Cubli to get to a standup position.[37] A picture of the servo can be seen in Fig. A.9.



Figure A.9: Hitec HS-5245MG servo [38]

B Estimation of friction coefficient for the reaction wheel

Response bearers and test facilities

Test report made by	Group 733
Test done by	Group 733
Test facilities	B3-209
Address	Fredrik Bajers Vej 7, 9220 Aalborg East, Denmark

Journal information

Unique journal number	JN01	
Relevant dates	Date of initiation: 17.09.2020	Date of completion: 17.09.2020
Number of pages in journal	3	

B.1 Test equipment

Manufacturer	Function	Model no.	Location	AAU-I8-no.
Maxon Group	Motor Driver Software	ESCON Studio 2.2	N/A	N/A
AAU	Cubli	N/A	N/A	N/A

B.2 Purpose of the test

The purpose of the test is to determine the viscous friction coefficient between the joint of the frame and the reaction wheel.

B.3 Test setup

The frame of the Cubli was secured to a steady position. After the wheel had achieved steady state with a constant velocity, sets of approximately 100 current and velocity measurements were taken. The motor control, as well as the measurements, were done/taken by Maxon driver ESCON 50/5 and using manufacturer's software ESCON Studio.

B.4 Measure procedure

Equation 3.5 was used to calculate the torque from the motor τ_m and Equation 3.4 to calculate the viscous friction b_m .

The torque constant K_t for the motor Maxon 251601 was found in the manufacturer's datasheet[39]:

$$K_t = 33.5 \cdot 10^{-3} \text{ N} \cdot \text{m} \cdot \text{A}^{-1} \quad (\text{B.1})$$

The test was repeated with 500 RPM intervals and results can be seen in Table B.1.

Table B.1: Maxon motor 251601 friction test results.

RPM	i_a	ω_m	τ_m	b_m
RPM	A	$\text{rad} \cdot \text{s}^{-1}$	$\text{N} \cdot \text{m}$	$\text{N} \cdot \text{m} \cdot \text{s} \cdot \text{rad}^{-1}$
500	0.080	52.4	$2.67 \cdot 10^{-3}$	$5.09 \cdot 10^{-5}$
1000	0.105	104.8	$3.52 \cdot 10^{-3}$	$3.35 \cdot 10^{-5}$
1500	0.148	157.1	$4.95 \cdot 10^{-3}$	$3.15 \cdot 10^{-5}$
2000	0.156	209.6	$5.23 \cdot 10^{-3}$	$2.49 \cdot 10^{-5}$
2500	0.159	262.0	$5.33 \cdot 10^{-3}$	$2.04 \cdot 10^{-5}$
3000	0.239	314.3	$7.99 \cdot 10^{-3}$	$2.54 \cdot 10^{-5}$
3500	0.235	366.7	$7.86 \cdot 10^{-3}$	$2.14 \cdot 10^{-5}$
4000	0.309	419.2	$1.04 \cdot 10^{-2}$	$2.47 \cdot 10^{-5}$

B.5 Results and conclusion

Using linear regression modelling, the relationship between τ_m and ω_m can be described by the Equation B.2, with measures $R^2 = 0.933$ and $p_{\text{value}} = 0.041$. Thus, it can be concluded there is a good fit between the model and the results. The results can be seen on Figure B.1.

$$\tau_m = 1.93 \cdot 10^{-5} \cdot \omega_m + 1.45 \cdot 10^{-3} \quad (\text{B.2})$$

It can be concluded, that the relationship between the motor torque and the angular velocity is approximated to be linear. The friction coefficient of the motor, which can now be viewed as a constant, equals the slope of the Equation B.2 and can be found by differentiation.

$$b_m = 1.93 \cdot 10^{-5} \text{ N} \cdot \text{m} \cdot \text{s} \cdot \text{rad}^{-1} \quad (\text{B.3})$$

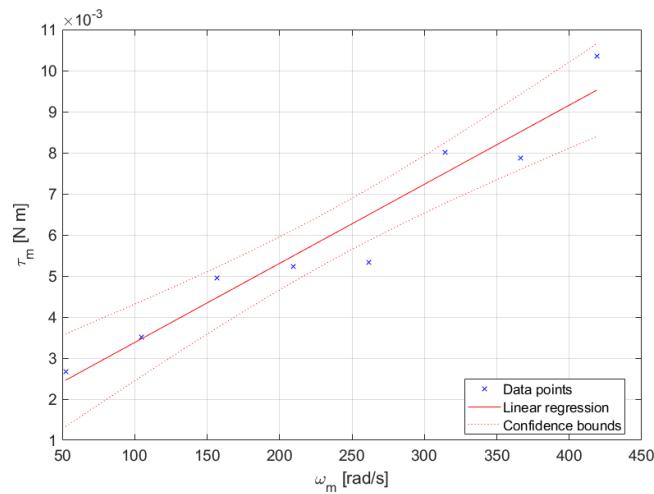


Figure B.1: Linear regression of the motor torque and the angular velocity of the reaction wheel.

Compared to the previous AAU bachelor project [5], the friction coefficient in this report is approximately a factor of 10 bigger, because in the previous project's linear regression modeling the angular velocity ω_m was in RPM, not in $\frac{\text{rad}}{\text{s}}$.

C Center of mass of the frame

Response bearers and test facilities

Test report made by	Group 733
Test done by	Group 733
Test facilities	B3-209
Address	Fredrik Bajers Vej 7, 9220 Aalborg East, Denmark

Journal information

Unique journal number	JN02	
Relevant dates	Date of initiation: 21.09.2020	Date of completion: 22.09.2020
Number of pages in journal	2	

C.1 Test equipment

Manufacturer	Function	Model no.	Location	AAU-I8-no.
AAU	Cubli	N/A	N/A	N/A

C.2 Purpose of the test

The purpose of this test is to find the center of mass of the frame in order to determine l_f which is the distance from the center of mass of the frame to the origin of the coordinate system.

C.3 Test setup

The Cubli is hung upside down with enough room to swing freely and the base plate of the Cubli is leveled using a spirit level. The Cubli is moved and allowed to swing until it comes to a standstill. After taking the angle measurements with respect to the vertical, the Cubli is detached from its' origin and turned 90°in order to take the angle measurements from the adjacent vertex.

This will give the angles for a triangle as can be seen on Figure C.1. The distance between the two vertices is 0.13 m.

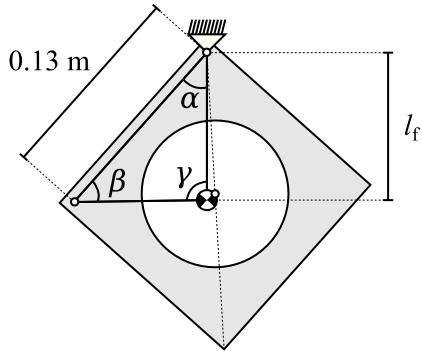


Figure C.1: Sketch of the Cubli hung upside down with the triangle, which is used for finding the length from the origin to the CoM. The CoM of the frame is denoted by the \bullet symbol.

C.4 Measure procedure

The angular position measurements are taken from the potentiometer of the Cubli by printing the values on the serial port of the Arduino.

C.5 Results and conclusion

Table C.1 shows the results for the measured angles α and β .

Angle	Measured value
α	41.69°
β	47.49°

Table C.1: Measured values for angles α and β

Using the sum of angles of a triangle formula in order to calculate the missing angle:

$$180^\circ = \alpha + \beta + \gamma \Rightarrow \gamma = 90.82^\circ \quad (\text{C.1})$$

The law of sines can be used to solve the scalene triangle:

$$\frac{\sin(\beta)}{B} = \frac{\sin(\gamma)}{C} \quad (\text{C.2})$$

In the case of this experiment the distance between the two vertices is $C = 0.13$ m and $B = l_f$. Now l_f can be calculated:

$$l_f = \frac{C \cdot \sin(\beta)}{\sin(\gamma)} = 0.0958 \text{ m} \quad (\text{C.3})$$

D Estimation of friction coefficient and inertia of the frame

Response bearers and test facilities

Test report made by	Group 733
Test done by	Group 733
Test facilities	B3-209
Address	Fredrik Bajers Vej 7, 9220 Aalborg East, Denmark

Journal information

Unique journal number	JN03	
Relevant dates	Date of initiation: 21.09.2020	Date of completion: 21.09.2020
Number of pages in journal	2	

D.1 Test equipment

Manufacturer	Function	Model no.	Location	AAU-I8-no.
AAU	Cubli	N/A	N/A	N/A

D.2 Purpose of the test

The purpose of this test is to determine the friction coefficient of the bearing between the frame and the base plate, as well as the moment of inertia of the frame itself.

D.3 Test setup

The Cubli is hung upside down with enough room to swing freely. The base plate of the Cubli is leveled using a spirit level.

The Cubli is then raised up to one of its extreme positions.

D.4 Measure procedure

The angular positional measurements from the potentiometer of the Cubli are started and the Cubli is released allowing it to swing freely like a pendulum. The positional measurements are continued until it comes to a halt.

D.5 Results

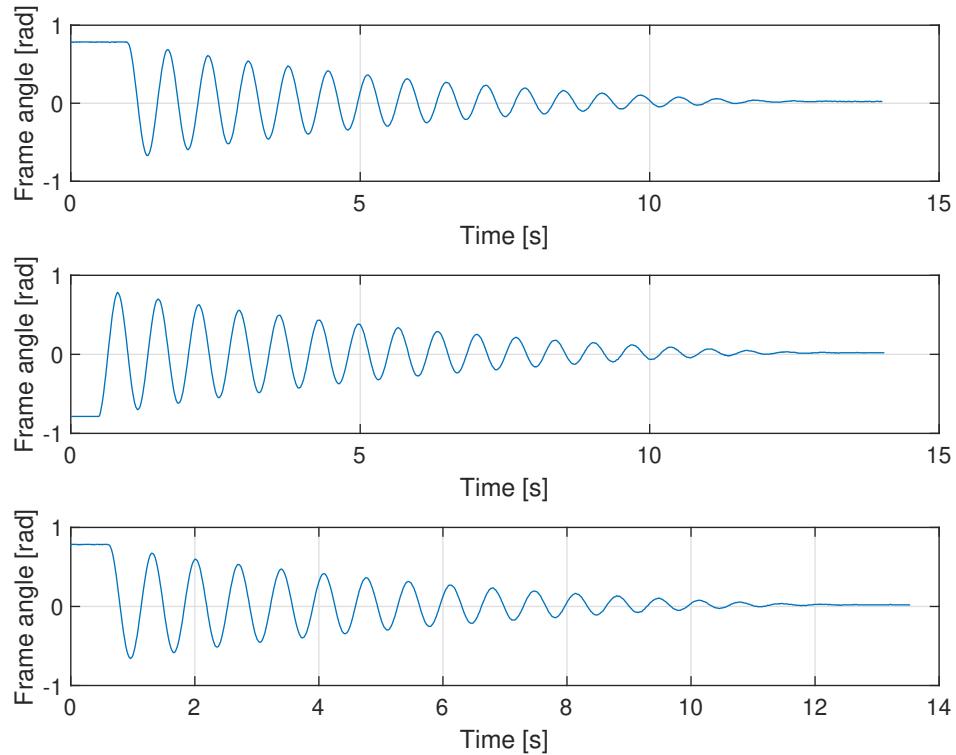


Figure D.1: Test results from the three runs.

E LQR controller tuning

Response bearers and test facilities

Test report made by	Group 733
Test done by	Group 733
Test facilities	B3-209
Address	Fredrik Bajers Vej 7, 9220 Aalborg East, Denmark

Journal information

Unique journal number	JN04	
Relevant dates	Date of initiation: 19.10.2020	Date of completion: 20.10.2020
Number of pages in journal	2	

E.1 Test equipment

Manufacturer	Function	Model no.	Location	AAU-I8-no.
AAU	Cubli	N/A	N/A	N/A
Group 733	Block of wood with $\approx 10^\circ$ angle	N/A	N/A	N/A

E.2 Purpose of the test

The purpose of this test is to determine the transient response of the system for different controllers. The controllers under observation are the controller designed by [6], using Bryson's Rule as described in subsection 5.2.1, and the tuned controller specified in subsection 5.2.2.

E.3 Test setup

The Cubli is placed at an angle of $\approx 10^\circ$ by leaning it against an angled piece of wood.

The Cubli is then turned on and due to the non-zero initial angle, the designed controller will attempt to reach the balancing point. As the Cubli is not in its resting position, it will not attempt the start up procedure.

E.4 Measure procedure

The angular position θ_f measurement is taken from the IMU of the Cubli. The Arduino code is modified by adding printing functions to print the measured values on the serial port of the Arduino.

E.5 Results and conclusion

The results of the tests are illustrated on Figure E.1. As can be seen from the figure, the tuned controller has a faster response and a smaller overshoot than the other two controllers.

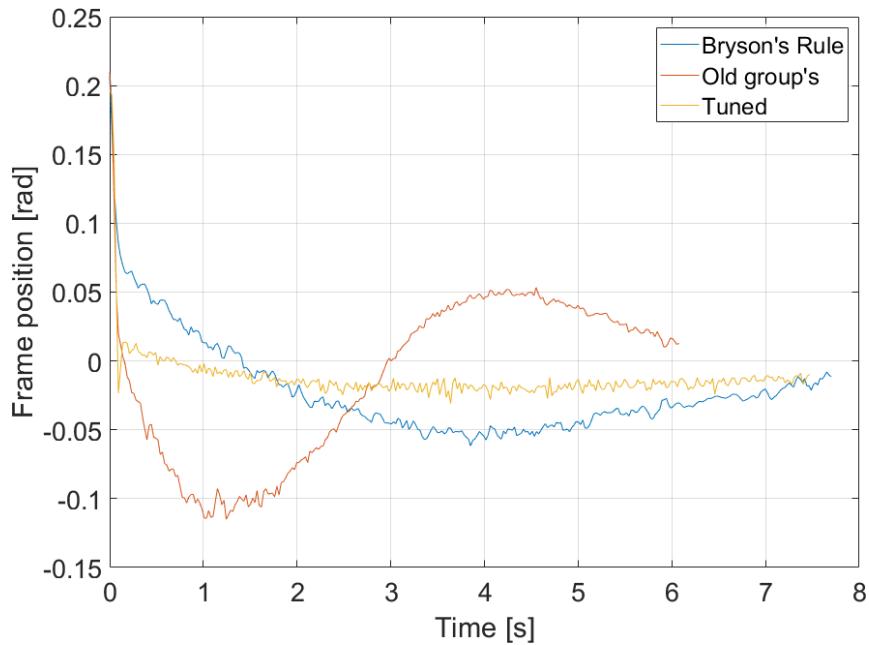


Figure E.1: Test results for the three different controllers.

F Required velocities for the start up procedure using speed controller

Response bearers and test facilities

Test report made by	Group 733
Test done by	Group 733
Test facilities	B3-209
Address	Fredrik Bajers Vej 7, 9220 Aalborg East, Denmark

Journal information

Unique journal number	JN05	
Relevant dates	Date of initiation: 05.10.2020	Date of completion: 07.10.2020
Number of pages in journal	2	

F.1 Test equipment

Manufacturer	Function	Model no.	Location	AAU-I8-no.
AAU	Cubli	N/A	N/A	N/A

F.2 Purpose of the test

The purpose of this test is to determine the start up velocities required to perform the start up procedure for the old, modified Cubli and the new Cubli.

F.3 Test setup

The tests are performed with the Cubli initially lying down on one side and the initial speed of the reaction wheel is 0 RPM. The test is initiated by switching on the Cubli for 3 seconds, the velocity of the wheel is controlled by the speed controller.

F.4 Measure procedure

The speed of the reaction wheel $\dot{\theta}_w$ is measured through the built-in tachometer of the motor driver and the angular position θ_f measurement is taken from the potentiometer of the Cubli.

F.5 Results and conclusion

Using the calculated estimate from Equation 5.24 as an initial guess, the required reaction wheel velocities can be found using a simple bisection-based approach. If the frame does not reach the balancing point, the reaction wheel's velocity is increased, and if it overshoots, the velocity is decreased. The determined values for start up for the modified, old Cubli and the new Cubli can be found in Table F.1:

Table F.1: The old, modified Cubli and new Cubli's start up velocities

Cubli	Left side	Right side
Old, modified	1400 RPM	1540 RPM
New	1320 RPM	1597 RPM

Figure F.1 presents the final test result of the start up procedure for the old, modified Cubli for the left side. The reference value for the speed controller is set to 1400 RPM.

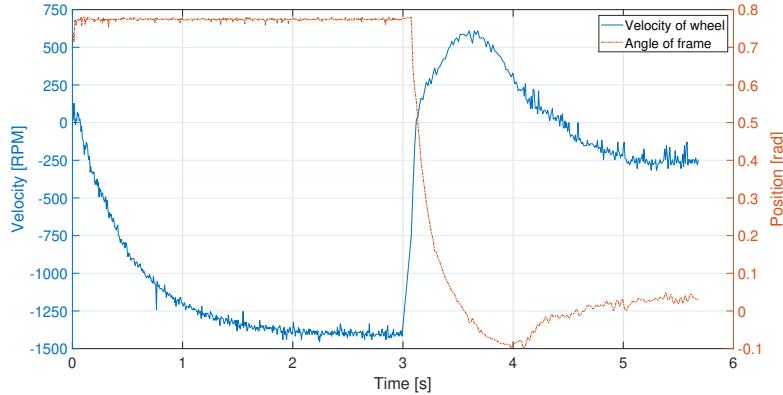


Figure F.1: The test results for the implemented start up procedure for the old, modified Cubli for the left side. Graph depicts the angular velocity of the reaction wheel $\dot{\theta}_w$ as well as the angular position of the frame θ_f .

The overshoot of 0.1 rad is considered to be a satisfactory result for the start up procedure, as the LQR controller takes over after the brake has been released and attempts to balance the Cubli.

G Shut down procedures

Response bearers and test facilities

Test report made by	Group 733
Test done by	Group 733
Test facilities	B3-209
Address	Fredrik Bajers Vej 7, 9220 Aalborg East, Denmark

Journal information

Unique journal number	JN06	
Relevant dates	Date of initiation: 12.10.2020	Date of completion: 15.10.2020
Number of pages in journal	5	

G.1 Test equipment

Manufacturer	Function	Model no.	Location	AAU-I8-no.
AAU	Cubli	N/A	N/A	N/A

G.2 Purpose of the test

The purpose of this test is to determine the compatibility of the two new shut down procedures, namely braking before impact and deceleration, with the requirement set in section 5.5. Additionally, each of the methods' performance is evaluated by observing the interaction with disturbances.

G.3 Test setup

In order to test the compatibility of the shut down procedures with the requirements, the Cubli is set to balance around its equilibrium by using the designed LQR controller. The test is initiated by switching off the controller.

Evaluating the interaction with disturbances is done by letting the Cubli balance and right before switching off the controller, the Cubli is disturbed by applying force in arbitrary direction, changing the Cubli's states θ_f , $\dot{\theta}_f$, and $\dot{\theta}_w$ to non-zero initial conditions right before the shut down procedure.

G.4 Measure procedure

The speed of the reaction wheel $\dot{\theta}_w$ is measured through the built-in tachometer of the motor driver and the angular position θ_f measurement is taken from the potentiometer of the Cubli. Speed of frame $\dot{\theta}_f$ is not measured as its effects are considered to be insignificant for this test compared to the other system states.

The Arduino code is modified by adding printing functions to print the measured values on the serial port of the Arduino. The first measurements are taken right as the controller is switched off.

G.5 Results and conclusion

Firstly, the two new shut down methods are tested with initial conditions being zero and are compared to free fall. The results can be seen on figure Figure G.1.

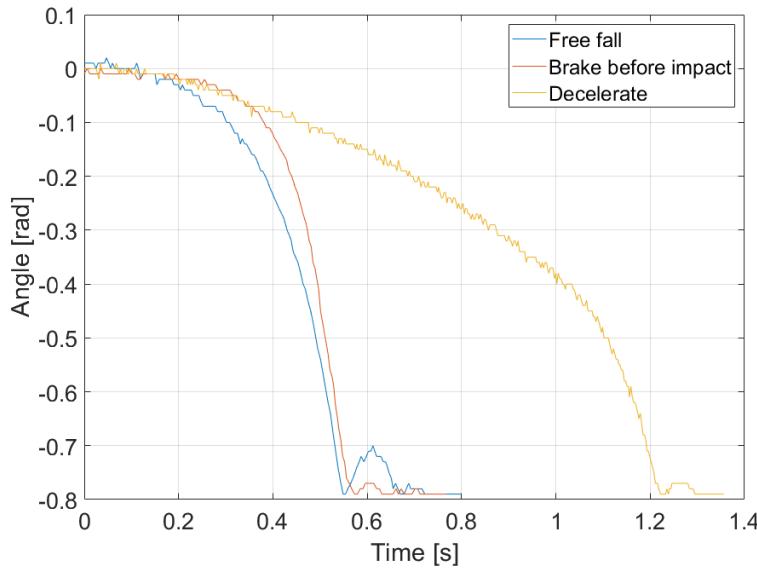


Figure G.1: The test results for the two new methods and free fall.

It can be seen that in the case of free fall, the bounce up after impact is approximately 0.1 rad, while in the cases of braking before impact and decelerating, it is 0.03 rad. Both of the new methods fulfil the requirement, which is to lessen the impact.

In order to determine which method is more reliable, it is necessary to identify the differences between the two methods by testing their robustness through the introduction of disturbances right before initiating the shut down procedures.

Two different types of disturbances are applied to the Cubli right before initiating the shut down procedures. Firstly, the Cubli is disturbed such that the velocity

of the reaction wheel is about 1000 RPM in the same direction as is the direction which is required to spin the reaction wheel in order to perform the shut down procedure. In the graphs this is denoted as "Initial wheel velocity POS". In the second case, the disturbance asserts the velocity of the reaction wheel to be about 1000 RPM in the opposite direction compared to the required direction. In the graphs this is denoted as "Initial wheel velocity NEG". Both graphs also include the case without any disturbances being applied for comparison.

Figure G.2 illustrates the test results for the deceleration method and Figure G.3 shows the test results for the braking before impact method. Please note that in the case of braking before impact, the reaction wheels velocity graph shows the velocity after applying the break to converge around $|500|$ RPM for some time, while in reality this was not the case. In reality, this is not the case and the velocity does converge to 0 RPM as intended, thus it does not affect the results of the test. The reason for this error is currently unknown.

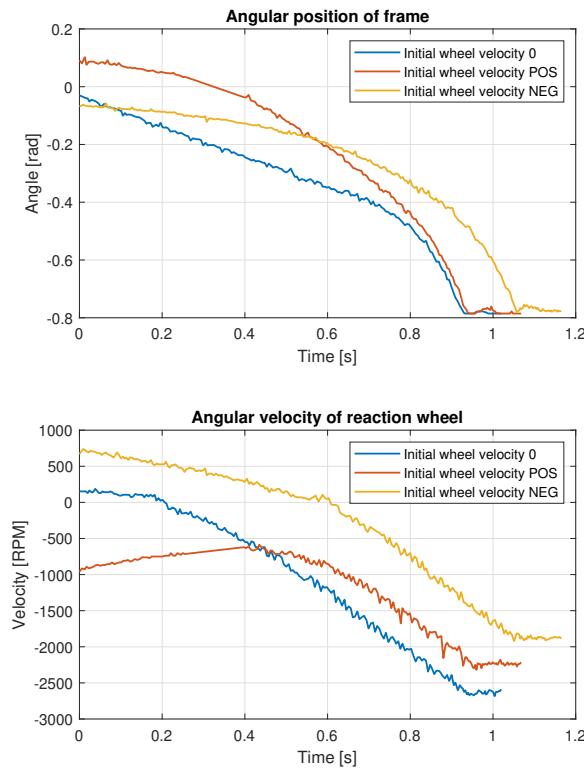


Figure G.2: Deceleration method with disturbances. Upper graph shows the angular position of the frame θ_f , while lower graph depicts the angular velocity of the reaction wheel $\dot{\theta}_w$.

In the case of deceleration it can be seen that the method is robust against

disturbances. The bounce up after impact is about 0.03 rad in all 3 cases, meaning that the method fulfils the specified requirements.

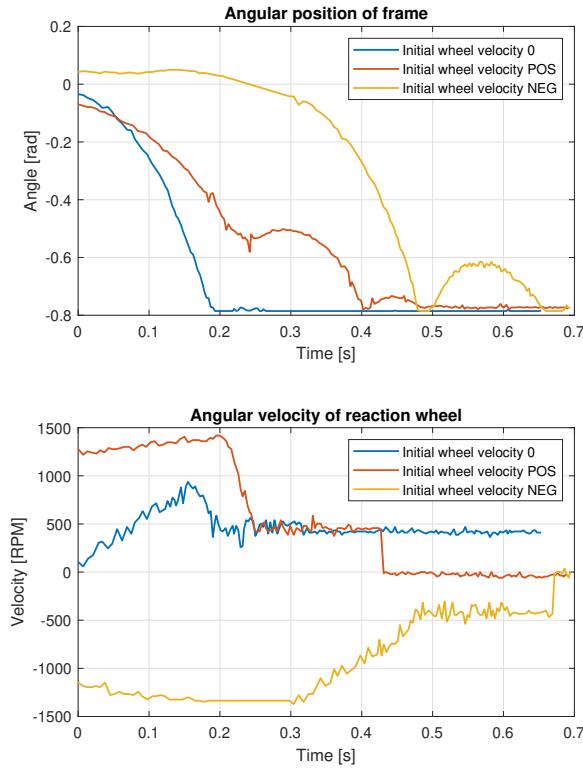


Figure G.3: Braking before impact method with disturbances. Upper graph shows the angular position of the frame θ_f , while lower graph depicts the angular velocity of the reaction wheel $\dot{\theta}_w$.

When it comes to braking before impact, if the initial wheel velocity is in the same direction as the required direction, after the wheel reaches the 1400 RPM limit, there is no additional torque applied to the Cubli and the only acting torque is from gravity. This means that the acceleration of the fall is slower. The brake is actuated after the frames angle has surpassed 0.33 rad. This results in the Cubli changing the direction of the fall at 0.5 rad for a moment, making the movement of the frame look uneven.

If the initial wheel velocity is in the opposite direction as the required direction, during the fall the Cubli manages to reach -500 RPM before applying the brake. This means that when applying the brake, the Cubli crashes the ground harder, making the bounce up after impact to be about 0.2 rad, which is 0.1 rad higher when comparing to free fall in Figure G.1, meaning, that the method is counter-productive to its purpose.

From these test results, even though both methods have shown the ability to fulfil the specified requirements if there are no disturbances acting on the system, the deceleration method for shut down procedure has proven to be more robust when disturbances are acting on the system. Thus, the deceleration method has been chosen to be used for shut down.

H Synchronised Start Up

Response bearers and test facilities

Test report made by	Group 733
Test done by	Group 733
Test facilities	B3-209
Address	Fredrik Bajers Vej 7, 9220 Aalborg East, Denmark

Journal information

Unique journal number	JN07	
Relevant dates	Date of initiation: 18.11.2020	Date of completion: 18.11.2020
Number of pages in journal	2	

H.1 Test equipment

Manufacturer	Function	Model no.	Location	AAU-I8-no.
AAU	Cubli	N/A	N/A	N/A
OnePlus	Camera	OnePlus 6 A6003	N/A	N/A

H.2 Purpose of the test

The purpose of this test is to determine the delay between the two Cublis when attempting a synchronised start up procedure.

H.3 Test setup

The code of the Cublis are changed such that an LED is turned on when the brake is applied to the reaction wheel during the start up procedure.

The Cublis are placed next to each other lying down to allow easy view of the two LEDs during the start up procedure by a single camera running at 480 frames per second.

H.4 Measure procedure

The two Cublis are then turned on and the motion is recorded by the camera. 10 synchronised start up procedures were recorded. These videos are then analysed by finding the number of frames between the two LEDs turning on. This can then be converted to a maximum delay by dividing the number of frames with the frame rate of the video.

H.5 Results and conclusion

Table H.1 presents the results from the tests and the conversion to max. delay in m.

Table H.1: Measurement of the delay between two Cublis.

Test #	1	2	3	4	5	6	7	8	9	10
Frames	6	12	84	2	4	4	6	5	5	5
Max. delay [ms]	12	24	175	4	8	8	12	10	10	10

I Estimation of delay between input and output of a Cubli

Response bearers and test facilities

Test report made by	Group 733
Test done by	Group 733
Test facilities	B3-209
Address	Fredrik Bajers Vej 7, 9220 Aalborg East, Denmark

Journal information

Unique journal number	JN08	
Relevant dates	Date of initiation: 27.11.2020	Date of completion: 27.11.2020
Number of pages in journal	2	

I.1 Test equipment

Manufacturer	Function	Model no.	Location	AAU-I8-no.
AAU	Cubli	N/A	N/A	N/A
OnePlus	Camera	OnePlus 6 A6003	N/A	N/A
Group 733	Block of wood with $\approx 10^\circ$ angle	N/A	N/A	N/A

I.2 Purpose of the test

The purpose of this test is to determine delay between a measured input by the Cubli to when the torque is applied to the reaction wheel using the DC-motor.

I.3 Test setup

The Cubli is placed at an angle of $\approx 10^\circ$ by leaning it against an angled piece of wood.

The code of the Cubli is changed, such that a LED is turned on, when the Cubli starts using the full state feedback controller.

The camera is placed, such that both the reaction wheel and the LED is in frame.

The Cubli is then turned on, and due to the initial angle it will not attempt the start up procedure, but instead only use the LQR controller.

I.4 Measure procedure

The camera, running at 480 frames per second, is used to record the motion from when the Cubli's power is turned on.

The amount of frames from when the LED is turned on, to when the wheel starts moving is then counted.

I.5 Results and conclusion

According to the video it took 5 frames from when the LED turned on to when the wheel started moving. Using the frame rate of the camera, an upper bound for the delay can be found:

$$\text{delay} = \frac{5 \text{ frames}}{480 \text{ FPS}} \approx 10.4 \text{ ms} \quad (\text{I.1})$$

J Requirement for synchronisation

Response bearers and test facilities

Test report made by	Group 733
Test done by	Group 733
Test facilities	B3-209
Address	Fredrik Bajers Vej 7, 9220 Aalborg East, Denmark

Journal information

Unique journal number	JN09	
Relevant dates	Date of initiation: 13.11.2020	Date of completion: 13.11.2020
Number of pages in journal	2	

J.1 Test equipment

Manufacturer	Function	Model no.	Location	AAU-I8-no.
Arduino	Microcontroller	Arduino MKR Vidor 4000	N/A	N/A
N/A	Signal light	Red LED	N/A	N/A

J.2 Purpose of the test

The purpose of the test is to find the lower bound for noticeable delay for a human observer.

J.3 Test setup

The physical test setup is an Arduino connected with two LEDs. The two LEDs are placed 1 cm between each other. The following Arduino code is used to make the LEDs blinking with a changeable delay between them.

```
1
2 #define LED1 2
3 #define LED2 4
```

```
4
5 void setup() {
6     // put your setup code here, to run once:
7     pinMode(LED2,OUTPUT);
8     pinMode(LED_BUILTIN,OUTPUT);
9     pinMode(LED1,OUTPUT);
10 }
11
12 void loop() {
13     digitalWrite(LED_BUILTIN,LOW);
14     digitalWrite(LED2,HIGH);
15     delay(30);
16     digitalWrite(LED1,HIGH);
17     //Serial.print ("1");
18     delay(1000);
19     //Serial.print ("2");
20     digitalWrite(LED_BUILTIN,HIGH);
21     digitalWrite(LED1,LOW);
22     delay(30);
23     digitalWrite(LED2,LOW);
24     delay(1000);
25 }
```

J.4 Measure procedure

The delay between two LEDs is set to start at 50 ms. If the observer could identify which LED lights up first at this delay time, the delay is reduced.

J.5 Results and conclusion

According to the experiment, the delay can not be detected when it is less than 25 ms.

K Configuration of XBEE S6B communication modules

This document describes the configuration of the each of the XBEE S6B modules.

Following parameters has to be set to the following values in the XCTU configuration software[24].

	XBEE module 1	XBEE module 2
Network:		
AH Network type	IBSS Joiner	IBSS Creator
CE Infrastructure Mode	STA mode	STA mode
ID SSID	xbee	xbee
IP IP Protocol	UDP	UDP
MA IP Addressing Mode	Static	Static
NS DNS Address	192.168.1.1	192.168.1.1
DL Destination IP Address	192.168.1.13	192.168.1.12
GW IP Address of gateway	192.168.1.1	192.168.1.1
MK IP Address Mask	255.255.0.0	255.255.0.0
MY Module IP Address	192.168.1.12	192.168.1.13
Serial Interfacing:		
BD Baud Rate	115200	115200
AP API Enable	Transparrent Mode	Transparrent Mode

Furthermore after configuring the Creator module, it is important to choose a WIFI channel for it. If this is not done, the module will choose an invalid channel and the modules will not be able to communicate. The channel is chosen by sending the following commands:

```
+++ //type this to enter AT command mode
OK //module responds now were in AT command mode.
ATCH1 //type in end command with Carriage return .
OK //module responds that it has accepted the command.
```

L Communication delay mean and standard deviation.

Response bearers and test facilities

Test report made by	Group 733
Test done by	Group 733
Test facilities	B3-209
Address	Fredrik Bajers Vej 7, 9220 Aalborg East, Denmark

Journal information

Unique journal number	JN10	
Relevant dates	Date of initiation: 23.10.2020	Date of completion: 23.10.2020
Number of pages in journal	3	

L.1 Test equipment

Manufacturer	Function	Model no.
XBEE	Communication module	DIGI XBEE S6B
Arduino Vidor	Microcontroller	Arduino MKR Vidor 4000

L.2 Purpose of the test

The purpose of the test is to determine the communication delay between two Arduino Vidor modules over XBEE modules in the system, over large numbers of packets.

L.3 Test setup

The physical test setup is shown in Figure L.1 and the code used for the test can be found in https://github.com/KristianHLarsen/AAU-Cubli-2D-2020/tree/main/communication/Delay_estimator_v2.

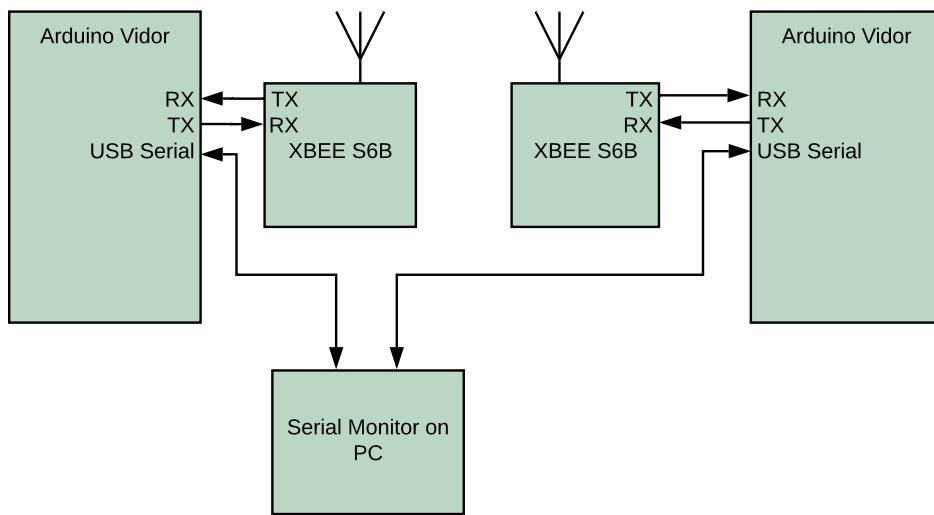


Figure L.1: Physical test setup. Supply voltage and GND has not been included in the diagram for simplicity.

L.4 Measure procedure

To measure the communication delay one has to follow the measurement procedure:

1. Upload code to both Arduino Vidor Microcontrollers
2. Run code on each device.
3. Save data from Serial Monitor.

L.5 Results and conclusion

The result for the test is shown in Figure L.2, where the transmission delay for transmitting and receiving packets is measured.

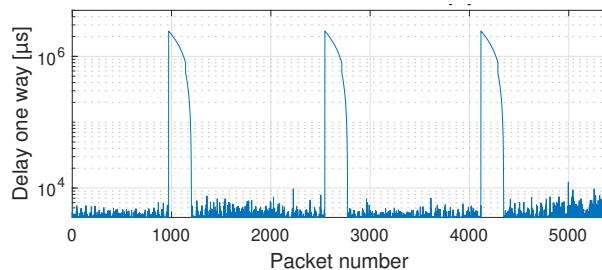


Figure L.2: Test results for 5400 packets transmitted and received.

In Figure L.2 the delay spikes periodically. The reason for these delay spikes on the communication channel is currently unknown, but presumed to be caused by the hardware itself. Different packet sizes and packet rates have been tested, but the delay spikes remain.

Omitting the delay spikes, the delay has a mean of 3.8 ms and a standard deviation of 0.33 ms.

M Communication bandwidth

Response bearers and test facilities

Test report made by	Group 733
Test done by	Group 733
Test facilities	B3-209
Address	Fredrik Bajers Vej 7, 9220 Aalborg East, Denmark

Journal information

Unique journal number	JN11	
Relevant dates	Date of initiation: 23.10.2020	Date of completion: 23.10.2020
Number of pages in journal	3	

M.1 Test equipment

Manufacturer	Function	Model no.	Location	AAU-I8-no.
PC	Computer	N/A	N/A	N/A
XBEE	communication module	DIGI XBEE S6B	N/A	N/A
Arduino Vidor	Microcontroller	Arduino MKR Vidor 4000	N/A	N/A

M.2 Purpose of the test

The purpose of the test is to determine the Communication bandwidth from end to end in between the Arduino Vidor microcontrollers.

M.3 Test setup

The physical test setup is shown in Figure M.1 and the code used for the test can be found in <https://github.com/KristianHLarsen/AAU-Cubli-2D-2020/tree/>

main/communication/test/tx_rx_speedtest.

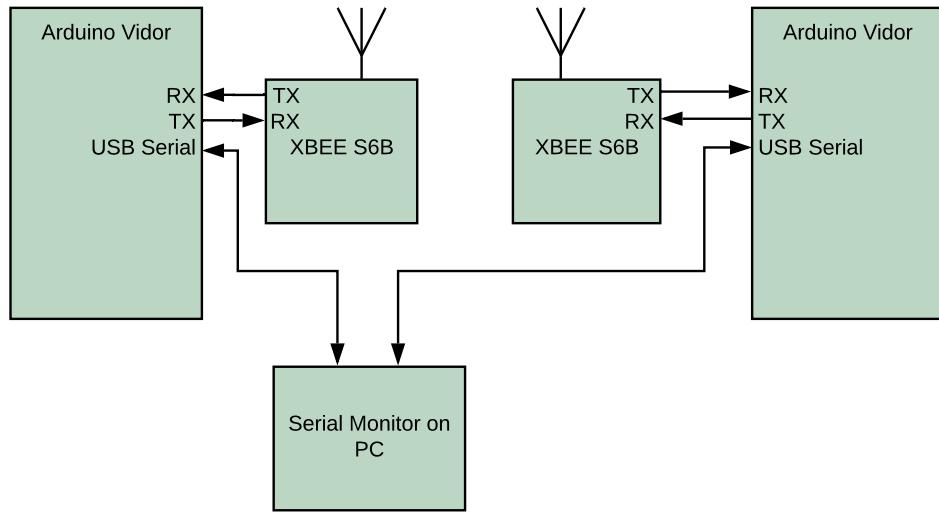


Figure M.1: Physical test setup. Supply voltage and GND has not been included in the diagram for simplicity.

M.4 Measure procedure

To measure the the communication delay one has to follow the measurement procedure:

1. Upload code to both Arduino Vidor microcontrollers
2. Open serial monitor in Arduino IDE on a computer for the receiver module.
3. Start a transmission by touching ground terminal with D1 on Arduino Vidor.
4. Save measurements from serial monitor.

M.5 Results and conclusion

The result for the test is shown in Figure M.2 where 15000 packets were sent and received.

```

Size of packet13
1000 packages received in: 1259318 uSeconds
Bad packets: 0
1000 packages received in: 1259331 uSeconds
Bad packets: 0
1000 packages received in: 1259480 uSeconds
Bad packets: 0
1000 packages received in: 1259529 uSeconds
Bad packets: 0
1000 packages received in: 1259580 uSeconds
Bad packets: 0
1000 packages received in: 1259539 uSeconds
Bad packets: 0
1000 packages received in: 1259570 uSeconds
Bad packets: 0
1000 packages received in: 1259451 uSeconds
Bad packets: 0
1000 packages received in: 1259607 uSeconds
Bad packets: 0
1000 packages received in: 1259584 uSeconds
Bad packets: 0
1000 packages received in: 1259647 uSeconds
Bad packets: 0
1000 packages received in: 1259701 uSeconds
Bad packets: 0
1000 packages received in: 1259825 uSeconds
Bad packets: 0
1000 packages received in: 1259789 uSeconds
Bad packets: 0
1000 packages received in: 1259783 uSeconds
Bad packets: 0

```

Figure M.2: Test result for bandwidth test.

From Figure M.2 an average of all tests will be used for estimating the bandwidth of the throughput between the Arduino Vidors.

$$\begin{aligned}
bandwidth &= \frac{\text{packet} \cdot \text{packet size} \cdot 8}{\text{transmission time}} \\
&= \frac{15000 \cdot 14 \text{ B} \cdot 8}{18893734 \mu\text{s}} \\
&= 88.918 \text{ Kbit/s}
\end{aligned} \tag{M.1}$$

Thus it can be concluded the bandwidth of the system is 88.91 Kbit/s.

N Networked control

Response bearers and test facilities

Test report made by	Group 733
Test done by	Group 733
Test facilities	B3-209
Address	Fredrik Bajers Vej 7, 9220 Aalborg East, Denmark

Journal information

Unique journal number	JN12	
Relevant dates	Date of initiation: 10.11.2020	Date of completion: 10.11.2020
Number of pages in journal	3	

N.1 Test equipment

Manufacturer	Function	Model no.
AAU	Cubli	Old, modified
AAU	Cubli	New

N.2 Purpose of the test

The purpose of the test is to evaluate whether networked control between the two Cublis is feasible.

N.3 Test setup

The code is altered to enable the following operation: the two Cublis are started up and balanced for 1 second as usual, using their local controller. After that, the two controllers start exchanging their sensor data and compute the controller inputs to each other.

The following pseudo code snippets present the key changes applied to the Arduino program to enable networked control. First the packet for sending is populated with the sensor measurements:

```
// populate packet with data measurements for this Cubli
txdata.packet.val1 = ang_pos_frame;
txdata.packet.val2 = ang_vel_frame;
txdata.packet.val3 = ang_vel_wheel;
```

Then the sensor data and from the other Cubli is extracted from the received packet. Furthermore, the controller input (*current_rx*) calculated for this Cubli is also extracted from the packet:

```
// extract data measurements from the other Cubli
ang_pos_frame_rx = rx_data.val1;
ang_vel_frame_rx = rx_data.val2;
ang_vel_wheel_rx = rx_data.val3;

// extract controller input for this Cubli
current_rx = rx_data.val4;
```

Using the received sensor data, this Cubli calculates the other Cubli's controller input:

```
// calculate the current for the motor of the other Cubli
current_tx = (((k1 * ang_vel_wheel_rx + k2 * ang_pos_frame_rx + k3 * ←
ang_vel_frame_rx / kt);
```

Finally, the calculation is normalised and inserted into the packet for sending:

```
// normalise
if (current_tx >= CURRENT_MAX) current_tx = CURRENT_MAX;
if (current_tx <= -CURRENT_MAX) current_tx = -CURRENT_MAX;

// populate packet with the control input for the other Cubli
txdata.packet.val4 = current_tx;
```

N.4 Measure procedure

The position data and the controller input of one Cubli are logged through a USB cable from one Arduino.

N.5 Result and conclusion

As can be seen in Figure N.1, the delay between the input and output of the controller for the Cubli, gets too large when using the communication in the control

loop.

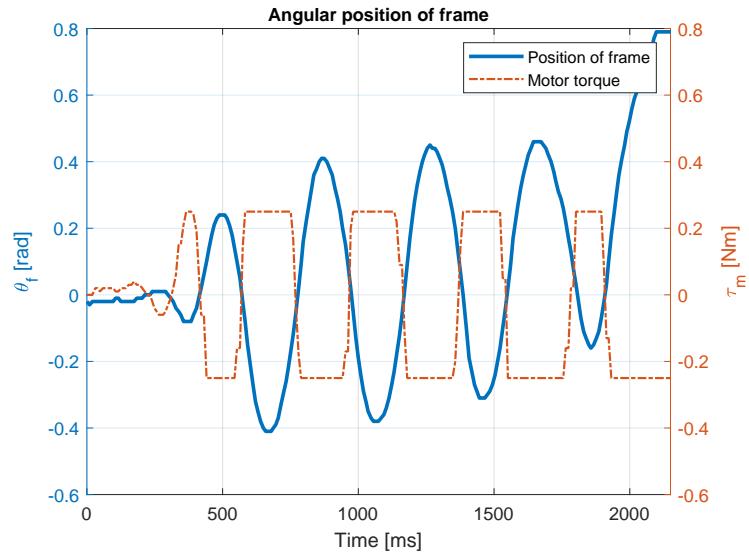


Figure N.1: Networked control results

O Kalman filter data gathering

Response bearers and test facilities

Test report made by	Group 733
Test done by	Group 733
Test facilities	B3-209
Address	Fredrik Bajers Vej 7, 9220 Aalborg East, Denmark

Journal information

Unique journal number	JN13	
Relevant dates	Date of initiation: 13.11.2020	Date of completion: 13.11.2020
Number of pages in journal	2	

O.1 Test equipment

Manufacturer	Function	Model no.	Location	AAU-I8-no.
AAU	Cubli	N/A	N/A	N/A
N/A	PC	N/A	N/A	N/A

O.2 Purpose of the test

The purpose of the test is to gather data for the Kalman filter's simulation in MATLAB.

O.3 Test setup

A PC is connected with a USB cable to the Arduino to access the serial port.

O.4 Measure procedure

When the Cubli stands up, the output of the complementary filter, the potentiometer data, frame velocity, wheel velocity, input to the motors and timestamp are transmitted to the computer through the serial port. Some disturbances are applied by pushing the Cubli manually.

O.5 Results and conclusion

Using MATLAB the test result are analysed and later used for simulation. Figure O.1 shows part of the gathered data of one test.

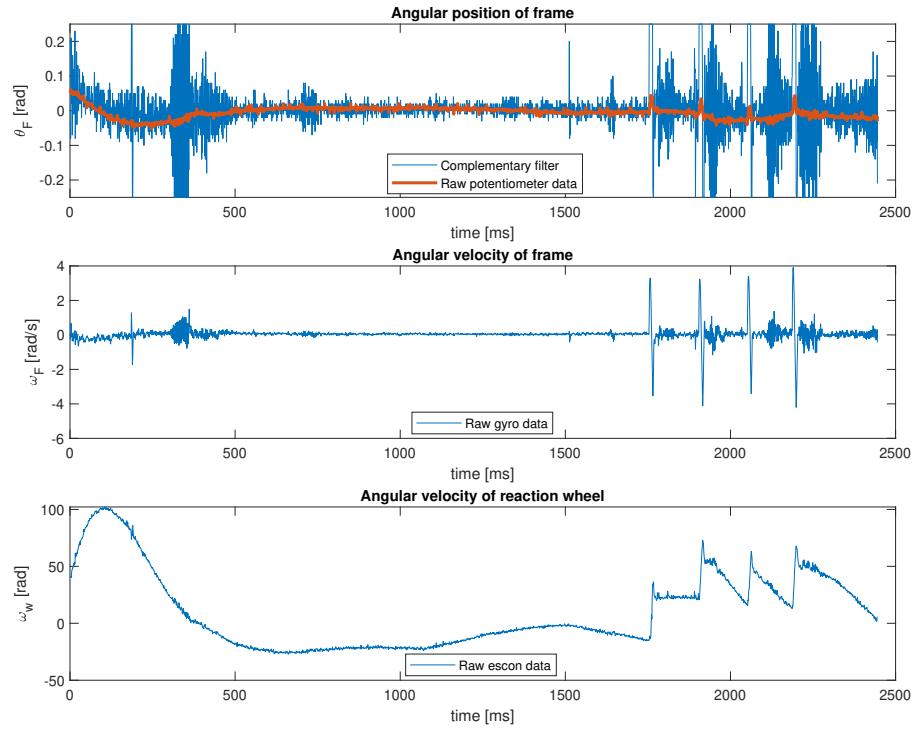


Figure O.1: Test result for Kalman filter.

P Review from group 730

P.1 Overall Comments

The paper about the 2D Cubli is generally well written and easy to read.

Furthermore, the amount of specific comments is a result of your work having high standards, not the other way around as one might think.

The purpose of the article, as we understand it, is to synchronize the startup and shutdown of two similar 2D Cublis, with the addition of a smoother shutdown procedure and better tuned balancing controller.

Although, a description of the novelty is needed, since the application of your improvement is not immediately obvious to us.

Otherwise, great job.

P.2 General Comments

1. *Layout: The column spacing is a bit too narrow (4mm vs 1cm IEEE), pages of the article are not numbered, caption text and callout needs to be in the font "Helvetica".*

When following the IEEE style guide for conference papers, the above mentioned formatting should not be followed. It therefore has not been changed.

2. *You should omit specific hardware descriptions throughout the text, they shouldn't be relevant for your results.*

3. *Use consistent matrix and vector notation. For instance, in figure 7 x is written with an arrow and B in eqn. 5 is written with bold text. Consider using lines instead of boldface, to make a better distinction between vectors and matrices.*

Vectors have been changed so that they are all boldface italic.

4. *Indexes in equations should not be in italic, that is only for subscripts. E.g., eqn. 11.*

Following the IEEE standard for editing mathematics, the indices are kept in italic.

5. *Figures should have a higher resolution. Figures 5, 6 and 7 for instance are hard to read (especially the text) and the grid on figure 8, 9 and 10 is hard to see.*

Figure 5, 6, and 7 are vector graphic images, and therefore have infinite resolution. The rest have been changed to vector graphics and the font size on all has been increased to ease readability.

6. *Consider reducing the number of subsections. This especially applies to the 7 subsections in section II and III*

Subsections in Materials and Methods:

1. Dynamic Model of the System
2. Controller Design
3. Shut Down
4. Start Up
5. Synchronised Cubli motion
6. Communication Setup and Infrastructure
7. Networked Control

Each of the subsections deals with separate topics. It could be argued that Controller Design, Shut Down, and Start Up are all part of the controller. However, the theory, and modelling for the latter two, make these subsections fairly different from each other. Therefore, it has been chosen not to combine these into one subsection, to not loose clarity in the section.

For Results:

1. Controller
2. Start Up
3. Shut Down
4. Delay
5. Requirement for Synchronisation
6. Synchronised Motion
7. Networked Control

Requirement for Synchronisation and Synchronised Motion has been combined into one subsection under the title Synchronised Motion.

7. *Consider a nomenclature for all your variables.*

Adding a nomenclature might facilitate the understanding of the equations. However, due to the relatively large amount of variables and matrices, a nomenclature would take up relatively large space, resulting in the need to reduce valuable descriptions.

8. *The main purpose/goal of a test or a design should be mentioned. E.g., the upper bound of the delay in startup must not be notable to a person. Then you should also remember to compare the result with your goals/specifications (same example).*

The following has been added or changed in the subsections in Results:

Controller: "This is done to investigate the transient responses of the different controllers."

Start Up: "The following tests are performed to find suitable start up velocities for Cublis to reach their balancing point."

Shut Down: "Experiments have been conducted to find which of the two methods reduce the impact with the ground the most."

Delay: "To evaluate the performance of the playout buffer, the delay in the system with and without the buffer has been examined."

Synchronised Motion: "Before testing the synchronisation of the two Cublis, an acceptable limit first needs to be determined using the following experiment:"

Network Control: No changes.

P.2.1 Specific Comments

1. *Title: Capitalize "operation" in the headline*

It has been changed.

2.a *Abstract: §1, line 17: Is it the state machine that facilitates the network or the other way around?*

The state machine is an implementation architecture which makes it easier for the developer to implement a certain set of functionalities. Therefore it has not been changed, since "the state machine facilitates the network implementation".

2.b *Abstract: §1, line 18: Elaborate on the extension part, and tell why you make an extension.*

The sentence has been changed to improve understanding: "Networked control is also tested on the systems, but communication delays cause instability."

2.c *Abstract: §1, line 22: It's probably the communication and not the delay that drops out.*

"the synchronisation is" has been added to clarify.

3.a *Intro: Nice background check*

No comment.

3.b *Intro: §2, line 3: [...]discrepancy in reference[...] It seems strange to develop a method of continuously changing the reference. One could argue that the reference is either not correctly set or that the controller needs to be able to handle this*

The actual quotation is: "[...] a discrepancy of the balancing angle[...]" . The sentence has been changed to: "A method was developed for continually correcting the reference angle to compensate for a discrepancy of the balancing angle of the Cubli".

4.a *Material and Methods.A: §3, line 6: The "this" reference is not clear. Explicitly state what results in the following two equations.*

"this" has been changed to a reference to equation 1.

4.b. *Material and Methods.A: §4, line 1: Linearization method should be presented (Taylor, Small angle approximation, etc.).*

Linearisation is always a first order Taylor expansion. For trigonometric functions, this is sometimes called small angle approximation. Thus, it has not been changed.

4.c Material and Methods.A: §5, line 2: Consider merging the two sentences into one to remove double references.

It has not been done, as the first reference is with respect to finding the same model, and the second is that the same methods are used for the parameter estimation. These two things are considered too separate to combine.

4.d Material and Methods.A: §6, line 1: What is meant by "comparable", is it the same unit? Value?

"Comparable" has several meanings, one of which is a synonym to "similar", which is what it means in this sentence. Therefore, it has not been changed.

The sentence reads: "These parameter estimates are comparable to those found by [5]." I.e the estimates are comparable.

4.e Material and Methods.A: Eqn. 5: The equation is not needed when values and eqn. 4 is presented. Pole location is nice and sufficient

It has been changed as suggested.

4.f Material and Methods.A: §7, line 2: "for for" is written under eqn. 5

It has been changed.

4.g Material and Methods.B: §1, line 4: Describe your controller design goals and weighing of your states

It is stated what the goal of the controller is: "[...]to decrease the settling time while allowing a small overshoot.". There are no hard limits set for either of these, which is also one of the reasons why optimal control is used.

The exact weighing of the states are based on experiments, and thus are discussed under Results.

4.h Material and Methods.C: §1 & 2: The difference between methods only seems to be the application of brakes. Description should be more clear as to the different method of spinning the reaction wheel, since accelerating and applying torque seems the same.

It has been added that the direction of the wheel acceleration are different in the two methods.

4.i Material and Methods.C: §3, line 2: You should state whether the factor α is found from simulation or experiment. Also consider if there are any design goals (reduced speed, reduced acceleration). Is compensation of gravity by more than a factor of 1 possible? The method of "trial and error(empirically)" does not seem convincing.

It is stated that α is found empirically and the goal is to lessen the impact.

4.j Material and Methods.D: §1, line 2: [...]needs to get off the ground[...], aren't there more conditions that have to be fulfilled, like how far off?

"and to the balancing point" has been added.

4.k Material and Methods.E: §1, line 1: Rephrase into: [...]method, as just described. Consider merging sentence one and two.

The reference to the specific subsection has not been removed, to not reduce clarity.

4.l Material and Methods.F: Fig. 4: Is 17 bytes really the whole packet and not just the payload?

It has been changed to, so it is the payload.

4.m Material and Methods.F: §2, line 1: We believe delay variations is just equal to jitter. Also, by buffering you are not reducing jitter, you are altering the effect of the jitter. Combine the last two sentences into one, to reduce repeating yourself.

From the course Distributed Realtime Systems, jitter was defined as the minimum and maximum variation of the delay. Since this subsection deals with the standard deviations of the delay, it has not been called jitter. However, the sentence has been changed to "[...]reduce the impact of delay variations[...]".

4.n Material and Methods.G: §1, line 9: The note is not that relevant.

It is considered relevant, as the networked control is not shown on the state machine diagram.

5.a Results.A: §4, line 1: You present a nice method (with no controller design goals as described earlier), then you take the result and change by factor of 100 and 0.05, without commenting. This does not seem convincing, elaborate on why you do it as well as why it is valid.

"The entries $Q_{i,i}$ is the value of the response of the state x_i . R is the value of the input to the system." has been added to the Materials and Methods section.

It is a tuning process, so the validity is found empirically.

5.b Results.A: Fig. 8 & 9: Show the reference, to make it easier to see the performance of the controller. This is especially important if the reference is changing.

The controller will go to position zero. To clarify this, " A reference input of 0 is given." has been added.

5.c Results.B: §2, line 1: Rephrase the sentence since it is difficult to read. You should also comment on why the theoretical value is not used.

The theoretical value is used as the initial value in the bisection method. The sentence has been reformulated slightly to:

The initial reaction wheel velocities in Table II were found empirically using a bisection-based approach with (13) as a starting point.

5.d Results.B: §3, line1: Specify that it's the difference between left and right.

It has been changed.

5.e Results.C: §1, line 2: Make some consideration regarding the value of α If you compensate for gravity 100% it should stand still.

"Note, according to the model, the Cubli would not move with this value, however the model is imperfect." has been added.

5.f Results.C: Fig. 9: What is your performance criteria; no jump up, low force, speed at which it hits the ground,no holes in the table? If you want to reduce the impact, measuring the acceleration could be of interest.

Since the purpose has been added earlier it should be more clear now.

5.g Results.C: §2, line 4: *State in which way the method is unreliable.*

It is unreliable in the way that is not reliable, i.e. it does not always work. The sentence has not been changed.

5.h Results.E: §1, line 3: *The way it is stated, it seems like the delay also varies during one test. Consider removing "variable"*

It has been changed.

5.i Results.E: §1, line 4: *Be more specific about the observer is(person,photo sensor, etc.). Have you considered the delay of the observer?*

It is a human, and the sentence has been updated.

5.j Results.E: §1, line 6: *It should be the upper bound of the delay, instead of the lower.*

Added "noticeable"; then it is the lower bound (which is the same as the upper bound of the unnoticeable delay).

5.k Results.F: *Relate the results of table 4 to the test is table 3.* This has been changed.

5.l Results.G: *Refrain from calling two subsections the same (II-G& III-G)*

It has been changed.

5.m Results.G: *Nice tests. You should make a comment why it is not stable. Is it due to the delay? The controller?*

"This is presumed to be caused by the increased delay" has been added.

5.n Results.G: *Fig. 11:The motor clearly saturates, which you should comment on. How does the controller perform in simulation?*

Yes, the motor saturates, but that will always happen when the system becomes unstable. The sentence has not been changed.

No simulations have been presented at all in the paper.

6.a Conclusion: §1, line 3: *Less impact seems to be a good thing here.State it earlier.*

It is now stated.

6.b Conclusion: §2, line 3: *The comment regarding the out lying third test is not needed, because it doesn't obscure the general findings.*

The delay spikes in the network is an important problem of the paper, and thus the comment about its impact remain.

6.c Conclusion: §3, line 7: *You state that the Smith predictor is not applicable, but what about other predictors e.g., Kalman filter?*

To the best of our knowledge, the Kalman filter will not work.

Q Review from group 731

Q.1 Overall Comments

The article was written clearly and without any general mistakes/problems. Only a small amount of specific comments could be made.

Q.2 General comments

N/A

Q.3 Specific comments:

SC1: In the title of the article "operation" should be capitalized.

This has already been fixed.

SC2: Consider writing the full word followed by the acronym, for example, in Introduction "IMU" to keep consistency, because beforehand in the abstract LQR was written in both forms.

This has been fixed

SC3: To keep consistency with the paper instead of writing "old Cubli", in the introduction, consider writing Cubli 1.

It is already consistent. It is only mentioned twice.

R Review from group 732

R.1 Overall Comments

Very good article. Hard to find any substantial problems. The article is very well written, understandable, correctly structured and appropriately referenced. Great work!

R.2 General comments

GC1. The subsection titles should have the first letter of words capitalised instead of just the first word.

This has now been fixed.

GC2. When writing complex numbers, consider writing $2j$ instead of $2\cdot j$ (4th page, right column, under equation 12 and 13).

We have in the paper decided to use \cdot for all multiplications, including between real numbers and the complex variable. Thus it has not been changed for consistency.

R.3 Specific comments

SC1. First page, right column: You mention two approaches for a feedback controller: Pole placement and LQR. Maybe mention something about the pole-placement approach and why LQR was chosen.

The pole placement and LQR was used by the previous group, who ended up using LQR in the end. We only ever used LQR. Thus, the sentence has not been changed.

SC2. Consider elaborating on the method of communication between Cublis i.e., specifying that you are using Wi-Fi as well as the bandwidth available.

This has now been added: "[...]an XBEE S6B Wi-Fi module is implemented on each of them, providing a bandwidth of 320 kbps."

SC3. Figure 8: Equation 12 and equation 13 are separated from the rest of the text with commas, which gives the impression that they are individual tests. Consider using parenthesis instead.

The extra commas have been removed.

SC4. Reference 5 and 6 should not have the author's full name written. Just the initials.

This has been fixed.

SC5. Consider elaborating on the way the playout buffer is implemented and its effect on reducing delay variation.

The playout buffer will be removed from the paper.

SC6. Under equation 7, consider writing "where subscripts a and b" instead of what's currently there as it looks a bit odd.

It has been fixed as suggested.

SC7. Just before the section about controller design on page 2, you write "for" twice.

The extra "for" has been removed.

SC8. Have you considered the impact of the filter for the IMU on the dynamics of the Cublis?

Installing the IMU and the filtering is not a subject of this paper. It has been done by the previous AAU projects. However, descriptions of the different filters have been added in the worksheets.