

PROJ-H-405  
PROJECT IN ELECTROMECHANICAL ENGINEERING

---

## Project report

Identification and validation of the model of an in-scale prototype of a boom crane for didactic purposes.

---

*Authors:*

Maxime JONGEN  
Dinh-Hao NGUYEN

*Promoter:*

Emanuele GARONE

*Supervisor:*

Michele AMBROSINO

May 15, 2022

Academic year 2021-2022

## Abstract

Nowadays, boom cranes are widely used in several industrial sectors, especially on construction sites where their automation can have a direct impact on productivity. In this project, we aim to identify and validate a model of a previously-built prototype [6]. Although fairly simple mechanically, from a control point of view, this type of crane is a non-linear under-actuated system which presents several challenges for identification and control. Throughout this project, a 5-degree-of-freedom dynamic model for a boom crane has been developed based on a few assumptions. In order to predict the motion of the prototype, the parameters appearing in this model must first be identified. By sending a certain trajectory to the prototype, capturing its motion with high-precision cameras and measuring the torque applied by the actuators, these dynamic parameters can be estimated using a regression approach that can be solved with a least-squares optimization algorithm. Based on this dynamic model, a Simulink model was built to validate it by comparing, for the same requested trajectory, the motion of the physical crane with the motion predicted by the model. Discrepancies between the theoretical and real curves were observed: the shape of the curves is quite similar but an offset was observable. It has been evaluated to be due to the fact that the base of the crane is not rigid enough, which causes it to rotate around the vertical axis. This small rotation of the base is responsible for this offset and makes parameter identification harder.

**Keywords:** Boom cranes ; Robotics ; Parameter identification ; Model validation ; Under-actuated systems ; Non-linear systems.

# Acknowledgments

This MA1 project would not be what it is if it would not have been for the help and assistance of those who have guided us professionally and those who have supported us personally.

First of all, we would like to acknowledge and give our warmest thanks to our promoter, Prof. Emanuele GARONE, for his expert advice and the opportunity he offered us to work on this exciting project.

We are also deeply grateful to Ir. Michele AMBROSINO for his implication in this project from day one. No matter the difficulties or challenges, he always encouraged and helped us – especially when we encountered personal problems and things were not going well. We cannot express enough how reassuring it has been to be guided and advised by him.

In addition, we are also grateful to Sabri EL AMRANI for the time he spent teaching us how the OptiTrack cameras and Motive software work. His sharing of experience and demonstration allowed us to quickly manipulate these advanced tools that have been at the heart of this project.

We would also like to thank all the members of the SAAS department who helped us in one way or another in the realization of this project. Their contribution allowed us to solve many problems and to move forward in this project.

Finally, we would like to express our gratitude to our family, friends and colleagues who have supported and encouraged us throughout this journey. Their support helped us to persevere and get to the end of this project.

# External links

All MATLAB codes presented in this report can be found in the following open GitHub repository:

[https://github.com/Dinh-Hao-Nguyen/MA1\\_Project\\_Crane.git](https://github.com/Dinh-Hao-Nguyen/MA1_Project_Crane.git).

In addition, the video available at the following link:

<https://www.youtube.com/shorts/0ZIyQ80FdGc>

shows the crane prototype in operation during the parameter identification phase. A sinusoidal trajectory has been sent to the crane and the data of the position measurements are collected from the OptiTrack cameras.

# Contents

<b>Abstract</b>	i
<b>Acknowledgments</b>	ii
<b>External links</b>	iii
<b>1 Introduction</b>	1
1.1 Motivation . . . . .	1
1.2 State of the art . . . . .	1
1.3 Related work . . . . .	2
<b>2 Presentation of the project</b>	3
2.1 Scope & Objectives . . . . .	3
2.2 Methodology . . . . .	3
2.3 Presentation of the available material . . . . .	4
2.3.1 The crane . . . . .	4
2.3.2 The payload . . . . .	5
2.3.3 The actuators . . . . .	6
2.3.4 The OptiTrack system . . . . .	7
<b>3 Modeling</b>	8
3.1 Model of the boom crane . . . . .	8
3.2 Equations of motion . . . . .	9
<b>4 Dynamic parameter identification</b>	12
4.1 Different identification techniques . . . . .	12
4.2 The data-driven regression approach . . . . .	12
<b>5 Experimental procedure</b>	15
5.1 Data collection . . . . .	15
5.1.1 Data acquisition from the OptiTrack cameras . . . . .	15
5.1.2 From Motive to MATLAB . . . . .	17
5.1.3 Processing and storage of the data on MATLAB . . . . .	17
5.2 From the data to the least-squares algorithm . . . . .	22
5.3 Simulink model . . . . .	23
5.3.1 Presentation of the Simulink model . . . . .	23
5.3.2 Validation of the identification code with Simulink . . . . .	24
5.3.3 Use of the Simulink model to help identifying problems in the data .	25
5.3.4 Potential use to validate the model of the crane and the controller .	25

<b>6 Discussion of the experimental results</b>	<b>26</b>
6.1 Comparison of predicted and actual behavior of the crane . . . . .	26
6.2 Results obtained with the least-squares algorithm . . . . .	28
6.3 Identification and resolution of a problem related to the definition of the payload oscillation angles . . . . .	29
6.4 Effect of including the counterbalance in the model . . . . .	30
6.5 Identification of a problem related to the non-rigidity of the crane base . . . . .	32
<b>7 Work distribution</b>	<b>34</b>
<b>8 Learning outcomes of the project</b>	<b>35</b>
<b>9 Future possible improvements</b>	<b>37</b>
<b>References</b>	<b>39</b>
<b>A Dynamic model of a boom crane</b>	<b>41</b>

# List of Figures

2.1	The in-scale prototype of a boom crane used for this project: (a) the base, (b) the arm, (c) the hoist rope, (d) the hoisting platform. . . . .	4
2.2	Another view of the crane used for this project. . . . .	5
2.3	The two different payloads used for this project. . . . .	5
2.4	The HEBI Robotics' X8-16 Actuator [14]. . . . .	6
2.5	Control strategy implemented in the HEBI Robotics' X8-16 Actuator [10]. .	6
3.1	Model of a boom crane, inspired by [2]. . . . .	9
5.1	Example of OptiTrack camera calibration. . . . .	15
5.2	Results obtained from the calibration. . . . .	16
5.3	Definition of the rigid bodies on Motive (Rigid Body 1: the payload and Rigid Body 2: the crane). . . . .	17
5.4	Illustration of the methodology used to store data in MATLAB. Only a fraction of the MATLAB workspace is shown here, more variables are actually stored from an experiment. . . . .	18
5.5	Example of filtering for the $\theta_2$ angle . . . . .	19
5.6	Illustration of a problem linked to the discontinuity in the angle definition. .	20
5.7	Illustration of irregularities in the sampling time. . . . .	20
5.8	Simulink model of the crane . . . . .	23
6.1	Comparison between the experimental measurements (in blue) and the simu- lation results (in red), for the same trajectory. . . . .	27
6.2	Example of a sinusoidal function used as a command in velocity for one of the actuators in order to get the crane to move randomly. . . . .	28
6.3	Comparison of the results obtained with and without the consideration of the counterbalance in the model. . . . .	31
6.4	Illustration of the problem of the non-rigidity of the crane base: the base axis is not aligned with the gravity vector resulting in an undesired torque on the actuator. . . . .	33

# List of Tables

3.1	Generalized coordinates of the crane-payload system. . . . .	10
3.2	Parameters of the crane-payload system. . . . .	10
3.3	Some particular position coordinates of the crane-payload system. . . . .	10

# Chapter 1

## Introduction

### 1.1 Motivation

Despite the abundance of boom cranes in construction sites, shipyards and various other industrial applications, there is still a great demand for research and progress in this field. There is still a need for larger, safer, faster and more efficient cranes. Crane automation could enable such improvements. As of today, crane automation is primarily limited to small cranes due to the complexity of safely controlling cranes on site. However, automation of boom cranes in the transportation sector has several benefits, including increased productivity, reduced labor costs, and the potential for safer operation by eliminating human error.

Furthermore, there is currently a significant lack of studies on boom crane control in the literature and most of the existing studies are based only on simulated models without any experimental validation. In this project, the use of an in-scale prototype of a boom crane [6] enables to experiment on a real physical crane in order to validate the developed dynamic model.

### 1.2 State of the art

Crane automation has been widely studied in recent years, but boom cranes have been the subject of only a few articles [2] although they are one of the most widely-used types of cranes due to their simple design [3]. They are mainly used to vertically lift heavy loads in various industrial applications such as factories, construction sites and marine industries. Moreover, this type of crane is non-linear and under-actuated. As a result, the control of a boom crane is rather challenging.

Several studies on the modeling and control of a 5-degree-of-freedom (5-DoF) boom crane have been conducted. Many control schemes for the boom crane have been studied in recent years, including:

- control based on the Explicit Reference Governor (ERG) for constraints consideration [1],
- non-linear Model Predictive Control scheme [4],
- control based on a PD plus gravity compensation [6],
- and Linear Quadratic Regulator (LQR) [6].

## 1.3 Related work

This MA1 project is a continuation of several previous projects. First, the in-scale prototype of a crane was designed and built for a master thesis [6]. In addition to constructing the prototype, the student also developed a dynamic model and some control schemes for this crane. Many of the CAD models and MATLAB codes that were provided to us at the beginning of the project come from his work and constitute a solid basis for this project.

We can also mention the work of another student who focused on measuring boom crane's payload vibrations using the OptiTrack system as well as trajectory tracking using different control strategies. His work provided us with some MATLAB codes for data collection from the OptiTrack cameras.

# Chapter 2

## Presentation of the project

### 2.1 Scope & Objectives

This work is part of a wider project to develop a robotic methodology to build walls using an in-scale prototype of a crane. The in-scale prototype used for this project is presented in Section 2.3.1. In order for this crane to move the payload to a desired position, a control scheme is required. The movement should not be too fast in order to avoid as much as possible the oscillation of the payload which would make the placement more difficult and less accurate. However, to move the payload to the correct position, one must first be able to predict the motion of the crane. This can be achieved by developing a dynamic mathematical model. Given certain assumptions, this model can be obtained using the Lagrange-Euler approach. The various parameters of the model can then be identified – or rather, estimated – by experimenting with the prototype.

### 2.2 Methodology

For this project, we focus on the identification and validation of the crane model, keeping in mind the further objective of developing a robotic methodology to build walls using this crane. Therefore, the methodology proposed for this project can be described as follows:

1. Obtain the equations of motion of the crane.
2. Set-up the environment and the prototype to perform realistic experiments.
3. Identify the dynamic parameters of the crane model.
4. Validate the dynamic model in order to develop a control scheme to move the crane to the desired position.

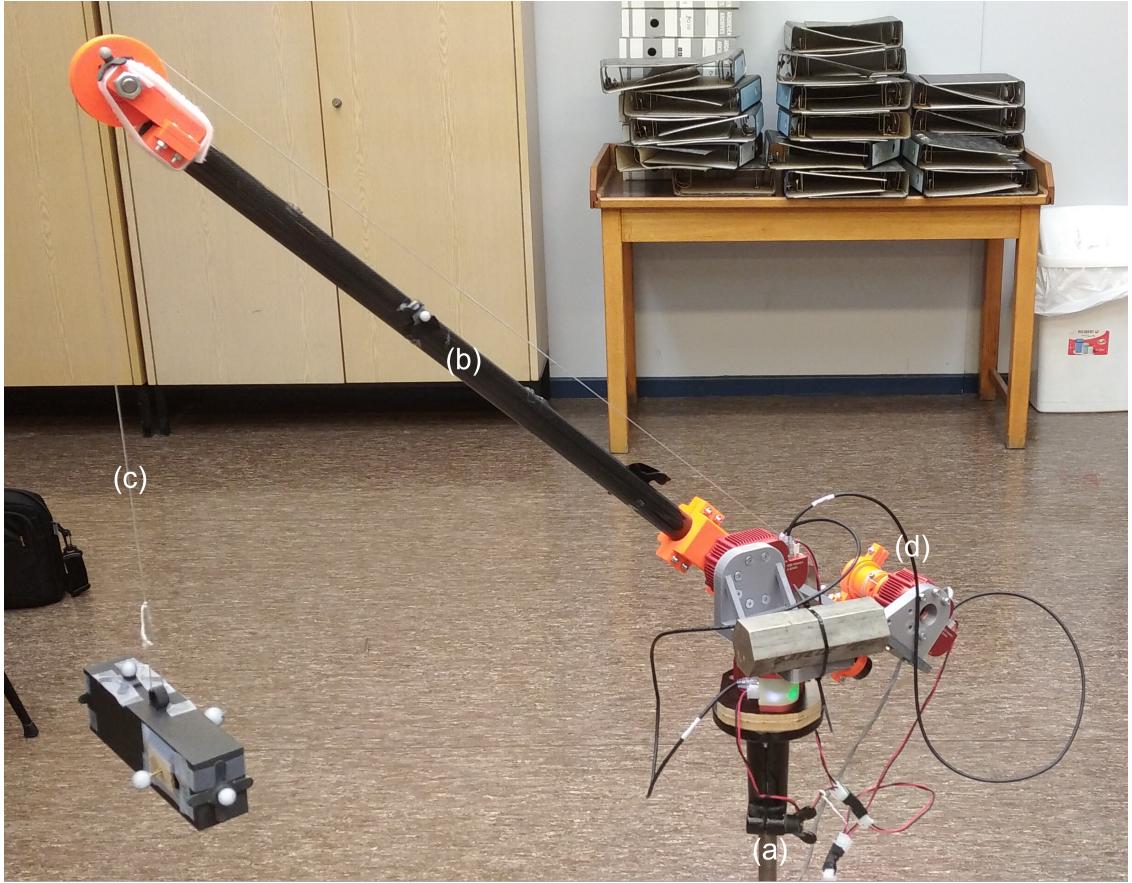


Figure 2.1: The in-scale prototype of a boom crane used for this project: (a) the base, (b) the arm, (c) the hoist rope, (d) the hoisting platform.

## 2.3 Presentation of the available material

### 2.3.1 The crane

Figures 2.1 and 2.2 are pictures of the in-scale prototype used for this project. The crane was designed and built during a previous master thesis [6]. As illustrated in Figure 2.1, it consists of:

- a *base* – also called *tower*,
- an *arm* – also called *boom*,
- a hoist rope,
- a hoisting platform with hoisting mechanism and counterbalance.

At the connection point between the base and the arm, an actuator enables the arm to rotate around the vertical  $z$ -axis of the base. This movement is called the *slew motion*. At that same point, another actuator – connected to the previous one by a  $90^\circ$  mechanical link – enables the free end of the arm to move vertically (up and down). This is the so-called *luff motion*. Finally, the crane has to hoist a payload which is suspended at the free end of the arm using a rope. To do this, the length of the hoisting rope varies with the *hoist movement*. This motion is accomplished by an actuator mounted on the other end of the arm, on the hoisting platform [6].



Figure 2.2: Another view of the crane used for this project.

### 2.3.2 The payload

Figure 2.3 shows the two different payloads used for this project. The first one (Figure 2.3a) is a block of particular shape. It has been designed in order to facilitate the placement of one block on top of another: small disturbances causing the block not to be at the exact anticipated position might in that way be disregarded as the block will slide and end up in the right position. The second payload (Figure 2.3b) is a simple rectangular block directly connected to a ring allowing the payload to be easily attached and removed from the crane.



(a) The blue payload.



(b) The black payload.

Figure 2.3: The two different payloads used for this project.

### 2.3.3 The actuators

The HEBI Robotics' X8-16 Actuators – already mounted on the crane – are used for this project. One such actuator is shown in Figure 2.4. These actuators have the particularity to include multiple built-in sensors allowing the measurement of various parameters such as position, velocity, acceleration and torque [14].



Figure 2.4: The HEBI Robotics' X8-16 Actuator [14].

Each actuator can be controlled in position, velocity and/or torque (also called effort by HEBI) using the already implemented control strategy<sup>1</sup> shown in Figure 2.5 [10]. Then, the goal is to correctly tune the parameters present in the control scheme.

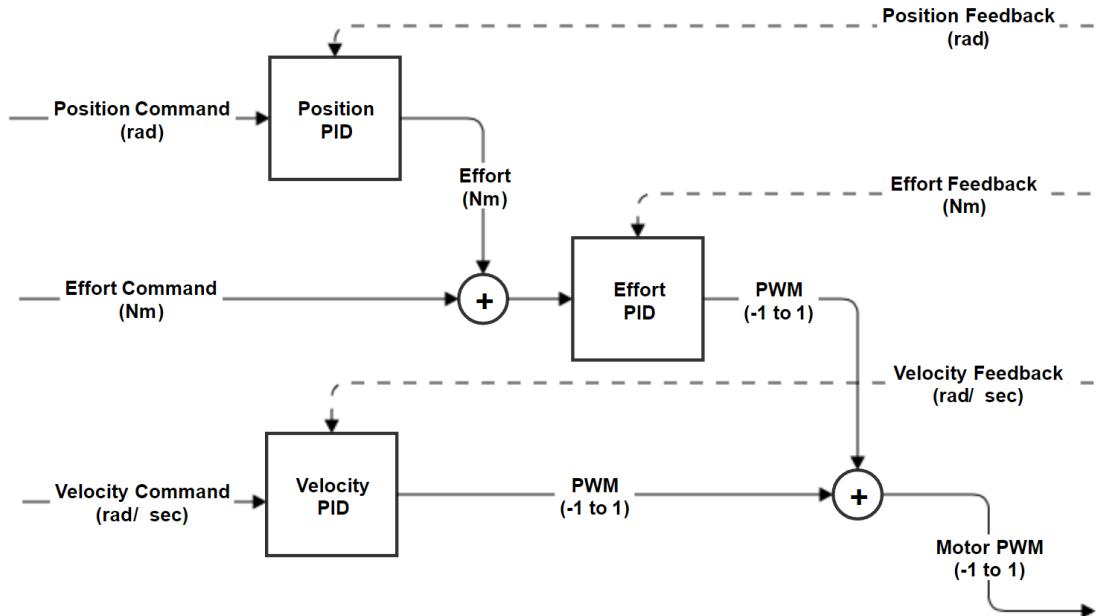


Figure 2.5: Control strategy implemented in the HEBI Robotics' X8-16 Actuator [10].

To use and control HEBI's hardware, various APIs can be used including MATLAB, Python and ROS [11]. For this project, the actuators are controlled using MATLAB and HEBI Robotics Scope – an app specifically dedicated to the real-time control and feedback of HEBI's actuators.

<sup>1</sup>Control strategies allow to combine simultaneous input commands (position, velocity, torque) to provide a single command that is sent to the actuator.

### 2.3.4 The OptiTrack system

The SAAS lab is equipped with an OptiTrack system, which is a set of 10 high-precision cameras that can locate the position of small reflective balls [9]. In order to track both the crane and the payload, several of these reflective balls have been placed on each of them. The position measurements returned by the cameras are read using MATLAB. This enables a direct processing of the read data. Moreover, the HEBI's actuators used to actuate the crane's joints can also be controlled from MATLAB. Therefore, the use of MATLAB allows the centralization of the sending of commands and the reading and processing of data [7].

# Chapter 3

## Modeling

This chapter describes the dynamic model of a boom crane and derives its equations of motion. These equations are non-linear and provide a relationship between the joint actuator torques (inputs) and the motion of the crane (output). A possible way to obtain these equations of motion is through the Euler-Lagrange formulation. Once the dynamic model of the crane is established, it can be used for simulation of motion and for the development of control algorithms [6, 15].

### 3.1 Model of the boom crane

For the modeling, the following assumptions are considered [3, 4]:

- **Assumption 1:** All the links and joints are considered to be rigid.
- **Assumption 2:** The rope is supposed to be massless and rigid, therefore the hoisting mechanism can be represented as a prismatic joint.
- **Assumption 3:** The payload and the counterbalance can be represented as point masses.

A schematic representation of a boom crane is shown in Figure 3.1. This crane – of inertia  $I_{\text{tot}}$  around its base axis ( $z$ -axis) – consists of a boom of length  $l_b$ , mass  $m_b$  and inertia  $I_b$  around the  $y$ -axis, connected to the base with two revolute joints – one for slew motion (i.e. rotation  $\alpha$  around the  $z$ -axis), the other for luff motion (i.e. rotation  $\beta$  around the  $y$ -axis). The payload of mass  $m$  is represented as a lumped mass. This can be done since the rope is assumed to be rigid (see Assumption 2 above). Therefore, the yaw, pitch and roll motions of the payload can be neglected. In addition, compared to the other models developed in [2], [4] and [6], this model takes into account the contribution of the counterbalance<sup>1</sup> present on the side of the hoisting platform. As will be shown in Section 6.4, the inclusion of this counterbalance in the model is necessary since it has a major effect on the gravity and inertia of the crane and therefore on its dynamics. This counterbalance is modeled as a lumped point of mass  $m_c$  located in the extension of the boom at a distance  $l_c$  from the connection point between the boom and the base.

---

<sup>1</sup>The term "counterbalance" is used here to refer to the combination of the hoisting platform and the counterbalance itself: see Figure 2.1 (d).

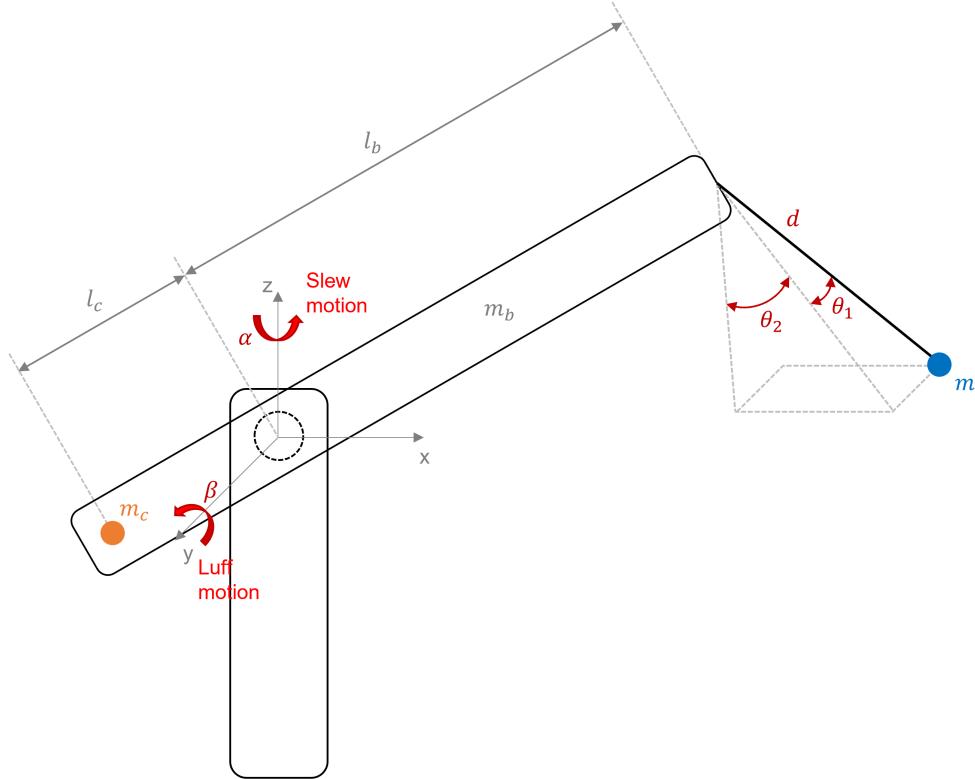


Figure 3.1: Model of a boom crane, inspired by [2].

Since the crane under study has five degrees of freedom<sup>2</sup>, its configuration is appropriately defined by five generalized coordinates, in which,  $\alpha$  is the slew angle of the base,  $\beta$  is the luff angle of the boom,  $d$  is the length of the hoisting rope,  $\theta_1$  is the payload tangential pendulation – mainly due to the rotation of the base – and  $\theta_2$  is the payload radial sway – mainly due to the motion of the boom. This is illustrated in Figure 3.1.

## 3.2 Equations of motion

The equations of motion of the boom crane shown in Figure 3.1 can be obtained using the Euler-Lagrange approach. First, the total kinetic energy  $T$  of the system must be expressed, which consists of three contributions: the boom kinetic energy, the payload kinetic energy and the counterbalance kinetic energy. For a rigid body, the kinetic energy can be decomposed into the sum of two terms: the body's center-of-mass translational kinetic energy and the rotational kinetic energy (i.e. the kinetic energy due to the rotation of the body around its center of mass). As a result, the total kinetic energy of the crane-payload system is expressed as:

$$T = \frac{1}{8} m_b (\dot{x}_b^2 + \dot{y}_b^2 + \dot{z}_b^2) + \frac{1}{2} m (\dot{x}_M^2 + \dot{y}_M^2 + \dot{z}_M^2) + \frac{1}{2} m_c (\dot{x}_c^2 + \dot{y}_c^2 + \dot{z}_c^2) + \frac{1}{2} I_{\text{tot}} \dot{\alpha} + \frac{1}{2} I_b \dot{\beta}. \quad (3.1)$$

---

<sup>2</sup>In reality, the crane should be represented with six degrees of freedom since the load could rotate on itself. However, we consider a model in which the rope is assumed rigid (see Assumption 2) and therefore the possible rotations of the payload are neglected.

Then, the total potential energy  $V$  of the system consists similarly of the boom gravity energy, the payload gravity energy and the counterbalance gravity energy [3, 6, 15] and is expressed as:

$$V = m_b g \frac{z_b}{2} + m g z_M - m_c g l_c \sin \beta . \quad (3.2)$$

The different variables appearing in these two expressions are defined in Tables 3.1-3.3.

Notation	Definition	Units
$\alpha$	Slew angle of the base	rad
$\beta$	Luff angle of the arm	rad
$d$	Length of the hoisting rope	m
$\theta_1$	Tangential oscillation angle of the payload	rad
$\theta_2$	Radial oscillation angle of the payload	rad

Table 3.1: Generalized coordinates of the crane-payload system.

Notation	Definition	Units
$g$	Standard acceleration due to gravity	$\text{m.s}^{-2}$
$I_b$	Moment of inertia of the boom including the payload, around the $y$ -axis	$\text{kg.m}^2$
$I_{\text{tot}}$	Moment of inertia of the whole crane, around the $z$ -axis	$\text{kg.m}^2$
$l_b$	Length of the boom	m
$l_c$	Distance from the counterbalance to the connection point between the base and the boom	m
$m$	Mass of the payload	kg
$m_b$	Mass of the boom	kg
$m_c$	Total mass of the hoisting platform and counterbalance	kg

Table 3.2: Parameters of the crane-payload system.

Notation	Definition	Units
$x_b, y_b, z_b$	Position of the tip of the boom	m
$x_M, y_M, z_M$	Position of the payload's center of mass	m
$x_c, y_c, z_c$	Position of the counterbalance's center of mass	m

Table 3.3: Some particular position coordinates of the crane-payload system.

With the generalized coordinates  $q = [\alpha, \beta, d, \theta_1, \theta_2]^T$ , the Lagrangian of the system can be defined as:

$$\mathcal{L}(q, \dot{q}) = T(q, \dot{q}) - V(q) . \quad (3.3)$$

Developing subsequently the Euler-Lagrange equations:

$$\frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{q}}(q, \dot{q}) - \frac{\partial \mathcal{L}}{\partial q}(q, \dot{q}) = \tau, \quad (3.4)$$

with  $\tau = [\tau_1, \tau_2, \tau_3, 0, 0]^\top$  the *generalized forces* applied to the joints [15, 19], one obtains a system of five non-linear equations that describes the motion of the crane. The mathematical development leading to these equations is fully detailed in Appendix A.

In practice, to obtain these equations, a Matlab code was used [12]. It computes the equations of motion of the crane, in particular the matrices  $M(q)$ ,  $C(q, \dot{q})$  and  $G(q)$  present in Equation 3.5. This code was provided at the beginning of the project, but some modifications have been brought. In particular, the contribution of the counterbalance has been added in this code in the form of kinetic and potential energy.

Furthermore, the system dynamics can also be rewritten in matrix form as:

$$M(q) \ddot{q} + C(q, \dot{q}) \dot{q} + G(q) = \tau, \quad (3.5)$$

with  $M(q) \in \mathbb{R}^{5 \times 5}$ ,  $C(q, \dot{q}) \in \mathbb{R}^{5 \times 5}$  and  $G(q) \in \mathbb{R}^5$  representing the inertia matrix, the centripetal-Coriolis forces and the gravity term, respectively. Note that in this equation, friction is not considered. However, for better performance, it is useful to take it into account. If this is the case, these parameters must also be identified.

As one can observe from the expression of  $\tau = [\tau_1, \tau_2, \tau_3, 0, 0]^\top$ , the boom crane presented in Figure 3.1 is an under-actuated mechanical system, having fewer independent actuators (i.e. the 3 control inputs) than system degrees of freedom (i.e. the 5 generalized coordinates). The base, boom and hoisting actuator – generating respectively torques  $\tau_1$ ,  $\tau_2$  and  $\tau_3$  – correspond to the three actuated joints, while the payload oscillation angles  $\theta_1$  and  $\theta_2$  are the non-actuated degrees of freedom [2, 4, 6].

# Chapter 4

## Dynamic parameter identification

In Chapter 3, the dynamic model (i.e. the equations of motion) of the boom crane has been established. The use of this dynamic model to solve simulation and control problems requires the knowledge of the values of the dynamic parameters of the crane. In this chapter, a procedure is proposed to estimate these parameters.

### 4.1 Different identification techniques

There are different approaches to identify the parameters of the model. One approach is to compute these parameters from the design data (i.e. the CAD model) of the crane, which is available since the crane was completely built in the SAAS lab [6]. In this way, the values of the inertial parameters of the different components can be determined on the basis of their geometry and the material employed. However, the estimates obtained by such a technique are inaccurate due to the simplifications typically introduced by geometric modeling<sup>1</sup>. In addition, complex dynamic effects, such as joint friction, cannot be taken into account.

An alternative, more heuristic, approach might be to disassemble the various components of the crane and make a series of measurements to evaluate the mass and inertia parameters. Such a technique is difficult to implement and may cause problems in measuring the quantities involved [15].

In order to find accurate estimates of the dynamic parameters, it is worth resorting to data-driven regression approaches. These techniques have the advantage of giving sufficiently accurate results using a very large amount of data and taking into account dynamic effects such as joint friction. Such an approach is presented in Section 4.2.

### 4.2 The data-driven regression approach

It was previously seen that the equations of motion of the crane can be described by the following matrix model (Equation 3.5):

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau,$$

---

<sup>1</sup>Some simplifications can be introduced when modeling certain parts, such as the geometric simplification on the shape of the actuators for instance.

known in robotics as the *manipulator dynamic model*. This equation depends on the generalized coordinates  $q = [\alpha, \beta, d, \theta_1, \theta_2]^\top$  (Table 3.1), but also on other parameters of mass, length and inertia (Table 3.2). These seven parameters – namely  $I_b$ ,  $I_{\text{tot}}$ ,  $l_b$ ,  $l_c$ ,  $m$ ,  $m_b$  and  $m_c$ <sup>2</sup> – must be determined in order to obtain a model which, from certain torque inputs  $\tau$ , returns the motion of the crane (i.e.  $q$ ,  $\dot{q}$ ,  $\ddot{q}$ ). Note that to be totally accurate, one should have included the viscous friction coefficient and the Coulomb friction coefficient in the model since they also influence the dynamics of the crane. In doing so, these two additional parameters are also to be estimated.

One must therefore have a way to obtain equations where the unknowns are these dynamic parameters (contained in a certain vector  $\pi$ ), and where the variables depending on  $q$ ,  $\dot{q}$  and  $\ddot{q}$  are considered known. For this project, the proposed procedure follows the approach described in [15] and exploits the linearity property of the manipulator model with respect to an appropriate set of dynamic parameters:

$$\tau = Y(q, \dot{q}, \ddot{q})\pi . \quad (4.1)$$

Given  $n$  the number of degrees of freedom and  $p$  the number of groups of unknown parameters,  $\pi$  is the parameter vector of dimension  $(p \times 1)$  and  $Y$  is a  $(n \times p)$  matrix – called *regressor* – that is a function of joint positions, velocities and accelerations. This is simply a reformulation of Equation 3.5, highlighting the unknown constant parameters of the system to determine. A large number of measurements then allows for a very large set of equations in these parameters, which leads to a least-squares optimization problem to find their numerical value.

First, one needs to identify the unknowns present in Equation 4.1. This is done by inspecting the analytical expression of Equation 3.5. For example, if the factor  $l_b m_b$  multiplies a term – whose parameters depend only on  $q$ ,  $\dot{q}$  and/or  $\ddot{q}$  – in one of the equations, it constitutes one of the parameters that needs to be found<sup>3</sup>. All these parameters are then stacked in the vector  $\pi$  which, multiplied by the corresponding numerical regressor  $Y$ , should give the joint actuators torque inputs  $\tau$ . Concerning the studied crane, 9 groups of parameters can be extracted. Therefore, the  $\pi$  vector can be expressed analytically by the  $(9 \times 1)$  following vector:

$$\pi = [I_{\text{tot}}, l_b^2 m_b, l_b^2 m, l_b m, m, l_b m_b, I_b, l_c^2 m_c, l_c m_c]^\top . \quad (4.2)$$

It is then interesting to take a closer look at the regressor  $Y$ . This matrix must contain the terms multiplying the unknown parameters contained in  $\pi$  in the equations. It is represented as a symbolic matrix in MATLAB, and its term in the  $i^{\text{th}}$  row and  $j^{\text{th}}$  column is the coefficient multiplying the  $j^{\text{th}}$  unknown parameter in the  $i^{\text{th}}$  equation. This symbolic expression depends only on the variables  $q$ ,  $\dot{q}$  and  $\ddot{q}$ , so that a numerical expression of the matrix  $Y$  can always be found if the numerical values for these three vectors are available. Since the collected data contains the values of these three vectors (see Section 5.1.3), it is indeed possible. The

---

<sup>2</sup>The parameter  $g$  (standard gravity) does not appear in the above list since it is considered constant and equal to  $9.81 \text{ m.s}^{-2}$ .

<sup>3</sup>It is important to note that if  $\pi_1 = m_b$  and  $\pi_2 = l_b$  are already parameters contained in  $\pi$ , it is wrong to consider that a term  $l_b m_b$  in an equation is equivalent to  $\pi_2 \pi_1$ . Instead, a new parameter  $\pi_3 = l_b m_b$  should be defined and also append to the vector  $\pi$ .

matrix  $Y$  has been found once and for all along the vector  $\pi$  by inspecting the equations of the crane in MATLAB. Its analytical expression can be found in [12].

As mentioned before, the general idea here is to obtain a large system of equations where the unknowns are the groups of constant parameters of the crane equations that need to be calculated. The purpose is to collect a large amount of data – in which the positions  $q$  can be measured and the velocities  $\dot{q}$  and accelerations  $\ddot{q}$  can be computed from the position measurements – and to use each of these collected samples to obtain a set of five equations in the form of Equation 4.1<sup>4</sup>. From there, a least-squares optimization problem can be solved with the many equations obtained to find the values of the unknown parameters that best fit the data.

---

<sup>4</sup>For example, a set of data with 5,000 samples implies a total of 25,000 equations. The value of the actuator torques must also be known for each sample.

# Chapter 5

## Experimental procedure

### 5.1 Data collection

#### 5.1.1 Data acquisition from the OptiTrack cameras

As mentioned in Section 2.3.4, the lab is equipped with a set of high-precision cameras, which enables an efficient trajectory tracking of the reflective balls. It is therefore possible to follow the movements of both the crane and the payload by distributing these reflective balls on them.

In order to reduce the impact of the environment in which the crane operates, the device must first be calibrated. To do this, a wand with three reflective balls was first moved around the room for a sufficient amount of time. Then, a ground plane was placed to define the floor and the different axes on Motive<sup>1</sup> [5]. Figures 5.1 and 5.2 show the result of such a calibration.

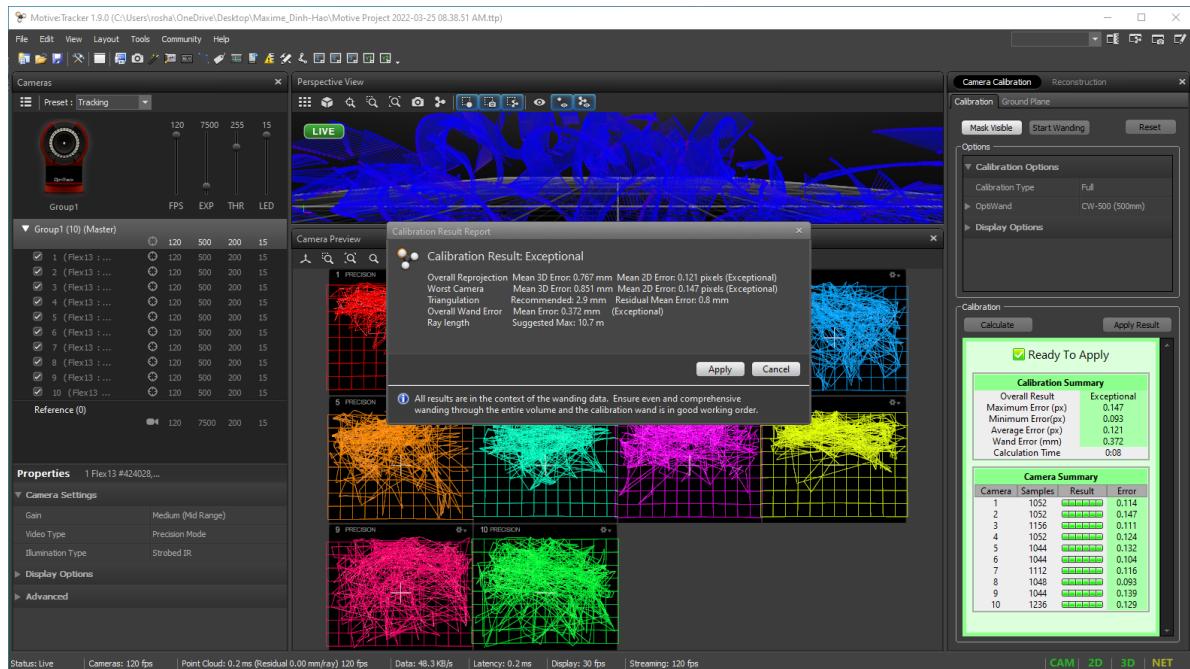


Figure 5.1: Example of OptiTrack camera calibration.

<sup>1</sup>Motive is a specific software for OptiTrack cameras that allows to visualize in real-time the information returned by the cameras.

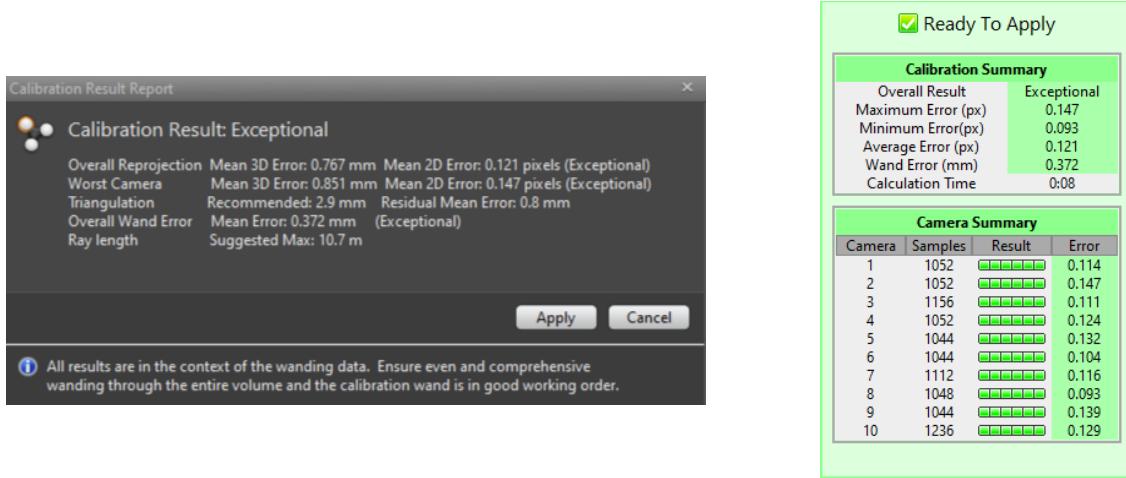


Figure 5.2: Results obtained from the calibration.

Once this step is completed, the crane can be brought in front of the cameras. Motive can then detect the reflective balls scattered all over them. Their movements can be tracked with great precision<sup>2</sup> and can be visualized in real-time with Motive.

However, at this stage, this does not provide direct information about the crane and the payload themselves. For this purpose, a set of reflective balls can be regarded as a rigid body, which allows the motion of each reflective ball to be associated with the crane or the payload. Once this is done, information about the motion of each rigid body is available.

It is important to note that Motive initializes the current pitch, yaw and roll angles of any newly defined rigid body to zero. This is of great importance for data manipulation and for checking that everything is defined correctly with respect to the model. The pivot point can also be moved, which is very interesting for the rigid body associated with the crane. Indeed, the default pivot point is the center of mass of the rigid body, whereas it is useful to have a pivot point at the top of the boom<sup>3</sup>. Figure 5.3 shows a screenshot from Motive where both the crane and the payload have been defined as rigid bodies and the crane's pivot point has been moved to the top of the boom.

Finally, some settings need to be changed on Motive in order to get the data on MATLAB.

- In Data Streaming, set *Stream Rigid Bodies* to *True*,
- In Data Streaming, set *Local Interface* to *loopback*<sup>4</sup>.

Once these parameters have been modified, if the cameras have been properly calibrated and if the rigid bodies have been correctly defined, the data are ready to be used in MATLAB.

<sup>2</sup>The precision depends on the quality of the calibration.

<sup>3</sup>This is very useful for easily calculating many parameters, such as the length of the rope  $d$  and the payload oscillation angles  $\theta_1$  and  $\theta_2$ .

<sup>4</sup>to collect the data back to the computer

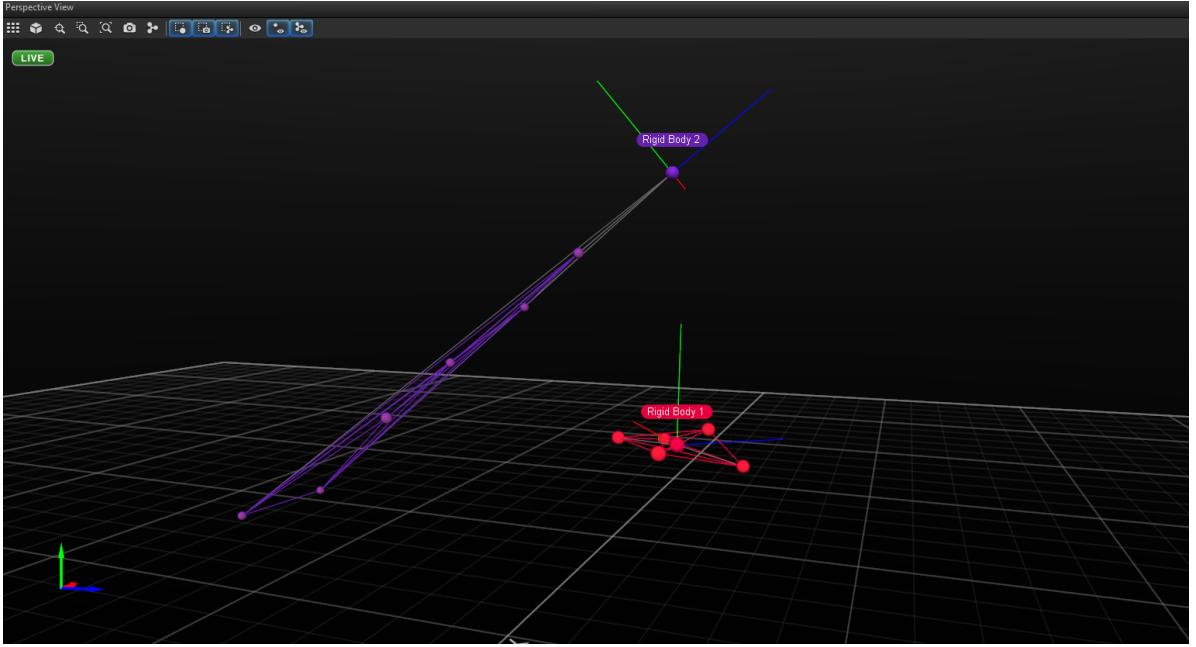


Figure 5.3: Definition of the rigid bodies on Motive (Rigid Body 1: the payload and Rigid Body 2: the crane).

### 5.1.2 From Motive to MATLAB

Once Motive is properly configured, it is possible to obtain its data in real-time on MATLAB. Throughout this project, a previously existing code [7] has been used to translate the information from the cameras. This code also provided a real-time plot of certain quantities, which was not necessary for the model identification procedure, but was nonetheless very useful for verifying that the measured data corresponded to reality.

However, an error was noticed in this code: the definition of the angles  $\theta_1$  and  $\theta_2$  was done in an absolute frame when it should be done in a relative frame attached to the crane. This problem is discussed in details in Section 6.3. The solution is to simply implement a rotation matrix. The code has also been improved on other aspects [12].

### 5.1.3 Processing and storage of the data on MATLAB

The data obtained from Motive contain the roll, pitch and yaw angles of each rigid body, and the position of their respective pivot point. The pitch and yaw angles of the rigid body corresponding to the crane can be used directly, as they are already degrees of freedom of interest –  $\alpha$  and  $\beta$  respectively. The length of the rope  $d$  and the swing angles  $\theta_1$  and  $\theta_2$  (shown in Figure 3.1) are deduced from the measured positions of the pivots of the crane and the payload. The length of the rope corresponds to the distance between these two pivot points. As for the oscillation angles, they can be constructed using the trigonometric formulas. Equations 5.1, 5.2 and 5.3 detail these computations.

$$d = \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2}, \quad (5.1)$$

$$\theta_1 = -\arcsin\left(\frac{\Delta y}{\sqrt{\Delta x^2 + \Delta z^2}}\right), \quad (5.2)$$

$$\theta_2 = \arcsin \left( \frac{\Delta x}{\sqrt{\Delta y^2 + \Delta z^2}} \right), \quad (5.3)$$

where  $[\Delta x, \Delta y, \Delta z]$  represents the vector connecting the crane's pivot to the payload's pivot, expressed in a relative reference frame.

The value of each degree of freedom varies with the samples. All the collected data can be stored in vectors. For example, as shown in Figure 5.4, the value of the time for each sample is collected and stored in a vector, which is useful to compute time derivatives and to compare results with simulations. In the case of the degrees of freedom, their values are also stored in vectors during data acquisition, but when it comes to processing the data, these vectors are overlaid in a matrix for easier manipulation [12].

theta1	-1.5212
theta1_vec	1x5000 double
theta2	1.4253
theta2_vec	1x5000 double
time	1x5000 double
torque	[-0.6224, 1.1285, ...
torque_vec	5000x3 double

Figure 5.4: Illustration of the methodology used to store data in MATLAB. Only a fraction of the MATLAB workspace is shown here, more variables are actually stored from an experiment.

All the parameters mentioned above are directly obtained from Motive data. The last important values to measure in Equation 4.1 are the torques applied by the actuators. These values are obtained directly in MATLAB by reading the actuator's torquemeter, and are stored in matrices like the rest of the data. This idea of progressive storage of real-time measured data in matrices is illustrated in Figure 5.4, which partially shows what a workspace looks like once the data acquisition has been completed.

On the one hand, the objective for each sample is to obtain an equation of the form of Equation 4.1 with numerical values for in the matrix  $Y$  and in the vector  $\tau$  (see Section 4.2), so that a least-squares algorithm can be used with the ( $5 N_{\text{samples}}$ ) obtained equations. The vector  $\tau$  is already filled with the numerical values read from the actuators<sup>5</sup>, but the matrix  $Y$  depends on velocities and accelerations. As the measurements from the cameras do not provide them, they need to be computed from the measured positions and the time measurements. A MATLAB function has been created for that purpose [12]. This function computes the time derivative of a matrix  $A$  following the classical formula:

$$dA_i = \frac{A_i - A_{i-1}}{t_i - t_{i-1}} \quad (5.4)$$

where the index  $i$  corresponds to the column of the matrix  $A$ , in other words to the index of the sample. The first column of the  $dA$  matrix is initialized to 0.

---

<sup>5</sup>Two lines are added with a zero value for dimensional consistency, as the system is under-actuated.

On the other hand, the data also needs to be filtered. Indeed, the presence of noise could ruin the obtained data when the time derivatives are computed, and it could lead to bad results for the least-square optimization problem. To solve this issue, a Matlab function filtering the data has been created [12]. However, the measurements to filter don't have all the same characteristics, so that it is very complicated to write a single function processing everything correctly. For example, the torque measurements show a lot of noise, and the useful frequencies for the swing angles  $\theta_1$  and  $\theta_2$  are sensibly higher than the ones for  $d$ . In addition, robots are prone to pink noise, which make the definition of the cut-off frequency more delicate. But since common points can still be found among all these signals, many solutions have been investigated to try having a function automatically building the right filter, based on the spectral content of the input signal. As explained in Chapter 9, finding such a function can considerably fasten the whole identification procedure. For now, the filtering is done on a case-by-case basis. In general, Butterworth filters are preferred because of the low distortion they induce in their passband. The data were chosen to be filtered when it is still in degrees, because this entails a larger scale as in radians. Converting into radians before filtering might therefore lead to loss of information. The Figure 5.5 shows an example of filtering, in this case for the  $\theta_2$  angle.

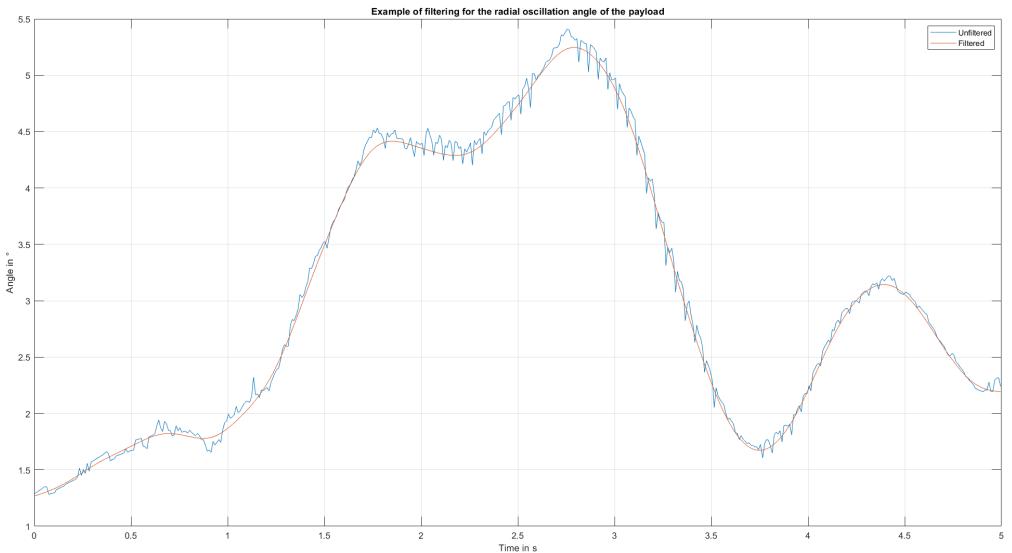


Figure 5.5: Example of filtering for the  $\theta_2$  angle

Another function has been added after noticing that the data could be altered by discontinuities [12]. In some cases, some of the angles obtained from Motive could jump from  $180^\circ$  to  $-180^\circ$ , leading to very high and narrow peaks in the velocity. Filtering this uncorrected data leads to highly inaccurate results, as those exceptional peaks happen in very particular situations and are therefore not associated with a high frequency (making the low-pass filter inefficient to treat them). A function has been written to make sure to avoid such big discontinuities, wherever they come from. The function looks at the samples one by one. If one sample has a value separated of more than  $300^\circ$  from the previous one, the code corrects the data by adding or subtracting  $360^\circ$  to this value. This is done before filtering the data. The Figure 5.6 illustrates this problem. It is important to note that this situation has been encountered very early in the project. Careful initialization of the rigid bodies on Motive avoids reaching such high values for the angles.

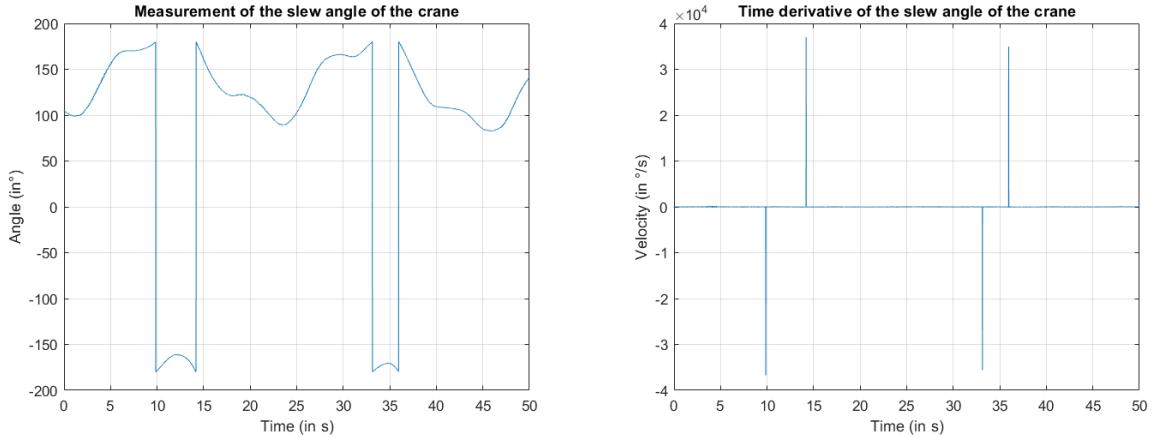


Figure 5.6: Illustration of a problem linked to the discontinuity in the angle definition.

Finally, the time vector retrieved from Matlab was showing some irregularities. Plotting the time lapse between each sample allowed to highlight this phenomena (see Figure 5.7). Those discontinuities are not problematic as long as the values stored in the time vector indeed correspond to the moments where the values have been measured. However, seeing this kind of problem, it is conceivable to have unexpected technical issues, leading to irregular sampling time in the stored time vector which would not correspond to the actual sampling time. To be sure to avoid this kind of problem, another Matlab function has been created [12]. This function analyses the amount of time between the  $i^{\text{th}}$  and  $(i-1)^{\text{th}}$  samples, and it compares it to the amount of time between the two previous samples, which are the  $(i-1)^{\text{th}}$  and  $(i-2)^{\text{th}}$  ones. If the value is considered abnormally high or low compared to the previous one, it is fixed to a lower or upper bound and the values of the collected data is reconstructed by extrapolating the evolution that led to the previous sample. This is of course done before computing any time derivative.

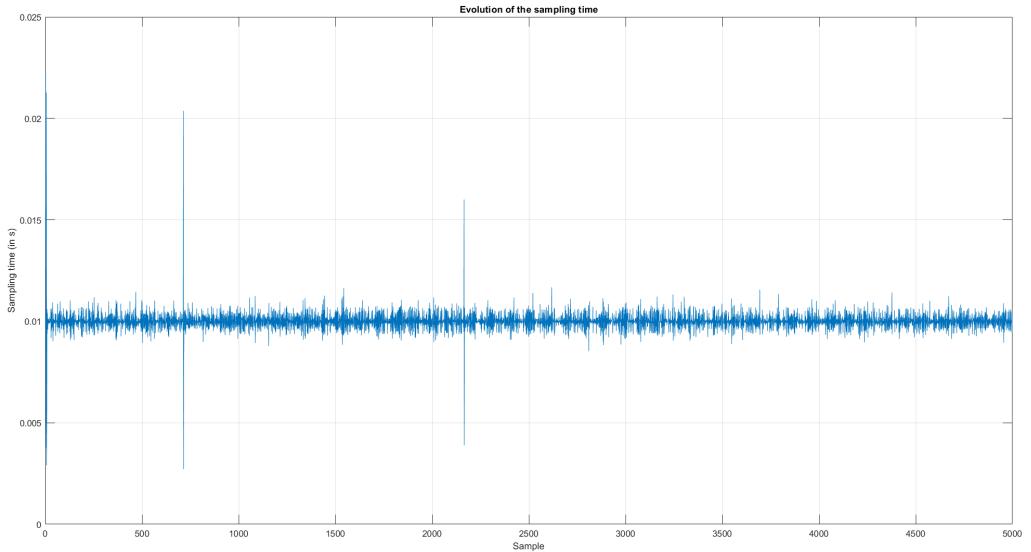


Figure 5.7: Illustration of irregularities in the sampling time.

It is also worth noting that a relatively small fraction of the very first samples systematically has to be discarded for two reasons: on the one hand, the sampling time shows

important irregularities in the beginning, and on the other hand, the real crane sometimes shows very slight unwanted motions when it begins to follow a trajectory. The number of samples to discard is determined by looking at plots of the raw data. It typically represents 10 to 15 samples, with a certain margin of security.

To summarize, the treatment of the data and preliminary computations are done in the following order :

1. Vectors containing the values of the variables of interest for each sample are created.
2. A small number of the first samples is discarded.
3. The vectors corresponding to the degrees of freedom are grouped into a matrix for easier manipulation of the data.
4. The data is first corrected to get rid of unwanted discontinuities.
5. The data is filtered, for now on a case-by-case basis.
6. The angles are converted to radians<sup>6</sup>.
7. The velocity matrix and acceleration matrix are computed **and** filtered.

---

<sup>6</sup>This corresponds to 4 out of the 5 degrees of freedom, the length of the rope  $d$  being expressed in meters.

## 5.2 From the data to the least-squares algorithm

As explained in Section 4.2, a set of five equations of the form 4.1 can be associated to each sample, the unknowns of the problem being the coefficients of the vector  $\pi$ . The values of the vector  $\tau$  were directly obtained from the actuators and filtered, and the values of the positions, velocities and accelerations obtained after the treatment of the data can be substituted in the symbolic matrix  $Y$  in order to obtain a matrix with numerical coefficients.

The objective is to solve a least-squares algorithm with the  $(5 N_{\text{samples}})$  equations obtained with the data. To do so, a  $(5 N_{\text{samples}} \times 1)$  vector  $\tau_{\text{tot}}$  and a  $(5 N_{\text{samples}} \times 9)$  matrix  $Y_{\text{tot}}$  are created and progressively filled sample by sample thanks to 'for' loops [12].

The  $\tau_{\text{tot}}$  column vector is constructed by simply superposing the measurements from the actuators. But as the boom crane is an under-actuated system, two lines with null values need to be added for each samples. This corresponds to the absence of actuators for the degrees of freedom concerning the payload and ensures dimensional consistency.

The same principle goes for the matrix  $Y_{\text{tot}}$ , except that it is now  $5 \times 9$  matrices that are being superposed for each sample. As each position, velocity and acceleration have to be substituted by the measured values for each sample in the 'for' loop, this operation is more time consuming. A waitbar has therefore been added in the code in order to make sure that everything is running properly, and to avoid waiting for nothing in case of problem.

In line of principle, once the matrices  $\tau_{\text{tot}}$  and  $Y_{\text{tot}}$  have been obtained, the least-squares algorithm can be launched thanks to the Matlab function `lsqr`. However, as the rank of  $Y_{\text{tot}}$  is lower than the number of parameters to be determined, three very simple equations involving easily measurable parameters have been added. The validity of this approach has been verified with Simulink, as it is explained in Section 5.3.2.

## 5.3 Simulink model

A virtual model of the crane has been implemented on Simulink. The use of this model has had multiple benefits, from validating the identification code to validating the model of the crane. This Simulink model can be found in [12].

### 5.3.1 Presentation of the Simulink model

The Simulink model of the boom crane is shown in Figure 5.8.

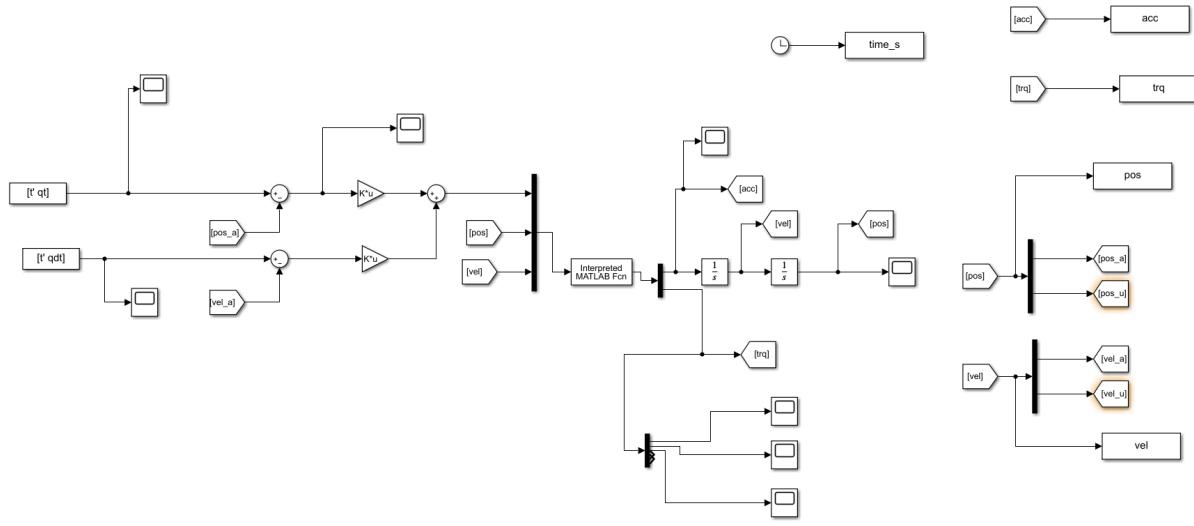


Figure 5.8: Simulink model of the crane

The left part of the model contains the controller. The only purpose of this controller is to make sure that the crane follows the imposed trajectory<sup>7</sup>. The interpreted MATLAB function block is where everything takes place. This allows to use a MATLAB function, which translates how the crane is supposed to behave using the boom crane dynamic model with consideration of the counterbalance in which the unknown parameters have been fixed [12]. The torques computed with the simple PD controller are put as inputs to this function, along the positions and velocities. The gravity compensation part is added in the code, as this is an easier way to implement it. As the code fixes the values of the unknown parameters, the accelerations are the only remaining unknowns and can be computed from the rest. The function has those accelerations as outputs, which allows to obtain the positions and the velocity thanks to integrators<sup>8</sup>. The torque is also an output, as it is required for some purposes like the validation of the regressor approach and the identification of potential issues.

In this case, the controller is a PD controller with gravity compensation. The PD part can be seen on Figure 5.8, while the gravity part is added in the MATLAB function.

Scopes can be found everywhere, as they help to visualize what the model is doing and to identify potential problems.

<sup>7</sup>Which is the qt and the qdt on Figure 5.8, during the time t.

<sup>8</sup>Initial positions and/or velocities can be set with those integrators.

Finally, the useful data from the simulation is sent to the MATLAB workspace thanks to "To Workspace" blocks. This allows for example to try using the regressor approach with the Simulink data, or to compare this data to real collected data.

### 5.3.2 Validation of the identification code with Simulink

The Simulink model can be used to verify that the code putting everything in place for the least square optimization problem works adequately. To do so, the values of the unknown parameters are arbitrarily fixed in the Matlab function from the Interpreted MATLAB Function bloc. Some trajectory is then sent to the virtual crane on Simulink. The positions, velocities, accelerations and torques are retrieved from Simulink, and the identification code (see [12]) can be run with that data. If the identification code works adequately, the least square algorithm should result in finding the arbitrary values of the parameters that were initially fixed in the MATLAB Function.

In the end, this procedure was successfully completed, showing that the identification code works as intended. But it also allowed to identify and solve problems along the way.

#### Problem linked to the rank of the matrix $Y$

Using this approach allowed to identify more effectively and to solve a serious issue with the identification code. It is worth noting that this was done when the counterweight part was still not taken into account<sup>9</sup>. When using the approach explained above, the vector  $\pi$  resulting from the least square approach with the Simulink data systematically had the same two wrong components. As the other 5 components were correct, this implied that the rank of the regressor  $Y$  was too low in order to determine all the parameters. After verification on MATLAB, this was indeed the case: the rank of the  $Y_{\text{tot}}$  matrix (obtained from the identification code) is 6 instead of 7. To deal with this issue, as one of the wrong parameters appeared to be easily measurable, it was chosen to add an equation<sup>10</sup> fixing this parameter to the measured value. The procedure with Simulink then lead to the expected results.

It is worth noting that the identification code used with real data was leading to wrong results before **and** after this correction. The use of Simulink therefore allowed to identify and solve one of several issues in the identification code. This problem would have been harder to notice among the others.

This reasoning can be easily extended to the case where the counterweight part is taken into account. In this case, the vector  $\pi$  contains 9 elements instead of seven, and the regressor  $Y$  is a  $5 \times 9$  matrix<sup>11</sup>. As a quick look at MATLAB showed that the rank of  $Y_{\text{tot}}$  is still 6, this means that two additional equations are required to find the vector  $\pi$ . As the new parameters depend on  $m_c$  and  $l_c$  (see Figure 3.1), which are both easily measurable, their value could be imposed. This also led to correct results with the Simulink data, confirming the validity of the approach.

---

<sup>9</sup>So the vector  $\pi$  only had 7 components. The incoming conclusion however extends to the model taking the counterweight into account.

<sup>10</sup>Adding an equation means adding a new line in the  $Y_{\text{tot}}$  and the  $\tau_{\text{tot}}$  matrices.

<sup>11</sup>NB : If  $Y$  is a  $5 \times 9$  matrix, then a set of data containing 5000 samples leads to a  $25,000 \times 9$  matrix for  $Y_{\text{tot}}$

### **5.3.3 Use of the Simulink model to help identifying problems in the data**

The virtual model of the crane is also useful to identify other potential problems. Simple trajectories can indeed be sent both to the real crane and to the Simulink model, in order to see whether there is a sufficient matching between the observed behaviors. Of course, the obtained curves are not going to match perfectly as arbitrary values of the unknown parameters are used in the Simulink model. But if those values make sense physically (with rough measures and estimations), it is interesting to look at the shape of the obtained curves and compare them to the ones obtained from the real measurements. Big mismatches can be indicators of what potential problems need to be investigated.

For example, this approach showed important mismatches for the torques and the angles  $\theta_1$  and  $\theta_2$ , whereas the 3 other degrees of freedom were more consistent with what was seen on simulation. This allowed to identify major problems in the way  $\theta_1$  and  $\theta_2$  were defined (see Section 6.3).

### **5.3.4 Potential use to validate the model of the crane and the controller**

Having a virtual model of the crane is very useful to verify that the simulations associated to the results match with reality. To do so, the same trajectory can be sent both to the crane and to the Simulink model, and the curves can be superposed to see whether a sufficient match has been reached. The controller can also be added in order to verify that it is working as expected.

# Chapter 6

## Discussion of the experimental results

This chapter discusses the current results of the experiments performed on the physical crane presented in Section 2.3.1. It is important to note that a very large number of graphs and experimental results have been obtained throughout the project, leading to all the codes and solutions presented in the previous chapters. The purpose of this chapter is first to show the experimental results currently obtained, and then to discuss the main problems encountered. All of them have been solved, except for the non-rigidity of the base (Section 6.5), which is a problem related to the design of the crane itself<sup>1</sup>. This latter results in significant offsets in the base actuator torque, making the least-squares approach inefficient to provide an accurate estimation of the model parameters. However, a good match between the actual behavior and the theoretical predictions from the Simulink model can already be observed.

Before presenting the results, it is important to note that the length of the rope  $d$  has been computed by multiplying the angular position of the hoisting actuator (i.e. the one responsible for the variation in the rope length) with the radius of the drum attached to it and around which the rope is winding.

### 6.1 Comparison of predicted and actual behavior of the crane

Many simple polynomial trajectories have been generated and sent to the physical crane. The same trajectories were then sent to the virtual model of the crane on Simulink to see whether some correspondence could be observed between the real experiment and the simulation.

All tests lead to the same conclusion: the data collected by the cameras and processed with the identification code are in excellent agreement with the simulation data for the first three degrees of freedom, namely  $\alpha$ ,  $\beta$  and  $d$ . The payload oscillation angles  $\theta_1$  and  $\theta_2$  also show experimental and theoretical curves of similar shapes, where the mismatches can be explained by inaccurate values of the dynamic parameters used in the Simulink model.

For comparison, a certain trajectory was sent to both the real crane and its virtual Simulink model. The measurements obtained by both are shown in Figure 6.1.

---

<sup>1</sup>It is therefore slightly outside the initial scope of this project.

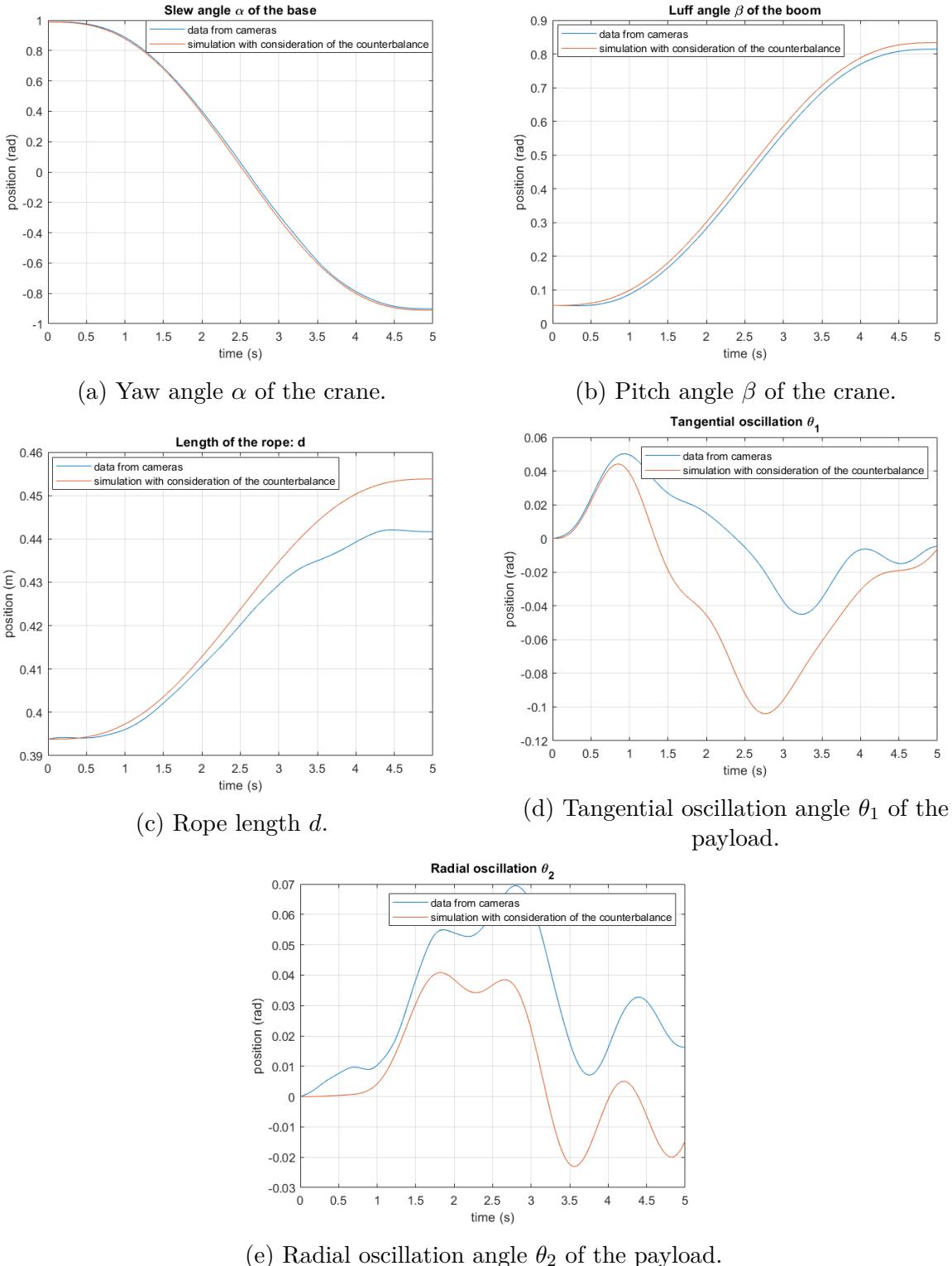


Figure 6.1: Comparison between the experimental measurements (in blue) and the simulation results (in red), for the same trajectory.

## 6.2 Results obtained with the least-squares algorithm

Sufficiently random<sup>2</sup> trajectories must be sent to the real crane<sup>3</sup> in order to identify the unknown parameters through a least-squares optimization problem. Sinusoidal functions have been used as commands for the actuators' velocity in order to obtain a sufficiently random motion of the crane (see Figure 6.2). During these experiments, data have been obtained and processed using the equipment and Matlab, as described in Sections 5.1.2 and 5.1.3.

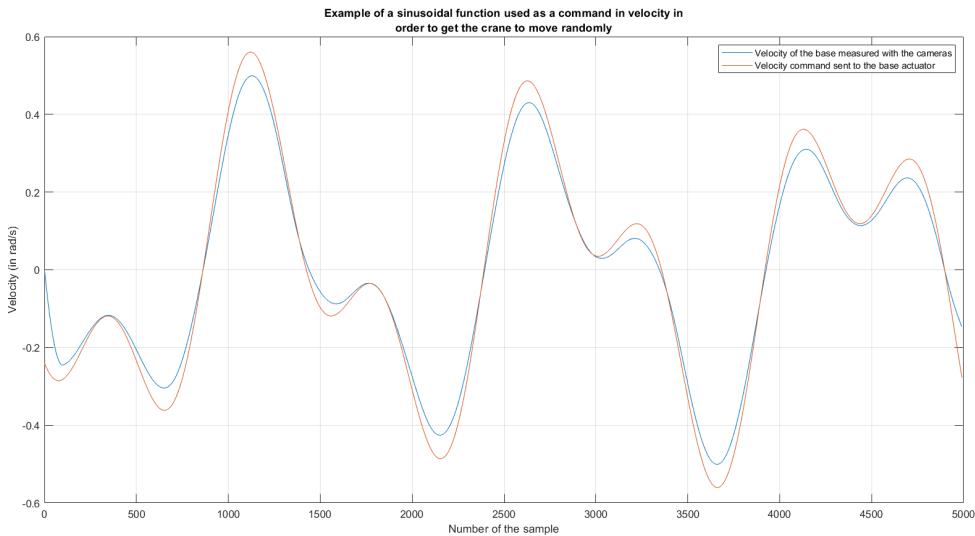


Figure 6.2: Example of a sinusoidal function used as a command in velocity for one of the actuators in order to get the crane to move randomly.

As explained in Section 5.2, once the crane positions, velocities and accelerations as well as the torques generated by the actuators have been computed and filtered, each sample of the data yields a set of five equations in the unknown variables (i.e. the model parameters to identify). Then, a least square optimization problem can be solved for a sufficiently large number of samples, which is indeed the case as shown in Figure 6.2.

Many different tests have been performed, but the least-squares algorithm has not yet led to physically acceptable results. Negative values are indeed found, whereas the inertia terms can not be negative. The least-squares algorithm has therefore been performed with additional constraints on the values of some unknown measureable parameters (such as the mass of the payload  $m$ , the mass of the counterbalance  $m_c$ , the length of the boom  $l_b$  and the distance of the counterweight to the connection point between the base and the boom  $l_c$ ), but without success yet.

However, the amount of experiments has led to increasingly satisfactory results and to the identification of the most-likely source of the problem, which is explained at Section 6.5.

---

<sup>2</sup>Those were non-optimal trajectories. Using optimal trajectories could have provided more accurate results.

<sup>3</sup>A video showing the crane in operation during this experiment can be found at the following link: <https://www.youtube.com/shorts/0ZIyQ80FdGc>.

## 6.3 Identification and resolution of a problem related to the definition of the payload oscillation angles

The payload oscillation angles  $\theta_1$  and  $\theta_2$  are directly computed in the code which collects the data from the cameras on Matlab[7]. However, it was noticed that these angles were defined in an absolute reference frame and not in a relative one, which led to completely wrong measurements. The problem was identified by performing the same experiment twice, but with a  $90^\circ$  difference for the initial slew angle  $\alpha$ . When comparing the data from those experiments, it has been noticed that the  $\theta_1$  angle of the first experiment corresponds to the  $\theta_2$  angle of the second experiment, and vice-versa. This means that the tangential and radial oscillation angles are always defined in the same orientation regardless of the orientation of the crane, which is not consistent. Since the code was provided at the beginning of the project, it took a long time to spot this issue, and a lot of investigation has been done throughout all the scope of this project before investigating the provided resources.

The angles  $\theta_1$  and  $\theta_2$  have been defined in Figure 3.1. As explained in Section 5.1.1, the data obtained from Motive does not provide them directly, but it provides the position of the pivots of the two rigid bodies. The pivot of the crane having been placed on the tip of its boom and the pivot of the payload at its center of mass, the oscillation angles were constructed in the provided Matlab code [7] using equations 5.2 and 5.3, which are recalled below

$$\theta_1 = -\arcsin \left( \frac{\Delta y}{\sqrt{\Delta x^2 + \Delta z^2}} \right), \quad (6.1)$$

$$\theta_2 = \arcsin \left( \frac{\Delta x}{\sqrt{\Delta y^2 + \Delta z^2}} \right), \quad (6.2)$$

where  $[\Delta x, \Delta y, \Delta z]$  represents the vector connecting the crane's pivot to the payload's pivot. As the position of those pivots obtained from Motive's data correspond to measurements in Motive's absolute frame, it is obvious that  $\theta_1$  and  $\theta_2$  were erroneously defined.

To solve this problem, it is sufficient to simply multiply the crane-pivot-to-payload-pivot vector  $[\Delta x, \Delta y, \Delta z]^\top$  by the rotation matrix around the  $z$ -axis:

$$R_z(\alpha) = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (6.3)$$

with  $\alpha$  the already-defined slew angle of the crane base. Each sample has a different rotation matrix, depending on  $\alpha$ .

This modification was directly made to the code collecting the data. In doing so, a new vector is defined:

$$\begin{bmatrix} \Delta x' \\ \Delta y' \\ \Delta z' \end{bmatrix} = R_z(\alpha) \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix}. \quad (6.4)$$

This rotation matrix allowed to express the crane-pivot-to-payload-pivot no longer in an absolute frame, but in the relative frame rotating with the crane base. By applying such a transformation, the following variables also need to be redefined:

$$d = \sqrt{\Delta x'^2 + \Delta y'^2 + \Delta z'^2}, \quad (6.5)$$

$$\theta_1 = -\arcsin\left(\frac{\Delta y'}{\sqrt{\Delta x'^2 + \Delta z'^2}}\right), \quad (6.6)$$

$$\theta_2 = \arcsin\left(\frac{\Delta x'}{\sqrt{\Delta y'^2 + \Delta z'^2}}\right). \quad (6.7)$$

This modification in the provided code allows to obtain results that make much more sense when compared with simulations.

## 6.4 Effect of including the counterbalance in the model

The Simulink model uses an interpreted Matlab function block which allows to use a function translating the predicted behavior of the crane for arbitrary values of the model parameters, as explained in Section 5.3.1. This function is not the one provided at the beginning of the project [6]. It is a modified version which allows to take into account the contribution of the counterbalance. Both functions work in the same way, the main difference resides in the matrices they use for the crane equations. It is possible to verify whether this modification of the model has had a beneficial effect on the results. To do this, the same simulation as in Section 6.1 was run, but this time, with the old function in the interpreted MATLAB function block. Of course, the unknown dynamic parameters were set to the same values so that the compared curves are indeed comparable. The comparison led to the comparative graphs shown in Figure 6.3.

The modification of the crane model to include the contribution of the counterbalance clearly leads to a better match between the model-based simulation and the actual behavior of the crane. The need to consider the counterbalance in the model is an important conclusion of this project.

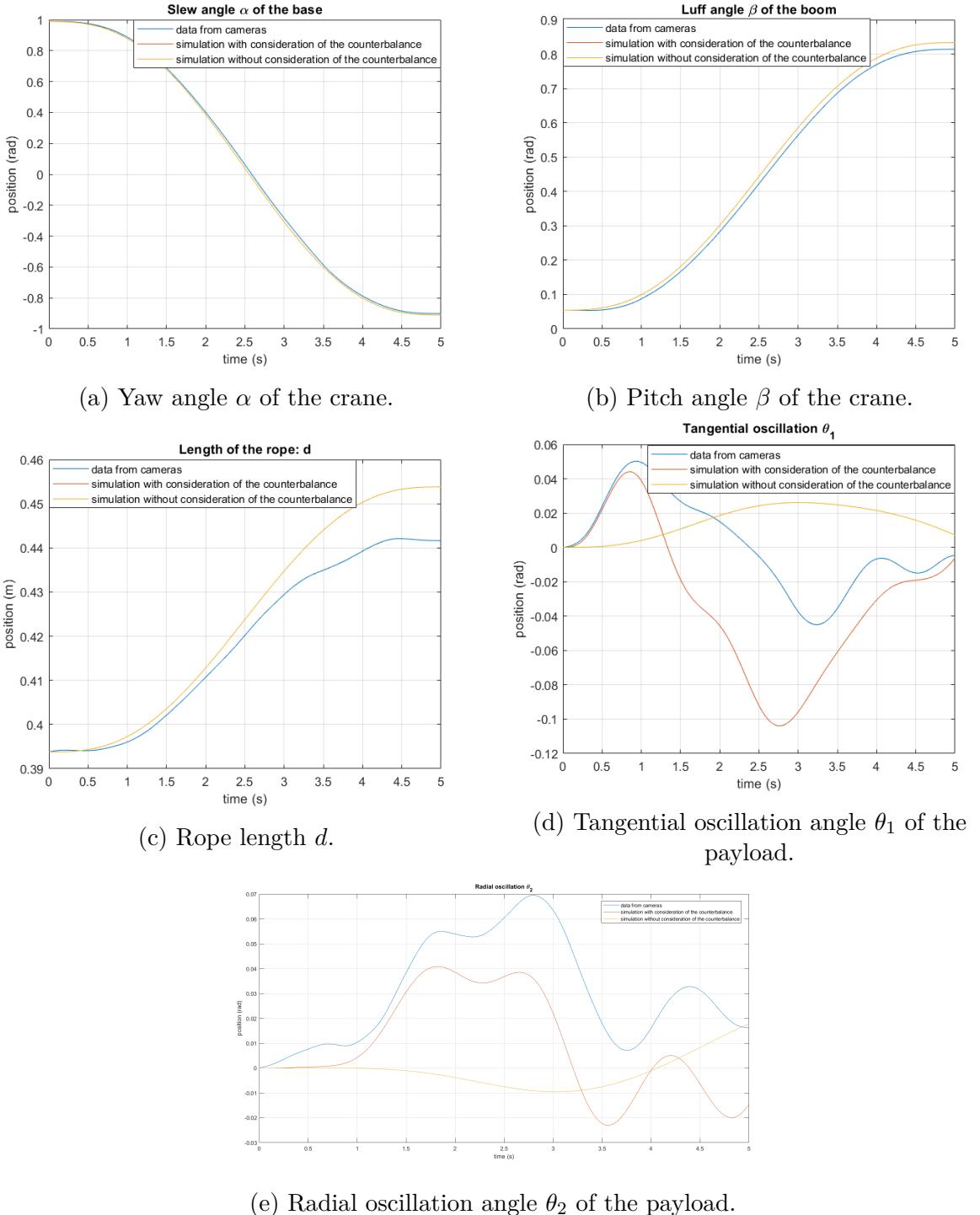


Figure 6.3: Comparison of the results obtained with and without the consideration of the counterbalance in the model.

## 6.5 Identification of a problem related to the non-rigidity of the crane base

This problem was identified by examining the torque of the base actuator when the base is at rest. When the crane is at rest, all velocities and accelerations are equal to zero. Under these conditions, Equation 3.5 can be rewritten as follows:

$$G(q) = \tau . \quad (6.8)$$

In this situation, the joint actuator torques should depend only on the gravity vector. In the model taking into account the counterbalance, the gravity vector is given by:

$$G(q) = \begin{pmatrix} 0 \\ \frac{1}{2} g \cos \beta (2 l_b m + l_b m_b - 2 l_c m_c) \\ -g m \cos \theta_1 \cos \theta_2 \\ g m \cos \theta_2 \sin \theta_1 d \\ g m \cos \theta_1 \sin \theta_2 d \end{pmatrix} . \quad (6.9)$$

Since the first equation of motion is related to the base actuator, the torque should be given by:

$$\tau = \begin{pmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ 0 \\ 0 \end{pmatrix} \implies \tau_1 = 0 , \quad (6.10)$$

which is an intuitive result since the gravity vector should be perfectly aligned with the axis around which the base actuator rotates. However, contrary to what was expected, the torque value read at rest is around - 0.2 Nm.

Investigations revealed that the problem stems from a lack of rigidity in the crane base. As a result, the surface on which the base actuator sits is not perfectly orthogonal to the z-axis, and the weight of the crane components generates some torsion on the actuator, resulting in undesired torque. To verify whether this explanation is correct or not, accelerometer measurements of the base actuator were observed. If the base actuator was on a perfectly stationary surface perpendicular to the axis of the base (i.e. the z-axis), the accelerometer should only measure  $9.81 \text{ m.s}^{-2}$  in the z-direction at all times, whatever the motion of the crane. However, non-zero values were observed in the x- and y-directions and these values were oscillating as the crane moves. The cause of the problem is illustrated in Figure 6.4. This proves that the crane base is not rigid enough for the base actuator to rotate rigorously around the z-axis, which causes undesired forces on the actuator due to the weight of the crane, resulting in undesired torque. This problem could have been avoided during the design phase by not connecting the boom directly to the base actuator, but by using a gearbox, for example, which would allow the gravity force not to be applied to the base actuator, but to other mechanical components instead.

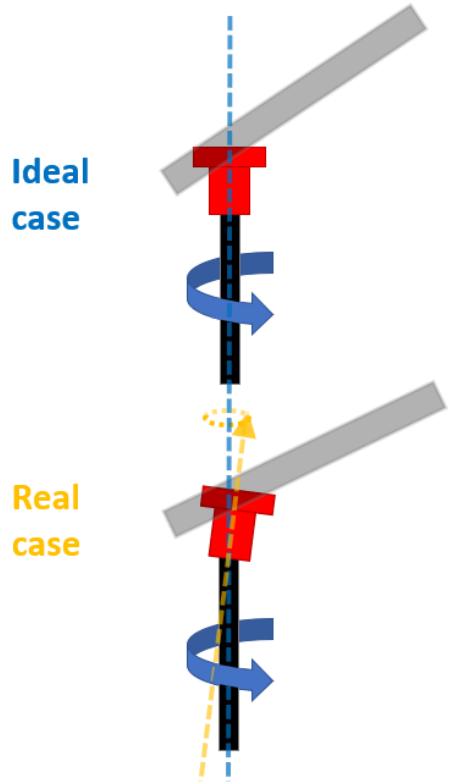


Figure 6.4: Illustration of the problem of the non-rigidity of the crane base: the base axis is not aligned with the gravity vector resulting in an undesired torque on the actuator.

The main difficulty with this problem is that the undesired non-constant offset on the torque is of the same order of magnitude as the typical torque values of the base actuator. As a result, the measured values are far off from what they should actually be, which ruins 20% of the equations used in the least-squares algorithm. This whole problem stems from the lack of rigidity of the crane base, which leads to the conclusion that the least-squares approach is not usable to identify the crane model as long as this problem remains. Nevertheless, this conclusion is a big step forward in the search for methodologies to identify the crane's model.

# Chapter 7

## Work distribution

This MA1 project and the related report are the result of the hard work of Maxime JONGEN, student in the Energy option, and Dinh-Hao NGUYEN, student in the Robotics option.

Initially, much of the work was done together in the SAAS lab<sup>1</sup>. First, we spent time together learning how to operate and control the HEBI's actuators and how to collect data using the OptiTrack cameras and the dedicated Motive software. After this learning phase, data collection for model identification constituted most of the time spent in the lab. We experimented on the prototype, sent many trajectories in order to collect data from the cameras, analyzed their relevance and tried to correct experimental problems that might occur and distort our results. This part was mainly done together, but sometimes it had to be done asynchronously because our respective schedules did not allow us to find a common free time slot.

During the second semester, the work was more clearly divided between us. Maxime was mainly in charge of the post-processing of the data provided by the cameras, the creation and implementation of the codes for model identification and the investigation of the multiple sources of inconsistencies in the data. Dinh-Hao was mostly responsible for structuring and writing the report, but he also helped in problem investigation.

---

<sup>1</sup>Since the crane prototype and the cameras are only available in the lab, we mainly had to work on site.

# Chapter 8

## Learning outcomes of the project

This project has allowed the development of numerous skills in several domains. The following list is not exhaustive, but aims to give an idea of the richness of knowledge that can be acquired from such a project.

- **Knowledge in robotics**

This project has allowed to develop skills primarily in robotics and in system modeling. For a student in the Energy option, the material of the project is quite far from what it is seen in the studies, making it an extremely enriching experience. Even for a student in the Robotics option, the experience was very rewarding because, since the robotics course does not start until the second semester, this project was a first experience of a practical application in this field.

- **Identification of the source of problems**

The purpose of the project is not that long to explain once everything is solved, but each solution to an encountered problem required investigation, and it sometimes took a very long time to find it. In total, a huge number of graphs were plotted, a huge number of simulations were performed, many trajectories were sent to the crane. As a result, we became much more efficient in finding potential sources of problems by knowing what to plot and where to look.

- **Working on a real robot with advanced equipment**

This project involved working with very expensive<sup>1</sup> and performing actuators and an advanced camera tracking system. Having tools of this quality available for a project that mixes theory and practice maximizes the learning outcomes, as the information is often very accurate and easily visualized. This allows a better understanding of the theory and simplifies the corresponding practical tests. In addition, it also provides the opportunity to discover and experiment with real equipment often used in engineering and industry.

- **Manipulation of data**

Data manipulation and filtering were key aspects of this project. On the one hand, we had to find solutions to collect and store adequately certain quantities, and to compare them with the simulation results. This led to very significant improvements in our Matlab skills and in our knowledge about filtering. On the other hand, it was necessary to manipulate and sort a large number of Matlab functions and workspaces, which pushed us to have a good and efficient way to organize all these files.

---

<sup>1</sup>Each HEBI's actuator costs \$3,500.00 [14].

- **Skills in Matlab and Simulink**

A lot of work in this project has been done with Matlab and Simulink, which greatly improved our skills and knowledge of these software. We learned many ways to use them more efficiently.

- **Model identification techniques**

This project has provided a good overview of existing methodologies for identifying the model of a robot, including regression techniques for estimating the parameters of the model. Such techniques are very useful to accurately estimate some dynamic parameters and are widely-used in engineering.

- **Time and resource management**

This project taught us the importance of time and resource management, including sharing and organizing the use of cameras. Indeed, we were not the only ones to use them and other researchers and students also needed them. We had to make arrangements and plan our lab sessions properly.

- **Perseverance**

A lot of problems has occurred during this project, especially technical ones:a router that did not connect to the WiFi, markers that are blinking on Motive, differences in daylight in winter that required to restart the calibration several times, multiple crashes of Matlab and Motive etc. This project taught us that technical problems are part of experimentation, and that whatever the problem, we must persevere to find a solution.

# Chapter 9

## Future possible improvements

At the end of this project, some results for parameter identification have been obtained. However, improvements in the results and methodology can also be brought in the future to obtain more accurate results. The following is a non-exhaustive list of some possible improvements.

- **Consideration of the friction in the model**

As explained in Section 4.2, in order to be 100% rigorous, it is necessary to take into account the friction – especially of the actuators – in the crane model. For this purpose, two parameters can be introduced in the model: the viscous friction coefficient and the Coulomb friction coefficient. These new parameters will also need to be estimated experimentally [15].

- **A 6-degree-of-freedom model**

The modeling of the crane can be improved by considering a 6-DoF model which also takes into account the possible rotation of the payload on itself. A mechanism to control the orientation of the payload can also be added with an additional actuator.

- **Filtering of data**

The way the data are filtered can be improved by systematizing this process. At the moment, the choice of filter is done on a case-by-case basis because some variables such as oscillation angles have a wider range of useful frequencies than other degrees of freedom. It may be interesting and useful to set up a systematic method to filter any type of signal. Several possible solutions have been studied during this project, but such a systematic data filtering method has yet to be found.

# Conclusions

The objective of the project was to identify the model of the boom crane. Many problems have been encountered and solved throughout this project, each time narrowing the area to be investigated to find the potential source of the problems. Ultimately, the dynamic model of the crane could not be found due to the way the crane was designed, but the procedure to estimate the parameters through a regression and then a least-squares approach was established and many improvements have been brought to the resources provided at the beginning of the project.

The payload swing angles have been redefined so that they are based on a relative frame, and not an absolute frame. This problem was present in a code that was provided to us at the beginning of the project. Implementing a rotation matrix solved this problem and led to data consistent with the simulations.

The crane model was modified to take into account the contribution of the counterbalance. The modification of the Lagrangian allowed to obtain the new equations of motion of the crane. Simulations comparing the new model and the old one have been performed. They have shown that taking into account the counterbalance leads to simulation results closer to the actual behavior of the crane.

The regressive matrix  $Y$  of the crane was found from its equations. It was verified that no error was made by obtaining zero when subtracting the product  $Y \cdot \pi$  from the equations of motion.

A Simulink model of the crane has been created, allowing comparison of experimental data with simulations and helping to investigate potential sources of problems.

The data were adequately filtered and processed in MATLAB. All potential sources of discontinuities have been investigated, and the obtained curves could be compared to the commands and simulations, showing a good agreement.

Everything has been set up to perform a least-squares algorithm with the data collected from sending random trajectories to the crane. The code was checked for correct operation with Simulink, but the data collected does not lead to a physically acceptable solution. The lack of rigidity of the base may explain this problem, as it causes a significant offset in the base actuator torque. This problem is present no matter what, and a key conclusion of this project is that a least-squares algorithm cannot lead to a good solution until the base is made more rigid. This currently seems to be the last big problem to fix before getting good results from the least-squares approach. If not, at least this project has solved many procedural problems.

# References

- [1] Michele Ambrosino, Arnaud Dawans, and Emanuele Garone. “Constraint Control of a Boom Crane System”. In: 37th International Symposium on Automation and Robotics in Construction. Kitakyushu, Japan, Oct. 14, 2020. DOI: 10.22260/ISARC2020/0069. URL: [http://www.iaarc.org/publications/2020\\_proceedings\\_of\\_the\\_37th\\_isarc/constraint\\_control\\_of\\_a\\_boom\\_crane\\_system.html](http://www.iaarc.org/publications/2020_proceedings_of_the_37th_isarc/constraint_control_of_a_boom_crane_system.html) (visited on 04/29/2022).
- [2] Michele Ambrosino et al. “Modeling and Control of 5-DoF Boom Crane”. In: 37th International Symposium on Automation and Robotics in Construction. Kitakyushu, Japan, Oct. 14, 2020. DOI: 10.22260/ISARC2020/0071. URL: [http://www.iaarc.org/publications/2020\\_proceedings\\_of\\_the\\_37th\\_isarc/modeling\\_and\\_control\\_of\\_5\\_dof\\_boom\\_crane.html](http://www.iaarc.org/publications/2020_proceedings_of_the_37th_isarc/modeling_and_control_of_5_dof_boom_crane.html) (visited on 04/29/2022).
- [3] Michele Ambrosino et al. “Oscillation Reduction for Knuckle Cranes”. In: 37th International Symposium on Automation and Robotics in Construction. Kitakyushu, Japan, Oct. 14, 2020. DOI: 10.22260/ISARC2020/0221. URL: [http://www.iaarc.org/publications/2020\\_proceedings\\_of\\_the\\_37th\\_isarc/oscillation\\_reduction\\_for\\_knuckle\\_cranes.html](http://www.iaarc.org/publications/2020_proceedings_of_the_37th_isarc/oscillation_reduction_for_knuckle_cranes.html) (visited on 04/29/2022).
- [4] Zemerart Asani et al. “Nonlinear Model Predictive Control of a 5-DoFs Boom Crane”. In: *2021 29th Mediterranean Conference on Control and Automation (MED)*. 2021 29th Mediterranean Conference on Control and Automation (MED). PUGLIA, Italy: IEEE, June 22, 2021, pp. 867–871. ISBN: 978-1-66542-258-1. DOI: 10.1109/MED51440.2021.9480259. URL: <https://ieeexplore.ieee.org/document/9480259/> (visited on 04/29/2022).
- [5] *Calibration - NaturalPoint Product Documentation Ver 2.2*. URL: <https://v22.wiki.optitrack.com/index.php?title=Calibration> (visited on 05/15/2022).
- [6] Christophe Dushimyimana. *Design and Realization of an in scale prototype of a boom crane for teaching purposes*. ULB.
- [7] Sabri El Amrani. *Report: Measurement of the vibrations of a boom crane's payload with OptiTrack*. ULB.
- [8] Sabri El Amrani. *Short report: Trajectory following by a boom crane using various control strategies*. ULB.
- [9] *Hardware*. OptiTrack. URL: <http://optitrack.com/cameras/index.html> (visited on 05/01/2022).
- [10] *HEBI Robotics / Documentation*. HEBI Robotics. URL: [https://docs.hebi.us/core\\_concepts.html#motor\\_control](https://docs.hebi.us/core_concepts.html#motor_control) (visited on 04/30/2022).
- [11] *HEBI Robotics / Documentation*. HEBI Robotics. URL: <https://docs.hebi.us/tools.html#matlab-api> (visited on 04/30/2022).

- [12] Dinh-Hao Nguyen and Maxime Jongen. *MA1\_Project\_Crane*. original-date: 2022-05-14T23:11:42Z. May 15, 2022. URL: [https://github.com/Dinh-Hao-Nguyen/MA1\\_Project\\_Crane](https://github.com/Dinh-Hao-Nguyen/MA1_Project_Crane) (visited on 05/15/2022).
- [13] Liyana Ramli et al. “Control strategies for crane systems: A comprehensive review”. In: *Mechanical systems and signal processing* 95 (2017). Place: Berlin Publisher: Elsevier Ltd, pp. 1–23. ISSN: 0888-3270. DOI: 10.1016/j.ymssp.2017.03.015.
- [14] *Robotic Actuation Hardware*. HEBI Robotics. URL: <https://www.hebirobotics.com/hardware> (visited on 04/30/2022).
- [15] Bruno Siciliano et al. *Robotics*. Red. by Michael J. Grimble and Michael A. Johnson. Advanced Textbooks in Control and Signal Processing. London: Springer London, 2009. ISBN: 978-1-84628-642-1. DOI: 10.1007/978-1-84628-642-1. URL: <http://link.springer.com/10.1007/978-1-84628-642-1> (visited on 04/29/2022).
- [16] Mark W. Spong. *Robot modeling and control*. In collab. with Seth Hutchinson and M. Vidyasagar. Hoboken, NJ: John Wiley & Sons, 2006. 478 pp. ISBN: 978-0-471-64990-8.
- [17] Yvonne R. Stürz, Lukas M. Affolter, and Roy S. Smith. “Parameter Identification of the KUKA LBR iiwa Robot Including Constraints on Physical Feasibility”. In: *IFAC-PapersOnLine* 50.1 (July 2017), pp. 6863–6868. ISSN: 24058963. DOI: 10.1016/j.ifacol.2017.08.1208. URL: <https://linkinghub.elsevier.com/retrieve/pii/S2405896317317147> (visited on 04/29/2022).
- [18] Ning Sun et al. “Nonlinear Motion Control of Underactuated Three-Dimensional Boom Cranes With Hardware Experiments”. In: *IEEE Transactions on Industrial Informatics* 14.3 (Mar. 2018), pp. 887–897. ISSN: 1551-3203, 1941-0050. DOI: 10.1109/TII.2017.2754540. URL: <https://ieeexplore.ieee.org/document/8047336/> (visited on 04/29/2022).
- [19] Michaël Van Damme. “Robotics 1 - Dynamics”.
- [20] Viorica Spoiala et al. “Position and Payload Oscillations Control of Cranes - A Review”. In: *Journal of computer science and control systems* 3.1 (2010). Place: Oradea Publisher: University of Oradea, pp. 221–. ISSN: 1844-6043.
- [21] Jun Wu, Jinsong Wang, and Zheng You. “An overview of dynamic parameter identification of robots”. In: *Robotics and computer-integrated manufacturing* 26.5 (2010). Publisher: Elsevier Ltd, pp. 414–419. ISSN: 0736-5845. DOI: 10.1016/j.rcim.2010.03.013.

# Appendix A

## Dynamic model of a boom crane

As presented in Section 3.2, the kinetic and potential energy of a boom crane can be expressed as:

$$T(t) = \frac{1}{8} m_b (\dot{x}_b^2 + \dot{y}_b^2 + \dot{z}_b^2) + \frac{1}{2} m (\dot{x}_M^2 + \dot{y}_M^2 + \dot{z}_M^2) + \frac{1}{2} m_c (\dot{x}_c^2 + \dot{y}_c^2 + \dot{z}_c^2) + \frac{1}{2} I_{\text{tot}} \dot{\alpha} + \frac{1}{2} I_b \dot{\beta}, \quad (\text{A.1})$$

$$V(t) = m_b g \frac{z_b}{2} + m g z_M - m_c g l_c \sin \beta. \quad (\text{A.2})$$

Developing these expressions in terms of the generalized coordinates  $q = [\alpha, \beta, d, \theta_1, \theta_2]^\top$ , it follows:

$$\begin{aligned} T(t) = & \frac{1}{2} m \dot{d}^2 + \frac{1}{2} I_{\text{tot}} \dot{\alpha}^2 + \frac{1}{2} I_b \dot{\beta}^2 + \frac{1}{2} m d^2 \dot{\alpha}^2 + \frac{1}{2} m d^2 \dot{\theta}_2^2 + \frac{1}{2} l_b^2 m \dot{\beta}^2 \\ & + \frac{1}{8} l_b^2 m_b \dot{\beta}^2 + \frac{1}{2} l_c^2 m_c \dot{\beta}^2 + \frac{1}{2} l_b^2 m C_\beta^2 \dot{\alpha}^2 + \frac{1}{8} l_b^2 m_b C_\beta^2 \dot{\alpha}^2 + \frac{1}{2} l_c^2 m_c C_\beta^2 \dot{\alpha}^2 \\ & + \frac{1}{2} m C_{\theta_2}^2 d^2 \dot{\theta}_1^2 - m S_{\theta_1} d^2 \dot{\alpha} \dot{\theta}_2 - \frac{1}{2} m C_{\theta_1}^2 C_{\theta_2}^2 d^2 \dot{\alpha}^2 - l_b m S_\beta S_{\theta_2} \dot{\beta} \dot{d} \\ & + l_b m C_\beta S_{\theta_2} d \dot{\alpha}^2 - l_b m S_\beta C_{\theta_2} d \dot{\beta} \dot{\theta}_2 + m C_{\theta_1} C_{\theta_2} S_{\theta_2} d^2 \dot{\alpha} \dot{\theta}_1 \\ & - l_b m C_\beta C_{\theta_1} C_{\theta_2} \dot{\beta} \dot{d} + l_b m C_\beta C_{\theta_2} S_{\theta_1} \dot{\alpha} \dot{d} + l_b m C_\beta C_{\theta_1} C_{\theta_2} d \dot{\alpha} \dot{\theta}_1 \\ & + l_b m C_\beta C_{\theta_2} S_{\theta_1} d \dot{\beta} \dot{\theta}_1 + l_b m C_\beta C_{\theta_1} S_{\theta_2} d \dot{\beta} \dot{\theta}_2 + l_b m S_\beta C_{\theta_2} S_{\theta_1} d \dot{\alpha} \dot{\beta} \\ & - l_b m C_\beta S_{\theta_1} S_{\theta_2} d \dot{\alpha} \dot{\theta}_2, \end{aligned} \quad (\text{A.3})$$

$$V(t) = g l_b m S_\beta + \frac{1}{2} g l_b m_b S_\beta - g l_c m_c S_\beta - g m C_{\theta_1} C_{\theta_2} d, \quad (\text{A.4})$$

using the abbreviations  $C_\phi \triangleq \cos \phi$  and  $S_\phi \triangleq \sin \phi$ .

Defining the Lagrangian as:

$$\mathcal{L}(q, \dot{q}) = T(q, \dot{q}) - V(q), \quad (\text{A.5})$$

and developing the Euler-Lagrange equations:

$$\frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{q}}(q, \dot{q}) - \frac{\partial \mathcal{L}}{\partial q}(q, \dot{q}) = \tau, \quad (\text{A.6})$$

one obtains the five following equations:

$$\begin{aligned} & I_{\text{tot}} \ddot{\alpha} + m d^2 \ddot{\alpha} - m S_{\theta_1} d^2 \ddot{\theta}_2 + 2 m d \dot{\alpha} \dot{d} + l_b^2 m C_\beta^2 \ddot{\alpha} + \frac{l_b^2 m_b C_\beta^2 \ddot{\alpha}}{4} \\ & + l_c^2 m_c C_\beta^2 \ddot{\alpha} - 2 m C_{\theta_1} d^2 \dot{\theta}_1 \dot{\theta}_2 - m C_{\theta_1}^2 C_{\theta_2}^2 d^2 \ddot{\alpha} - 2 m S_{\theta_1} d \dot{d} \dot{\theta}_2 \\ & - m C_{\theta_2} S_{\theta_1} S_{\theta_2} d^2 \dot{\theta}_1^2 + 2 l_b m C_\beta S_{\theta_2} d \ddot{\alpha} + 2 l_b m C_\beta S_{\theta_2} \dot{\alpha} \dot{d} \\ & + m C_{\theta_1} C_{\theta_2} S_{\theta_2} d^2 \ddot{\theta}_1 - 2 m C_{\theta_1}^2 C_{\theta_2}^2 d \dot{\alpha} \dot{d} + 2 m C_{\theta_1} C_{\theta_2}^2 d^2 \dot{\theta}_1 \dot{\theta}_2 \\ & + l_b m C_\beta C_{\theta_2} S_{\theta_1} \ddot{d} - 2 l_b^2 m C_\beta S_\beta \dot{\alpha} \dot{\beta} - \frac{l_b^2 m_b C_\beta S_\beta \dot{\alpha} \dot{\beta}}{2} - 2 l_c^2 m_c C_\beta S_\beta \dot{\alpha} \dot{\beta} \\ & + 2 m C_{\theta_1} C_{\theta_2}^2 S_{\theta_1} d^2 \dot{\alpha} \dot{\theta}_1 + 2 m C_{\theta_1}^2 C_{\theta_2} S_{\theta_2} d^2 \dot{\alpha} \dot{\theta}_2 + l_b m C_\beta C_{\theta_2} S_{\theta_1} d \dot{\beta}^2 \\ & - l_b m C_\beta C_{\theta_2} S_{\theta_1} d \dot{\theta}_1^2 - l_b m C_\beta C_{\theta_2} S_{\theta_1} d \dot{\theta}_2^2 + 2 l_b m C_\beta C_{\theta_2} d \dot{\alpha} \dot{\theta}_2 \\ & + 2 m C_{\theta_1} C_{\theta_2} S_{\theta_2} d \dot{d} \dot{\theta}_1 - 2 l_b m S_\beta S_{\theta_2} d \dot{\alpha} \dot{\beta} + l_b m C_\beta C_{\theta_1} C_{\theta_2} d \ddot{\theta}_1 \\ & + l_b m S_\beta C_{\theta_2} S_{\theta_1} d \ddot{\beta} - l_b m C_\beta S_{\theta_1} S_{\theta_2} d \ddot{\theta}_2 + 2 l_b m C_\beta C_{\theta_1} C_{\theta_2} \dot{d} \dot{\theta}_1 \\ & - 2 l_b m C_\beta S_{\theta_1} S_{\theta_2} \dot{d} \dot{\theta}_2 - 2 l_b m C_\beta C_{\theta_1} S_{\theta_2} d \dot{\theta}_1 \dot{\theta}_2 = \tau_1, \end{aligned} \quad (\text{A.7})$$

$$\begin{aligned} & I_b \ddot{\beta} + l_b^2 m \ddot{\beta} + \frac{l_b^2 m_b \ddot{\beta}}{4} + l_c^2 m_c \ddot{\beta} + g l_b m C_\beta + \frac{g l_b m_b C_\beta}{2} - g l_c m_c C_\beta \\ & + l_b^2 m C_\beta S_\beta \dot{\alpha}^2 + \frac{l_b^2 m_b C_\beta S_\beta \dot{\alpha}^2}{4} + l_c^2 m_c C_\beta S_\beta \dot{\alpha}^2 - l_b m S_\beta S_{\theta_2} \ddot{d} \\ & - l_b m S_\beta C_{\theta_2} d \ddot{\theta}_2 - 2 l_b m S_\beta C_{\theta_2} \dot{d} \dot{\theta}_2 - l_b m C_\beta C_{\theta_1} C_{\theta_2} \ddot{d} + l_b m S_\beta S_{\theta_2} d \dot{\alpha}^2 \\ & + l_b m S_\beta S_{\theta_2} d \dot{\theta}_2^2 + l_b m C_\beta C_{\theta_1} C_{\theta_2} d \dot{\theta}_1^2 + l_b m C_\beta C_{\theta_1} C_{\theta_2} d \dot{\theta}_2^2 \\ & + l_b m C_\beta C_{\theta_2} S_{\theta_1} d \ddot{\theta}_1 + l_b m C_\beta C_{\theta_1} S_{\theta_2} d \ddot{\theta}_2 + l_b m S_\beta C_{\theta_2} S_{\theta_1} d \ddot{\alpha} \\ & + 2 l_b m C_\beta C_{\theta_2} S_{\theta_1} \dot{d} \dot{\theta}_1 + 2 l_b m C_\beta C_{\theta_1} S_{\theta_2} \dot{d} \dot{\theta}_2 + 2 l_b m S_\beta C_{\theta_2} S_{\theta_1} \dot{\alpha} \dot{d} \\ & + 2 l_b m S_\beta C_{\theta_1} C_{\theta_2} d \dot{\alpha} \dot{\theta}_1 - 2 l_b m C_\beta S_{\theta_1} S_{\theta_2} d \dot{\theta}_1 \dot{\theta}_2 - 2 l_b m S_\beta S_{\theta_1} S_{\theta_2} d \dot{\alpha} \dot{\theta}_2 = \tau_2, \end{aligned} \quad (\text{A.8})$$

$$\begin{aligned} & m \ddot{d} - m d \dot{\alpha}^2 - m d \dot{\theta}_2^2 - m C_{\theta_2}^2 d \dot{\theta}_1^2 - g m C_{\theta_1} C_{\theta_2} - l_b m C_\beta S_{\theta_2} \dot{\alpha}^2 \\ & - l_b m C_\beta S_{\theta_2} \dot{\beta}^2 + m C_{\theta_1}^2 C_{\theta_2}^2 d \dot{\alpha}^2 - l_b m S_\beta S_{\theta_2} \ddot{\beta} + 2 m S_{\theta_1} d \dot{\alpha} \dot{\theta}_2 \\ & + l_b m S_\beta C_{\theta_1} C_{\theta_2} \dot{\beta}^2 - l_b m C_\beta C_{\theta_1} C_{\theta_2} \ddot{\beta} + l_b m C_\beta C_{\theta_2} S_{\theta_1} \ddot{\alpha} \\ & - 2 m C_{\theta_1} C_{\theta_2} S_{\theta_2} d \dot{\alpha} \dot{\theta}_1 - 2 l_b m S_\beta C_{\theta_2} S_{\theta_1} \dot{\alpha} \dot{\beta} = \tau_3, \end{aligned} \quad (\text{A.9})$$

$$\begin{aligned} & m C_{\theta_2}^2 d^2 \ddot{\theta}_1 + 2 m C_{\theta_2}^2 d \dot{d} \dot{\theta}_1 + g m C_{\theta_2} S_{\theta_1} d - 2 m C_{\theta_2} S_{\theta_2} d^2 \dot{\theta}_1 \dot{\theta}_2 \\ & - m C_{\theta_1} C_{\theta_2}^2 S_{\theta_1} d^2 \dot{\alpha}^2 + m C_{\theta_1} C_{\theta_2} S_{\theta_2} d^2 \ddot{\alpha} + 2 m C_{\theta_1} C_{\theta_2}^2 d^2 \dot{\alpha} \dot{\theta}_2 \\ & - l_b m S_\beta C_{\theta_2} S_{\theta_1} d \dot{\beta}^2 + 2 m C_{\theta_1} C_{\theta_2} S_{\theta_2} d \dot{\alpha} \dot{d} + l_b m C_\beta C_{\theta_1} C_{\theta_2} d \ddot{\alpha} \\ & + l_b m C_\beta C_{\theta_2} S_{\theta_1} d \ddot{\beta} - 2 l_b m S_\beta C_{\theta_1} C_{\theta_2} d \dot{\alpha} \dot{\beta} = 0, \end{aligned} \quad (\text{A.10})$$

$$\begin{aligned}
& m d^2 \ddot{\theta}_2 - m S_{\theta_1} d^2 \ddot{\alpha} + 2 m d \dot{d} \dot{\theta}_2 - 2 m S_{\theta_1} d \dot{\alpha} \dot{d} + g m C_{\theta_1} S_{\theta_2} d \\
& + m C_{\theta_2} S_{\theta_2} d^2 \dot{\theta}_1^2 - l_b m S_\beta C_{\theta_2} d \ddot{\beta} - m C_{\theta_1}^2 C_{\theta_2} S_{\theta_2} d^2 \dot{\alpha}^2 \\
& - 2 m C_{\theta_1} C_{\theta_2}^2 d^2 \dot{\alpha} \dot{\theta}_1 - l_b m C_\beta C_{\theta_2} d \dot{\alpha}^2 - l_b m C_\beta C_{\theta_2} d \dot{\beta}^2 \\
& - l_b m S_\beta C_{\theta_1} S_{\theta_2} d \dot{\beta}^2 + l_b m C_\beta C_{\theta_1} S_{\theta_2} d \ddot{\beta} - l_b m C_\beta S_{\theta_1} S_{\theta_2} d \ddot{\alpha} \\
& + 2 l_b m S_\beta S_{\theta_1} S_{\theta_2} d \dot{\alpha} \dot{\beta} = 0 .
\end{aligned} \tag{A.11}$$