

**MASTER**

**Delta Robot Motion Stabilization for Flexible Packaging of Viscous Fluid Products in a Bag**

van Noorden, A.C.H.

*Award date:*  
2021

[Link to publication](#)

**Disclaimer**

This document contains a student thesis (bachelor's or master's), as authored by a student at Eindhoven University of Technology. Student theses are made available in the TU/e repository upon obtaining the required degree. The grade received is not published on the document as presented in the repository. The required complexity or quality of research of student theses may vary by program, and the required minimum study period may vary in duration.

**General rights**

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain

GRADUATION REPORT

---

Delta Robot Motion Stabilization for Flexible  
Packaging of Viscous Fluid Products in a Bag

---

2020 - 2021

THIS REPORT WAS MADE IN ACCORDANCE WITH THE TU/E CODE OF SCIENTIFIC CONDUCT FOR  
THE MASTER THESIS.

**Supervisors:**

---

ir. R.J. van der Kruk  
dr.ir. M.J.G. van de Molengraft

**Committee members:**

---

ir. R.J. van der Kruk  
dr.ir. M.J.G. van de Molengraft  
prof.dr. H.P.J. Bruyninckx  
dr.ir. A.A.J. Lefeber

**Student:**

---

A.C.H. van Noorden (1378147)    a.c.h.v.noorden@student.tue.nl

Eindhoven, October 16, 2021

## Abstract

The worldwide food demand is rapidly increasing due to the growing population. To supply this demand further automation of the food industry is required. Multiple companies are involved in this process. One of these companies is Blueprint Automation (BPA). BPA is specialized in handling stage two food products at high volumes. Stage two products are individually packed food products that are picked from a conveyor belt and placed inside a box for shipping. Delta robots equipped with a vacuum gripper are used for manipulating the food products.

This graduation project is part of the FlexCRAFT program and is focused on planning and control (P3) and packaging integration (P7). This project investigates the handling behavior of viscous food products inside a bag. During the pick and place motion the product starts to oscillate due to the flexible structure of the vacuum gripper. This oscillating behavior can result in the product peeling off from the gripper, or the oscillation complicates the placement inside a box. The main goal of this project is to find a method (alternative control strategy) to reduce the oscillating behavior without facially changing the robot.

To determine the most suitable control strategy for this application a literature research is executed. Based on this research the method named input shaping is chosen, which is developed for canceling out oscillations from a load of a construction crane. To apply input shaping the eigenfrequency and relative damping of the product attached to the gripper (gripper-product system) are required. These parameters are derived based on measurements executed on a test setup which is provided by BPA, which contains a Spider DPR01 delta robot. With the derived parameters is the performance of input shaping determined in simulations and in practice. Both show an effective reduction of the oscillating behavior.

Thus, input shaping can effectively reduce the oscillating behavior of the gripper-product system for this application.

# Contents

<b>Abstract</b>	<b>i</b>
<b>Contents</b>	<b>ii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Problem definition . . . . .	2
1.2 Research questions . . . . .	2
1.3 Report outline . . . . .	3
<b>2 Background</b>	<b>4</b>
2.1 Mechanics . . . . .	4
2.2 Electronics . . . . .	5
2.3 Pneumatics . . . . .	5
2.4 Software . . . . .	6
<b>3 Literature Research</b>	<b>8</b>
3.1 System identification . . . . .	8
3.2 Vibration reduction control strategies . . . . .	9
<b>4 System Identification</b>	<b>13</b>
4.1 Spider identification . . . . .	13
4.2 Gripper-product system identification . . . . .	14
4.3 Trajectory frequency analysis . . . . .	19
<b>5 Modeling</b>	<b>21</b>
5.1 Spider model . . . . .	21
5.2 Gripper-product system . . . . .	22
<b>6 Simulation</b>	<b>23</b>
6.1 Simulation software structure . . . . .	23
6.2 Pendulum parameters estimation . . . . .	23
6.3 Input Shaping Trajectories . . . . .	24
6.4 Input Shaping results . . . . .	26
<b>7 Implementation</b>	<b>29</b>
7.1 Beckhoff interface . . . . .	29
7.2 MATLAB external trajectory generator . . . . .	29
<b>8 Experiments and Results</b>	<b>30</b>
8.1 Validation of simulation results . . . . .	30
8.2 Input shaping performance for varying maximum accelerations . . . . .	32
8.3 Use case test . . . . .	34
<b>9 Conclusions and Recommendations</b>	<b>36</b>
<b>Appendix A</b>	<b>39</b>

## 1 Introduction

The worldwide food demand is rapidly increasing due to the growing population. To supply this demand, further automation of the food industry is required. Blueprint Automation (BPA) is specialized in handling stage two food packages. In this stage products are already packed in individual bags. BPA uses delta robots to automate the handling process. The products are picked from a conveyor belt and stacked into shipping boxes. Delta robots are parallel structured robots as shown in Figure 1.1. Using this type of robot for handling stage two food products is the state of the practice [1]. The products are picked with a compliant vacuum gripper, which means the gripper can deform to match the product's shape. This gripper's main advantage is that it can be used to handle a great variety of products. Due to this property, there is no need to design a specific gripper for every single product type. The connection between the delta and the product is not fixated resulting in an undesired oscillating behavior of the product. This behavior results in two problems: on the one hand, the product can peel off the vacuum gripper. On the other hand, the product oscillates during placement. These two issues appear if the delta acceleration becomes too high. Therefore the current acceleration is limited to prevent these issues.



Figure 1.1: Double Spider DPR01 setup, BPA

Currently it is desired to improve the robot efficiency by increasing the acceleration. The mechanical and electrical components of the delta have been upgraded resulting in higher accelerations. The next step is to investigate how the product in combination with the gripper (gripper-product system) can be improved to withstand these higher accelerations. The challenge is to improve the system behavior without physically changing the gripper or the product. This means that alternative control strategies must be investigated to reduce the oscillating behavior. This topic is the primary research of this graduation project. This graduation project is part of the FlexCRAFT Project which is divided into four generic robot capabilities (P1 to P4) and three robotic system integration topics (P5 to P7). This project is focused on planning and control (P3) and packaging integration (P7). An alternative trajectory generator and the inverse kinematics of the delta is obtained from an internship titled: "Delta Robot B-Spline based Trajectory Generation Inverse Kinematics for Food Packaging Applications". Its results will be used to find alternative delta control strategies. This research focuses on viscous fluid products in a bag, for example soup or cheese in a fluid such as mozzarella packages. This last type is chosen due to its challenging characteristics, for example, the changing center of mass during movement due to the mozzarella ball that moves in the fluid. Besides, the fragile structure of mozzarella requires careful handling. A Spider DPR01 delta robot will execute the handling of the product and BPA develops this delta. The Spider is equipped with a standard Piab vacuum gripper designed for handling bags filled with viscous content. Figure 1.1 shows two Spiders DPR01 stacking food products from a conveyor into boxes.

## 1.1 Problem definition

Based on the preliminary measurements as described in the phase one report titled, it can be concluded the two named problems are caused by the excitation of the eigenfrequency ( $f_n$  in  $Hz$ ) of the gripper-product system during the pick and place motion. The excitation results in an oscillating product behavior during the movement and placement; this behavior can result in:

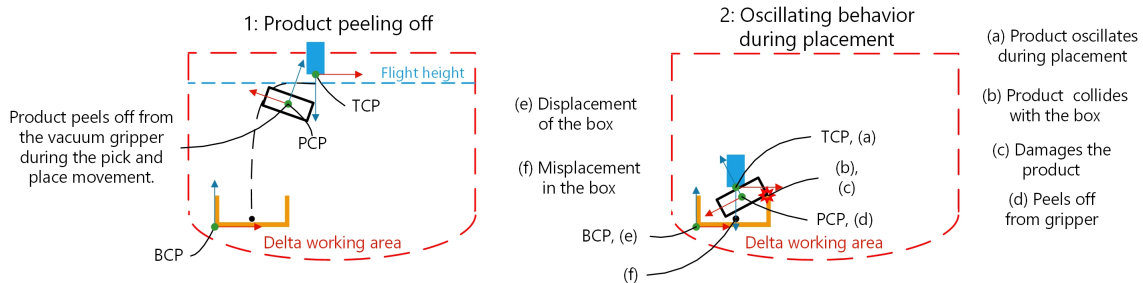


Figure 1.2: Schematic representation of the problem definition

1. If the brake-away force between the lip of the vacuum gripper and the product exceeds the gripper retention force, the product will peel off from the vacuum gripper.
2. Inaccurate placement or collision with the box can result in misplacement in the box and the prescribed number of products will not fit inside the box. Displacement of the box due to collision with the product causes a misaligned placement of the neighboring products. Finally, the product can get damaged or peel off from the gripper by hitting the side of the box.

## 1.2 Research questions

Based on the problem definition, the main and side research questions were determined. These are presented below.

How can the control strategy be changed to reduce the oscillating behavior of the mozzarella package attached to the vacuum gripper (gripper-product system) during the pick and place motion executed by the Spider DPR01 delta robot without physically changing the Spider, gripper, or product?

- Which method is suitable to apply system identification to the Spider to determine its modal characteristics?
- How can the oscillating behavior of the gripper-product system be analyzed to acquire the modal parameters of this gripper-product system?
- How can the Spider and end-effector be modeled?
- Which simulation environment is suitable for simulating the Spider and gripper-product system?
- How can the oscillating behavior of the gripper-product system underneath the Spider during the pick and place motion be simulated?
- Which control strategy is suitable to limit the oscillating behavior of the gripper-product system?
- How can the found control strategy be tested on the Spider test setup?

### 1.3 Report outline

The outline of this report is as follows, Chapter 2 describes the measurement setup which is used for this project. Chapter 3 shows the existing identification and control strategies described in the literature. The Spider and gripper-product system are identified in Chapter 3.1. Chapter 5 describes the obtained Spider- and gripper-product system model required for simulating the oscillating behavior. This simulation is shown in Chapter 6, together with the performance of input shaping. Chapter 7 discusses the implementation process of input shaping on the measurement setup. The experiments and results executed on this setup are shown in Chapter 8. Finally, the derived conclusions and recommendations are presented in Chapter 9.

## 2 Background

This chapter describes the Spider DRP01 in detail based on the sections: Mechanics, electronics, pneumatics, and software. For executing measurements, a setup is provided by BPA which is located at its headquarters in Woerden. Figure 2.1 shows the setup. The main components are the Spider in the center and a stationary pick and place table below, both connected by the frame.

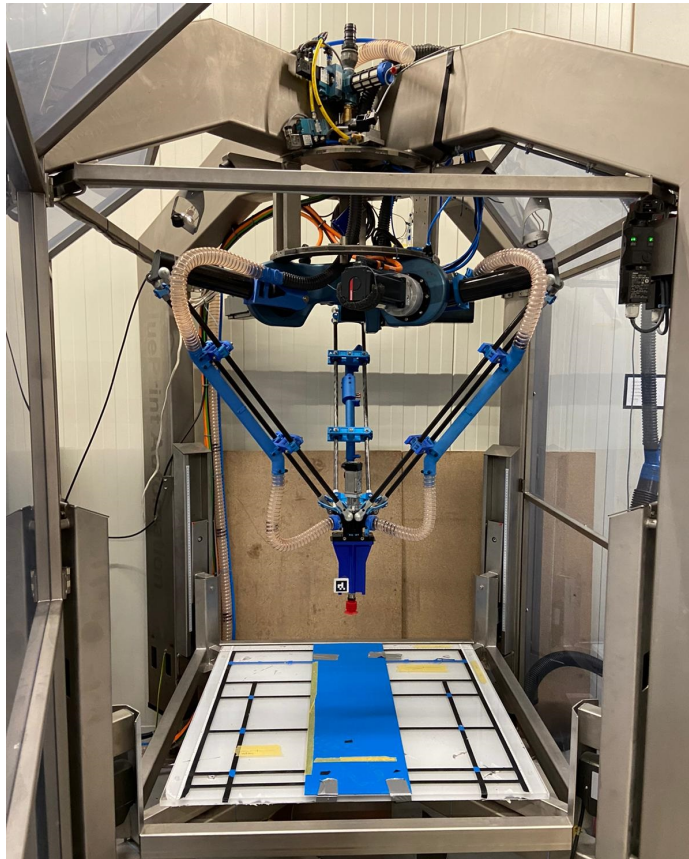


Figure 2.1: Single Spider DPR01 measurement setup

### 2.1 Mechanics

The Spider DRP01 is a parallel structured robot and is actuated by three rotary joints [2]. Figure 2.3 shows the mechanical structure of the Spider and the physical parameters are expressed in Table 2.1. The Spider is equipped with three servo motors in combination with planetary gearboxes which are connected to the upper arms (Proximal link). The forearms exist out of two parallel distal links connected by two spherical joints to the upper arms. The other ends of the distal links are connected to the traveling platform with two spherical joints. The traveling platform connects the three forearms, and this results in a closed kinematic chain. The traveling platform can move in the Cartesian space, with three translational degrees of freedom (DOF) ( $x, y, z$ ). A fourth servo motor is installed on the traveling platform to realize an extra rotational decoupled DOF along the  $z$  axis. For this project, this fourth DOF is left out of consideration because the gripper-product system's oscillating behavior is mainly present during the translating movement of the Spider. Finally, the end-effector (vacuum gripper) is connected to the motor shaft of the fourth joint. The working space is defined such that all singularities will be avoided resulting in a square cuboid space of  $797 \times 797 \times 228$  (W x L x H) [mm].



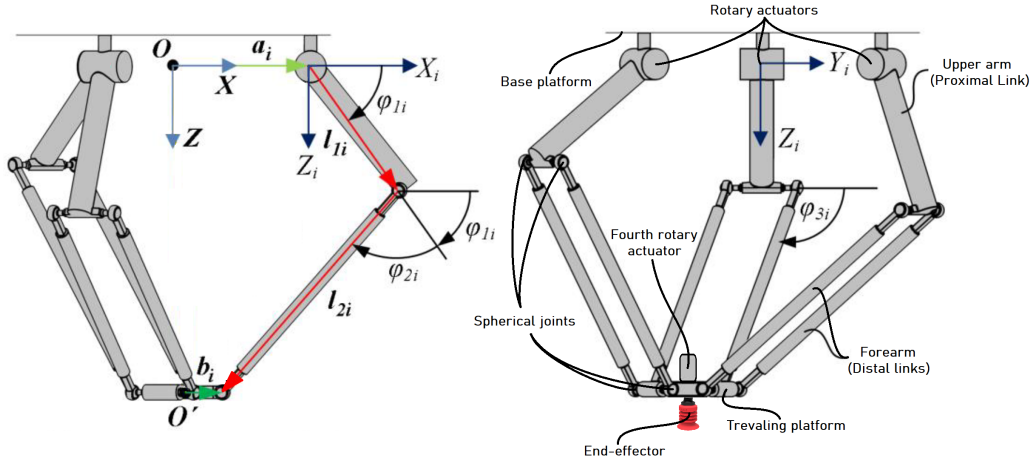


Figure 2.2: Mechanical structure delta robot

Table 2.1: Physical parameters delta robot

Kinematic Parameters			Dynamic Parameters		
Sym.	Parameters	Value	Sym.	Parameters	Value
$l_{1i}$	Length Upper arm [mm]	348.088	$m_{1i}$	Mass Upper arm [kg]	0.949
$l_{2i}$	Length Forearm [mm]	800	$m_{2i}$	Mass Forearm [kg]	0.308
$\alpha_i$	Offset rotary joints [mm]	200	$m_3$	Mass End-effector [kg]	0.026
$\beta_i$	Offset Travailing platform [mm]	45	$m_4$	Mass travelling platform [kg]	1.728

## 2.2 Electronics

The electronics for controlling the Spider are provided by a supplier named Beckhoff. Figure 2.3 shows is an overview of the electronic components. This system uses two dual axes motor drives powered by a single supply unit. The motor drives are controlled by an industrial PC (IPC). The software controlling the Spider runs on this IPC. The two motor drives control the three servo motors to actuate the rotary joints of the Spider and the servo to actuate the fourth joint for rotating the end-effector around the z-axis. The rotary joint servos have a rated power of 1.38 kW and torque of 2.2 Nm. The absolute encoders with an 18-bit resolution measure the joint positions and are integrated in the servos.

## 2.3 Pneumatics

To realize the vacuum pressure for the gripper, a dry claw vacuum pump actuated by an asynchronous motor is used, controlled by a frequency converter. A buffer vessel is used to realize a constant vacuum pressure. A pressure sensor is installed inside this vessel and is connected to an EtherCAT I/O module. This module communicates with the IPC over an EtherCAT Bus. The IPC contains a PID controller that generates a velocity input for the frequency inverter based on a vacuum setpoint and the measured vacuum pressure inside the vessel. The vacuum pressure on the gripper can be activated by a vacuum valve that the IPC controls the EtherCAT I/O module. Likewise, a relief valve can be activated to release the product at the place position by injecting a short pulse of pressurized air inside the gripper.

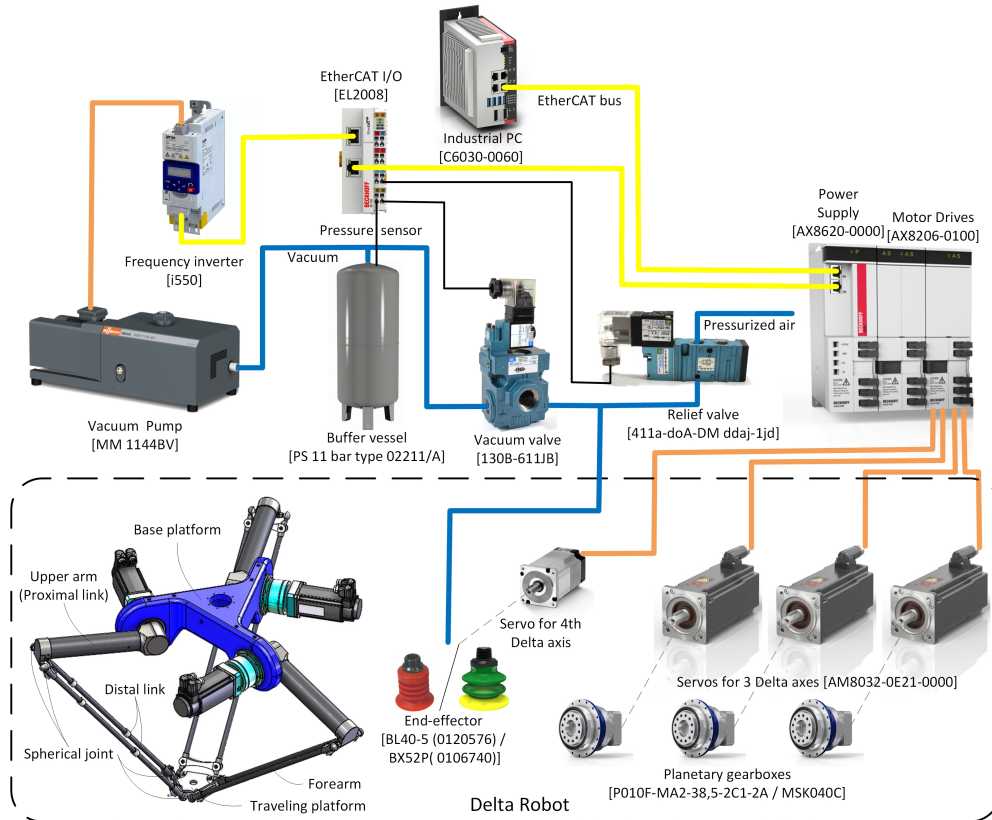


Figure 2.3: Delta robot schematic electronics and pneumatics overview

## 2.4 Software

The software for controlling the Spider consists out of high-level and low-level modules. The high-level modules are executed by the IPC where the low-level modules are executed by the motor drives. Figure 2.5 shows the software levels. The high-level modules exist out of a task scheduler that determines the order of the pick and place tasks. Next, a state machine executes these tasks by executing multiple states. One of the states contains a path planner that generates control and blend points ( $x, y, z, C.P.$ ) in Cartesian space based on manually set pick and place coordinates. Besides, the generated control points are used by a trajectory generator to fit a trajectory along with these control points with a sample rate of 1 kHz. Every millisecond, a trajectory point ( $x, y, z, Traj.$ ) is generated. Once the complete trajectory is generated, it is translated from Cartesian space to joint space by applying an inverse kinematic transformation. This results in three individual joint position trajectories, one for each joint ( $q_1, q_2, q_3, Traj.$ ). Finally, these joint trajectories are used as setpoints for the joint controllers. The joint controller is described in detail in Section 2.4.a. The current control strategy is located in the marked green area of Figure 2.4. The most logical choice is to find a new control strategy that can be implemented in this area.

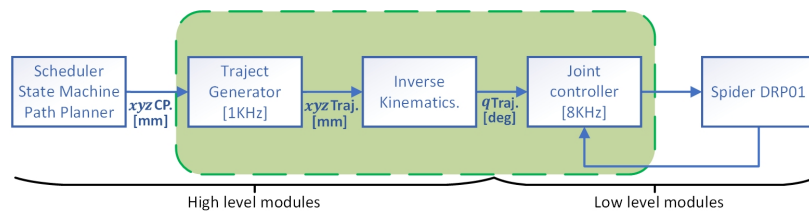


Figure 2.4: Software architecture

### 2.4.a Joint Controller

The joint controller contains a cubic interpolator and four control components: a feedforward, position, velocity, and current controller [3]. Every joint has a separate joint controller containing this control structure.

First, the joint controller interpolates the 1 *kHz* joint position trajectory with 8 *KHz* based on cubic interpolation. This is done to increase the trajectory resolution and improve the trajectory smoothness between each data point. Eight extra points are generated. Besides, the interpolated trajectory is used to derive the velocity and acceleration joint trajectory required for the feedforward controller. The increased resolution logically improves the smoothness of these two trajectories.

Second, the feedforward controller contains velocity and acceleration feedforward. The interpolated velocity ( $\dot{q}_{Traj.}$ ) and acceleration ( $\ddot{q}_{Traj.}$ ) trajectories are multiplied with feedforward velocity gain ( $K_{fv}$ ) and acceleration gain ( $K_{fa}$ ). Third, the position error is multiplied with the position proportional gain ( $K_{pp}$ ). This gain exists out of the interpolated position trajectory ( $q_{Traj.}$ ) minus the measured actual joint position ( $q_{Actual}$ ). The actual position is determined based on the servo encoder signals.

Fourth, the input of the velocity controller exist out of the output of the position controller ( $\omega_{fb}$ ) in degrees per second, plus the velocity feedforward output ( $\omega_{ff}$ ), minus the actual joint velocity ( $\omega_{Actual}$ ). This input results in the velocity error and is applied to a PI controller containing the velocity proportional gain ( $K_{pv}$ ) and integral velocity gain ( $K_{vi}$ ). The actual velocity is derived by differentiating the measured joint position. This method has a limitation that high-frequency mechanical vibrations will realize a noisy velocity signal. To prevent high-frequency noise enters the velocity signal, a velocity observer is used [3]. This observer uses the method of differentiating the joint position for frequencies up to a set bandwidth. The servo current is measured and converted to the actual joint acceleration for higher frequencies based on the servo motor characteristics curve. Next, this joint acceleration is numerically integrated to obtain the actual joint velocity. This velocity signal is less susceptible to mechanical resonances.

Finally, the current controller converts the torque output of the velocity controller ( $\tau_{fb}$ ) added to the acceleration feedforward ( $\tau_{fb}$ ) into a current determined by the inverse servo motor characteristics curve. The error between this current and the actual measured current is multiplied by the current proportional gain ( $K_{pc}$ ). This output of the current controller is the input voltage ( $U_{fb}$ ) for the servo motor.

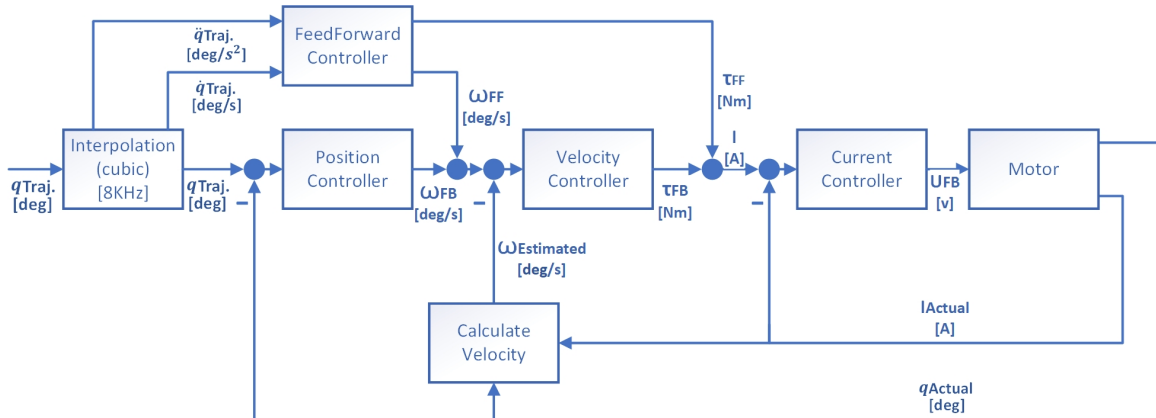


Figure 2.5: Joint controller architecture

### 3 Literature Research

Literature research is performed to investigate the topics system identification and vibration reduction control strategies. The results are described in section 3.1 and 3.2.

#### 3.1 System identification

Several papers in this field are analyzed to investigate the existing system identification. This is used for finding a suitable method to identify the Spider and the gripper-product system.

##### 3.1.a Spider system identification

System identification of the Spider is required because the first eigenfrequency of the Spider must be identified to compare with the eigenfrequency of the gripper-product system. If these frequencies are close to each other, extra measures must be considered to increase the Spider first eigenfrequency. Subsequently this prevents excitation of the eigenfrequency of the gripper-product system by the mechanical resonances caused by the first eigenmode of the Spider. To identify the Spider first eigenfrequency three methods can be used. First, a theoretical approach by creating a dynamic model of the Spider by deriving the equations of motion with Lagrange's equations for example. Next, linearizing the equations of motion and determining the eigenfrequency by solving the characteristic equation [4]. Second, an experimental method by applying system identification to the Spider, for example, with a closed-loop indirect measurement, by injecting a chirp signal between the output of the joint controller and the input of the servo's [5]. The problem with this method is the interface between the controller and the servo, which is unavailable for the used motor drive. Third, the Spider can be identified by applying an impulse with a modal hammer to the tool center point (TCP) and measure the response over time by an accelerometer applied to the TCP [6]. A simplified version of this method is to measure the impulse response with the internal encoders of the servo motors. The measured displacement on joint level is also suitable to determine the first eigenfrequency of the Spider. This last method is chosen because of its simplicity and is as effective as the theoretical approach [6].

##### 3.1.b Gripper-product system identification

To investigate the oscillating behavior of the gripper-product system, an identification method is required. With this method the eigenfrequency and relative damping of this system can be determined. These modal parameters give a full description of the dynamic characteristics of the gripper-product system. First, to determine these parameters, a monitoring method is required to track the oscillating behavior. Two types of tracking are possible: with and without direct contact to the system [7]. Examples of methods with direct contact are: applying an accelerometer, transducer, or strain gauge to the system. This has a downside. On the one hand, the mass of the system will be increased which changes the dynamic behavior [8]. On the other hand, cables are needed to read out the sensor data, reducing the Spider movement freedom and introducing extra springs and dampers to the system. The second approach is based on monitoring the dynamic behavior with a high-speed camera [7] and tracks an AprilTag that is attached to the system [7] [9]. This tag exists out of a small piece of printed paper which will not significantly increase the system's weight. AprilTags are developed to precisely track the 3D position and orientation of a moving object. This last approach is chosen to track the oscillating behavior of the gripper-product system. Second, tracking the oscillating behavior results in displacement data over time; this data can be used to derive the modal parameters of the gripper-product system. The relative damping can be determined by the logarithmic decrement method [10]. This method is suitable for underdamped exponentially decaying systems. Based on the preliminary measurements it is determined that this method can be applied to the gripper-product system [11]. The relative damping can be calculated by solving Equation 3.1, with  $A_{tn}$  and  $A_{tn+1}$  as the amplitude of two successive peaks and  $\zeta$  as the relative damping.

$$\zeta = \frac{\delta}{\sqrt{((2\pi)^2 + \delta)}}, \text{ where, } \delta = \ln\left(\frac{A_{tn}}{A_{tn+1}}\right) \quad (3.1)$$

The eigenfrequency can be derived in two ways: by obtaining the vibration period between two successive peaks in the displacement data [7] or by converting the displacement data to the frequency domain by a fast Fourier transform (FFT) [12]. Based on the successive peaks, the eigenfrequency can easily be obtained by solving Equation 3.2, with  $\tau_v$  as the vibration period,  $f_v$  as the damped eigenfrequency, and  $f_n$  as the undamped eigenfrequency. If  $\zeta$  is small, it can be assumed  $f_v$  is equal to  $f_n$ .

$$f_n = \frac{f_v}{\sqrt{1 - \zeta^2}}, \text{ where, } f_v = \frac{1}{\tau_v} \quad (3.2)$$

The second method provides a frequency representation of the displacement data, including the power of each frequency present in the time data. In this way, the dominant frequency can easily be determined to find the eigenfrequency of the gripper-product system. These two methods are both suitable for determining the eigenfrequency. In chapter 4 are both used to determine the eigenfrequency and is the most convenient method chosen.

## 3.2 Vibration reduction control strategies

To investigate the existing vibration reduction control strategies, a literature research is executed. In this research, three possible control strategies are found. These are: digital filtering, jerk-limiting, and input shaping. All methods can be applied to the Cartesian space trajectory of the Spider.

### 3.2.a Digital filtering

The first method applies a digital filter to the trajectory to cancel out a specific frequency. By applying, for example a notch- or low-pass filter the excitation of the eigenfrequency of the gripper-product system by the movement of the Spider can be reduced [13] [14].

### 3.2.b Jerk-limiting

Another method can be applied to cubic (third order) and higher-order trajectories to limit the jerk (third derivative). For a cubic trajectory, the acceleration profile is trapezoidal or triangular-shaped. The maximum jerk determines the slope of these shapes ( $J_{max}$ ) and the jerk time ( $T_J$ ) as shown in Figure 3.1. The eigenfrequency can be suppressed by setting the jerk time to an integer multiple of the period of the eigenfrequency [15].

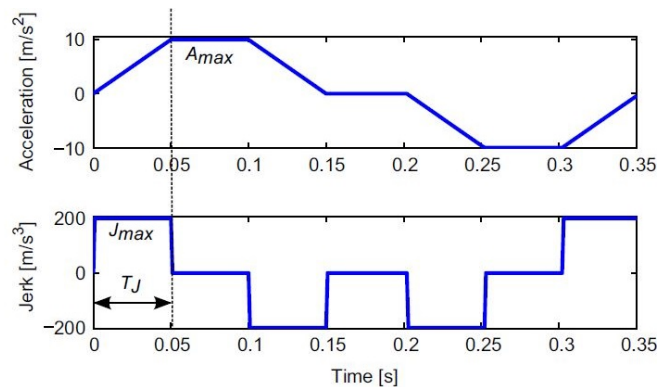


Figure 3.1: Cubic trajectory with jerk-limiting, [15].

### 3.2.c Input shaping

Input shaping is developed for reducing vibrations of flexible underdamped dynamic systems. The principle of the method is based on canceling vibrations by applying multiple impulses to the system's input. This process is illustrated in Figure 3.2. By applying the first impulse, the system will start to oscillate. The second impulse will cancel the oscillation induced by the first impulse if it has the correct amplitude and is applied after half the vibration period. [16].

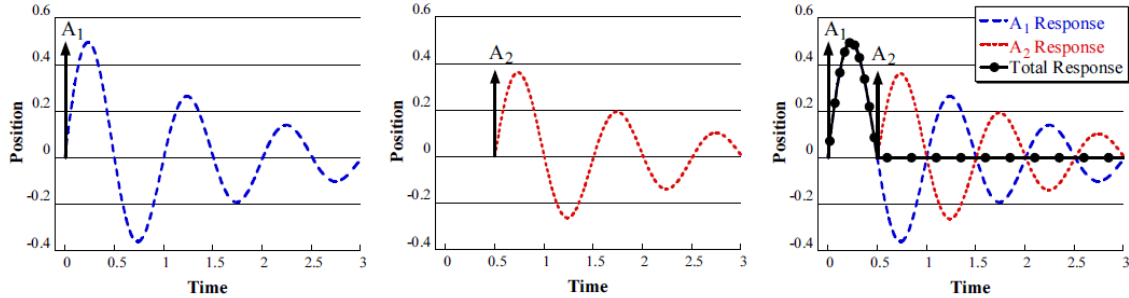


Figure 3.2: Input shaping principle, [16].

This principle can be used to suppress the excitation of a particular frequency caused by a robot's movement. This can be realized by convoluting the trajectory of the movement with the set of impulses. Convolution of impulses with a continuous or discrete trajectory can easily be executed in two steps. First by multiplying the trajectory with the amplitude of the first impulse. In the second step the trajectory is multiplied with the amplitude of the second impulse and delayed by the time of the second impulse. The shaped trajectory can be realized by adding the result of the two steps. If the input shaper contains more than two impulses, then the shaper can be realized by repeating step two for the additional impulses as shown in Figure 3.3. Convoluting the trajectory with the impulses will increase the duration of the resulting shaped trajectory with the duration of the set of impulses ( $\Delta t$ ) [16].

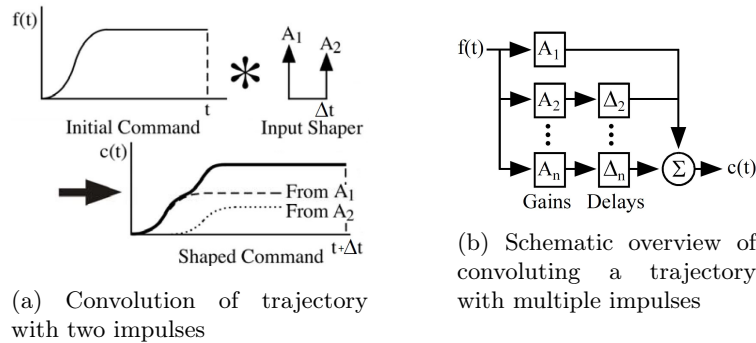
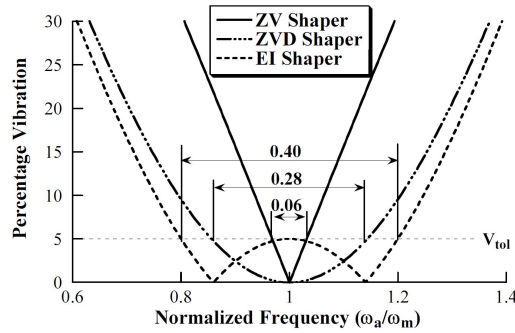


Figure 3.3: Convolution to obtain the shaped trajectory, [16]

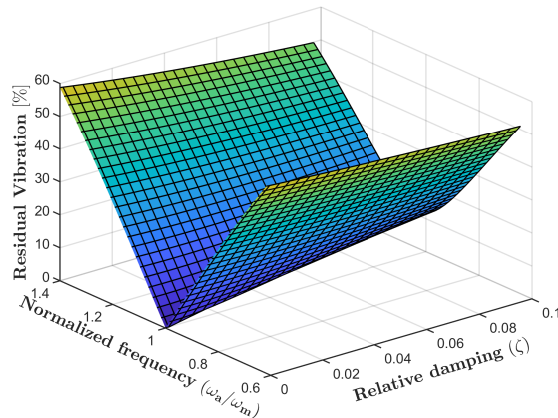
The number of impulses determines the duration of the shaper and the robustness to eigenfrequency variations of the system. For every impulse added to the minimum number of two, the shaper duration increases with half a vibration period and increases the robustness. For choosing the most suitable shaper for a certain system, the consideration between shaper time and robustness must be taken into account. The simplest shaper named the zero vibration shaper (ZV shaper), contains two impulses. This shaper can be formulated based on Equation 3.3. The amplitude of the two impulses is dependent on the eigenfrequency ( $\omega_n$ ) in *rad/s* and relative damping ( $\zeta$ ) of the system. Logically, the timings are at zero seconds and after half a vibration period [16].

$$\begin{bmatrix} A_i \\ t_i \end{bmatrix} = \begin{bmatrix} \frac{1}{1+K} & \frac{K}{1+K} \\ 0 & \frac{\pi}{\omega\sqrt{1-\zeta^2}} \end{bmatrix}, \quad \text{where, } K = e^{\left(\frac{-\zeta\pi}{\sqrt{1-\zeta^2}}\right)} \quad (3.3)$$

The robustness of the ZV shaper can be compared to the higher-order shapers by evaluating the percentage vibration Equation [16]. This equation gives insight into the residual vibration after applying input shaping to a system with a certain variation of the eigenfrequency which specifies the insensitivity of the shaper for frequency changes. Figure 3.4 compares the ZV Shaper to the zero vibration derivative shaper (ZVD Shaper) and the extra insensitivity shaper (EI shaper). The last two shapers have both a duration of one period instead of half a period. This leads to an insensitivity improved for the higher-order shapers, for example, if the maximum allowed residual vibration is five percent ( $V_{tol}$ ). The ZV shaper can suppress the vibration with a maximum frequency variation of  $\pm 3$  percent, instead of  $\pm 14$  and  $\pm 20$  for the higher-order shapers.



(a)



(b)

Figure 3.4: (a) Shaper insensitivity comparison [16], (b) ZV shaper residual vibration with varying eigenfrequency and relative damping

To choose the correct shaper for the gripper-product system, the comparison between insensitivity and shaper time must be made. Based on the preliminary measurements the eigenfrequency is estimated at  $2.5 \text{ Hz}$ . The resulting vibration period is 0.4 seconds. Due to the relatively short pick and place time (0.5 to 1 second), extending this time by half a period has a significant impact. Because the lost time due to the shaper duration must be compensated by increasing the acceleration, this means that duration-wise, the shortest shaper, the ZV shaper, is the best choice. On the other hand, the variation of the eigenfrequency of the gripper-product system and the required vibration reduction must be determined to conclude if the ZV shaper has a sufficient vibration reduction for this application.

### 3.2.d Strategy comparison

Two criteria are used to compare the three control strategies discussed: extended trajectory duration in vibration periods and the residual vibration reduction. The found parameters obtained by the literature study are expressed in Table 3.1

Table 3.1: Comparison between the control strategies based on the residual vibration and extended trajectory duration, expressed in vibration periods[13][15][16].

Control strategy Type	Input shaping			Jerk-limiting	Digital filer	
	ZV	ZVD	IE		Notch	Low-pass
<b>Residual vibration with <math>\omega_n \pm 15\%</math> Variation [%]</b>	23	6	1.4	16.7	18 - 73	10.8 - 54
<b>Extended trajectory duration [periods]</b>	1/2	1	1	2 for every acceleration/deceleration	1 - >10	1 - 8.9

First, the duration of digital filtering varies between one and ten periods, based on which notch or low-pass filter type is used [13]. The duration of the jerk limiting strategy is at least two periods for every acceleration and deceleration in the trajectory [15]. This means that at least four cycles will be added to the trajectory duration for a pick place motion. Compared to the discussed input shapers, the trajectory duration will only be increased with a half or one period [16]. This concludes that duration-wise input shaping is the most suitable strategy for this application. Secondly, a  $\pm 15\%$  variation in frequency is considered to compare the vibration reduction effectiveness between the strategies. The residual vibration by applying eleven different digital filters varies between 10.8% and 54% [13]. The residual vibration after applying jerk-limiting is 16.7%. Finally, the performance of the discussed input shapers is as follows: ZV shaper 23%, ZVD shaper 6%, EI shaper 1.4% [16]. Input shaping is the superior control strategy, due to the short trajectory extension and high vibration reduction. Besides, the ZV shaper with the shortest duration is chosen because the shaper time is more important than the vibration reduction in this case, due to the relatively short trajectory time. Chapter 4 describes the determination of the actual variation in the eigenfrequency of the gripper-product system. Based on this result, the exact vibration reduction for this shaper can be determined. Chapter 6 and 8 describes the ZV shaper tested in a simulation and on the measurement setup. These measurements can determine if the ZV shaper is the most suitable for this application.



## 4 System Identification

In this chapter the Spider and gripper-product system are identified based on the methods as discussed in chapter 3. This identification is done to obtain the Spider's first eigenmode and the modal parameters of the gripper-product system. Finally, a frequency analysis of the Spider trajectory and the effect on the gripper-product system is executed.

### 4.1 Spider identification

The identification of the Spider is executed by positioning the Spider in its homing position  $(0, 0, 600)$   $(x, y, z)$ . During the measurement the Spider is actively controlled to this position by the motor drives. The impulse is applied at the traveling platform with the modal hammer in three directions  $(x, y, \text{and } -z)$ . The impulse is applied three times in every direction as shown in Figure 4.1.

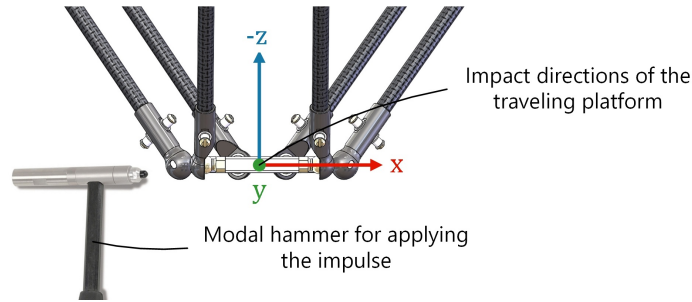


Figure 4.1: schematic overview of the Spider identification, based on a modal hammer measurement

The motor drives measure the response in the three joint accelerations with a sample rate of  $1 \text{ kHz}$ . These accelerations are translated to the frequency domain by a FFT as shown in Figure 4.2. The impulse response in the frequency domain is applied three times in every direction and is shown for every joint. The first peaks indicates the first eigenfrequency. these are for all responses determined by analyzing the frequency data and averaged, this results in  $36.59 \text{ Hz}$  with a deviation of  $\pm 13.25\%$ . Zero padding is applied to increase the frequency resolution because the impulse responses are relatively short (0.5 to 1 s). This measure increases the resolution from 4 to  $0.23 \text{ Hz}$ .

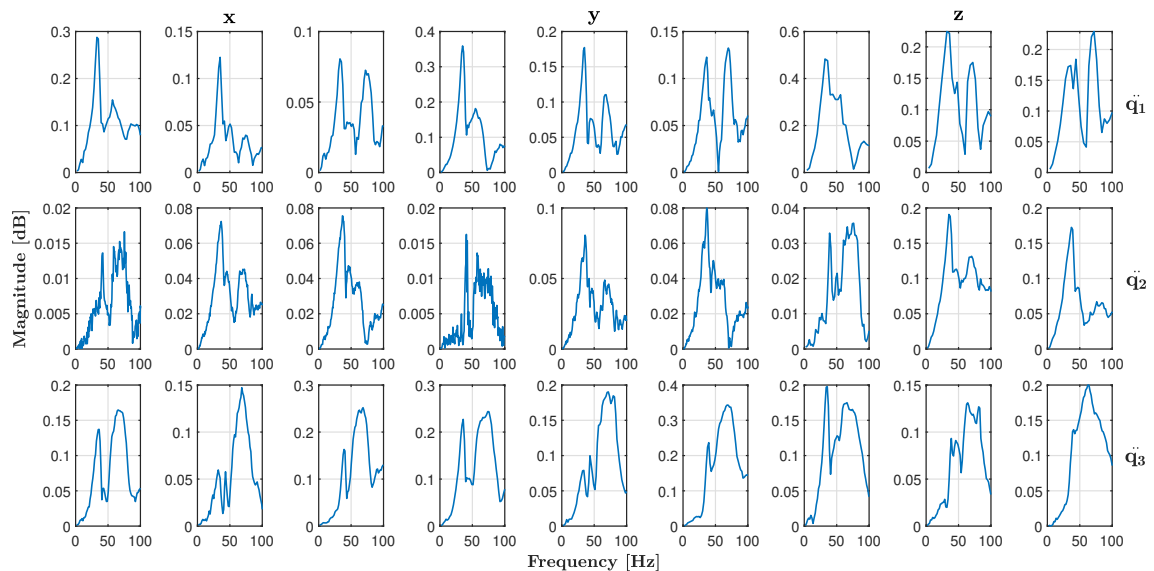


Figure 4.2: Frequency representation of joint acceleration impulse response.

## 4.2 Gripper-product system identification

The gripper-product system is identified to derive the eigenfrequency ( $f_n$ ) in  $Hz$  and relative damping. These modal parameters are required to design the ZV Shaper. Besides, the variation of the two parameters is determined to investigate the actual vibration reduction of the ZV shaper. The preliminary measurements indicate that the vibration is caused by five factors: the gripper type, product variations, vacuum level, pick position, and the trajectory. First, the gripper type defines the distance between the product center point (PCP) and the tool center point (TCP), the connection rigidity between the two, the vacuum force applied to the product related to the size of the gripper opening. Second, the product can vary in mass, viscosity, and shape. Third, the vacuum level determines the force on the product, the amount of plastic that is sucked into the gripper, and the compression of the vacuum gripper. The compression influences the distance between the TCP and the PCP. Fourth, the variations in the pick position determine the orientation of the product underneath the Spider and also influence the TCP to PCP distance. Last, the length of the trajectory determines the maximum velocity that can be reached. Therefore, the longest trajectory possible (in the  $xz$  plane) is chosen with the highest velocity. This fifth factor is chosen constant, so only the variations of the first four factors are considered in this research.

The following equipment is required for the identification: AOS Promon U1000 high-speed camera, LED lighting, Galbani mozzarella packages, Piab BL40-5, and BX52P grippers. Figure 4.3 shows the measurement setup. The high-speed camera is used with a frame rate of  $200 Hz$ , this means that frequencies up to  $100 Hz$  can be detected without aliasing. Besides, the frequency band of interest is between  $2.5$  and  $42.35 Hz$  based on the preliminary measurements and the first eigenmode of the Spider. Based on this information it can be assumed that  $200 Hz$  is a suitable sample frequency. Second, due to the high frame rate of the camera and the resulting short shutter time, extra lighting is required and is provided by six led panels powered by a  $24$  volts DC power supply. Third, one type of mozzarella package is chosen for all measurements in this research. To fully understand the oscillating behavior, variation in the modal parameters and the effect of input shaping on the residual vibration is estimated. The project phase's complete duration is required. The Galbani Mozzarella is chosen due to its broad availability. This product contains a mozzarella ball of  $125 g$  and weighs  $225 g$  in total. The size of the ball is  $70 \times 40 \times 60 mm$ , and the total dimension are  $140 \times 30 \times 125 mm$  (WxHxD). Finally, two types of grippers are chosen. The BL40-5 is relatively small in diameter ( $40 mm$ ) and flexible, which results in an undue oscillating behavior compared to the behavior as described in chapter 1 and gives insight into the nature and causes of the oscillating behavior. The BX52P gripper is chosen in collaboration with BPA and can be used in actual mozzarella packaging machines. This gripper is less compliant and has a larger vacuum area. Figure 4.4 shows the chosen product attached to the two gripper types.

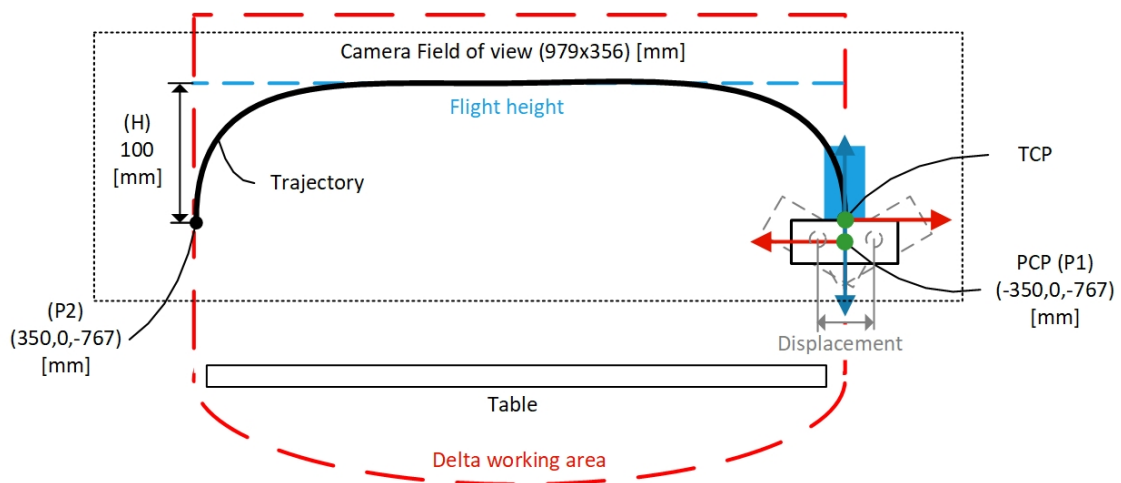


Figure 4.3: Schematic representation of the measurement setup

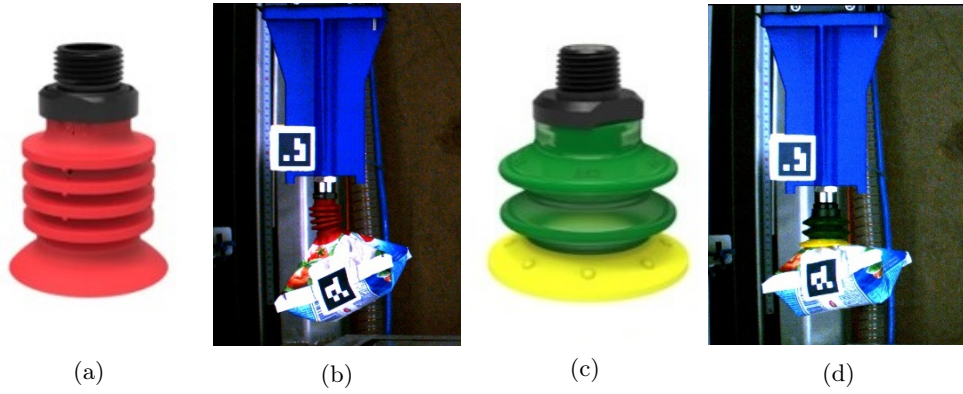


Figure 4.4: (a) BL40-5 gripper, (b) Galbani mozzarella attached to the BL40-5, (c) BX52P gripper, (d) Galbani mozzarella attached to the BX52P.

To identify the contribution of the four named factors to the variation of the modal parameters, two test sets are determined as shown in Table 4.1. The first set determines the variation in the pick position with the two types of grippers and three different vacuum levels. This set exists out of 25 repetitions of every combination between the gripper and vacuum level (subset). This results in 150 measurements for the first set, executed with a single product to exclude the product variation. The second set determines the product variation and contains the same combinations between the gripper and the vacuum level but is executed with 25 different Galbani mozzarella packages. This set also results in 150 tests. The used trajectory is equal for all measurements, and the accelerations for the two grippers are set to  $5 \text{ m/s}^2$  and  $10 \text{ m/s}^2$ .

Table 4.1: Test sets parameters

Measurement sets	Pick position variation		Product variation	
<b>Gripper</b>	<b>BL40-5</b>	<b>B52XP</b>	<b>BL40-5</b>	<b>B52XP</b>
<b>Trajectory</b>	(-350,0,-767), (350,0,-767)			
<b>Product</b>	1 Sample		25 Samples	
<b>Amax [m/s<sup>2</sup>]</b>	5	10	5	10
<b>Vacuum levels [mBar]</b>	-125, -250, -500	-250, -375, -500	-125, -250, -500	-250, -375, -500

The video footage captured with the high-speed camera is processed with a MATLAB AprilTag detection function. It results in displacement data over time in two directions ( $x, z$ ) of the PCP and TCP. The PCP displacement in the horizontal ( $x$ ) direction is used to derive the modal parameters. The measurement data is processed as shown in Figure 4.5, which is necessary because the data contains false tag detections, which results in spikes. A spike- (or median) filter removes these detections from the data. Besides, the displacement data is used to determine the velocity and acceleration by deriving the first and second derivatives. The acceleration is directly related to the forces applied to the PCP, so this gives a good insight in how the Spider's movement translates in the forces applied to the PCP. The relatively low sample frequency of the high-speed camera for calculating derivatives, results in a noisy velocity and acceleration signal. By applying a second spike filter (10th order median) the noise is removed from the signals. Figure 4.6 shows the raw and processed position, velocity, and acceleration data. The spikes in the unfiltered data are clearly present in the position data as well as in the acceleration data due to the two times deriving of the position data.

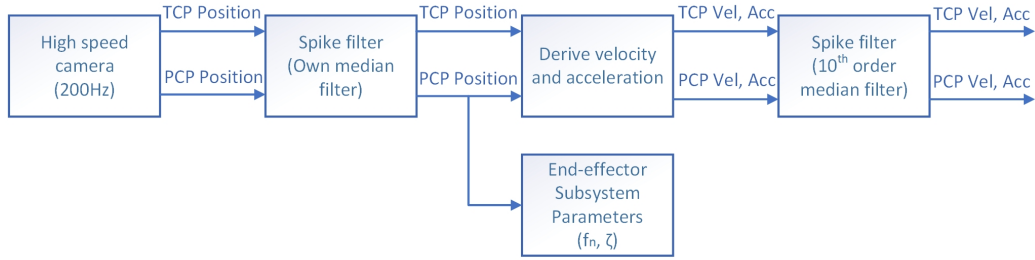


Figure 4.5: Displacement data processing structure

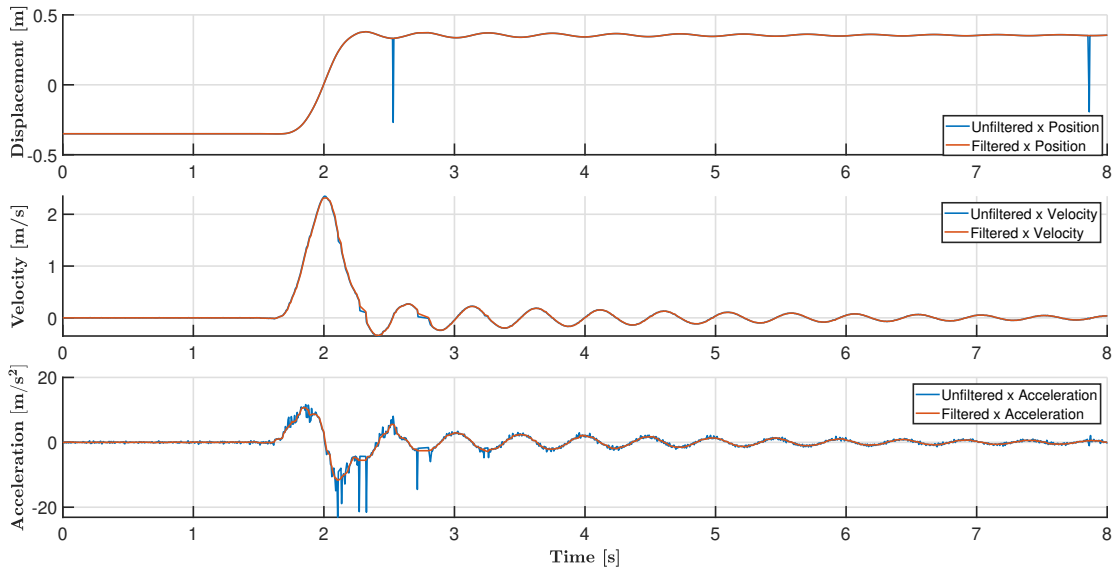


Figure 4.6: Raw and processed position, velocity, and acceleration PCP data

The processed position data is used to determine the modal parameters. The relative damping is derived by Equation 3.1 based on the detected peaks as shown in Figure 4.7. The eigenfrequency is determined in two ways: based on Equation 3.2 and by computing the FFT as shown in Figure 4.7. The two methods are compared and concludes that the FFT method is superior to the other method. The peaks are sometimes not fully and accurately detected caused by false tag detections or absence of a detectable tag. For determining the relative damping this effect can be neglected. Because the amplitude is less effected by the inaccurate peak detection. The advantage of the FFT method is that the frequency can also be determined if a small part of the oscillation is missing. The movement from the pick to the place position is excluded from the FFT data to prevent the frequencies introduced by this movement appear in the frequency spectrum. The FFT data has a minimum frequency resolution of  $0.66 \text{ Hz}$

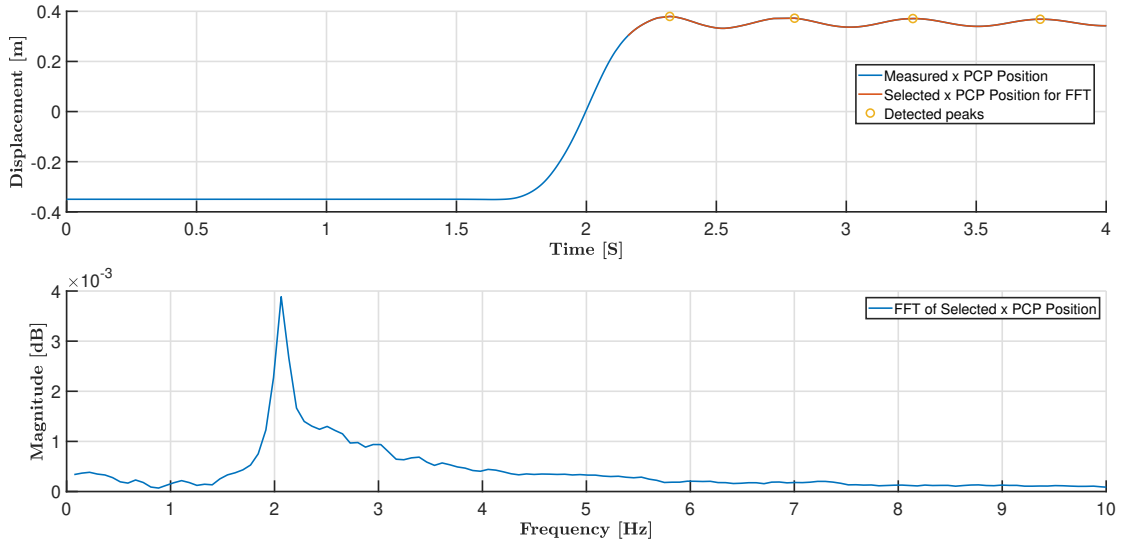


Figure 4.7: Detected peaks for determining the relative damping and Frequency representation of the selected position data

The results of the two measurements sets are presented in Table 4.2 and 4.3. During the computation of this data, became clear that the derived relative damping's and eigenfrequencies contain outliers due to inaccurate or false tag detections. These are removed from the data by determining the mean of every test subset and removing data outside  $\pm 3\sigma$ .

Table 4.2: Product variation measurement results

Measurement set	Pick position variation					
	BL40-5			BX52P		
Gripper	1	2	3	4	5	6
Measurement subset	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>
Vacuum [ <i>mBar</i> ]	<b>-125</b>	<b>-250</b>	<b>-500</b>	<b>-250</b>	<b>-375</b>	<b>-500</b>
$f_n$ median [Hz]	2.05	2.39	2.78	3.33	3.97	3.90
$f_n$ deviation [ $\pm$ %]	3.05	8.45	12.12	17.71	10.18	11.12
$\zeta$ median	3.11e-04	5.52e-04	8.10e-04	0.0019	0.0013	0.0012
$\zeta$ deviation [ $\pm$ %]	98.03	96.78	99.03	98.49	98.22	99.45

Table 4.3: Test sets results

Measurement set	Product variation					
	BL40-5			BX52P		
Gripper	1	2	3	4	5	6
Measurement subset	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>
Vacuum [ <i>mBar</i> ]	<b>-125</b>	<b>-250</b>	<b>-500</b>	<b>-250</b>	<b>-375</b>	<b>-500</b>
$f_n$ median [Hz]	2.02	2.30	2.47	3.32	3.59	3.68
$f_n$ deviation [ $\pm$ %]	3.96	12.87	11.46	22.04	23.13	26.19
$\zeta$ median	2.96e-04	4.80e-04	3.51e-04	5.52e-04	0.0012	9.92e-04
$\zeta$ deviation [ $\pm$ %]	97.41	98.02	98.61	99.31	99.50	99.06

The results as shown in Table 4.2 and 4.3 and the product behavior underneath the gripper as shown in Figure 4.4 concludes that the product can be compared to a second-order pendulum. With the revolute joint at the bending point of the gripper (defined as the TCP) and the point mass at the PCP of the product. Because when the distance between the TCP and PCP decreases, the

eigenfrequency increases as formulated in the eigenfrequency formula of a pendulum. This formula is shown in Equation 4.1, with gravity  $g$  and arm length  $l$  (TCP to PCP). This effect matches with the measurement results.

$$f_n = \frac{\sqrt{\frac{g}{l}}}{2\pi} \quad (4.1)$$

The relative damping is determined by the spring and damping constant inside the revolute joint. The influence of the four named factors on the modal parameters based on the pendulum principle is discussed below. Chapter 5 reviews the pendulum principle in detail.

First, The flexible BL40-5 gripper results in a lower eigenfrequency ( $f_n$ ) and relative damping ( $\zeta$ ) compared to the stiffer BX52P gripper. Besides, the gripper has only influence on the  $f_n$  deviation. This is caused by the difference in gripper openings. The small opening of the BL40-5 gripper can suck a small amount of foil from the package into the gripper. Due to the flexible structure of the product, will it align itself underneath the gripper, which results in less variation of the PCP position relative to the TCP. Compared to the bigger gripper that consumes more foil, the product has less the potential to align itself resulting in more variation in the PCP position.

Second, increasing vacuum level results in an increase in  $f_n$  and  $\zeta$ , which is caused by the decrease in distance and stiffening of the gripper. This is not the case for the BX52P gripper at a vacuum level of  $-500 \text{ mBar}$ , because the gripper is at  $-375 \text{ mBar}$  is already fully compressed and the distance can not be further reduced. Besides, this highest vacuum level deforms the foil and gives visible traces after handling. The aligning effect as described above, is amplified by increasing the vacuum level resulting in more deviation of  $f_n$ .

Third, the influence of the product variation on the  $f_n$  deviation is clearly visible by comparing the two sets because the first set is without product variation and the second with product variation. The difference between the two is the contribution by the product variation.

Fourth, the influence of the pick position to the  $f_n$  deviation is present in both sets. The deviation of the first set gives the quantity of influence caused by the pick position. This shows the pick position contributes for average 69.2% to the total deviation, which seems logical because the pick position has much influence on the distance between the PCP and TCP.

Finally, the combinations between the BL40-5 gripper and  $-500 \text{ mBar}$  (third measurement subset) and the BX52P gripper and  $-375 \text{ mBar}$  (fifth measurement subset) are chosen for further investigation. these subsets are hereinafter referred to as the flexible (third measurement subset) and stiff (fifth measurement subset) subsets.

#### 4.2.a ZV shaper theoretical vibration reduction

With the deviations in  $f_n$  and  $\zeta$  of the two subsets known, the theoretical residual vibration can be determined after applying the ZV shaper. Figure 3.4 shows that the vibration reduction is mainly dependent on the variation in  $f_n$ . The influence of  $\zeta$  can be neglected if  $\zeta$  is smaller than 0.01. This means that only the deviation of  $f_n$  is considered. Figure 4.8 shows the distribution of  $f_n$  deviation illustrated in a histogram, compared with the residual vibration curve. The majority of the  $f_n$  deviation in the flexible subset is distributed close to the set shaper frequency; the most extreme deviation results in a 20% residual vibration. The distribution is similar in the stiff subset but the outer frequencies result in a 50% residual vibration. This concludes that the ZV shaper theoretically realizes a minimum reduction in vibration of 50%. Chapter 6 describes the shaper effect tested in a simulation, and chapter 8 the effect tested on the measurement setup.

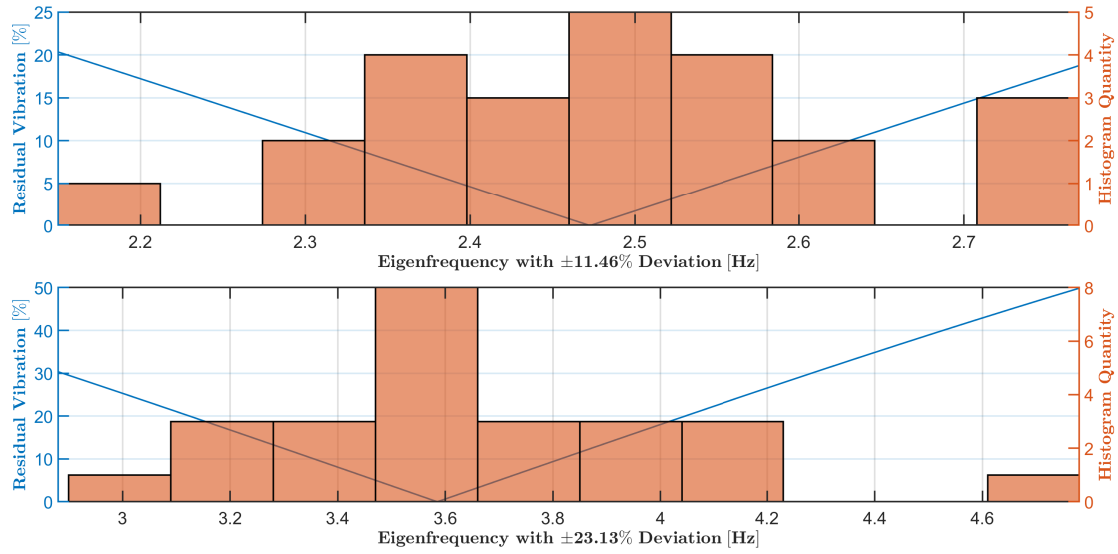


Figure 4.8: (above) Distribution of the eigenfrequency compared to the residual vibration of the flexible subset, (below) of the stiff subset.

### 4.3 Trajectory frequency analysis

With the first eigenfrequency of the Spider and the gripper-spider system known, it is clear that these are not close together so additional mechanical improvements to the Spider are unnecessary. Besides, the trajectory executed by the Spider also introduces a range of frequencies. Investigating if the movement of the Spider excites the eigenfrequency of the gripper-product system, gives insight into the cause of the oscillating behavior. This analysis is executed for both subsets and the same trajectory is used as in Section 4.2, but with higher accelerations ( $10 \text{ m/s}^2$ ,  $32.5 \text{ m/s}^2$ ) to amplify the excitation. Figure 4.9 shows the result of the measurement based on the flexible subset, the TCP position, velocity, and acceleration are measured in two ways: by the AprilTag on the TCP and by the joint controller. Comparing the two concludes that the AprilTag tracking is accurate.

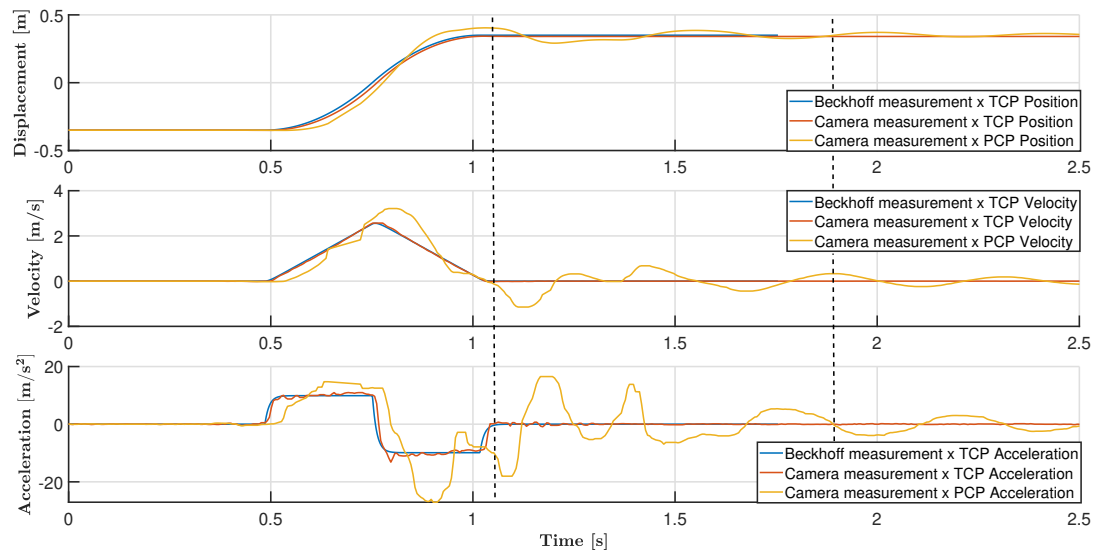


Figure 4.9: Horizontal TCP and PCP displacement, velocity, and acceleration with configuration of the flexible subset

The acceleration data is converted to the frequency domain by a FFT as shown in Figure 4.10. The top figure contains the FFT of the complete length of the time data. In the middle, the FFT is shown of the data from the end of the Spider's movement (first dashed line) until 2.5 s. The bottom plot contains the frequency data of the last part (second dashed line to 2.5 s). The first plot clearly shows that the frequency-band excited by the Spider overlaps with the eigenfrequency of the gripper-subsystem (2.7 Hz). This concludes that the excitation of the oscillating behavior of the gripper-product system is a result of the movement of the Spider. The middle plot shows multiple frequency peaks in the PCP data. This is a result from the non-linear behavior and higher eigenmodes that appear directly after the spider movement. This can also be recognized by the non-sinusoidal behavior between the two dashed lines. The ZV Shaper will not reduce these higher frequencies, so there is always a residual vibration even if there is no deviation in the set shaper frequency. The bottom plot clearly shows the first eigenmode of the gripper-product system. With these results, it can be concluded that the oscillating behavior can be reduced by suppressing the frequency in the Spider trajectory with input shaping. In the next chapter is the Spider and gripper-product system modeled for simulating the effect of the shaper on the residual vibration.

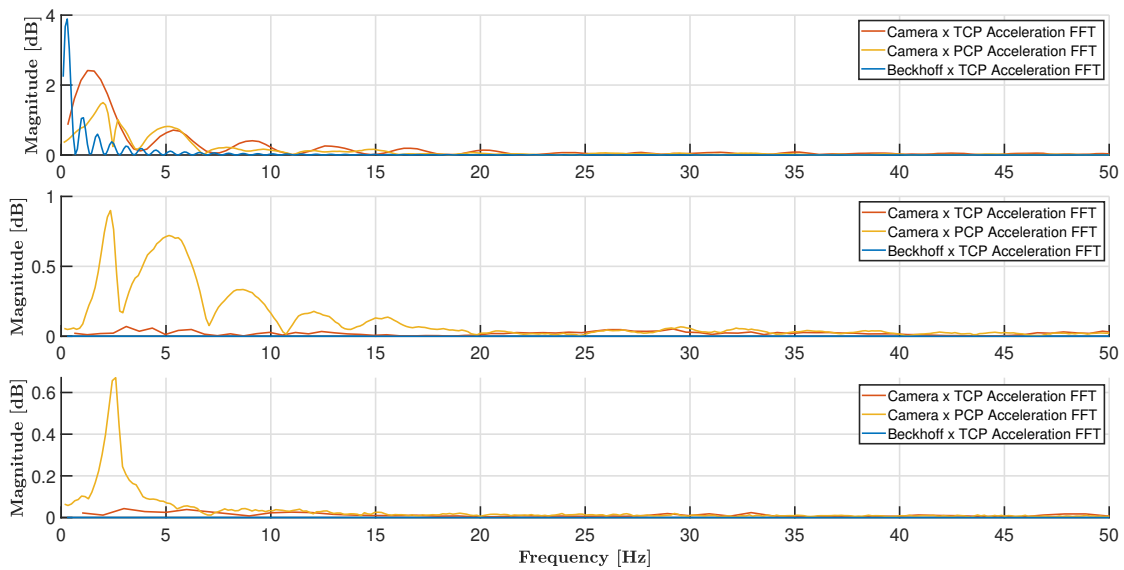


Figure 4.10: (above) FFT of the complete time data, (middle) FFT from the first dashed line to the end of the time data, (below) FFT from the second dashed line to the end of the time data.



## 5 Modeling

In this chapter the Spider and gripper-product system is modeled in MATLAB for simulation purposes. The location of the measurement setup limits the possibility of testing in practice, so a simulation environment is desirable. If the designed ZV shaper is proven in the simulation, the shaper will be tested on the actual measurement setup. An accurate model of the Spider and gripper-product system is required to simulate the ZV shaper performance.

### 5.1 Spider model

To simulate the Spider a dynamical model is required. This model is derived from an accurate CAD model. This model contains all mechanical parts with the correct material properties, which defines the mass, center of mass, moments of inertia, and products of inertia of every part. These properties are required to derive an accurate dynamical model. The parts are converted to Simulink SimScape, a toolbox for generating dynamic models. In SimScape the parts are connected with joints to create the parallel structure of the Spider as shown in Figure 5.1. A MATLAB control signal can actuate the three revolute joints of the upper arms. Applying the appropriate joint trajectories to the revolute joints lets the traveling platform (TCP) follow the desired pick and place trajectory. This SimScape model is a rigid body model, based on the results of the frequency analysis in section 4.3, shows a limited excitation of frequencies around the first eigenfrequency, so the Spider be assumed as a rigid body system, for this application.

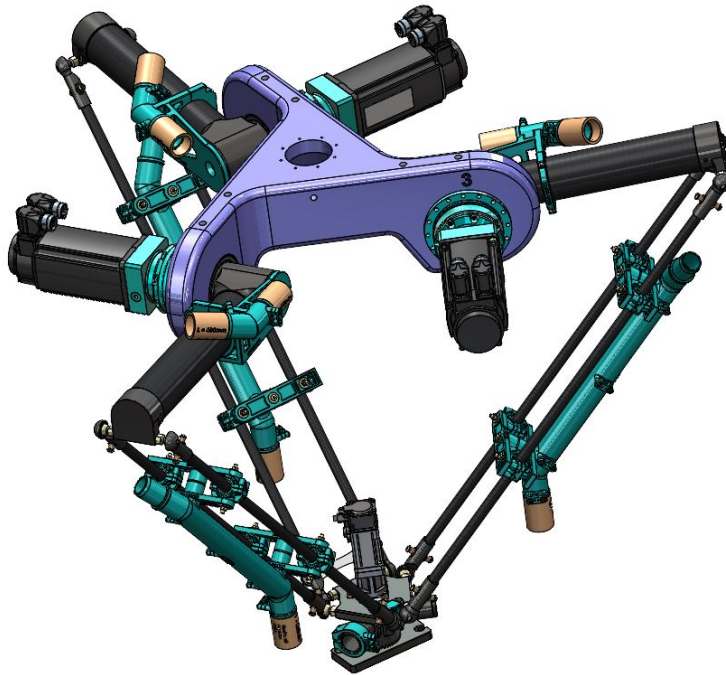


Figure 5.1: Accurate dynamic model of the Spider

#### 5.1.a Joint controller model

The joint controller for simulating the Spider model is modeled as an ideal controller; this can be assumed because the tracking error of the Beckhoff joint controllers is negligible. The ideal controller is selected in the revolute joint actuation settings and is driven by the Simulink control signal containing the joint position trajectory.

## 5.2 Gripper-product system

Simulating the performance of the ZV-shaper requires an accurate model of the gripper-product system. This model can be attached to the TCP of the Spider to simulate the oscillating behavior. The gripper-product system can be compared to a second-order pendulum as determined in Section 4.2. Figure 5.2a shows is this pendulum model schematically. The modal parameters  $f_n$  and  $\zeta$  are dependent on the mass ( $m$ ) in  $g$ , arm length ( $l$ ) in  $mm$ , the torsional spring constant ( $k$ ) in  $\frac{Nm}{deg}$  and the torsional damper constant ( $b$ ) in  $\frac{Nms}{deg}$ . Figure 5.2b presents the gripper-product system modeled in SimScape and based on the second-order pendulum principle.

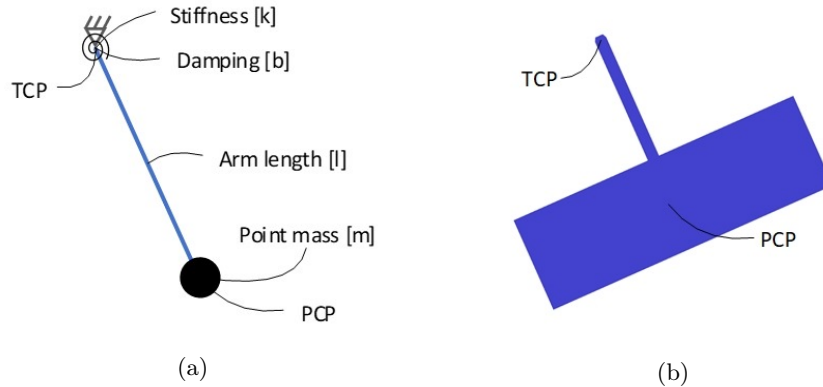


Figure 5.2: (a) Schematic second order pendulum, (b) modeled second order pendulum.

The blue rectangle of the SimScape model indicates the outer dimension of the mozzarella package. The arm and rectangle are modeled massless. At the endpoint of the arm (PCP) a point mass of  $255\text{ g}$  is added. The parameters  $l$ ,  $k$ , and  $b$  are estimated in Section 6.2 using a simulation model. This model is based on the measurement data of Section 4.2 and an optimization algorithm. This results in a model with the same modal parameters as the flexible and stiff subset of Table 4.3. In the next chapter, the Spider in combination with the gripper-product system is simulated with and without input shaping.

## 6 Simulation

In this chapter the simulation structure is discussed, the pendulum parameters are estimated, the effect of input shaping to a trajectory is shown and based on simulations the performance of input shaping is determined.

### 6.1 Simulation software structure

Realizing the simulation environment requires several components. These are schematically shown in Figure 6.1. The first four components are generated in MATLAB, and the actual simulation model is built in Simulink.

First, the pick and place trajectory is realized by the trajectory generator described in: "Delta Robot B-Spline based Trajectory Generation Inverse Kinematics for Food Packaging Applications". This generator has the maximum acceleration ( $A_{max}$ ), the pick ( $P1$ ) and place ( $P2$ ) position, and the trajectory height ( $H$ ) as inputs. The resulting trajectory in Cartesian space is generated on position, velocity, and acceleration level.

Second, Input shaping is applied to the acceleration trajectory because the acceleration is directly related to the forces applied to the gripper-product system. The forces must be limited to prevent peeling off from the gripper, so applying input shaping to the acceleration trajectory gives direct insight into the applied forces by evaluating the shaped trajectory. Input shaping is only applied to the  $x$  axis because the oscillating effect is mainly present in this direction. Besides, applying input shaping on both axes has drawbacks as described in Section 6.3. The  $z$  axis is synchronized with the shaped  $x$  axis by delaying the trajectory before the place movement.

Third, the shaped acceleration trajectory is numerically integrated twice (Forward Euler and Trapezoidal) to obtain the shaped position trajectory.

Fourth, this trajectory is translated to the joint space by applying an inverse kinematic transformation.

Finally, the shaped joint position trajectory is applied to the simulation model, which produces the simulated position, velocity, acceleration, and force data on TCP and PCP levels.

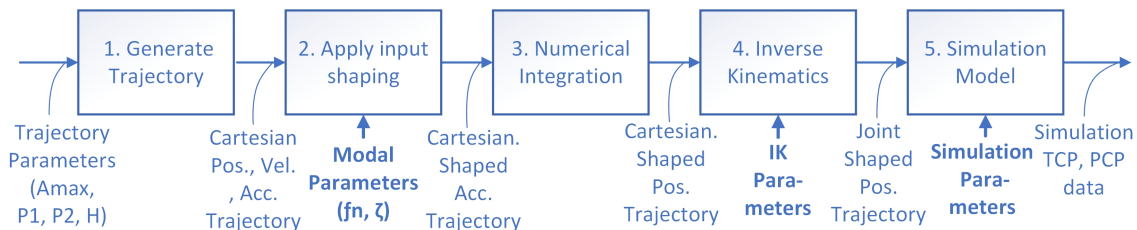


Figure 6.1: Schematic representation of the simulation architecture.

### 6.2 Pendulum parameters estimation

As discussed in Section 5.2, the pendulum parameters  $l$ ,  $k$ , and  $b$  are estimated to obtain two pendulum models containing the modal parameters of the flexible and stiff subset as shown in Table 4.3. The parameters are derived with a self-written simulation-optimization algorithm and are validated by comparing the simulation response with the measurements obtained in Section 4.2, containing the determined modal parameters. The algorithm derives the arm length ( $l$ ) based on the eigenfrequency as explained in Equation 4.1, the torsional damping constant ( $b$ ) based on the relative damping, and the torsional spring constant ( $k$ ) based on the oscillation amplitude of the measurement data. The relation between the amplitude and the torsional spring constant is empirically determined. Figure 6.2 shows the optimization result for the two pendulums compared with the two specific measurements containing the modal parameters. The measurements and simulations deviate with a maximum position error of  $32\text{ mm}$ . Besides, the velocity and acceleration profiles

are comparable and the modal parameters have a deviation of a maximum of 0.16% which indicates that the obtained models are accurate.

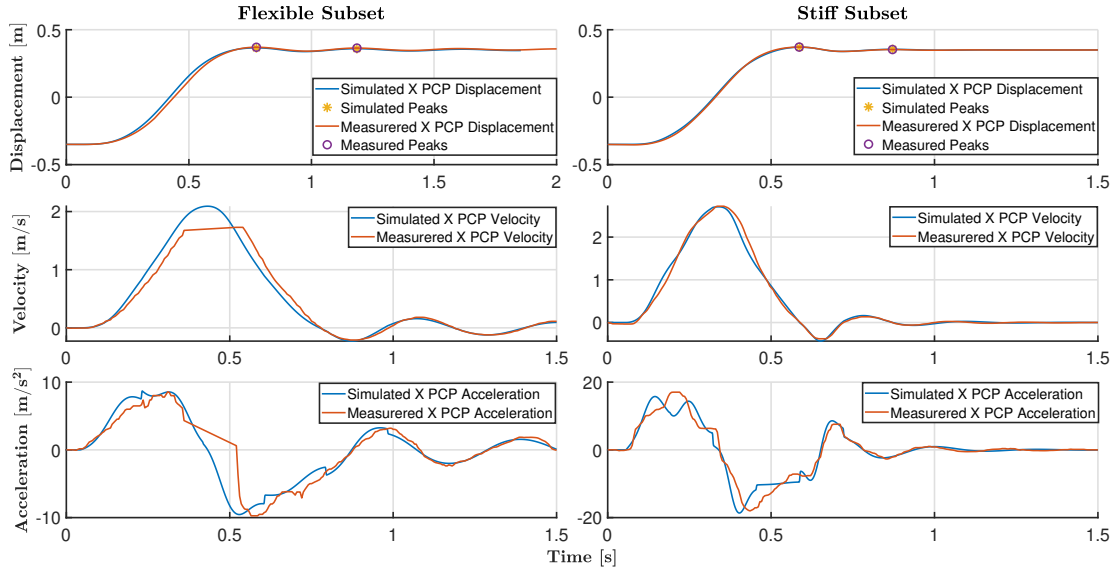


Figure 6.2: Pendulum simulation and measurement comparison of the flexible and stiff subset.

The derived pendulum parameters are presented in Table 6.1. The arm length scales inversely proportional with the eigenfrequency as expected. The stiffness and damping for the stiff subset is higher compared with the flexible. This can be explained by the stiffer BX52P gripper compared with the BL40-5 gripper.

Table 6.1: Pendulum parameters of the subset flexible and stiff.

Subset	Flexible	Stiff
$l$ [mm]	45.85	22.95
$k$ [Nm/deg]	2.00e-05	2.40e-05
$b$ [Nms/deg]	1.00e-07	2.00e-05

### 6.3 Input Shaping Trajectories

As mentioned in Section 3.2.c is the original trajectory deformed by convoluting with the impulses. For a rectangular acceleration profile, the shaped trajectory results in a stepped profile as shown in Figure 6.3. The interval between the steps is a result of the ratio ( $\tau$ ) between the acceleration time ( $t_{Acc}$ ) and the shaper length ( $t_{Shaper}$ ), which is equal to half a vibration period. The shaped trajectory can have three variations based on this ratio :

- $\tau < 1$  for  $t_{Acc}$  longer than  $t_{Shaper}$ .
- $\tau = 1$  for  $t_{Acc}$  is equal to  $t_{Shaper}$ .
- $\tau > 1$  for  $t_{Acc}$  short than  $t_{Shaper}$ .

Figure 6.3 shows how the shaped acceleration profile is affected by the three values of  $\tau$ . These values of  $\tau$  indicate if the product is actuated in the  $x$  direction lower, equal, or higher than its eigenfrequency. Suppose the product is actuated lower than  $f_n$ . In that case, the product will follow the trajectory easily. If  $f_n$  is equal, the product will start to oscillate, and if it is actuated higher than  $f_n$ , the product wants to stay stationary and will cost more effort to move it. This concludes that  $\tau$  must be smaller than one. Besides, if  $\tau$  is equal or higher than one, the maximum reference

acceleration is not reached. Finally, if  $\tau$  is larger than one, the position trajectory is deformed to an S-curve. This impacts the 2D trajectory if input shaping is also applied to the  $z$  axis. Figure 6.4 shows the 2D trajectory for the values of  $\tau$  with input shaping on the  $x$  axis, and on both axes.

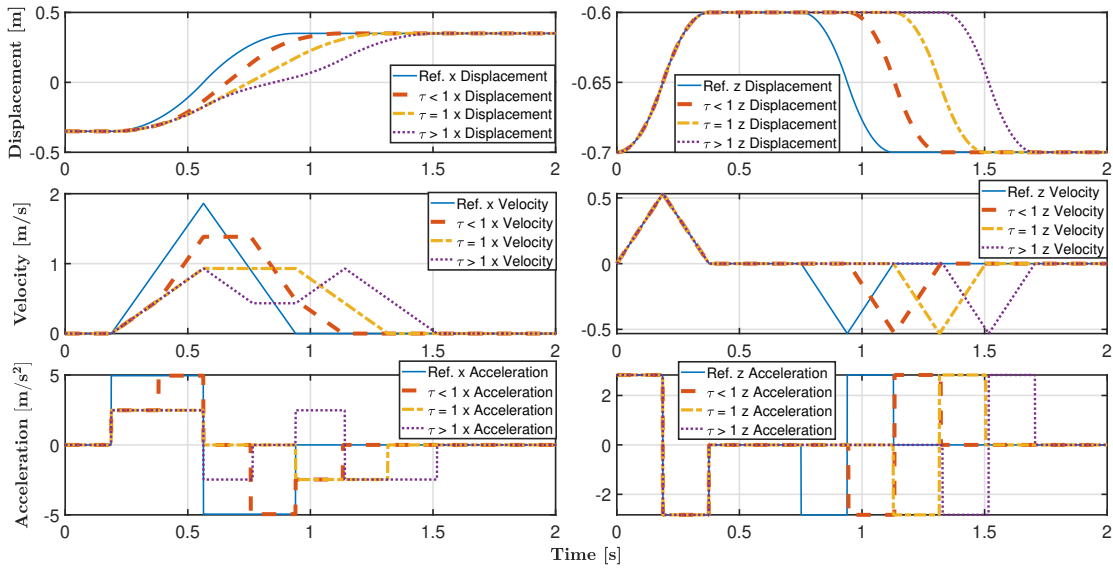


Figure 6.3: Input shaping applied to the  $x$  axis for  $\tau > 1$ ,  $\tau = 1$ , and  $\tau < 1$ , presented on displacement, velocity, and acceleration level. Compared with the unshaped (Ref.) trajectory.

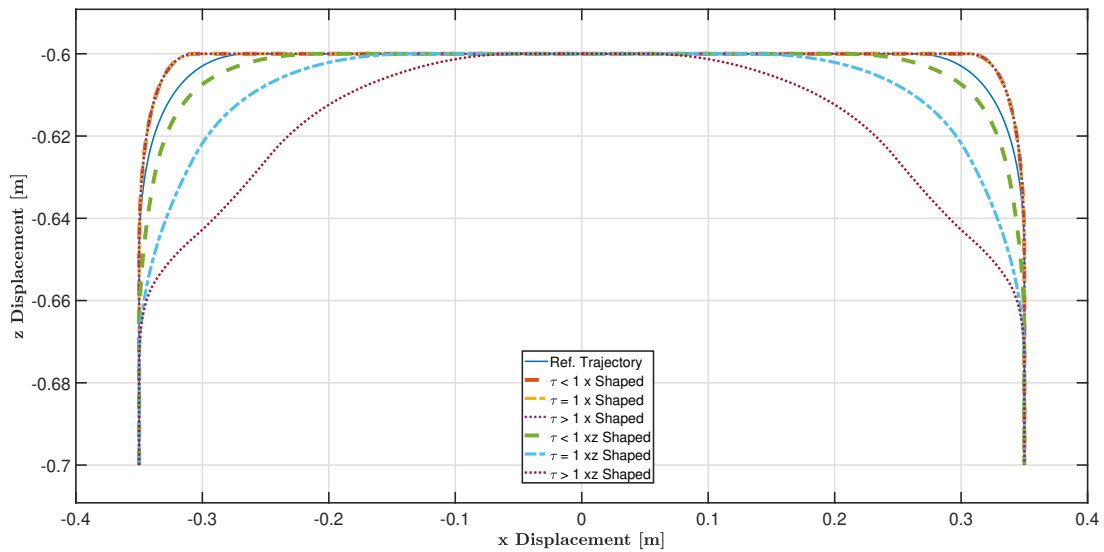


Figure 6.4: 2D trajectory with input shaping on the  $x$  axis and on both axes for  $\tau > 1$ ,  $\tau = 1$ , and  $\tau < 1$ , compared with the unshaped (Ref.) trajectory.

It is clear input shaping on the  $x$  axis reduces the blending radius; this can negatively affect the handling behavior of the product and must be investigated in future research. Applying input shaping to both axes increases the blending radius; this can violate the vertical height constraint. For example, to guarantee the product will be placed vertically inside a box. This vertical height is further reduced if  $\tau$  increases. To prevent the reduction of the vertical placement it is chosen to apply input shaping only on the  $x$  axis.

## 6.4 Input Shaping results

in this section the derived pendulums combined with the Spider model of Section 5.1 are simulated to determine the performance of the ZV shaper. The simulation is executed based on the parameters of Table 4.1 and the flexible and stiff subset of Table 4.3. The performance of the shaper is determined based on the following indicators: settling time (the time it takes for the horizontal position error to stay below  $\pm 2\%$  relative to the final position), overshoot (the percentage overshoot of the largest horizontal position peak relative to the final position), trajectory time (the total duration of the setpoint trajectory provided to the Spider), and the residual vibration compared to the unshaped response. These indicators are graphically explained as shown in Figure 6.5.

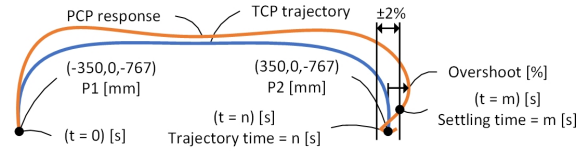


Figure 6.5: graphically representation of the the ZV shaper performance indicators.

The motion time is determined based on the trajectory- and settling time and is equal to the longest of the two. The overshoot and residual vibration both indicate the degree of horizontal vibration, the difference is the overshoot is only relative to the end position of the trajectory. The residual vibration is determined by comparing the overshoot of the unshaped versus the shaped response. The most applicable of the two is chosen in the coming simulations and measurements. The variation of the eigenfrequency ( $f_n$ ) as shown in Figure 4.8 is taken into account for the simulation to investigate if the theoretical vibration reduction of the ZV shaper can be validated in the simulation. Figure 6.6 shows for the flexible subset, the shaped and unshaped TCP  $x$  and  $z$  trajectory (dashed lines). This trajectory is presented on displacement, velocity, and acceleration level. Besides, the shaped and unshaped PCP response is shown. The thin purple lines indicate the response for the maximum deviation  $f_n$  of  $\pm 11.46\%$  (2.15 to 2.77 Hz). Table 6.6 shows the performance of the ZV shaper is evaluated based on the named position criteria. The 2D position trajectories and responses are shown in Figure 6.7.

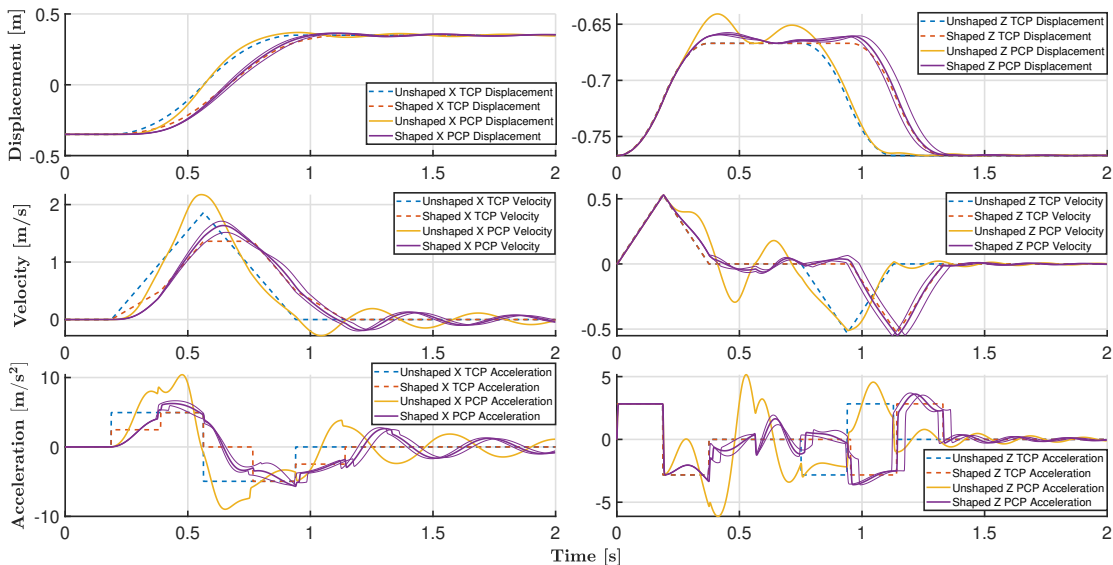


Figure 6.6: The simulated unshaped- (dashed blue line) and shaped (x) (dashed orange line) trajectory and the measured unshaped response (yellow line) and double shaped (purple line) of the flexible subset on position, velocity and acceleration level, in the  $x$  and  $z$  direction.

Table 6.2: Simulated performance indicators of the ZV shaper applied on the  $x$  axes with the total motion time, maximum acceleration on PCP level, and residual vibration in the  $x$  direction, for the flexible and Stiff subset.

Subset Shaping	Flexible				Stiff			
	Unsh.	Shaped (x)			Unsh.	Shaped(x)		
Deviated $f_n$ [Hz]	(2.47)	2.11	(2.47)	2.77	(3.59)	2.90	(3.59)	4.80
Motion time [s]	1.41	1.33	1.33	1.33	0.80	0.94	0.94	0.94
Max. PCP acc. [ $m/s^2$ ]	10.41	5.96	6.26	6.66	22.67	12.84	12.48	14.44
Residual Vibration [%]	(100)	67	71	75	(100)	77	81	88

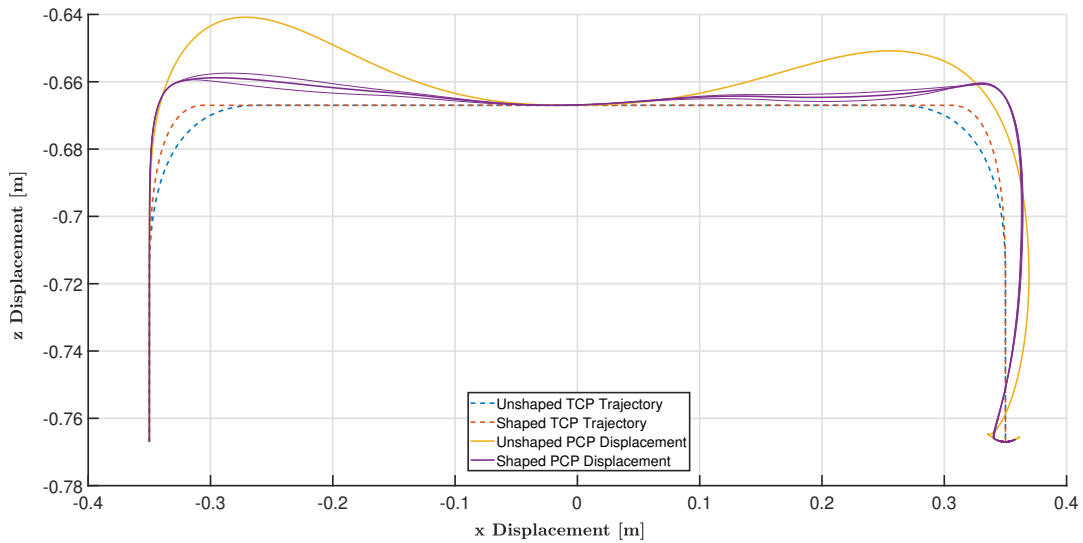


Figure 6.7: The shaped and unshaped trajectory and response of the flexible subset in the  $xz$  plane.

In figure 6.6 and 6.7, the vibration on position level is reduced for both axes and indicated by the accelerations are the forces on the PCP in the  $x$  direction significantly reduced with 38% between the maximum shaped and unshaped accelerations. Evaluated by comparing the unshaped and shaped acceleration peaks. The difference on position level is as shown in Table 6.2, is clear that the minimum theoretical residual vibration (flexible: 18%, stiff: 50%) for both subsets is not reached in the simulation (flexible: 75%, stiff: 88%). This is caused by only applying input shaping to the  $x$  axis instead to both axes (flexible: 18%, stiff: 57%). The oscillating behavior is mainly present in the  $x$  direction, but as shown in Figure 6.6 is the residual vibration mainly caused by the step in the acceleration of the  $z$  axis at 1.14 s. This means that the oscillating behavior in the  $x$  direction is greatly affected by the acceleration in the  $z$  direction. This concludes that the pendulum system is a coupled system, and the oscillating behavior can be actuated by the movement of both axes.

Figure 6.8 shows the trajectory and response with input shaping on both axes presented, and indicates that the residual vibration is effectively reduced. Table 6.3 shows the corresponding parameters related to the named position performance criteria. Besides, the forces on the PCP in the  $x$  direction are reduced with 52% compared to 38% for input shaping applied only to the  $x$  axis.

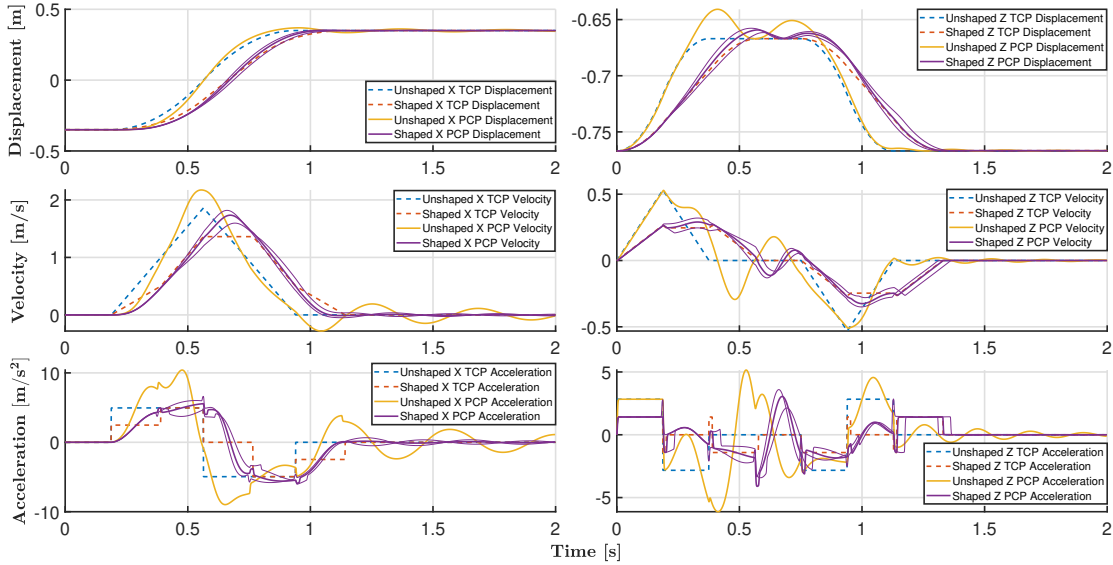


Figure 6.8: The simulated unshaped- (dashed blue line) and double shaped ( $x, z$ ) (dashed orange line) trajectory and the measured unshaped response (yellow line) and double shaped (purple line) of the flexible subset on position, velocity and acceleration level, in the  $x$  and  $z$  direction.

Table 6.3: Simulated performance indicators of the ZV shaper applied on both axes ( $x, z$ ) with the total motion time, maximum acceleration on PCP level, and residual vibration in the  $x$  direction, for the flexible and Stiff subset.

Subset	Flexible				Stiff			
	Unsh.	Shaped ( $x, z$ )			Unsh.	Shaped( $x, z$ )		
<b>Deviated <math>f_n</math> [Hz]</b>	<b>(2.47)</b>	<b>2.11</b>	<b>(2.47)</b>	<b>2.77</b>	<b>(3.59)</b>	<b>2.90</b>	<b>(3.59)</b>	<b>4.80</b>
Motion time [s]	1.41	1.33	1.33	1.33	0.80	0.94	0.94	0.94
Max. PCP acc. [ $m/s^2$ ]	10.41	6.04	5.91	6.62	22.67	11.25	11.06	14.12
<b>Residual Vibration [%]</b>	<b>(100)</b>	<b>11</b>	<b>3</b>	<b>18</b>	<b>(100)</b>	<b>34</b>	<b>32</b>	<b>56</b>

Chapter 8 evaluates the effect of input shaping of the  $x$  axis only on oscillations during placement and the chance of peeling off. Simulation results are compared with measurements and quantified to determine if applying this method to the  $x$  axis only, is sufficient. If this is not the case, input shaping on both axes must be considered.



## 7 Implementation

this chapter describes the implementation of input shaping on the measurement setup and is divided into two parts. First, the interface with the Beckhoff system is discussed. Second the MATLAB program to generate the shaped trajectories is explained.

### 7.1 Beckhoff interface

The existing Spider software programmed in Beckhoff TwinCAT 3.1 is structured as shown in blue presented in Figure 7.1. To implement input shaping in this structure it is chosen to integrate an external setpoint interface, which realizes that an externally generated trajectory could be loaded into the Beckhoff system. This method is chosen because it is less time consuming and complicated than applying input shaping to the trajectory generated by Beckhoff or replacing this generator with the MATLAB code, because the MATLAB code must be translated and implemented to the PLC program. The external setpoint interface is controlled by a high-level function block named "FBP-XzGantryExtSetpoint". The function moves the Spider to the start position of the external setpoint and executes this setpoint by calling the Beckhoff "MC-ExtSetPointGenFeed" function. This feeds the position data to the virtual Cartesian axes of the Spider. The external setpoint is loaded from a CSV file on an external USB drive. The Beckhoff "FBP-ReadCSVFile" function executes the loading process of this file. After executing the two function blocks, the data is transferred to the inverse kinematics, equal to existing software.

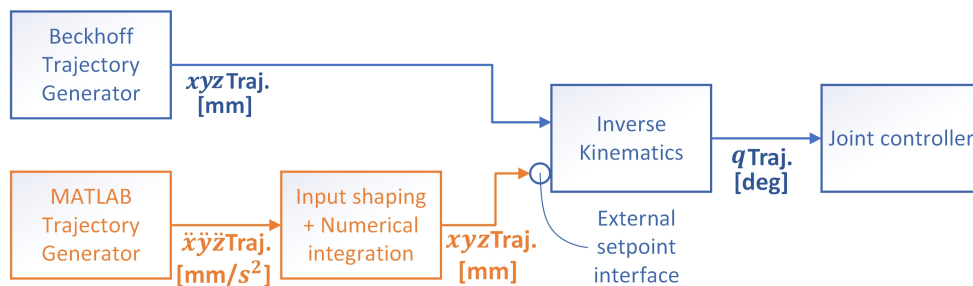


Figure 7.1: Schematic representation of the Beckhoff program and the added external setpoint interface in blue, and the MATLAB shaped trajectory generator in orange

### 7.2 MATLAB external trajectory generator

The MATLAB code generating the trajectory and applying input shaping (marked in orange) is equal to the software as described in Section 6.1. The shaped position trajectory in Cartesian space is in MATLAB exported to a CSV file. The MATLAB code for generating an external setpoint trajectory is provided in Appendix A.

## 8 Experiments and Results

This chapter discusses the experiments executed on the measurement setup discussed. First, the simulation results are validated. The performance of the input shaping using different trajectory accelerations is evaluated based on validated simulation results. Finally, a use case is determined and executed to validate the performance of input shaping in a comparable situation as during production.

### 8.1 Validation of simulation results

The obtained simulation results (Table 6.2) describing the performance of the ZV shaper for the flexible and stiff subset are validated based on measurements. The effect of the input shaping on the  $x$  axis and both axes are measured on the setup. The measurements for the four different situations are repeated 25 times, resulting in 100 measurements in total. This validation test is executed with the same equipment and method as described in Section 4.2. Figure 8.1 shows the comparison between the measured unshaped and  $x$  axis shaped response for the flexible subset. The two responses are plotted three times: the response with the median eigenfrequency (thick lines), the minimum, and the maximum eigenfrequency (thin lines).

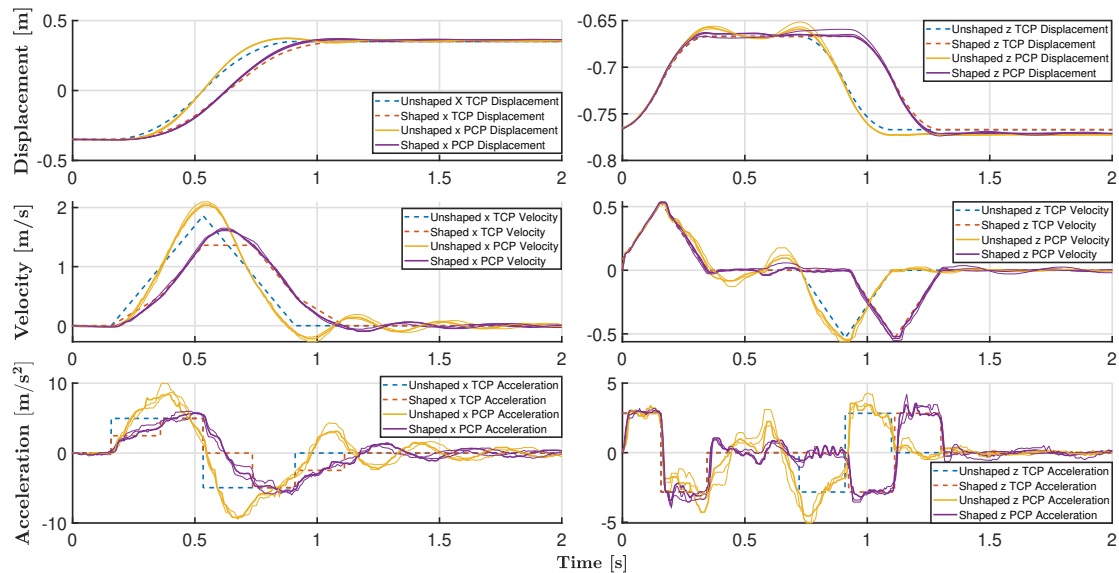


Figure 8.1: The measured unshaped- (dashed blue line) and shaped ( $x$ ) (dashed orange line) trajectory and the measured unshaped response (yellow line) and shaped (purple line) of the flexible subset on position, velocity and acceleration level, in the  $x$  and  $z$  direction.

The measured responses compared to simulated responses with and without input shaping on the  $x$  axis as shown in Figure 6.6 have similar oscillating behavior. The behavior of the maximum deviated responses are also comparable. The measured values show high-frequency behavior, which is visible on the acceleration level. This behavior is not present in the simulation results because the higher eigenmodes and non-linear behavior are not considered in the pendulum model. The measurement results of the double axes-shaped response are shown in Figure 8.2.

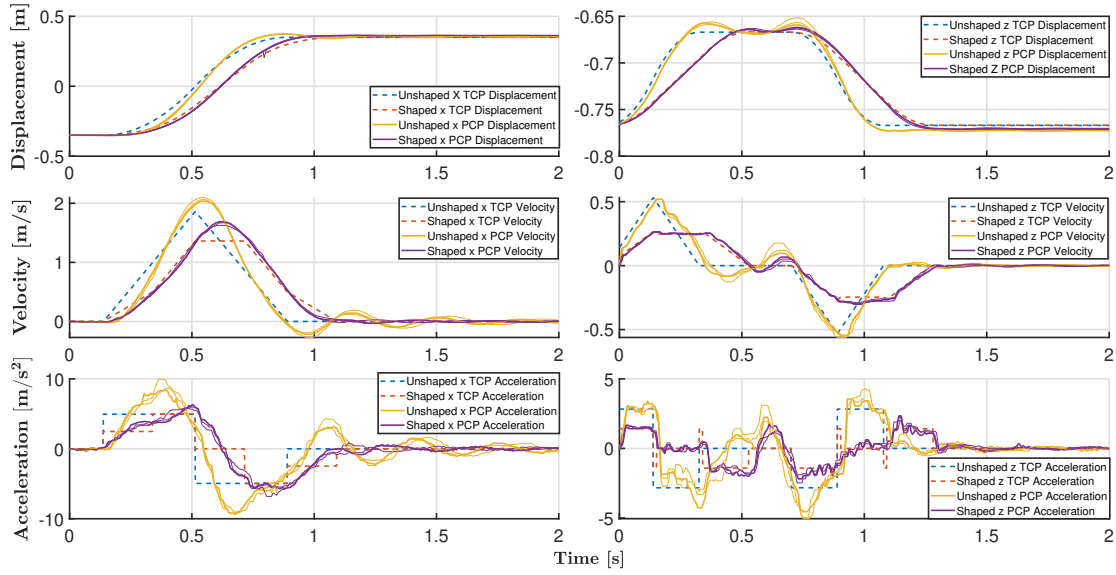


Figure 8.2: The measured unshaped- (dashed blue line) and double shaped ( $x, z$ ) (dashed orange line) trajectory and the measured unshaped response (yellow line) and double shaped (purple line) of the flexible subset on position, velocity and acceleration level, in the  $x$  and  $z$  direction.

The measurement results are again comparable with the simulation for input shaping on both axes, as shown in Figure 6.7. To compare the measured performance of the ZV shaper with the simulations (Figure 6.3), performance indicators are depicted in Table 8.1. These parameters are presented in Table 8.1 for the flexible and stiff subset. The parameters give the median value and the percentage of the deviation between brackets. The outliers are removed from the data by checking if the values are within  $\pm 3\sigma$  from the mean.

Table 8.1: Measured median performance indicators of the ZV shaper applied to the  $x$  axis and to both axes, with the total motion time, maximum acceleration on PCP level, and residual vibration in the  $x$  direction, for the flexible and Stiff subset. With the deviation percentage between brackets.

Subset	Flexible			Stiff		
	Unsh.	Sh. ( $x$ )	Sh. ( $x, z$ )	Unsh.	Sh. ( $x$ )	Sh. ( $x, z$ )
<b>Shaping</b>						
Motion time [s]	1.13	1.33	1.33	0.80	0.94	0.90
<b>Max. PCP acc. [<math>m/s^2</math>]</b>	9.63 (9%)	5.78 (6%)	6.00 (9%)	17.14 (11%)	12.06 (8%)	12.53 (6%)
<b>Residual Vibration [%]</b>	(100)	83	68	(100)	128	83

The motion time and maximum PCP acceleration are comparable with the simulation data. On the other hand, the measured residual vibrations are significantly higher, and the theoretical vibration reduction is not achieved. This is as expected due to the earlier mentioned non-linear behavior and higher eigenmodes present in the product-gripper system. In future research, the reduction of these effects must be investigated. In addition, the residual vibration for the single shaped stiff subset is increased by 28%. For this subset, input shaping on both axes is unavoidable. Finally, based on the comparable shaped and unshaped responses and the similar motion times and maximum PCP accelerations, it can be concluded that the simulation model is valid and applicable for this application. Additionally, it is determined that the ZV shaper can reduce the oscillating behavior of the gripper-product system.

## 8.2 Input shaping performance for varying maximum accelerations

The last section showed that the residual vibrations and the forces could be reduced in practice after applying the ZV shaper with equal accelerations. The next step is to investigate if the maximum acceleration of the shaped trajectory can be increased to compensate for the lost shaper time or even being faster than the unshaped trajectory, in combination with improved vibration reduction and reduced maximum PCP accelerations. This is investigated by measuring the shaped and unshaped response for six different accelerations and both subsets, as presented in Table 8.2. This measurement is executed with input shaping applied to the  $x$  axis. Future research must investigate the effect of input shaping on both axes with varying accelerations to evaluate its potential.

Table 8.2: Maximum accelerations for the flexible and stiff subset to determine the performance of the ZV shaper with varying accelerations.

Subset	Flexible	Stiff
Maximum trajectory acceleration [ $m/s^2$ ]	1.6, 2.5, 5, 8, 10, 11	10, 15, 20, 24.5, 30, 35

The measurements from table 8.2 are only conducted twice due to time limitations. This number must be increased in future measurements to derive more reliable results. Table 8.3 presents the results of the unshaped and shaped responses for the flexible subset, Table 8.4 for the stiff subset. The performance indicators are averaged over the two measurements.

Table 8.3: Measured performance indicators of the ZV shaper applied to the  $x$  axis and to both axes with varying trajectory accelerations, resulting in the total motion time, maximum acceleration on PCP level, and overshoot in the  $x$  direction, for the flexible subset.

Subset	Flexible					
Shaping	Unshaped/Shaped ( $x$ axis)					
Max. traj. acc. [ $m/s^2$ ]	1.6	2.5	5	8	10	11
Motion time [s]	1.98/2.19	1.59/1.79	1.13/1.33	1.60/1.09	1.80/1.18	2.64/0.98
Max. PCP acc. [ $m/s^2$ ]	3.8/2.2	4.6/3.2	9.0/5.5	20.4/8.6	26.7/11.8	31.3/12.2
Overshoot [%]	2.77/0.48	3.07/0.59	4.55/1.09	12.67/4.09	11.86/7.24	14.15/6.20

The indicators show that the performance improvement of input shaping is relatively at the highest for the lower accelerations. For example, the unshaped motion time at  $1.6 m/s^2$  is  $0.16 s$  lower than the shaped trajectory at  $2.5 m/s^2$  and has a reduced overshoot of 2.18%. This gives an improved vibration reduction of 370% with a comparable motion time and maximum PCP acceleration. The margins are smaller for higher accelerations. Evaluating the unshaped response at  $5 m/s^2$  shows a  $0.04 s$  lower motion time than the  $8 m/s^2$  shaped response. Besides, the overshoot and PCP acceleration is reduced with 0.46% and  $0.4 m/s^2$ . This shows an improved vibration reduction of 11% with a similar motion time and PCP acceleration. These two examples prove the possibility of compensating the lost shaper time for specific maximum trajectory accelerations with equal or decreased residual vibrations and maximum PCP accelerations.

Table 8.4: Measured performance indicators of the ZV shaper applied to the  $x$  axis and to both axes with varying trajectory accelerations, resulting in the total motion time, maximum acceleration on PCP level and overshoot in the  $x$  direction for the stiff subset.

Subset Shaping	Stiff					
	Unshaped/Shaped ( $x$ axis)					
Max. traj. acc. [ $m/s^2$ ]	10	15	20	24.5	30	35
Motion time [s]	0.98/0.94	0.66/0.79	0.57/0.70	0.51/0.65	0.46/0.62	0.47/0.60
Max. PCP acc. [ $m/s^2$ ]	16.1/11.4	23.8/16.1	54.9/20.3	42.3/21.5	50.2/23.6	58.7/24.9
Overshoot [%]	4.05/2.17	6.16/4.52	5.55/3.69	6.60/5.46	6.28/6.04	7.53/5.16

The results of the test for the stiff subset show just one case with equal or faster motion times and an equal or reduced overshoot and maximum PCP acceleration. This case is the combination of the trajectory accelerations  $15 m/s^2$  and  $24.5 m/s^2$ . These show an almost equal motion time and maximum PCP acceleration with a 1.41 [%] reduced overshoot, which shows an improvement in vibration reduction of 13 [%]. There is only one acceleration set that has an overall improved performance with input shaping for two reasons. First, the acceleration range is too limited to find a shaped response with a shorter motion time than 0.57 s. Second, the stiff subset has a bigger deviation in eigenfrequency, which limits the vibration reduction of this subset.

Next, the performance of the ZV-shaper related to the peeling off behavior is determined by finding at which maximum PCP accelerations the peeling off behavior appears and analyzing where this value appears with and without input shaping on the  $x$  axis. Based on the measurements presented in Table 8.3 and 8.4 it is determined that the behavior appears at  $20 m/s^2$  for the flexible- and  $50 m/s^2$  for the stiff subset. These values are reached at the unshaped trajectory accelerations of  $8 m/s^2$  and  $30 m/s^2$ . For the shaped response, these values are not reached in both acceleration sets. Additional measurements must determine the possibility of achieving similar motion times with improved vibration reduction and reduced PCP accelerations with input shaping, compared to the unshaped  $8 m/s^2$  and  $30 m/s^2$  trajectory accelerations. Finally, this is investigated for the flexible subset in the next section because this subset shows the effect of input shaping most clearly.

### 8.3 Use case test

The use case test is meant to identify if the ZV shaper can reduce the vibrating behavior, prevent the peeling off behavior, and obtain an equal or shorter motion time. All should be achieved in a comparable production environment with a standard pick and place motion combined with placement inside a box. The use case test is executed in two steps, first to determine how fast the product can be moved with input shaping without peeling off from the gripper. Second, the product is placed inside the box with a maximum overshoot of  $\pm 15\text{mm}$  in the  $x$  direction. This value is based on the findings of BPA. This is  $\pm 2\%$  of the horizontal displacement for the used trajectory. The two tests are schematically presented in Figure 8.3.

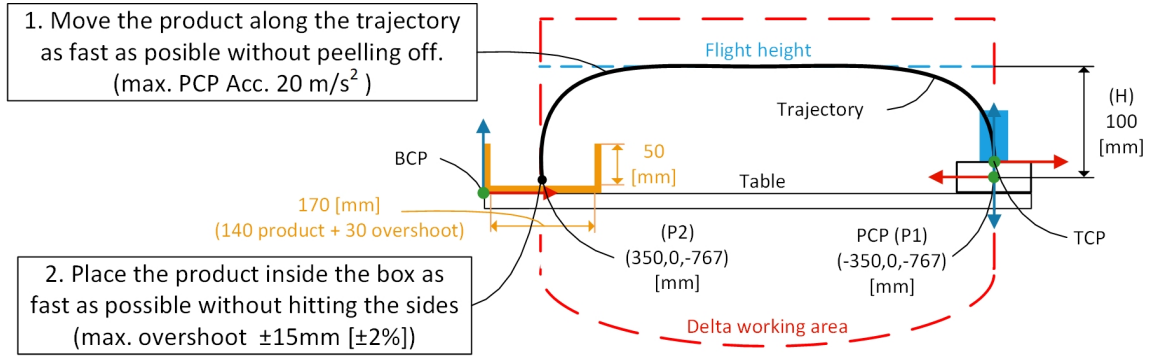


Figure 8.3: Schematic representation of the two use case tests.

First, the maximum allowable PCP acceleration of  $20\text{ m/s}^2$  is found by gradually increasing the trajectory acceleration until this value is reached, with and without input shaping. Second, the product placement test is executed based on a virtual box  $50\text{ mm}$  high and  $170\text{ mm}$  wide (product width  $140\text{ mm} + 30\text{ mm}$  overshoot). The maximum reachable trajectory acceleration is determined in the same manner as the first test. Furthermore, these tests are simulated first to determine the initial trajectory accelerations for the unshaped-, single-shaped- and double-shaped trajectories and to derive an indication of the expected outcome. The simulation results are presented in Table 8.6, the simulations are determined with the maximum  $f_n$  deviation of  $\pm 11.46\%$ .

Table 8.5: Simulation results in the  $x$  direction of the two use case tests, simulated with the unshaped-,  $x$  axis shaped-, and  $x, z$  axes shaped trajectories with the maximum  $f_n$  deviation of  $\pm 11.46\%$

Simulated use case test	1. Peeling off (Max. PCP acc. $20\text{ m/s}^2$ )			2. Overshoot ( $\pm 15\text{mm}$ ( $\pm 2\%$ ))		
	Unsh.	Sh. $x$	Sh. $x, z$	Unsh.	Sh. $x$	Sh. $x, z$
Max. Traj. acc. [ $\text{m/s}^2$ ]	6.5	15.85	16.17	1.14	3.65	7.28
Motion Time [s]	1.05	1.67	0.90	2.36	1.52	1.14
Max. PCP acc. [ $\text{m/s}^2$ ]	19.54	19.65	18.83	2.76	5.04	9.78
Overshoot [%]	6.27	10.82	4.97	1.99	1.98	1.89

The simulation results of the first test show a decrease in motion time for the two shaped situations. For the shaped  $x$  axis, the motion time is reduced with  $0.84\text{ s}$  (55%) and for the shaped  $x, z$  axes with  $1.22\text{ s}$  (107%) with equal overshoot but with the expense of increased maximum PCP accelerations. The second test shows only an improvement in motion time for the double-shaped trajectory of  $0.16\text{ s}$  (19%) with equal maximum PCP accelerations and a decreased overshoot. These findings are compared with the use case test executed on the measurement setup. Table 8.6 shows the obtained measurement results.

Table 8.6: Measurement results in the  $x$  direction of the two use case tests, measured with the unshaped,  $x$  axis shaped, and  $x, z$  axes shaped trajectories

Measured use case test	1. Peeling off (Max. PCP acc. 20 m/s <sup>2</sup> )			2. Overshoot ( $\pm 15$ mm ( $\pm 2\%$ ))		
	Unsh.	Sh. $x$	Sh. $x, z$	Unsh.	Sh. $x$	Sh. $x, z$
<b>Max. Traj. acc. [m/s<sup>2</sup>]</b>	<b>8</b>	<b>18</b>	<b>19</b>	<b>1.45</b>	<b>5.7</b>	<b>9</b>
Motion time [s]	1.57	1.71	1.35	2.09	1.26	1.04
<b>Max. PCP acc. [m/s<sup>2</sup>]</b>	<b>19.56</b>	<b>15.44</b>	<b>16.19</b>	3.25	6.26	9.70
<b>Overshoot [%]</b>	12.59	14.81	8.11	<b>1.65</b>	<b>1.90</b>	<b>1.89</b>

The measurements show an offset in trajectory accelerations that can be reached until the two criteria are violated compared with the simulations. This is probably a result of simulation model inaccuracies. Furthermore, the first test shows similar results compared to the simulations, with an increase in motion time of 0.14 s (9%) for the single-shaped case and a decrease of 0.22 s (16%) for the other case. Besides, the overshoot is slightly increased for the single-shaped case compared to a decreased overshoot for the double-shaped case. The second test shows similar motion time improvements as in the simulations, with a decrease in motion time of 0.83 s (65%) for the shaped  $x$  axis, and 1.05 (101%) s for the double shaped trajectory. These improvements come with the expense of increased PCP accelerations. Evaluating the results of the two use case tests show that the ZV shaper is performing similarly as in the simulations, with faster trajectory accelerations, maximum 16% faster motion time, and 5% less overshoot, for equal PCP accelerations. Besides, the shaper shows a maximum improvement of a 101% faster motion time for equal overshoot, but with the expense of increased PCP accelerations. The use case test indicates that input shaping on both axes shows the highest performance improvement. Before this method can be implemented in a machine for a customer must the associated reduced vertical trajectory height must be resolved.

## 9 Conclusions and Recommendations

The main research question of this project is: How can the control strategy be changed to reduce the oscillating behavior of the mozzarella package attached to the vacuum gripper (gripper-product system) during the pick and place motion executed by the Spider DPR01 delta robot without physically changing the Spider, gripper, or product? This research aims to find an alternative control strategy to prevent the product peeling off from the vacuum gripper or oscillating excessively during the placement inside a box.

Based on the literature research on alternative control strategies, input shaping is the most suitable strategy for this application. Specifically, the ZV Shaper is chosen due to the shortest shaper duration and reasonable theoretical vibration reduction. Next, based on measurements, it is determined that the gripper-product system can be modeled as a second-order pendulum. Besides, two types of grippers are used to investigate the influence of a relatively flexible and stiff gripper on the oscillating behavior. To validate the performance of the ZV shaper in a simulation environment based on the second-order pendulum principle, two Simulink models of the product-gripper system for flexible and stiff grippers are derived. The simulation results show that the theoretical residual vibration (flexible: 18% Stiff 50%) is only achieved if input shaping is applied to the horizontal  $x$  axis and to the vertical  $z$  axis, because the gripper-product system is a coupled system. The downside of applying input shaping to both axes is that the resulting 2D position trajectory is deformed, such that the blending radius is reduced. This results in a decreased vertical movement during placement. The simulation results are validated with measurements and show an decreased residual vibration (flexible: 68% Stiff 83%) with input shaping on both axes. The residual vibration is higher compared to the theoretical residual vibration, which is a result of the non-linear behavior and higher eigenmodes present in the gripper-product system. This behavior is not modeled in the simulation and is impossible to suppress with the ZV shaper.

Furthermore, additional tests are executed to investigate if the lost shaper time (half a vibration period) of the ZV shaper can be canceled by increasing trajectory acceleration without increasing the oscillating behavior. These tests are executed with input shaping only on the  $x$  axis to identify its potential without the drawback of the reduced vertical placement. These tests show that for a specific range (flexible: 1.6 to 10  $m/s^2$ , stiff: 10 to 15  $m/s^2$ ), a decrease in motion time without increasing the overshoot or forces on the product-gripper system (PCP). For higher accelerations, the vibration reduction comes with the expense of higher settling times and increased PCP forces. Besides, the horizontal PCP acceleration is correlated with the peeling off behavior. It indicates that the product starts to peel off at 20  $m/s^2$  for the flexible gripper and at 50  $m/s^2$  for the stiff gripper.

Finally, an use case test is executed to investigate the ZV shaper's performance in a comparable environment as during production. For this test, the flexible gripper is used. The use case is divided into two parts. The first test compares the unshaped, single axis-, and double axes shaped behavior during placement inside a box, with the constrain that the product can not have more horizontal overshoot than 15mm. This test shows a decrease in motion time of at least 65% and 101%, at the expense of higher PCP accelerations. The second test is meant to identify the maximum time decrease until the product starts to peel off. The used constraint for this measurement is that the horizontal PCP acceleration can not be higher than 20  $m/s^2$ . This test results in a 16% decrease in motion time without reaching the maximum PCP acceleration. The single-shaped test shows an increased overshoot and motion time compared with the unshaped response, in contrast with the double-shaped response, which shows a decrease in overshoot and motion time. The use case test showed the potential of input shaping, especially on both axes. Besides, the drawback of vertical height reduction should be taken into account with this result.

Thus, based on the literature results, simulations, and measurements can be concluded that the ZV shaper can reduce the oscillating behavior of the product-gripper system.



The next step is to implement the ZV-shaper on a production machine to identify the exact time gain in practice. Besides, multiple suggestions are given for future research to improve the performance of the ZV shaper. These recommendations are separated in two requirements to realize the implementation on a production machine and three optional research topics, these are listed below.

- develop a path planner that is able to generate shaped trajectories with dynamic end points.
- Determine if the reduced vertical movement during placement can be extended to overcome this limitation.
  
- To give more insight into the performance of the ZV shaper the influence of the following parameters must be determined: extended trajectory acceleration range, different trajectory dimensions, different products(different mozzarella types or soup).
- Investigate if there is a correlation between the physical parameters of the product and the maximum allowable PCP accelerations.
- Design an experimental gripper based on a standard vacuum gripper and a revolute joint at a fixed distance to the PCP to investigate if the performance of the ZV shaper can be improved.

## References

- [1] R. van der Kruk, “A review of the state of the art and outlook of packaging robots for agricultural and food applications,” *Automation and Robotics Magazine*, 2021.
- [2] R. Clavel, “Delta, a fast robot with parallel geometry,” *Proceedings of 18th International Symposium on Industrial Robot*, vol. 18th, pp. 91–100, 1988.
- [3] Beckhoff, “Ax5000 | servo drive function description | en,”
- [4] N. van de Wouw, *Multibody Dynamics*. Technische Universiteit Eindhoven, 2016, pp. 1–214.
- [5] K. J. Keesman, *System identification: an introduction*. Springer Science & Business Media, 2011.
- [6] G. J. Wiens and D. S. Hardage, “Structural dynamics and system identification of parallel kinematic machines,” in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, vol. 42568, 2006, pp. 749–758.
- [7] G. P. Villibor, F. L. Santos, D. M. de Queiroz, J. K. K. Júnior, and F. d. A. de Carvalho Pinto, “Determination of modal properties of the coffee fruit-stem system using high speed digital video and digital image processing,” *Acta Scientiarum. Technology*, vol. 38, no. 1, pp. 41–48, 2016.
- [8] M. N. Helfrick, C. Niezrecki, P. Avitabile, and T. Schmidt, “3d digital image correlation methods for full-field vibration measurement,” *Mechanical systems and signal processing*, vol. 25, no. 3, pp. 917–927, 2011.
- [9] E. Olson, “Apriltag: A robust and flexible visual fiducial system,” in *2011 IEEE International Conference on Robotics and Automation*, IEEE, 2011, pp. 3400–3407.
- [10] M. Casiano, “Extracting damping ratio from dynamic data and numerical solutions,” 2016.
- [11] A. van Noorden, “Phase 1: Graduation project preparation phase report: Delta robot motion stabilization for flexible packaging of viscous fluid products in a bag,” pp. 1–14, 2021.
- [12] T. Harčarik, J. Bocko, and K. Masláková, “Frequency analysis of acoustic signal using the fast fourier transformation in matlab,” *Procedia Engineering*, vol. 48, pp. 199–204, 2012.
- [13] N. Singer, W. Singhose, and W. Seering, “Comparison of filtering methods for reducing residual vibration,” *European Journal of Control*, vol. 5, no. 2-4, pp. 208–218, 1999.
- [14] W. Singhose and J. Vaughan, “Reducing vibration by digital filtering and input shaping,” *IEEE Transactions on Control Systems Technology*, vol. 19, no. 6, pp. 1410–1420, 2010.
- [15] R. Béarée, “New damped-jerk trajectory for vibration reduction,” *Control Engineering Practice*, vol. 28, pp. 112–120, 2014.
- [16] W. Singhose, “Command shaping for flexible systems: A review of the first 50 years,” *International journal of precision engineering and manufacturing*, vol. 10, no. 4, pp. 153–168, 2009.

## Appendix A

### Trajectory generator main program

```

1 %Simulation parameters
2 clc; clear; close all;
3 Folder = 'D:\Cloud storage\OneDrive - TU Eindhoven\TUE Mechanical
   Engineering\2020-2021\Q2-3-4\4ZM00 - Graduation project\MATLAB\';
4 TestFolder = 'D:\Cloud Storage\OneDrive - TU Eindhoven\TUE Mechanical
   Engineering\2020-2021\Q2-3-4\4ZM00 - Graduation project\Delta Tests\
   ';
5
6 % add library and Initialisation path
7 addpath('D:\Cloud storage\OneDrive - TU Eindhoven\TUE Mechanical
   Engineering\2020-2021\Q2-3-4\4ZM00 - Graduation project\MATLAB\
   FUNCTIONS');
8 addpath('D:\Cloud storage\OneDrive - TU Eindhoven\TUE Mechanical
   Engineering\2020-2021\Q2-3-4\4ZM00 - Graduation project\MATLAB\
   INITIALISATION');
9
10 % Set up Run simulation Parameters
11 %select inputshaper
12 SubSet = 5; % select subset 3 (B140-5 -500mBar) or 5 (BX75P -375mBar)
13 if SubSet == 3
14     MeasurementFolder = '20211004 - 7. Usecase Test\Trajectories\';
15 elseif SubSet == 5
16     MeasurementFolder = '20211004 - 7. Usecase Test\Trajectories\';
17 end
18
19 Omega = [0 0 2.4727 0 3.5906 0];
20 Zeta = [0 0 3.5146e-04 0 0.0012 0];
21 InputShaper.Type = "ZV Shaper"; % choose between "None", "Notch Filter"
   and "ZV Shaper"
22 InputShaper.Omega = Omega(SubSet); % choose shaper frequency in Hz.
   Thrid: 2.4727 Fifth: 3.5906
23 InputShaper.Zeta = Zeta(SubSet); % choose shaper relative damping. Thrid
   : 3.5146e-04 Fifth: 0.0012
24 SampleTime = 1.0000e-03;
25 RunTime = 4;
26 ExtendToRuntime = true;
27 ShapedAxes = [1 2]; % specify the input shaped axes x = 1, y = 2, z = 3
28
29 % select Trajectory
30 TrajectoryParameters.Amax = 15; % Maximal acceleration in m/s^2
31 TrajectoryParameters.Vmax = 10; % Maximal velocity in m/s
32 TrajectoryParameters.PickPoint = [-0.350 0 -0.767]; % Pick location [x y
   z] in m origin at [0 0 -767]?
33 TrajectoryParameters.PlacePoint = [0.350 0 -0.767]; % place location [x
   y z] in m origin at [0 0 -767]?
34 TrajectoryParameters.PickHeight = 0.050; % the approach heighth where de
   delta follows a strictly vertical trajectory in m
35 TrajectoryParameters.PlaceHeight = 0.050; % the approach heighth where de
   delta follows a strictly vertical trajectory in m
36

```

```

37 % Prepare the simulation parameters and run the simulation
38 [SimulationParameters] = PrepareSimulation(InputShaper ,
      TrajectoryParameters , SampleTime , RunTime , ExtendToRuntime ,
      ShapedAxes , SubSet);
39
40 if InputShaper.Type == "ZV Shaper"
41     % Data acquisition % Select the correct Path! Shaped or World
42     Data = [(0:size(SimulationParameters.Trajectory.Shaped.Pos,1)-1)'
      SimulationParameters.Trajectory.Shaped.Pos];
43     if size(ShapedAxes,2) == 2
44         writematrix(Data,append(TestFolder,MeasurementFolder,'
      CSV_ExtSetDelta_',num2str(TrajectoryParameters.Amax+1),'.csv
      '), 'Delimiter',';');
45     else
46         writematrix(Data,append(TestFolder,MeasurementFolder,'
      CSV_ExtSetDelta_',num2str(TrajectoryParameters.Amax+2),'.csv
      '), 'Delimiter',';');
47     end
48 elseif InputShaper.Type == "None"
49     % Data acquisition % Select the correct Path! Shaped or World
50     Data = [(0:size(SimulationParameters.Trajectory.World.Pos,1)-1)'
      SimulationParameters.Trajectory.World.Pos];
51     writematrix(Data,append(TestFolder,MeasurementFolder,'
      CSV_ExtSetDelta_',num2str(TrajectoryParameters.Amax),'.csv'), '
      Delimiter',';');
52 end

```

### Trajectory generator main function

```

1 function [SimulationParameters] = PrepareSimulation(InputShaper ,
      TrajectoryParameters , SampleTime , RunTime , ExtendToRuntime ,
      ShapedAxes , SubSet)
2
3 % 0. Run Initialization file
4 InitialisationParameters = Initialisation(SampleTime, RunTime, SubSet);
5 SimulationParameters.Controller = InitialisationParameters.Controller;
6 SimulationParameters.Pendulum = InitialisationParameters.Pendulum;
7 SimulationParameters.InverseKinematics = InitialisationParameters.
      InverseKinematics;
8
9 %1. create Trajectory
10 [SimulationParameters.Trajectory.World.Time, SimulationParameters.
      Trajectory.World.Pos, SimulationParameters.Trajectory.World.Vel,...
11     SimulationParameters.Trajectory.World.Acc] =
      Trajectory_BSpline_Function(TrajectoryParameters.Amax,
      TrajectoryParameters.Vmax, ...
12     TrajectoryParameters.PickPoint, TrajectoryParameters.PlacePoint ,
      TrajectoryParameters.PickHeight, TrajectoryParameters.
      PlaceHeight, SampleTime);
13
14 %2. select shaper
15 SimulationParameters.InputShaper = InputShaper;
16 if InputShaper.Type == "None"
17     SimulationParameters.InputShaper.Selected = 0;

```

```

18     SimulationParameters.InputShaper.t = [0 0];
19     SimulationParameters.InputShaper.A = [1 0];
20     elseif InputShaper.Type == "Notch Filter"
21         SimulationParameters.InputShaper.Selected = 1;
22         SimulationParameters.InputShaper.t = [0 0];
23         SimulationParameters.InputShaper.A = [1 0];
24     elseif InputShaper.Type == "ZV Shaper"
25         SimulationParameters.InputShaper.Selected = 2;
26         [A, t] = FindInputShaperParameters(InputShaper, SampleTime);
27         SimulationParameters.InputShaper.t = t;
28         SimulationParameters.InputShaper.A = A;
29     end
30
31     % 3. Apply input shaper on acceleration trajectory
32     [SimulationParameters.Trajectory.Shaped.Acc, SimulationParameters.
33     Trajectory.Shaped.Time] = ApplyInputShaping...
34     (SimulationParameters.Trajectory.World.Acc, SampleTime, RunTime,
35     SimulationParameters.InputShaper.A, SimulationParameters.
36     InputShaper.t, ShapedAxes, ExtendToRuntime);
37
38     % 4. Integrate Acceleration Profile with the 'Forward Euler' and '
39     Trapezoidal' method
40     SimulationParameters.Trajectory.Shaped.Vel = DiscreteIntegration('
41     Forward Euler', SimulationParameters.Trajectory.Shaped.Acc, [0;0;0],
42     1, SampleTime);
43     SimulationParameters.Trajectory.Shaped.Pos = DiscreteIntegration('
44     Trapezoidal', SimulationParameters.Trajectory.Shaped.Vel, ...
45     SimulationParameters.Trajectory.World.Pos(1,:), 1, SampleTime);
46
47     if ExtendToRuntime == false
48     % RunTime = SimulationParameters.Trajectory.World.Time(end)+
49     SimulationParameters.InputShaper.t(2)+SampleTime*2;
50     RunTime = SimulationParameters.Trajectory.Shaped.Time(end)+SampleTime
51     ;
52     end
53
54     % 5. extend trajectory based on the runtime
55     ExendedTimeWorld = (SimulationParameters.Trajectory.World.Time(end)+
56     SampleTime:SampleTime:RunTime)';
57     ExendedTimeShaped = (SimulationParameters.Trajectory.Shaped.Time(end)+
58     SampleTime:SampleTime:RunTime)';
59     SimulationParameters.Trajectory.World.Time = [SimulationParameters.
60     Trajectory.World.Time; ExendedTimeWorld];
61     SimulationParameters.Trajectory.Shaped.Time = [SimulationParameters.
62     Trajectory.Shaped.Time; ExendedTimeShaped];
63     for i = 1:3
64     PosWorld(:, i) = [SimulationParameters.Trajectory.World.Pos(:, i);
65     SimulationParameters.Trajectory.World.Pos(end, i)*ones(size(
66     ExendedTimeWorld,1),1)];
67     VelWorld(:, i) = [SimulationParameters.Trajectory.World.Vel(:, i); zeros(
68     size(ExendedTimeWorld,1),1)];
69     AccWorld(:, i) = [SimulationParameters.Trajectory.World.Acc(:, i); zeros(
70     size(ExendedTimeWorld,1),1)];
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99

```

```

55 PosShaped(:, i) = [SimulationParameters.Trajectory.Shaped.Pos(:, i);
    SimulationParameters.Trajectory.Shaped.Pos(end, i)*ones(size(
    ExendedTimeShaped,1),1)];
56 VelShaped(:, i) = [SimulationParameters.Trajectory.Shaped.Vel(:, i); zeros
    (size(ExendedTimeShaped,1),1)];
57 AccShaped(:, i) = [SimulationParameters.Trajectory.Shaped.Acc(:, i); zeros
    (size(ExendedTimeShaped,1),1)];
58 end
59
60 SimulationParameters.Trajectory.World.Pos = PosWorld;
61 SimulationParameters.Trajectory.World.Vel = VelWorld;
62 SimulationParameters.Trajectory.World.Acc = AccWorld;
63 SimulationParameters.Trajectory.Shaped.Pos = PosShaped;
64 SimulationParameters.Trajectory.Shaped.Vel = VelShaped;
65 SimulationParameters.Trajectory.Shaped.Acc = AccShaped;
66
67 % 6. Transform Trajectory to joint coordinates with Inverse Kinematics
68 SimulationParameters.Trajectory.Joint.Pos = InverseKinematics(
    SimulationParameters.Trajectory.Shaped.Pos, SimulationParameters.
    InverseKinematics);
69 SimulationParameters.Trajectory.Joint.Time = SimulationParameters.
    Trajectory.Shaped.Time;
70
71 % 7. Apply cubic interpolation at 8KHz and differentiate twice to obtain
    the vel and acc joint trajectories.
72 [SimulationParameters.Trajectory.Interpolated.Time,
    SimulationParameters.Trajectory.Interpolated.Pos] =
    CubicInterpolation...
73 (SimulationParameters.Trajectory.Joint.Pos, SimulationParameters.
    Controller, RunTime, SampleTime);
74 [Velocity_time, Velocity, Acceleration_time, Acceleration] =
    DeriveDerivatives(SimulationParameters.Trajectory.Interpolated.Time
    , SimulationParameters.Trajectory.Interpolated.Pos);
75 SimulationParameters.Trajectory.Interpolated.Time = Acceleration_time;
76 SimulationParameters.Trajectory.Interpolated.Pos = SimulationParameters
    .Trajectory.Interpolated.Pos((1:end-2),:);
77 SimulationParameters.Trajectory.Interpolated.Vel = Velocity((1:end-1)
    ,:);
78 SimulationParameters.Trajectory.Interpolated.Acc = Acceleration;
79
80 % Write variables to temporarily simulink format.
81 SimulationParameters.Trajectory.Interpolated.time = SimulationParameters
    .Trajectory.Interpolated.Time;
82 SimulationParameters.Trajectory.Interpolated.signals.values(:,(1:3)) =
    SimulationParameters.Trajectory.Interpolated.Pos;
83 SimulationParameters.Trajectory.Interpolated.signals.values(:,(4:6)) =
    SimulationParameters.Trajectory.Interpolated.Vel;
84 SimulationParameters.Trajectory.Interpolated.signals.values(:,(7:9)) =
    SimulationParameters.Trajectory.Interpolated.Acc;
85 SimulationParameters.Trajectory.World.time = SimulationParameters.
    Trajectory.World.Time;
86 SimulationParameters.Trajectory.World.signals.values(:,(1:3)) =
    SimulationParameters.Trajectory.World.Pos;
87 SimulationParameters.Trajectory.World.signals.values(:,(4:6)) =

```

```

SimulationParameters.Trajectory.World.Vel;
88 SimulationParameters.Trajectory.World.signals.values(:,(7:9)) =
SimulationParameters.Trajectory.World.Acc;
89 SimulationParameters.Trajectory.Shaped.time = SimulationParameters.
Trajectory.Shaped.Time;
90 SimulationParameters.Trajectory.Shaped.signals.values(:,(1:3)) =
SimulationParameters.Trajectory.Shaped.Pos;
91 SimulationParameters.Trajectory.Shaped.signals.values(:,(4:6)) =
SimulationParameters.Trajectory.Shaped.Vel;
92 SimulationParameters.Trajectory.Shaped.signals.values(:,(7:9)) =
SimulationParameters.Trajectory.Shaped.Acc;
93 SimulationParameters.Trajectory.Joint.time = SimulationParameters.
Trajectory.Joint.Time;
94 SimulationParameters.Trajectory.Joint.signals.values =
SimulationParameters.Trajectory.Joint.Pos;

```

### Initialization function

```

1 function InitialisationParameters = Initialisation(SampleTime, RunTime,
SubSet)
2
3 % Select Pendulum Configuration
4 if SubSet == 3
5 InitialisationParameters.Pendulum.Mass = 225;
6 InitialisationParameters.Pendulum.Damping = 2.00e-05;
7 InitialisationParameters.Pendulum.Stiffness = 1.00e-07;
8 InitialisationParameters.Pendulum.Length = 45.85;
9 elseif SubSet == 5
10 InitialisationParameters.Pendulum.Mass = 225;
11 InitialisationParameters.Pendulum.Damping = 2.40e-05;
12 InitialisationParameters.Pendulum.Stiffness = 2.00e-05;
13 InitialisationParameters.Pendulum.Length = 22.95;
14 end
15
16 InitialisationParameters.Pendulum.SampleTime = SampleTime;
17 InitialisationParameters.Pendulum.RunTime = RunTime;
18
19 % Set Ik Parameters
20 InitialisationParameters.InverseKinematics.r_Base = 200;
21 InitialisationParameters.InverseKinematics.r_End = 45; %45 small
trevaling platform 53 mm large platform
22 InitialisationParameters.InverseKinematics.l_Up = 348.087991749213;
23 InitialisationParameters.InverseKinematics.l_Low = 800;
24
25 % Set Controller Parameters
26 InitialisationParameters.Controller.InterpolationFrequency = 8000;
27 InitialisationParameters.Controller.Vlim = 51300; % Velocity limiter
value (deg/s) (8550 rpm)
28 InitialisationParameters.Controller.Tlim = 11.66*2; % 200% of the peak
motor torque limiter value (Nm) [AM8032-0E21-0000]
29 InitialisationParameters.Controller.Kfa = 0; %Acceleration feedforward
deg/s^2 -> Nm (Nm*s^2/deg)
30 InitialisationParameters.Controller.Kfv = 1; %Velocity feedforward deg/s
-> deg/s (1)

```

```

31 InitialisationParameters.Controller.Kp = 1100; %Position feedback deg ->
    deg/s (1/s)
32 InitialisationParameters.Controller.Kv = 0.25; %Velocity feedback
    Proportion step rad/s -> Nm ((Nm*s)/rad )
33 InitialisationParameters.Controller.Iv = 0.0067; %Velocity feedback
    Tntegration step rad/s -> Nm (s)

```

### Apply input shaping function

```

1 function [ShapedTrajectory ,Time] = ApplyInputShaping(Trajectory ,
    SampleTime, RunTime, A, t, ShapedAxes, ExtendToRuntime)
2 Axes = [1 2 3];
3
4 for i = ShapedAxes
5     S1 = Trajectory(:,i)*A(1);
6     S1 = [S1; zeros(round(t(2)*(1/SampleTime),0),1) ];
7     S2 = Trajectory(:,i)*A(2);
8     S2 = [zeros(round(t(2)*(1/SampleTime),0),1); S2];
9     S(:,i) = S1 + S2;
10 end
11
12 if isempty(setdiff(Axes, ShapedAxes)) == 0
13     for i = setdiff(Axes, ShapedAxes)
14         ExtendedSignal = zeros(size(S,1) - size(Trajectory,1),1) +
            Trajectory(round(size(Trajectory,1)/2),i);
15         S(:,i) = [Trajectory(1:(round(size(Trajectory,1)/2)),i);
            ExtendedSignal; Trajectory((round(size(Trajectory,1)/2)+1:
                end),i)];
16     end
17 end
18     ShapedTrajectory = S((1:end),:);
19     Time = (0:SampleTime:(size(ShapedTrajectory,1)-1)*SampleTime)';
20
21 end

```

### Discrete Integration function

```

1 function TrajectoryOutput = DiscreteIntegration(Method, TrajectoryInput ,
    InitialPos, Gain, SampleTime)
2
3 if Method == "Forward Euler"
4
5     for j = 1:3
6         for i = 1:size(TrajectoryInput,1)
7             if i == 1
8                 x(i,j) = InitialPos(i);
9                 TrajectoryOutput(i,j) = InitialPos(j);
10                x(i+1,j) = TrajectoryOutput(i,j) + Gain * SampleTime *
                    TrajectoryInput(i,j);
11            else
12                TrajectoryOutput(i,j) = x(i,j);
13                x(i+1,j) = TrajectoryOutput(i,j) + Gain * SampleTime *
                    TrajectoryInput(i,j);
14            end
15        end
16    end

```



```

16     end
17
18     elseif Method == "Trapezoidal"
19
20         for j = 1:3
21             for i = 1:size(TrajectoryInput,1)
22                 if i == 1
23                     x(i,j) = InitialPos(i);
24                     TrajectoryOutput(i,j) = InitialPos(j);
25                     x(i+1,j) = TrajectoryOutput(i,j) + Gain * (SampleTime /
26                         2) * TrajectoryInput(i,j);
27                 else
28                     TrajectoryOutput(i,j) = x(i,j) + Gain * (SampleTime / 2)
29                         * TrajectoryInput(i,j);
30                     x(i+1,j) = TrajectoryOutput(i,j) + Gain * (SampleTime /
31                         2) * TrajectoryInput(i,j);
32                 end
33             end
34         end
35     end
36 end

```

### Inverse Kinematics function

```

1 function q_Joint = InverseKinematics(p_World, IK_Parameters)
2
3 r_Base = IK_Parameters.r_Base;
4 r_End = IK_Parameters.r_End;
5 l_Up = IK_Parameters.l_Up;
6 l_Low = IK_Parameters.l_Low;
7
8 for i = 1:size(p_World,1)
9
10 x = p_World(i,1);
11 y = p_World(i,2);
12 z = p_World(i,3);
13
14 x_func = x;
15 y_func = y;
16 z_func = z;
17
18 q_Joint(i,1) = INV_KIN_TH(x_func, y_func, z_func, r_Base, r_End, l_Up, l_Low);
19 %.....
20 x_func = x*cos(2*pi/3) + y*sin(2*pi/3);
21 y_func = y*cos(2*pi/3) - x*sin(2*pi/3);
22 z_func = z;
23 q_Joint(i,2) = INV_KIN_TH(x_func, y_func, z_func, r_Base, r_End, l_Up, l_Low);
24 %.....
25 x_func = x*cos(4*pi/3) + y*sin(4*pi/3);
26 y_func = y*cos(4*pi/3) - x*sin(4*pi/3);
27 z_func = z;
28 q_Joint(i,3) = INV_KIN_TH(x_func, y_func, z_func, r_Base, r_End, l_Up, l_Low);
29 end
30 end

```

```

31
32 function [th_func] = INV_KIN_TH(x_func, y_func, z_func, r_Base, r_End, l_Up,
    l_Low)
33 a_abc = x_func^2+y_func^2+z_func^2+(r_Base-r_End)^2+l_Up^2-l_Low^2-2*
    l_Up*(y_func+(r_Base-r_End))+2*y_func*(r_Base-r_End);
34 b_abc = 4*z_func*l_Up;
35 c_abc = x_func^2+y_func^2+z_func^2+(r_Base-r_End)^2+l_Up^2-l_Low^2+2*
    l_Up*(y_func+(r_Base-r_End))+2*y_func*(r_Base-r_End);
36
37 t = (-b_abc-sqrt(b_abc^2-4*a_abc*c_abc))/(2*a_abc);
38 th_func = rad2deg(2*atan(t));
39 end

```

### Derive derivatives function

```

1 function [Velocity_time, Velocity, Acceleration_time, Acceleration] =
    DeriveDerivatives(Time, Position)
2 %using symmetric difference quotient method to derive the velocity and
    acceleration profile
3 for j = 1:size(Position,2)
4
5 for i = 2:size(Position,1)-1
6 Velocity(i,j) = (Position(i+1,j) - Position(i-1,j))/ (Time(i+1)-Time(i
    -1));
7 end
8 Velocity_time = Time(1:end-1);
9
10 for i = 2:size(Velocity,1)-1
11 Acceleration(i,j) = (Velocity(i+1,j) - Velocity(i-1,j))/ (Velocity_time
    (i+1)-Velocity_time(i-1));
12 end
13 Acceleration_time = Velocity_time(1:end-1);
14 end

```

### Cubic interpolation function

```

1 function [xq, q_Interpolated] = CubicInterpolation(q_Joint,
    InterpolationParameters, Runtime, SampleTime)
2
3 % capture Interpolation parameters
4 q1 = q_Joint(:,1);
5 q2 = q_Joint(:,2);
6 q3 = q_Joint(:,3);
7
8 InterpolationFrequency = InterpolationParameters.InterpolationFrequency;
9
10 % generatate timevectors for 1KHz and 8KHz
11 Time = (0:SampleTime:Runtime)';
12 xq = (0:(1/InterpolationFrequency):Runtime(end))';
13
14 % execute cubic interpolation
15 q_Interpolated(:,1) = spline(Time, q1, xq);
16 q_Interpolated(:,2) = spline(Time, q2, xq);
17 q_Interpolated(:,3) = spline(Time, q3, xq);

```