

Department of Precision and Microsystems Engineering

POI Position and Force Control in Industrial Wire Bonders

Rahul Puthukattil

Report no : 2023.077.
Coach : dr.ir. Dragan Kostić (ASMPT), ir. Luke van Eijk (ASMPT)
Professor : dr.ir. S. Hassan HosseinNia
Specialisation : Mechatronic System Design
Type of report : MSc Thesis
Date : 30 August 2023

MSc Thesis in Mechanical Engineering

**POI Position and Force Control in
Industrial Wire Bonders**

Rahul Puthukattil

August 2023

A thesis submitted to the Delft University of Technology in
partial fulfillment of the requirements for the degree of Master
of Science in Mechanical Engineering

Rahul Puthukattil: *POI Position and Force Control in Industrial Wire Bonders* (2023)

The work in this thesis was carried out in the:



MSD Group
Delft University of Technology

Center of Competency
ASMPT

Supervisors: dr.ir. S. Hassan HosseinNia (TU Delft)
dr.ir. Dragan Kostić (ASMPT)
ir. Luke van Eijk (ASMPT)

Abstract

This thesis investigates the application of force control techniques to enhance industrial wire bonding processes. It entails the design of a simulation model for the Z-axis interaction between bondhead and environment, followed by the implementation of a Parallel Force Control architecture in Simulink. The study evaluates different force and position control methods, selecting those suitable for single-degree-of-freedom wire bonding. Simulation model validation through system identification and inclusion of an impact transition phase ensures accurate bondhead-substrate interaction representation. Results show that the implemented controllers effectively eliminate force overshoot, mitigating substrate damage risks, even with suboptimal controllers. This thesis contributes to the understanding of force control's potential in industrial wire bonding processes. The developed simulation model serves as a platform for evaluating controller effectiveness, optimizing process outcomes, and aiding future research and experimentation in this domain.

Contents

1. Introduction	1
1.1. Motivation	1
1.1.1. Industrial Wire Bonders	1
1.1.2. Force Control	2
1.1.3. Ball Bonding	2
1.1.4. Ideal contact force output profile	3
1.1.5. Necessity of Force Control	4
1.1.6. ASMPT	4
1.2. Research Direction and Objectives	5
1.3. Thesis Overview	5
2. Literature Review	7
2.1. Classification of Force Control Methods	7
2.2. Direct Control Algorithms - Hybrid Force/Position Control	7
2.2.1. Proportional Force Control	7
2.2.2. Proportional Derivative Force Control	8
2.2.3. Proportional Integral Force Control	9
2.3. Indirect Force Control Algorithms - Impedance Control	9
2.4. Advanced Force Control Methods	11
2.4.1. Parallel Force Control	11
2.4.2. Hybrid Impedance Control and Adaptive Hybrid Impedance Control	12
2.5. Comparison of Force Control Methods	13
2.5.1. Important Parameters for Comparison	14
2.5.2. Grading Table for Comparison	15
2.5.3. Conclusion from grading	15
2.6. Contact Force Measurement	16
3. Modeling the Wirebonding Process in Simscape Multibody	19
3.1. Single Mass Contact Model	19
3.1.1. Model Parameter Selection	20
3.2. Single Mass Impact Transition Model	20
3.2.1. Free body diagram of impact transition model	21
3.2.2. Transfer functions before and after contact	21
3.2.3. Impact transition model representation in Simscape	22
4. Control Architecture Implementation in Simulink	25
4.1. Parallel force control - Schematic	25
4.2. Contact Detection	26
4.3. Reference Position Setpoint	27
4.3.1. Setpoint Flowchart	27
4.3.2. Implementation in Simulink	28
4.3.3. Generation of Position Setpoint	28

4.4. Reference Force Setpoint	29
4.4.1. Setpoint Flowchart	29
4.4.2. Implementation	29
4.4.3. Generation of Force Setpoint	31
4.5. Additional Parameters	31
4.5.1. Delays present in the system	31
4.5.2. Measurement Noise	32
4.6. Parallel Force Control Implementation in Simulink	32
4.7. Controller Design	33
4.7.1. Position Controller Design	33
4.7.2. Force Controller Design	35
5. Results And Discussion	37
5.1. Variables used for simulation	37
5.1.1. Search Velocity	37
5.1.2. Contact Detection Time	37
5.2. Simulation Results and Discussions	38
6. Conclusions and Future Recommendations	43
6.1. Conclusions	43
6.2. Future Recommendations	44
6.2.1. Modeling Refinement	44
6.2.2. Exploring Contact Detection Algorithms	44
6.2.3. Model Discretization	44
6.2.4. Experimental FRF Data	44
A. Appendix A - Single Mass Contact Model implementation and validation	45
A.0.1. Single mass contact model representation in Simscape	45
A.1. System Identification	46
A.1.1. Transfer functions for single mass contact model	46
A.1.2. Simulink and Matlab Setup for System Identification	47
A.1.3. System Identification Output	48
B. Appendix B - MATLAB Code - Position Setpoint Generation	51
C. Appendix C - Contact Time Detection	53

List of Figures

1.1. Z axis stage of an industrial wire bonder.	1
1.2. Steps involved in wire bonding process.	3
1.3. Flat top force profile.	4
2.1. Impedance control - Basic ideation.	10
2.2. FAB - Environment contact model.	12
2.3. HIC - Impedance modeling.	13
3.1. Single mass environment contact model.	19
3.2. Impact transition model.	21
3.3. Impact transition model - Simscape implementation.	23
4.1. Parallel force control.	25
4.2. Contact detection implementation.	26
4.3. Position setpoint flowchart.	27
4.4. Position setpoint implementation.	28
4.5. Generated position setpoint.	28
4.6. Force setpoint flowchart.	29
4.7. Force setpoint implementation.	30
4.8. Initial and intermediate force setpoints.	30
4.9. Generated force setpoint.	31
4.10. Single mass impact transition plant with sensor delays.	32
4.11. Parallel force control implementation in Simulink.	33
4.12. Position Frequency Response Function (FRF).	34
4.13. C_p - Bandwidth 100 Hz.	34
4.14. Position open loop.	35
4.15. Force FRF.	35
4.16. C_f - Bandwidth 300 Hz.	35
4.17. Force open loop.	36
5.1. $V_{search} = 5 \text{ mm s}^{-1}$ and Contact Detection time = 1.25 ms	38
5.2. $V_{search} = 8 \text{ mm s}^{-1}$ and Contact Detection time = 1.25 ms	38
5.3. $V_{search} = 10 \text{ mm s}^{-1}$ and Contact Detection time = 1.25 ms	39
5.4. $V_{search} = 12 \text{ mm s}^{-1}$ and Contact Detection time = 1.25 ms	39
5.5. $V_{search} = 15 \text{ mm s}^{-1}$ and Contact Detection time = 1.25 ms	39
5.6. $V_{search} = 10 \text{ mm s}^{-1}$ and Contact Detection time = 0 ms	40
5.7. $V_{search} = 10 \text{ mm s}^{-1}$ and Contact Detection time = 1.8 ms	40
5.8. $V_{search} = 10 \text{ mm s}^{-1}$ and Contact Detection time = 2.5 ms	41
A.1. Single mass contact model in Simscape.	45
A.2. FRF F_r to Z	47
A.3. FRF F_r to F_m	47

List of Figures

A.4. System identification schematic diagram.	47
A.5. Simulink implementation of system identification.	48
A.6. FRF F_r to Z - Analytical and System identification output.	49
A.7. FRF F_r to F_m - Analytical and System identification output.	50
C.1. Local variable - Contact time detection.	53
C.2. Enabled subsystem - Contact time detection.	53

List of Tables

2.1. Force control algorithms comparison.	15
---	----

Acronyms

DOF	Degree of Freedom	1
FAB	Free Air Ball	2
EFO	Electronic Flame Off	2
PD	Proportional Derivative	11
SCF	Spacial Contact Force	22
PID	Proportional Integral Derivative	12
FRF	Frequency Response Function	ix
FFT	Fast Fourier Transform	48
POI	Point of Interest	1

1. Introduction

1.1. Motivation

1.1.1. Industrial Wire Bonders

Wire bonding machines are among the most crucial pieces of equipment within the semiconductor industry [20]. They serve the purpose of accurately forming bonds between contact pads on the surface of silicon wafers. These machines execute their tasks by moving in three Degree of Freedom (DOF)s. The primary movements take place in the X and Y directions, which align with the plane of the substrate's surface on the chip where the wire bonding process is conducted. These movements enable the translation of the end-effector of the machine bondhead, also referred to as the Point of Interest (POI), to the precise position required for the bonding process. Once the bond head is positioned directly above the bond pad where the bonding operation is to be performed, a translation in the Z-direction, perpendicular to the substrate's surface, is executed to finalize the process. The Z-axis stage of a wire bonding machine is illustrated in Fig. 1.1.

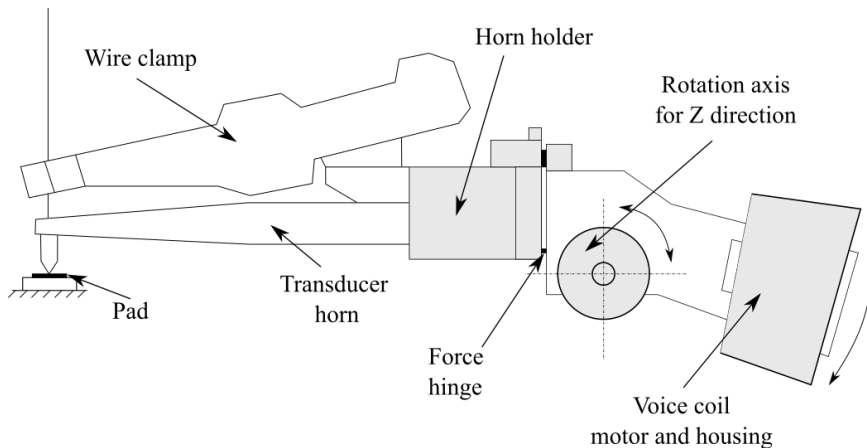


Figure 1.1.: Z axis stage of an industrial wire bonder.

Moore's Law states that the number of transistors on the surface of an integrated circuit doubles every two years [23]. As the technology has progressed, the number of connections on the surface of the substrate has also increased. To accommodate this, the number of bond pads on the surface has risen consequently leading to them becoming smaller and smaller. This along with the miniaturization of the chips themselves has led to finer and finer wires being used in the wire bonding process. Recent semiconductor chip housings require the pad pitch of the wires used to be as small as $45\text{ }\mu\text{m}$ [16]. Extensive research has been carried out to evaluate the performance of the wire bonding process to find the important process

1. Introduction

parameters and metallurgical conditions that lead to a good quality bond being undertaken [22].

One of the key factors affecting the bond quality is the impact force that is created when the bond head makes contact with the surface of the bond pad. The development of a higher magnitude impact force can lead to the surface being damaged. To ensure safe and proper bonding characteristics, the impact force needs to be controlled. Another factor is the position and consequently the velocity of the bond head in the Z-direction. The position plays a very important role as it is necessary to ensure that the bond head does not damage the substrate surface during the downward motion. Due to the large numbers of wires and fast production requirements, the velocity also needs to be high. Therefore, if the force and the position are accurately controlled during the wire bonding process, satisfactory performance can be ensured.

1.1.2. Force Control

A number of force/position control algorithms have been discussed in the literature [5; 12] in robotic applications. These applications generally involve a robotic manipulator performing a pick-and-place operation, a peg-in-hole operation, and other occurrences of a manipulator interacting with an environment [27]. The algorithms have been classified into active and passive based on whether the force is actively controlled. In the active force control algorithms a differentiation is observed between direct force control where the force is directly controlled and indirect force control. A common requirement among force control methods is the provision of a force feedback signal, which, in turn, necessitates the installation of a force sensor on the robot.

Volpe and Khosla [27] have tested different force control methods on a robotic arm where a feedback signal is provided. Kim et al. [16] have used a piezo force sensor in the wire-bonding machine to achieve desired performance using indirect force control. Hogan [6] states that indirect force control can be used in the absence of force feedback to get results similar to those achieved in the presence of the feedback.

Due to its applicability in a wide variety of scenarios where a manipulator comes in contact with the environment, force control is definitely a field of study that can prove useful for controlling contact force in wire bonding applications.

1.1.3. Ball Bonding

Ball bonding is the most common method of wire bonding carried out in the semiconductor industry. Kim et al. in [15] and Liu et al. [18] describe the process which involves six steps for the formation of a bond between two contact pads on the surface of the substrate.

Fig. 1.2 showcases the steps involved in the ball bonding process. The first step in the bonding process is the formation of the Free Air Ball (FAB) at the POI of the bond head which is in the form of a capillary using the method of Electronic Flame Off (EFO). EFO employs a high voltage spark generated between the wire and a metal wand, elevating the wire temperature and inducing its tip's melting. The wire's surface tension shapes the melted portion into a spherical configuration, resulting in the FAB. The FAB's size depends on the wire's pitch, and a higher impact force leads to a larger compressed FAB. It is important to note that an enlarged FAB size can negatively impact bonding characteristics [16].

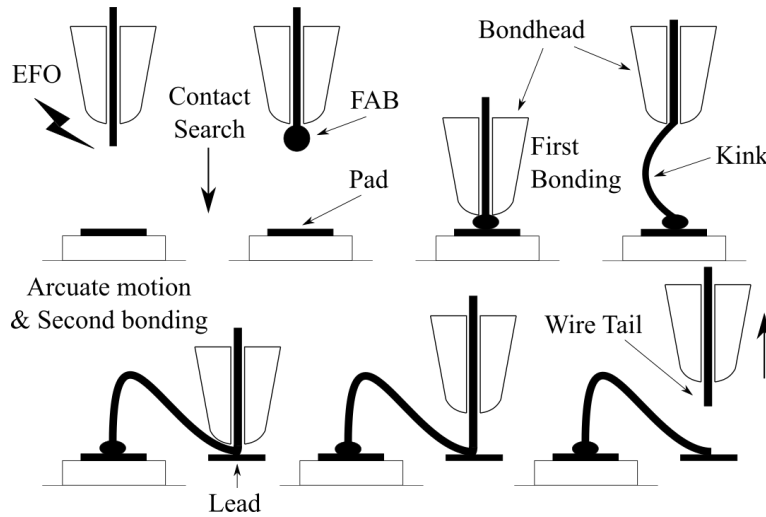


Figure 1.2.: Steps involved in wire bonding process.

After the formation of the FAB, the bond head descends downward in search mode until contact is established. Once the contact is established, the capillary is then moved further down to generate the contact force and squish the FAB to the desired size for a satisfactory bond. After the bond is created the capillary moves upward creating a kink in the wire as shown in the figure. The bond head then moves towards the second contact point in an arcuate (Bow-like) motion. It then descends and establishes contact to make the second bond. After the second bond, the capillary ascends and returns to the EFO level.

The reduction of impact or contact force stands as a significant advantage for the ball bonding process [16], primarily due to the consequent decrease in the size of the FAB along the substrate surface, as previously mentioned. Another advantage is the potential to employ a higher search speed when the impact force is minimized. The impact force directly dictates the maximum attainable search speed for the process. In scenarios without force control, search speeds must be curtailed to prevent exceeding contact force limits. The incorporation of force control empowers explicit control over the magnitude of the contact force.

1.1.4. Ideal contact force output profile

As discussed previously, controlling the force and the position in the wire bonding process greatly increases the quality of the output during the ball bonding operation. Fig. 1.3 shows both the conventional as well as the ideal force profile for the contact/impact force during the ball bonding process. The key difference between the two is the absence of overshoot. Force overshoot is highly undesirable in wire bonding. The steady-state contact force is decided depending on the wire pitch and ideally, once the force reaches this value, it should remain there to ensure proper bond quality. Overshoot, which essentially means that the force exceeds the desired contact force, might cause the bond head to exert excess force on the surface of the substrate and thus increase the FAB size ruining the bond or in the worst case might damage the substrate. In practice, according to input from ASMT experts, a 20% overshoot is deemed acceptable. Another crucial factor is settling time, which should

1. Introduction

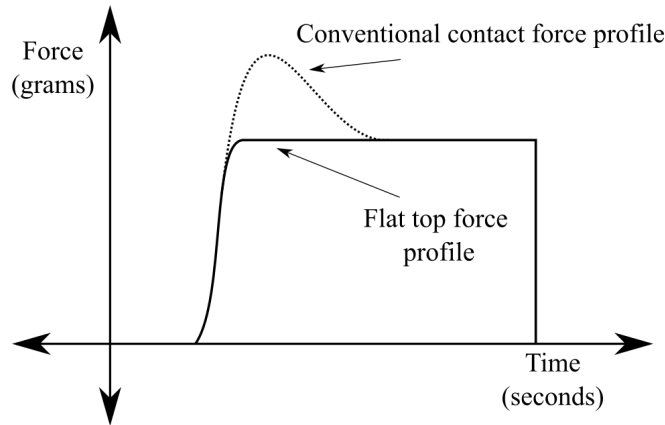


Figure 1.3.: Flat top force profile.

be minimized. Settling time is defined as the duration taken for the steady-state force to reach 90% and 110% of the intended value in wire bonding processes.

1.1.5. Necessity of Force Control

As mentioned in the previous section, force overshoot during impact is highly undesirable, and ideal performance would see a complete elimination of the overshoot. The industry achieves performance close to ideal without using force control by carefully tuning the feed-forward parameters provided to the actuators through trial and error. The main advantage of this approach is that it does not require the use of complex force control algorithms to obtain satisfactory results. They also forgo the need for a force sensor thus decreasing the cost of operation.

A key disadvantage of this method is that it has poor adaptability when it comes to changes in wire pitches and contact forces. Every time the needs of the operation change, a new set of feed-forward parameters needs to be calculated thus decreasing the speed of the operation. When a force controller is used, however, once the contact force is known for a specific wire pitch, it can be directly commanded in the algorithm and there is no delay in the operation time. It is for this reason, force control is much more preferable over the conventional method.

1.1.6. ASMPT

ASMPT holds a significant position in the semiconductor industry's backend manufacturing process, providing solutions for diverse processes including wafer and panel level packaging, MEMS, Discrete ICs, and Power Electronics. One of ASMPT's offerings is the wire bonding machine. This thesis has been conducted in collaboration with and under the guidance of experts from ASMPT. Throughout the report, particularly when it comes to parameter selection aligned with the wire bonding use case, consultation with ASMPT experts has been integral.

1.2. Research Direction and Objectives

As mentioned in the preceding section, there exists significant potential for the integration of force control in the context of industrial wire bonding. Literature showcases a diverse array of force control methods that have been experimentally tested to assess their feasibility in the wire bonding process. However, in the absence of actual setups, evaluating the controllers and control architecture's effectiveness becomes challenging. This gap can be addressed by simulating the wire bonding process within a digital environment. It is also important to highlight that employing a simulation environment for testing control algorithms provides a faster avenue compared to implementing them directly on physical setups. This approach not only expedites the testing phase but also ensures a safe experimentation environment, a crucial factor considering the potential high costs associated with any mishaps that could harm the machinery.

While digital modeling necessitates certain assumptions, it offers distinct advantages. Physical experimental setups entail inherent safety limits to prevent damage. In instances of damage, rectifying the setup can prove exorbitant. Preceding real-world experimentation with simulations can significantly reduce the risk of failures in the actual setup.

Hence, this thesis aims to achieve several objectives. Firstly, it seeks to comprehensively examine the wire bonding process and the dynamics of the wire bonding machine's Z-direction, culminating in the precise modeling of machine-environment interaction within a simulation environment. This modeling employs [Simscape Multibody](#). Secondly, the thesis aims to explore the array of force control types examined in the literature and select the most suitable algorithms based on criteria tailored to wire bonding requirements. Lastly, it strives to implement the most optimal force control algorithm within a [Simulink](#) environment, facilitating the evaluation of diverse force and position controllers in the pursuit of optimal wire bonding performance.

1.3. Thesis Overview

This thesis report is structured as follows:

- **Chapter 2:** This chapter presents various force/position control methods found in the literature. A comparison of these methods is conducted based on criteria specifically chosen for wire bonding.
- **Chapter 3:** In this chapter, the modeling of the wire bonding process within the Simscape Multibody environment is detailed. An environment contact model with a single mass is introduced and modeling parameters are selected. Furthermore, an impact transition model is formulated to simulate bondhead-substrate interaction and subsequently integrated into Simscape.
- **Chapter 4:** This chapter focuses on the Simulink implementation of the contact detection subsystem, along with the setpoint generators for reference force and position in the model simulation. The architecture's integration into Simulink is showcased, alongside the design of controllers for force and position regulation.

1. Introduction

- **Chapter 5:** Here, the variables chosen for simulating bondhead-substrate interaction are outlined, aligning with the criteria of the real-world use case. Subsequently, the simulation outputs from the model simulations are comprehensively discussed.
- **Chapter 6:** The concluding remarks are presented in this chapter, including suggestions for enhancing simulations and future experimental endeavors.
- **Appendices A, B, and C:** These appendices provide information on single mass contact model implementation and system identification, initial position setpoint generation, and calculation of contact detection time.

2. Literature Review

This chapter highlights various force control algorithms as discussed in the literature and their state-of-the-art implementations. These methods have been chosen to suit the specific use case of force control in a single [DOF](#) context, as elaborated in the subsequent chapters that present dynamic models for interactions between the wire bonder and its environment. The algorithms are subsequently subjected to a comparative analysis and evaluation, using predefined criteria tailored to the wire bonding application.

2.1. Classification of Force Control Methods

Force control in the literature is primarily categorized into two types: Active and Passive Force Control. Active force control methods involve commanding the force by providing a required signal to the actuators within the system. In contrast, passive force control does not explicitly command the force; instead, it restricts the force magnitude through passive compliance designed into the system's structure. An advantage of passive force control is its lack of dependence on a force sensor, rendering it relatively cost-effective. Moreover, it exhibits higher speed compared to active control methods since it doesn't require additional computational resources for force measurement and control. However, this absence of explicit control implies an inability to guarantee that the generated forces remain within acceptable limits [12].

Despite its slower pace and higher cost, Active Force Control ensures that the generated contact forces remain within predefined limits, making it a more reliable force control approach. Active methods can be divided into direct or explicit force control and indirect force control methods. Direct force control algorithms involve commanding the required force using a combination of force feedback and feed-forward techniques. In contrast, indirect force control methods operate by providing position set-points and imposing mechanical impedance on the environment to restrict undesired forces [6]. These methods do not incorporate an explicit force feedback loop. The following section presents a range of force control methods proposed in the literature. The primary focus here is on force control in a single [DOF](#), as the contact forces involved in the wire bonding process occur predominantly along the Z-direction.

2.2. Direct Control Algorithms - Hybrid Force/Position Control

2.2.1. Proportional Force Control

$$F = F_r + K_{fp}(F_r - F_m) - K_v\dot{x}_m \quad (2.1)$$

2. Literature Review

(2.1) shows the general control law for Proportional Force Control [5; 25] where K_{fp} represents the proportional gain, K_v represents the velocity gain (damping), F_r and F_m are the desired and measured force respectively and $\dot{x}_m$ is the measured POI velocity.

This control law is mainly divided into three sections which are the feedback represented by the force error term, the feed-forward represented by the F_r term, and finally the velocity damping term. The output of the control law F is applied directly to the actuator as the actuation force.

Gorinevsky [5] used a direct Lyapunov stability analysis method to prove that the proportional force control applied to a collocated system is globally asymptotically stable for any positive value of K_{fp} . However to reduce oscillatory behavior caused by large values of K_{fp} , the damping gain K_v also needs to be increased.

This control law was implemented by Volpe and Khosla [27] on a robot arm to test force control applicability for different values of K_{fp} . They discovered that while maintaining the same value of the damping gain K_v , increasing K_{fp} leads to a reduction in the steady-state error. However, at the same time, the overshoot in the step-response also increases consequently. Lowering K_{fp} which is stable at values greater than -1 results in lesser oscillations in exchange for a larger steady-state error. The larger steady-state error makes this controller less suitable for force-setpoint tracking.

2.2.2. Proportional Derivative Force Control

$$F = F_r + K_{fp}(F_r - F_m) + K_{fd}(\dot{F}_r - \dot{F}_m) - K_v\dot{x}_m \quad (2.2)$$

(2.2) showcases the control law for Proportional Derivative force control [27; 25] where K_{fd} is the derivative control gain and $\dot{F}_r$ and $\dot{F}_m$ are the derivatives of the desired and measured force respectively.

In the case where the sensor stiffness and the environment stiffness are high, when contact occurs, the motion is very small in magnitude. It is for this reason that the method of including the Force derivatives in the control law was proposed.

$$F_m = k_s x_m \rightarrow \dot{F}_m = k_s \dot{x}_m \quad (2.3)$$

Considering a system involving a mass connected to a fixed environment frame through a sensor spring with stiffness k_s , a relation between the mass displacement and the spring force can be derived as shown in (2.3). It can be observed that the force signal can also be considered as an amplified position signal, where the relation between them is the sensor stiffness. For the derivative of the measured force signal, the same holds true. Thus a variety of concepts have been proposed in the literature for using the force derivative feedback for application in active damping [27; 4; 28; 29].

When the measured force signal is obtained using a force sensor, the signal is also accompanied by significant measurement noise. This noise gets amplified when a derivative is taken without any pre-processing. Therefore, to obtain a viable derivative of the measured force signal, low-pass filtering is employed. This method was used by Volpe and Khosla [27] and Xu et al. [29] to obtain the force derivative signal which was then used in the Proportional

Derivative force control. However, this noise rejection advantage provided by the low-pass filtering also results in the trade-off where phase lag is introduced in the feedback signal. It is due to this reason that no significant improvement can be observed when compared to the Proportional control law mentioned in (2.1).

Another method to obtain the force derivative is to utilize an estimation model to calculate it rather than taking the actual derivative. This method was utilized by Qian et al. [21] thus overcoming the drawback of low-pass filtering as the estimation model has a lower sensitivity to noise due to having limited bandwidth. Nevertheless, significant noise was still observed in the feedback in their simulations and experiments. Ultimately, when compared, both the proportional and proportional derivative control laws showcase similar results.

2.2.3. Proportional Integral Force Control

$$F = K_{fp}(F_r - F_m) + K_{fi} \int (F_r - F_m) dt - K_v \dot{x}_m \quad (2.4)$$

(2.4) showcases the control law for Proportional Integral force control [27; 30; 25] where K_{fi} represents the Integral force gain. The key difference in this control law is that the force error is integrated over the time span, 't' of operation. This inclusion of this integral in the control law is what enables this control law to achieve a zero steady-state error value.

Gutz et al. [30] have used this law on a dynamic model in both impact and contact phase after impact. The proportional integral force control was utilized to achieve stable contact during the experimentation phase. The force response recorded however showcases multiple significant peaks and an overall quite oscillatory force response. This oscillatory behavior is not desirable as it would lead to damage when applied to sensitive equipment like the wire bonder. Integral force control has also been tested in [27] for impact control where it was found to be stable for low gains but still inapplicable in the transition phase due to undesirable response where impact transient is affected by the controller integral windup. Limiting the initial force peak can help prevent the undesirable impact transient.

Stable contact was also achieved by Volpe and Khosla [27] and they were able to establish excellent force set-point trajectory tracking. They note that increasing the integral gain K_{fi} to a limit leads to remarkable improvement in the rise time but increasing the value beyond the limit leads to oscillations and can thus be dangerous for sensitive equipment without modifications showcased in subsequent sections.

2.3. Indirect Force Control Algorithms - Impedance Control

Impedance control constitutes the majority of indirect-force control where the desired force is not explicitly commanded. Rather than an algorithm like previous methods, impedance is a type of approach to force control [1]. The main ideology of impedance control is showcased in figure 2.1 where the trapezoid block at the top represents the bond head POI and the flat grey block represents the surface of the substrate or the environment. In impedance control typically, a position setpoint is commanded which is physically not realizable. In the figure, it can be observed that the setpoint is below the surface of the substrate and thus the bond head cannot reach it. When the bond head descends during operation and makes contact, it

2. Literature Review

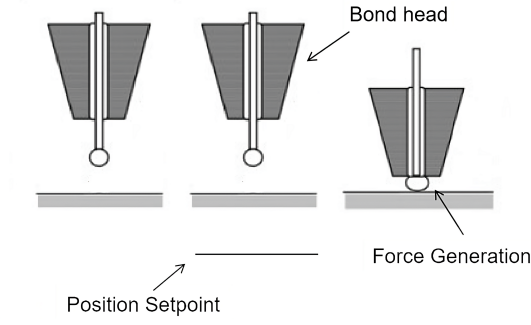


Figure 2.1.: Impedance control - Basic ideation.

tries to push further even though it is physically impossible causing position error. This is used to generate the necessary force required to squish the FAB.

$$E_f(s) = Z(s)E_x(s) = (Ms^2 + Cs + K)E_x(s) \quad (2.5)$$

Impedance control is realized by establishing a relationship between the aforementioned position error and the force error. This relationship is termed "impedance," denoted as $Z(s)$ in (2.5). Typically, a second-order mechanical impedance model is employed to connect the force error E_f and position error E_x [12; 26; 6; 7]. Here, the variable s represents the Laplace domain.

M , C , and K in (2.5) are the desired parameters for the impedance. M represents the desired inertia relating the acceleration to the force dependent on the mass of the bond head and the controller inertia. C is the desired damping parameter relating the velocity to the exerted force dependent on both the active (controller) and passive (system) damping. K is the desired impedance stiffness which describes the important relation between the position and force errors and is dependent on the controller and actuator stiffness. Provided the system is stable, these parameters can be adjusted to reach the application requirements. Impedance control is also used in combination with active compliance in the case of contact with a stiff environment. Due to this, sensor compliance is irreplaceable when stiff environment contact occurs [12].

$$F = K_p(x_r - x_m) + K_{fp}(F_r - F_m) - K_v\dot{x}_m \quad (2.6)$$

(2.6) showcases a control law to implement impedance control in terms of a force control algorithm proposed by Gorinevsky [5]. This control law basically considers only the stiffness relationship between the force and the position errors eliminating the terms M and C from (2.5) and thus can be classified as a zero-order impedance controller. As there is only stiffness taken into account in the impedance, it considers the manipulator to be behaving as a spring. The new term introduced in this law is the position error term where K_p is the proportional position gain. The parameters K_{fp} and K_p can be tuned to achieve different values of desired impedance stiffness and thus manipulate the relationship between the force and position errors.

De Schutter et al. [4] and Volpe and Khosla [26] have stated that even though impedance control differs in its structure and approach to explicit force control methods, an equivalence exists between these methods and they can be converted into each other. The equivalence of second-order impedance control and proportional explicit force control with acceleration feed-forward has been proven by Volpe and Khosla [26]. Hogan [7] proved the applicability of impedance control for ensuring stability when performing contact-based tasks and that it can be implemented during all phases i.e. both the free motion as well as during contact with the environment. Jeong et al. [9] performed a comparative study between classical explicit force control strategies and impedance control which concluded in impedance control being a superior strategy when the manipulator interacts with a hard or stiff environment while for soft environments, explicit force control becomes the better alternative. Impedance control has been implemented by Kim et al. [16] in the wire bonding process to achieve the flat top force profile in the presence of force feedback provided by a piezo force sensor. The force peak i.e. the overshoot occurring for the approach velocities of 3 and 5 mm s⁻¹ was significantly reduced. They noted that if the K_{fp} value is larger than the K_p and the environment stiffness, a steady-state force similar to the desired bonding contact force can be achieved.

Impedance control differs from classical explicit force control in such that the same controller can be used in both the search phase i.e. free motion and the contact phase without requiring any significant changes to the architecture proving to be advantageous. However, it is essential that the dynamics of the contact and environment are known for accurate calibration of the parameters [19]. A potential disadvantage of this method is that the force is not explicitly controlled but is also dependent on position error and hence the accuracy of the force trajectory tracking can be lower than the previous methods.

2.4. Advanced Force Control Methods

Although the explicit force control methods and second-order impedance control were successfully able to control the force during their respective studies and experiments, they had their own drawbacks. The main disadvantage of the explicit force control methods was that the control law doesn't account for the free motion displacement of the manipulator before the contact and thus they had to be used in collaboration with a position controller employing a methodology known as a hybrid force/position control. This was overcome by the impedance controller which could be used in both the free motion and the contact phases. However, impedance force control had its own disadvantage where it could only control the impedance i.e. the relation between the force and position errors, and not either separately hence reducing the setpoint trajectory tracking accuracy. To overcome the drawbacks of these controllers, two new methods have been implemented in the literature.

2.4.1. Parallel Force Control

$$F = K_p(x_r - x_m) + K_{fp}(F_r - F_m) + K_{fi} \int (F_r - F_m)dt - K_v \dot{x}_m \quad (2.7)$$

Equation 2.7 showcases the Parallel force control algorithm proposed to overcome the limitation of the classical explicit force control methods [2; 12]. From the control law it can be observed that it is a combination of a Proportional Integral Force controller and a Proportional

2. Literature Review

Derivative (PD) position controller. One key property of parallel force control is that it has force control as the dominating aspect. This means that priority is given to the force control over the position control in the event of there being any conflict between the two. For example, when the path of the position trajectory is blocked by an obstacle, hence making it impossible to achieve the position setpoint, the force controller dominates and ensures that the force setpoint is reached. This is established by using a Proportional Integral Derivative (PID) controller for force and a PD controller for position. The integrator component in the force controller ensures that the force setpoint tracking is achieved.

Parallel force control was rated the best in comparison to the other methods in a survey by Siciliano et al. [3] because of its superior position and force setpoint tracking. One key problem with Parallel force control is that it is very difficult to implement in multiple DOF systems where additional dynamics come into play. However, on single DOF systems like the contact force control in the wire bonding process, this method can be implemented relatively with ease.

2.4.2. Hybrid Impedance Control and Adaptive Hybrid Impedance Control

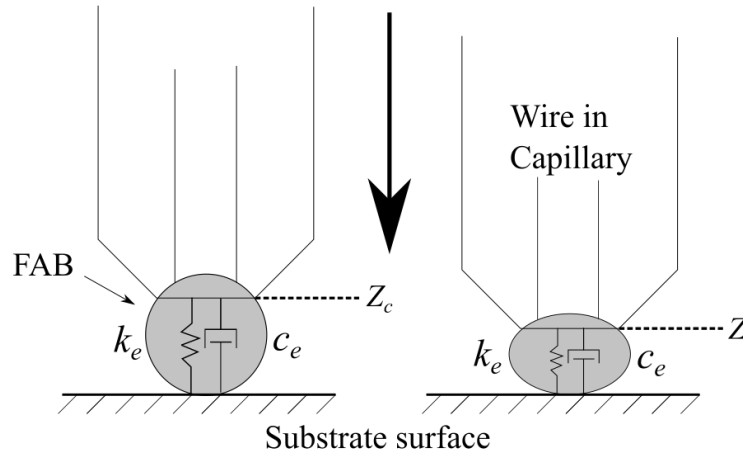


Figure 2.2.: FAB - Environment contact model.

Another method was proposed by Anderson and Spong [1] to overcome the shortcomings of the classical explicit force control and impedance control strategies and combine them together to reap the benefits of them both. Hybrid impedance control focuses mainly on impedance control with an addition that more priority can be given to force or position control specifically. This is done by modeling the impedances for the manipulator and environment interaction in a particular way. In the case of wire bonders, the interaction between the bond head, FAB, and the surface of the substrate can be defined as contact with a capacitive environment. An example of a capacitive environment can be observed in figure 2.2 where after the contact force is applied and the FAB is compressed, it has its own stiffness and damping. Anderson and Spong explain that when the environment is modeled as capacitive, the corresponding manipulator needs to be modeled as an inertial model. The circuit diagram for this interaction can be observed in figure 2.3. The inertial manipulator is represented as an impedance in series with an effort source i.e. the actuator force input and

the capacitive environment is represented as an impedance in parallel with a flow source i.e. a velocity output.

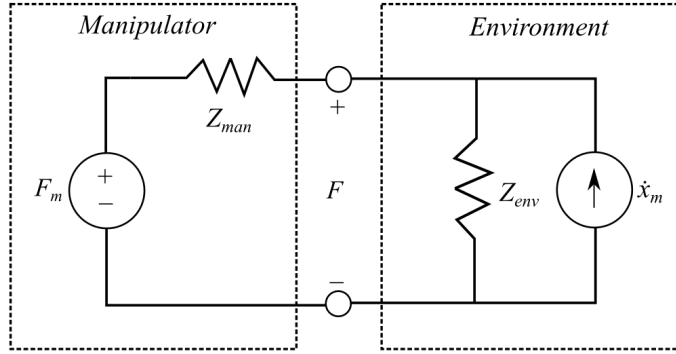


Figure 2.3.: HIC - Impedance modeling.

$$F = \frac{Z_{env}}{Z_{man} + Z_{env}} F_m + \frac{Z_{env} Z_{man}}{Z_{man} + Z_{env}} \dot{x}_m \quad (2.8)$$

The impedance values are calculated by accurately calculating the stiffness and damping parameters after dynamic modeling of the system. When the force control is considered of higher priority, the control law obtained is shown in (2.8) where Z_{man} and Z_{env} are the manipulator and environment impedances respectively.

A variety of implementations of Hybrid impedance control can be observed in recent literature [10; 8; 17]. Jung et al. [10] have implemented Hybrid Impedance control on a robot manipulator interactive with an environment. They were able to reduce the force overshoot significantly by adding an impedance function to hybrid explicit force control using the control framework derived by Anderson and Spong. Another drawback of classical impedance control proposed by Hogan was that the dynamics of the interaction needed to be known in detail for the modeling of the impedance. This was overcome by Hosseinzadeh et al. [8] and Li et al. [17] by using a method known as Adaptive Hybrid Impedance control. This method was proposed by Kelly et al. [11] to overcome the sensitivity of impedance control to the modeling errors caused by model-manipulator parameter mismatch which occurs when the dynamics of interaction are unknown. Both Hosseinzadeh et al. and Li et al. were able to implement Adaptive Hybrid Impedance control during robot interaction with an unknown environment. The Adaptive control was able to compensate for the unknown dynamics of the interaction successfully overcoming the shortcoming of classical impedance control in simulations. Furthermore, the asymptotic global stability of the system was proven using Lyapunov's theorem, even in the presence of modeling uncertainties of the manipulator.

2.5. Comparison of Force Control Methods

In the subsequent section, a comparative analysis of various force control algorithms is conducted. This comparison is tailored to the context of a single DOF application within the wire bonding process.

2.5.1. Important Parameters for Comparison

- **Force Reference Tracking**

As the control of the contact force that occurs during the impact is of paramount importance, the algorithm should be able to track a defined setpoint trajectory that is defined for the force during the operation.

- **Overshoot Suppression**

Overshoot of impact force that is outside the acceptable limit can be damaging when contact between the bond head and the surface of the substrate occurs. The algorithm should be able to minimize the overshoot and also have less oscillatory behavior after contact occurs.

- **Position Reference Tracking**

The tracking of the position of the POI is very important as very high speeds are involved. Accurate position or rather velocity control is required as the bond-head moves from high descending velocity to the comparatively lower search velocity.

- **Applicability in free motion and Contact**

Although a combination of control algorithms can be used to achieve the desired performance, it is not ideal as more computational power is required to change between algorithms when the bond head goes from the free motion to the contact phase. This could introduce additional delays in the operation and increase the time of operation. It is preferable that the same algorithm is able to be applied in both the free motion and the contact phases.

- **Ease of Tuning**

As the demand of the industry changes over time, the wire pitch and thus the contact force will also change. Moreover, a change in the wire bonding could also take place to increase the efficiency of the operation. The algorithm should be able to be intuitively and easily tuned for the changing needs of the operation.

- **Stability during Impact**

During the transition from free motion to the contact phase and the occurrence of impact, it is preferable, as described in the ideal force output section, to have minimal overshoot and a short settling time. To achieve this, it is important for the algorithms to exhibit predictable and stable behavior.

- **Robustness**

The force control should also be robust to operational variations and changes in the demands of the wire bonding process. For slight variations in the contact force taking place during the ball bonding when the FAB gets squished, the algorithm needs to be able to control the force without generating undesirable and unpredictable output.

2.5.2. Grading Table for Comparison

Control Algorithms	Force Reference Tracking	Overshoot Suppression	Position Reference Tracking	Applicability in Free motion and Contact	Ease of Tuning	Stability during Impact	Robustness
Proportional Force Control	+	-	-	-	++	++	++
Proportional Derivative Force Control	+	-	-	-	++	++	++
Proportional Integral Force Control	++	-	-	-	+	+	++
Impedance Control	+	+	++	++	-	++	-
Parallel Force Control	++	++	++	++	++	++	++
Hybrid Impedance Control	++	++	++	++	-	++	++
Adaptive Hybrid Impedance Control	++	++	++	++	++	++	++

Table 2.1.: Force control algorithms comparison.

Table 2.1 showcases the comparison between the force control algorithms presented in the previous sections. The algorithms are rated based on their performance with respect to the parameters discussed. In the table, ++ indicates that the algorithm performs extremely well, + indicates satisfactory performance, and – is used to showcase poor performance in regard to the criteria considered.

2.5.3. Conclusion from grading

From the table, it is evident that all the algorithms perform satisfactorily in terms of force setpoint trajectory tracking. The classical explicit force control algorithms, including pro-

2. Literature Review

portional, proportional derivative, and proportional integral methods, exhibit robust characteristics and stability. These methods are also relatively easier to tune. However, due to their control laws not incorporating the position error, they require a separate position controller. Consequently, this compromises their tracking performance in terms of position. Moreover, the need to switch controllers when transitioning from free motion to contact phase diminishes the overall applicability of these algorithms across the entire operation.

Impedance control addresses the position setpoint tracking limitations of classical explicit force control and delivers satisfactory force setpoint tracking results. However, it falls short in robustness when operational parameters change, as such changes could significantly alter the system's dynamics, and tuning parameters are highly dependent on these dynamics. This sensitivity to modeling errors makes impedance control challenging to tune.

These shortcomings are effectively addressed by parallel force control, which combines elements from both explicit and impedance force control methods while maintaining a focus on the explicit aspect. Parallel force control yields excellent outcomes across all parameters. It is applicable throughout the entire operation and can be intuitively tuned due to its architecture being rooted in explicit strategies, reducing its reliance on dynamic modeling of interactions.

Hybrid impedance control also demonstrates impressive results in force and position setpoint tracking, addressing the robustness deficit found in classical impedance control. Nonetheless, its susceptibility to modeling errors remains, preventing it from being the optimal solution. This limitation is overcome by adopting an adaptive algorithm capable of adjusting controller parameters autonomously, especially when dealing with unknown environments. Consequently, Adaptive Hybrid Impedance force control excels across all parameters.

In conclusion, the comparison reveals that Parallel Force Control and Adaptive Hybrid Impedance Force Control are the most suitable methods for the wire bonding process application. Considering that the wire bonding process involves bondhead-environment interaction in a single DOF and the interaction dynamics are partially known based on input from ASMP experts, the additional complexity of the Adaptive Hybrid Impedance Force Control would be unnecessary for this specific use case. Therefore, this thesis has implemented Parallel Force Control to assess force and position controllability during model simulations.

2.6. Contact Force Measurement

To regulate the contact force and achieve force setpoint trajectory tracking, it is essential to measure and minimize the force error. This error arises from the discrepancy between the commanded contact force and the real contact force that arises when the bond head comes into contact with the substrate surface, causing the FAB to deform. Accurate determination of the actual contact force is crucial for effective force control.

The most common and reliable approach for detecting and measuring contact force involves the utilization of a force sensor. This sensor is typically strategically positioned on the device to directly detect the contact force in a collocated manner or indirectly in a non-collocated manner. An example of this can be found in the work of Kim et al. [16], where they employed a piezo force sensor on an industrial wire bonder to accurately determine and regulate the contact force.

However, using a force sensor does come with certain significant drawbacks. Primarily, these sensors are expensive pieces of equipment, which in turn increases the operational costs. Moreover, due to their inherent fragility, they are prone to damage. The operational speeds involved in the process could potentially harm the sensor, leading to additional costs for replacements.

Zhou et al. [31] have incorporated a force sensor into the wire bonding process in conjunction with a simple force and position switching controller, achieving satisfactory results for force setpoint tracking. They have also discussed the suitable position on the machine for mounting the piezo sensor, as it cannot be mounted at the tip of the bond head due to its aforementioned fragile nature and the high speeds involved. For the purposes of this thesis, it is assumed that a force sensor is integrated into the system and can accurately gauge the contact force exerted across the FAB when it is compressed.

3. Modeling the Wirebonding Process in Simscape Multibody

3.1. Single Mass Contact Model

In Fig. 3.1, the model for a single mass in contact with the environment is presented. Kim et al. [16] employed this model in combination with Hogan's classical impedance force control to regulate the contact force in an industrial wire bonder. The stiffness coefficient k_e and the damping coefficient c_e represent the stiffness and damping of the FAB when it is compressed and m_A represents the mass of the bondhead. F_r shown in the figure represents the input force to the system. Z shows the displacement output for the POI of the bondhead and F_m showcases the contact force measured across their stiffness and damping of the FAB.

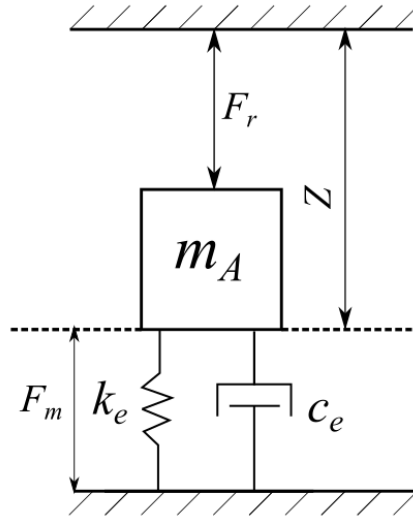


Figure 3.1.: Single mass environment contact model.

Using a combination of a contact detection algorithm [15] and a non-collocated force sensor for receiving contact force feedback, Kim et al. achieved a flat-top force profile as discussed in chapter 1. While the rise time was slower, the contact force overshoot was completely eliminated without adversely affecting bond quality.

This model can be utilized to simulate dynamics when the bond head makes contact with the environment. The contact force to be measured is the force across this spring and damper. The spring and damper represent the dynamics between the bondhead and the environment. In practical application, another set of spring and damper should be present to represent the

3. Modeling the Wirebonding Process in Simscape Multibody

dynamics between the bond head and the rest of the wire bonding machine. Considering the compliance in the Z-direction motion of the bond head, the stiffness and damper coefficient values between the bondhead and the wire bonding machine are negligible. Therefore, for the purpose of this thesis, only the contact dynamics are considered.

3.1.1. Model Parameter Selection

To ensure accurate simulation results, it is crucial that the model parameters closely resemble real-world conditions. In line with ASMPT's input, the following parameter values were chosen for the model:

- m_A : Mass of Bondhead = 0.13701 kg
- f_e : Resonance frequency of the additional mode when in contact = 50 Hz
- ζ_e : Relative damping of the additional mode when in contact = 0.03

These values can be utilized to calculate k_e and c_e using the following formulas:

$$k_e = (2\pi f_e)^2 m_A \quad (3.1)$$

$$c_e = 2m_A 2\pi f_e \zeta_e \quad (3.2)$$

(3.1) and (3.2) present the equations for the calculation of k_e and c_e [24]. Using the parameter values previously discusses, the values for k_e and c_e are $1.3522 \times 10^4 \text{ N m}^{-1}$ and 2.5826, respectively.

This single mass contact model was implemented in Simscape Multibody and system identification was performed to check the validity of the model. A detailed description of the steps involved has been provided in Appendix A.

3.2. Single Mass Impact Transition Model

While the previous section presented the model for the wire bonding process after the contact phase had taken place, it is equally vital to accurately simulate the initial impact of the bondhead on the substrate. This simulation is particularly critical because the region of initial impact is where catastrophic failures can occur if the initial impact force is excessively high.

This section introduces a comprehensive model that encompasses both the initial impact and the subsequent contact phase of the wire bonding process.

3.2.1. Free body diagram of impact transition model

Fig. 3.2 presents the Impact Transition Model, illustrating the various stages of the wire bonding process, including the initial impact and subsequent contact phases. The left figure portrays the model configuration prior to contact, showing the beginning of the descent. As the descent progresses, the center figure demonstrates the moment when the FAB initially contacts the substrate surface, and the right figure displays the model as the FAB becomes compressed.

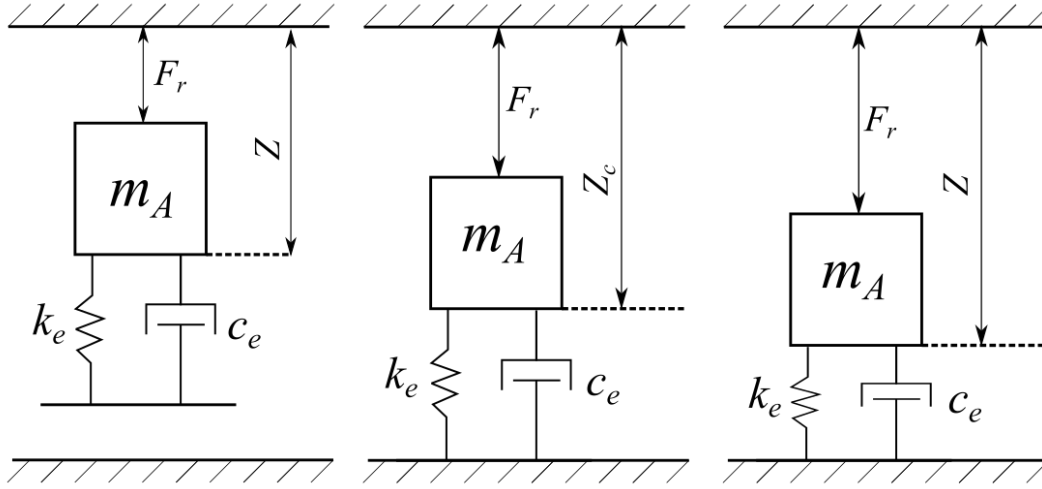


Figure 3.2.: Impact transition model.

During the descent, the FAB remains attached to the wire and subsequently the bondhead, leading to the modeling of the spring and damper components connected to the bondhead mass m_A . Upon contact, the lower portion of the FAB aligns with the substrate surface. Further downward movement results in the compression of the FAB, generating the necessary contact force across the spring and damper, crucial for the bonding process.

In the diagram, the POI displacement of the bondhead is represented as Z , and the position at which the initial contact is established is denoted as Z_c . As the FAB compresses, the bondhead displacement corresponding to the development of the contact force is given by $Z - Z_c$.

This model's integration into Simscape equips the control architecture with essential initial impact force information, enhancing its realism in replicating real-world wire bonding scenarios.

3.2.2. Transfer functions before and after contact

$$\frac{Z(s)}{F_r} = \frac{1}{m_A s^2} \quad (3.3)$$

$$\frac{F_m}{F_r} = 0 \quad (3.4)$$

3. Modeling the Wirebonding Process in Simscape Multibody

The transfer functions for the impact transition model prior to contact are given by (3.3) and (3.4). In the transfer function relating input force to output displacement (3.3), the model exhibits behavior akin to a single mass descending freely in space, resulting in a simplified single inertial force term. Moreover, before contact is established, the spring and damper remain uncompressed, precluding any force generation across these components. This circumstance leads to a measured force of zero, as exemplified in (3.4). It's important to note that, in practical scenarios, the force sensor is typically subject to measurement noise. This topic is elaborated upon in subsequent sections.

$$\frac{Z(s) - Z_c(s)}{F_r} = \frac{1}{m_A s^2 + c_e s + k_e} \quad (3.5)$$

$$\frac{F_m}{Z(s) - Z_c(s)} = c_e s + k_e \quad (3.6)$$

$$\frac{F_m}{F_r} = \frac{c_e s + k_e}{m_A s^2 + c_e s + k_e} \quad (3.7)$$

After contact takes place, the model closely resembles the contact model, with the only distinction being that the displacement is relative to the bonder frame. (3.5) displays the transfer function from input force to the output displacement of the bondhead, relative to the wire bonder frame. This outcome corresponds to the contact scenario discussed previously (see Appendix A). The other two equations of motion can be derived in a similar manner. Multiplying Equations (3.5) and (3.6) yields results akin to those obtained in the contact scenario, thereby preserving the system identification validity discussed in Appendix A.

3.2.3. Impact transition model representation in Simscape

Fig. 3.3 demonstrates the implementation of the impact transition model within Simscape Multibody. The system includes three primary masses, two of which remain consistent with the previous model: the bondhead mass and the substrate surface mass. A notable distinction lies in the bondhead mass configuration within this model—it functions as a free mass, solely attached to a compliant prismatic joint on its upper part, enabling unrestricted motion in the Z-direction. An additional mass has been introduced to symbolize the FAB. The setup for the rigid transforms and the transform sensor are the same as described for the single mass contact model (see Appendix A).

The principal contrast between the contact model and the impact transition model lies in the utilization of the Spatial Contact Force (SCF) block. In the impact transition model, the SCF block plays a pivotal role in detecting contact between the FAB masses and the surface. This is facilitated by defining unique geometries for each mass, as illustrated by the "G" labels in Fig. 3.3. The SCF block detects contact between these geometries and simulates the interaction.

The input remains consistent with the prior model. However, the output now encompasses more than just the bondhead's POI displacement; it also includes signals from the SCF block.

3.2. Single Mass Impact Transition Model

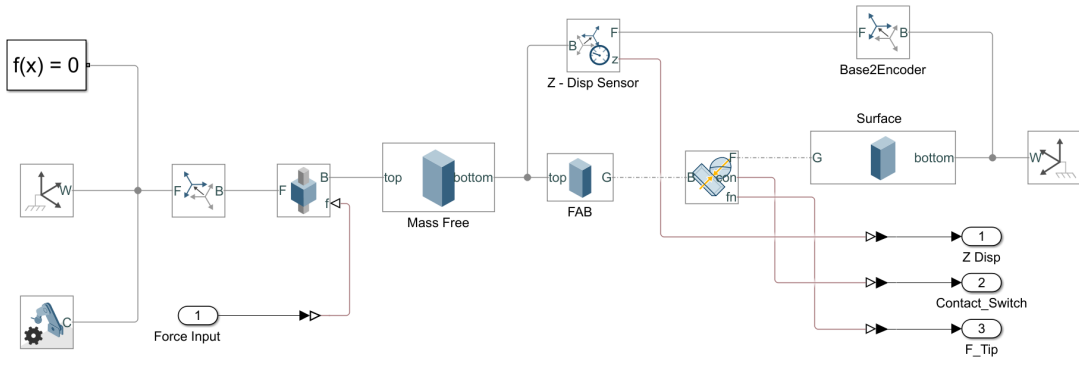


Figure 3.3.: Impact transition model - Simscape implementation.

These signals pertain to contact detection, elaborated upon in the following section, and the contact force perpendicular to the colliding geometries' surfaces. The SCF block's parameters define contact dynamics, incorporating the values of k_e and c_e .

Of notable importance is the intermass distance, which dictates the collision occurrence. In consultation with ASMPT, the intermass distance (or approach distance) between the FAB and the surface is set to 1 mm. Upon the FAB's traversal of this distance, facilitated by the SCF block, contact with the surface is established.

Another SCF block parameter is the transition region width, governing the extent of the geometrical intersection during contact and the point at which the spring's full force is activated. As the value for this parameter cannot be set to zero, for this thesis, this region has been set to 1e-10 m, ensuring the near-instantaneous engagement of the spring and damper upon contact. This configuration yields the highest impact force. In practical applications, with higher compliances and subsequently lower contact forces, the control architecture's capability to manage higher contact forces can potentially be applied to real-world systems.

4. Control Architecture Implementation in Simulink

4.1. Parallel force control - Schematic

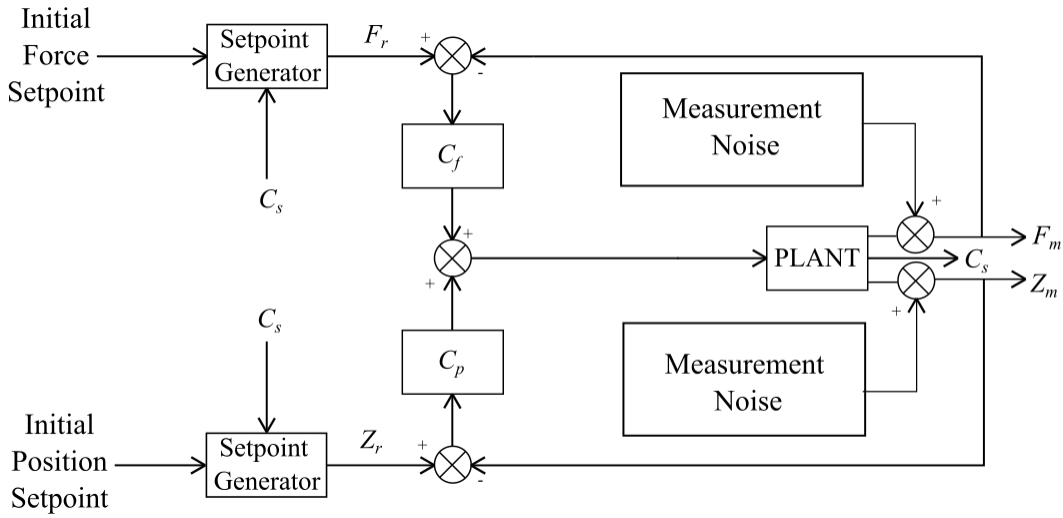


Figure 4.1.: Parallel force control.

In Chapter 2, it was established that Parallel Force Control (PFC) and Adaptive Hybrid Impedance Control were the most promising options for simultaneously controlling position and force during the wire bonding process. Fig. 4.1 illustrates the control structure schematic proposed by Chiaverini and Sciavicco[2] for the Parallel Force Control algorithm modified to include the setpoint generator used in this thesis and measurement noise for the force and position sensors. The diagram reveals two distinct loops within the control scheme: the upper loop, responsible for force control, and the lower loop, governing position control. The system's inputs comprise F_r , representing the reference force, and Z_r , signifying the reference position. In the context of the impact transition model explored in Chapter 3, three outputs emerge from the plant: the measured force F_m , the position Z_m , and the contact detection C_s signal.

In the ideal scenario devoid of measurement noise, before contact initiation, both the reference force and the measured force are zero, rendering the upper control loop inactive. During this phase, the primary input to the system remains the reference position. The positional error is processed by the position controller, subsequently serving as the principal input to the plant. As contact is established, the measured force assumes a non-zero

4. Control Architecture Implementation in Simulink

value, triggering the activation of the force controller. The force controller's role is to address the disparity between the reference force (which is zero during this stage) and the actual impact force, effectively mitigating the magnitude of the initial impact force. Once the contact detection algorithm confirms the presence of contact, the reference force trajectory is introduced to the system. The force controller is then responsible for overseeing the adherence of the system to this force trajectory. In practical application, with the presence of measurement noise in the system, both the force and position control loops remain active throughout the operation. The force controller is capable of addressing deviations from its zero reference value due to the influence of measurement noise, as will be further discussed in the subsequent sections.

After contact, it becomes necessary to prioritize force control while maintaining a degree of position control to avert potential drift. Parallel Force Control achieves this balance by employing a **PD** controller for position control and a **PID** controller for force control. Pre-contact, the **PD** controller is adequate for ensuring the bondhead adheres to the reference position trajectory. However, after contact, curbing the impact force becomes paramount to prevent substrate surface damage. This is where the **PID** force controller takes precedence due to its integration component, which enables it to exert sustained corrective actions to minimize force errors. The persistence of the position controller in the loop after contact prevents drift. Subsequent sections delve into the Simulink-based implementation of this strategy.

4.2. Contact Detection

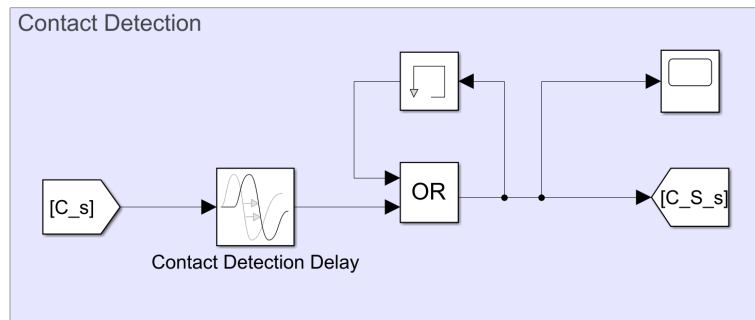


Figure 4.2.: Contact detection implementation.

The successful detection of contact represents a pivotal step in the operational effectiveness of the control structure detailed in the previous section. Contact detection serves to guide the control system in maintaining its trajectory and initializing the force setpoint trajectory only when contact is positively established. This becomes especially crucial in real-world scenarios like wire bonding, where the force sensor could be affixed to the wire bonding machine itself. Rapid movements and vibrations in the process introduce noise into the sensor data, potentially leading to erroneous force signals being sent to the controller. Depending on the controller's quality, the outcomes can range from innocuous to catastrophic. The implementation of a reliable contact detection mechanism addresses this challenge.

Contact detection also plays a crucial role in wire bonding scenarios by triggering adjustments in the input trajectories of position and force. During wire bonding, the position

trajectory is typically a constant velocity approach. However, a straightforward constant velocity ramp could lead to undesirable outcomes, such as the bondhead colliding with the surface and rebounding higher due to the elastic impact, which could damage the surface. Contact detection ensures that after contact is established, the position trajectory is altered to avert this issue. Similarly, the input force setpoint follows a smooth step rise to a desired value post-contact, but the timing of this transition depends on when contact is detected, highlighting the importance of accurate contact detection.

For the purpose of this thesis, it is assumed that a contact detection algorithm is available, capable of confirming contact within 1.25 ms from the initial impact. The contact detection signal produced by the SCF block is binary, transitioning from 0 to 1 upon contact detection. The Simulink implementation is illustrated in Figure 4.2. The contact detection signal C_s from the SCF block is subjected to a delay of 1.25 ms based on input from ASMPT, as per the aforementioned assumption. This signal may occasionally revert to zero if a high-force impact disrupts contact, causing fluctuations. To address this, a memory block captures the signal, and once it transitions to 1, the 'OR' logic block ensures that the contact detection signal remains at 1. This safeguards against potential issues in input setpoint generation for both force and position, as elucidated in subsequent sections. The initial contact detection signal is denoted by C_s and the new contact signal which remains at 1 after contact is denoted by CS_s .

4.3. Reference Position Setpoint

4.3.1. Setpoint Flowchart

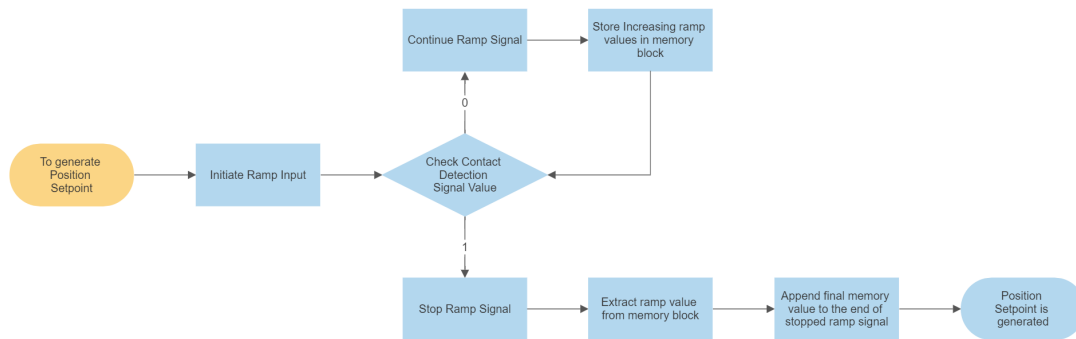


Figure 4.3.: Position setpoint flowchart.

The flowchart in Fig. 4.3 outlines the process involved in generating the position setpoint for the parallel force control structure used in the wire bonding process. First, an initial ramp input is created based on the desired approach velocity. This operation is performed within MATLAB, as depicted in Appendix B. Once the ramp input is established, the bondhead begins its downward movement at the specified velocity until contact is initiated. At each sampling point prior to the contact, the instantaneous position value is captured and stored in a memory block. Upon contact detection, the ramp movement halts, and the final value preserved in the memory block is appended to the end of the ramp. This process effectively

4. Control Architecture Implementation in Simulink

generates the necessary position trajectory, which starts as a ramp and, after contact, remains fixed at the position value where contact was established.

4.3.2. Implementation in Simulink

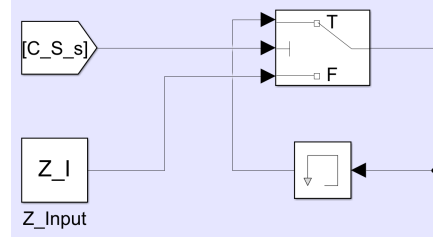


Figure 4.4.: Position setpoint implementation.

The implementation of the position setpoint generator within Simulink is depicted in Figure 4.4. Here, Z_I symbolizes the ramp input derived from MATLAB, and CS_s represents the contact detection signal. The ramp input is directed to the false terminal of a Simulink switch, with the output being fed into a memory block and simultaneously provided as input to the system. While the contact detection input remains at zero, the false terminal of the switch remains active, allowing the ramp values to be stored in the memory block at each sampling point. However, once the CS_s value shifts to one, the true terminal of the switch becomes active, interrupting the ramp signal. The memory block's output is then linked to the true terminal, enabling the last stored value before the interruption to become the constant value of the ramp. This approach ensures a smooth transition in the position setpoint generation after contact detection.

4.3.3. Generation of Position Setpoint

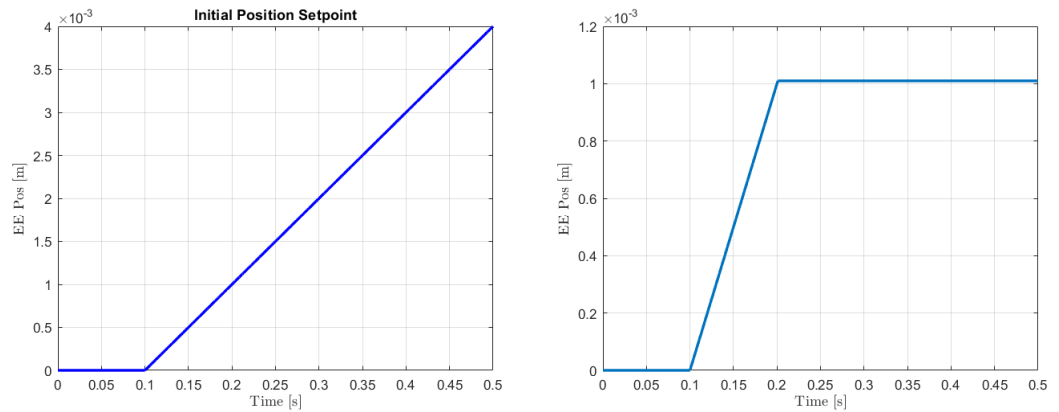


Figure 4.5.: Generated position setpoint.

The plot on the right of figure 4.5 illustrates the resulting trajectory generated by the previously discussed setpoint generator for the velocity of 10 mm s^{-1} . The initial ramp input

is set to zero for the initial 0.1 s based on input from ASMPT experts. The ramp's slope is determined by the velocity chosen in the MATLAB code. Additionally, due to the 1.25 ms delay in contact detection, a slight position overshoot is present in the setpoint. However, the controller effectively manages this overshoot, a topic that will be further elaborated upon in subsequent sections.

4.4. Reference Force Setpoint

4.4.1. Setpoint Flowchart

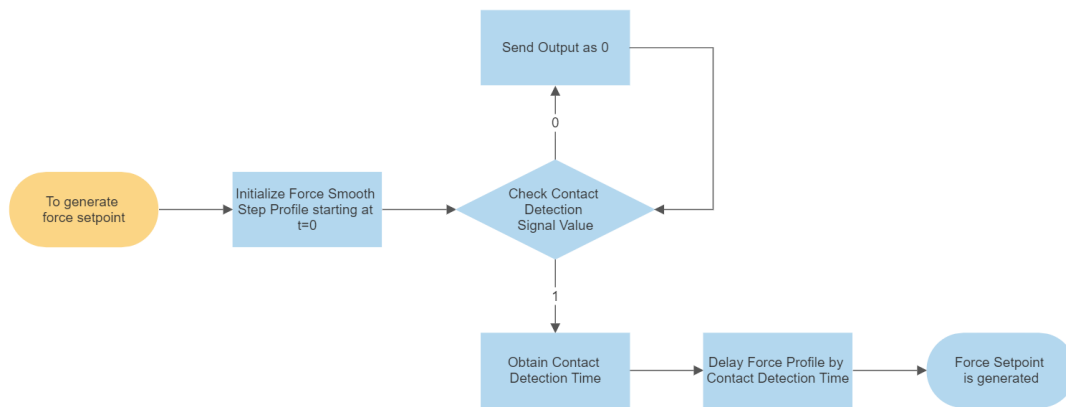


Figure 4.6.: Force setpoint flowchart.

Illustrated in Fig. 4.6, the flowchart outlines the procedural steps involved in generating the force setpoint trajectory. Similar to how contact detection is imperative to halt the position ramp and transition to a constant value, it also plays a pivotal role in initiating the force trajectory by transitioning from zero to the required trajectory. Initially, a smooth step profile for commencing at $t' = 0$ seconds and settling at the desired contact force is provided. During each sample interval, the contact detection value is assessed. If the value is zero, then zero force is sent as input to the system and the contact detection value is again assessed. This loop persists until the value changes. Once the value transitions to one, the time of contact detection is computed. Subsequently, the force trajectory is delayed by that calculated time, resulting in the trajectory's commencement at the point of contact.

4.4.2. Implementation

The implementation of the force setpoint generator in Simulink involves two crucial steps. The first step is the computation of the time when contact occurs, and the second step employs this time value to introduce a delay in the initial force trajectory, yielding the actual reference trajectory. The process of the contact detection time calculation has been highlighted in Appendix C.

4. Control Architecture Implementation in Simulink

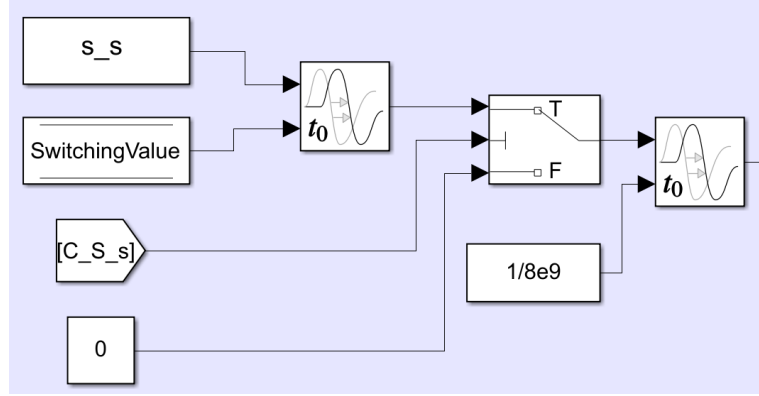


Figure 4.7.: Force setpoint implementation.

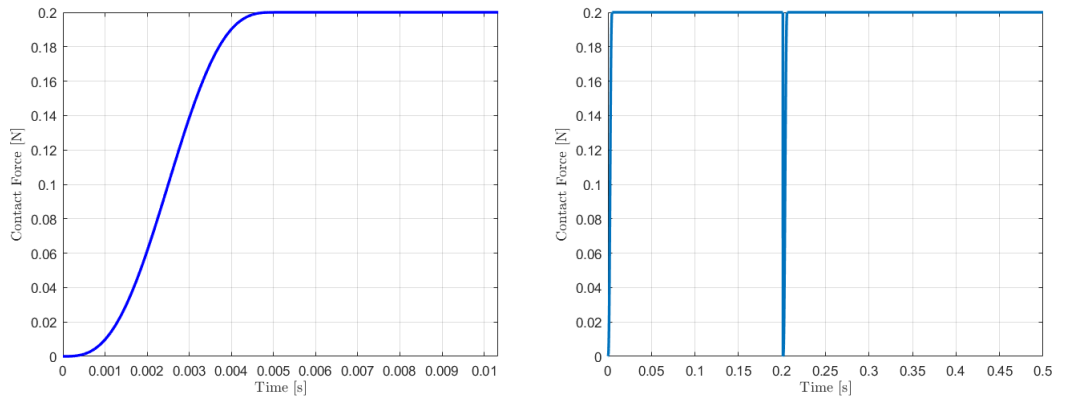


Figure 4.8.: Initial and intermediate force setpoints.

Fig. 4.7 showcases the Simulink implementation of the trajectory generator. The trajectory imported from MATLAB, shown on the left plot of figure 4.8, which is a smooth step beginning at 't=0 seconds', is utilized and denoted by 'SS'. This input signal for the force setpoint having a rise time of 5 ms and a final value of 0.2 N has been provided by ASMPT as the desired reference force trajectory. Additionally, the 'SwitchingValue' variable which is the calculated contact detection time is retrieved from the 'Data store memory - Read' block, and subsequently fed into the time delay block. This action introduces the delay into the force trajectory according to the contact detection time. However, an issue persists with the initial smooth step present at 't=0' as shown on the right plot of figure 4.8 where both trajectories appear simultaneously. To address this, a Simulink switch block is employed. This block selectively transmits the trajectory only after contact detection, effectively creating a force setpoint trajectory with zero values before contact detection, followed by a smooth step increase toward the desired force value. A small delay needs to be added due to the computation time required and the solvers involved. In the absence of this delay, the measured impact force magnitude increases further in the time when the computation for the values is occurring. The delay shown in the figure is $1/8e9$ seconds. This value can be made even smaller however the time required for the simulation then also increase due to very small values being calculated.

4.4.3. Generation of Force Setpoint

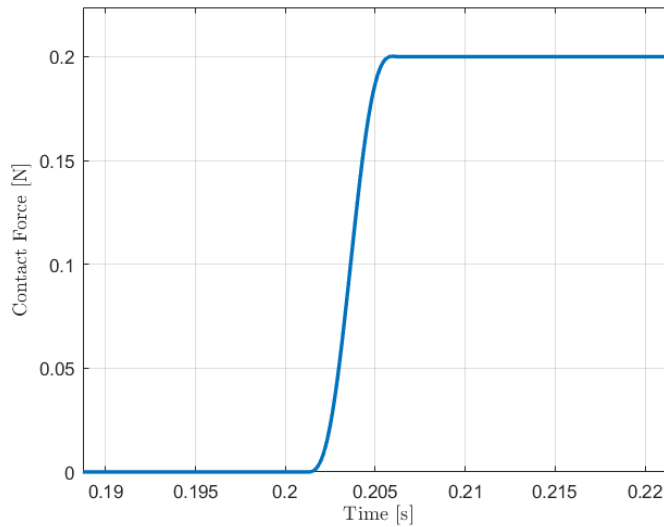


Figure 4.9.: Generated force setpoint.

Fig. 4.9 illustrates the final generated reference force trajectory. This trajectory exhibits a smooth step rise over a duration of 5 ms, ultimately reaching the desired force magnitude of 0.2 N or 20 g selected for the wire bonding use case under consultation from ASMPT. The trajectory initially remains at zero until the point of contact establishment. After the initial impact, 1.25 ms later, the contact detection time is computed. Subsequently, precisely at that calculated time, the trajectory transitions from zero to the intended smooth step setpoint value.

4.5. Additional Parameters

To enhance the accuracy of the wire bonding model and align it with the real-world ASMPT use case, several additional parameters are integrated into the control loop, as illustrated below. The selection of these parameter values is grounded in practical applicability.

4.5.1. Delays present in the system

In Figure 4.10, the Simulink implementation reveals the plant with force input and three outputs: bondhead POI displacement, measured contact force, and contact detection signal. In practical application, a number of delays get introduced into the system which include delays caused during computation, the software involved, and also the current loop of the actuators. To account for these delays, a transport delay was introduced as part of the plant. Under consultance from ASMPT experts, a delay of 0.25 ms was selected. Additionally, as depicted in the contact detection subsystem (Figure 4.2), a transport delay of 1.25 ms is applied to the contact detection signal.

4. Control Architecture Implementation in Simulink

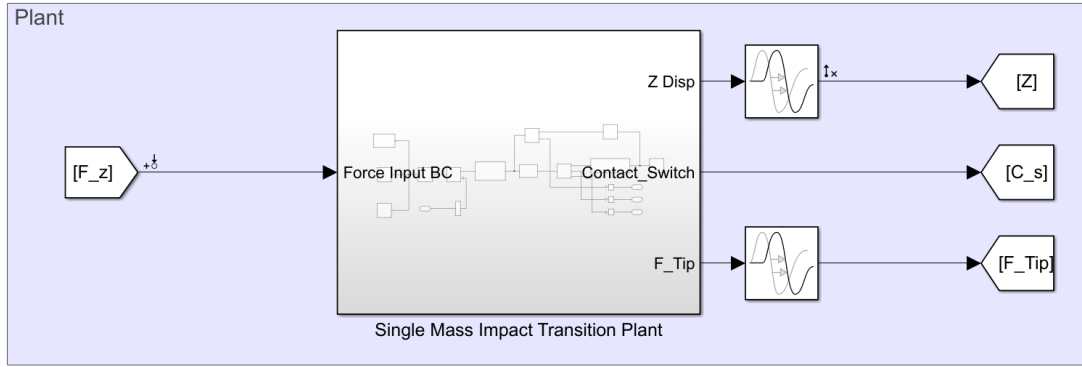


Figure 4.10.: Single mass impact transition plant with sensor delays.

4.5.2. Measurement Noise

As previously discussed, the measured outputs from both the position and force sensors in practical applications are never zero but are instead characterized by measurement noise, resulting in deviations from the true measurement values. These deviations are primarily caused by sensor quantization, calibration inaccuracies, and manufacturing imperfections. To create an accurate process model, it is crucial to account for these external influences. Ignoring these effects during the design of the control system could lead to the development of suboptimal controllers that struggle to effectively compensate for variations in sensor outputs.

To enhance the fidelity of simulation results, band-limited white noise blocks have been introduced into both the force and position output signals from the plant. These noise blocks replicate the real-world scenario of measurement noise, which originates from quantization errors and sensor imperfections. In collaboration with ASMPT, specific noise power values were selected for the noise integration. For the position sensor, a noise power of $5e-20$ was chosen, while a noise power of $5e-10$ was chosen for the force sensor. These selections were made based on consultation with experts to ensure that the noise levels accurately represent the practical measurement conditions and contribute to the realistic simulation of the wire bonding process.

4.6. Parallel Force Control Implementation in Simulink

The implementation of the parallel force control structure within the Simulink environment is depicted in Fig. 4.11. This figure portrays the integrated force and position control loops, similar to those illustrated in Fig. 4.1. On the right-hand side of the diagram, three ports are evident: F_z represents the input force applied to the impact transition model, while F_m and Z_m signify the measured contact force and the measured bondhead POI displacement, respectively both supplemented by measurement noise provided by the band-limited white noise blocks.

As elucidated in the preceding sections, the control structure operates with two simultaneous loops: the lower loop denotes the position control, while the upper loop signifies the force control.

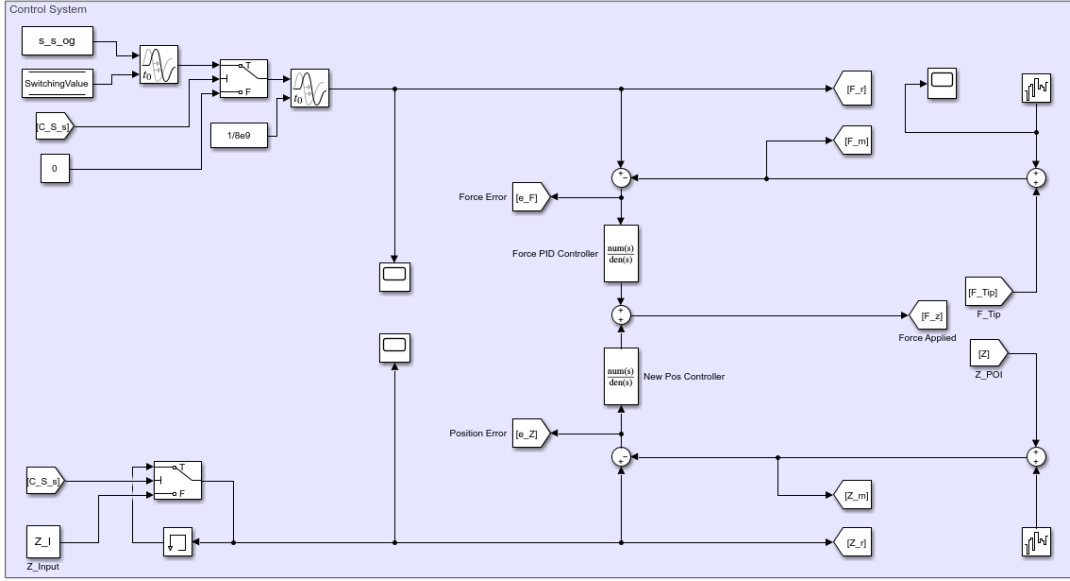


Figure 4.11.: Parallel force control implementation in Simulink.

The input references for both force and position are consistent with those detailed in previous sections for the respective force and position setpoint generators. In the central segment, the position PD and force PID controllers are aligned with the position and force errors, respectively.

4.7. Controller Design

This section presents the design of the controllers utilized in simulating the single mass impact transition model using the parallel force control architecture. As previously mentioned, the primary objective of this thesis is to model the interaction between the Z-axis bondhead and its environment during the wire bonding process and to implement a control architecture to explore different types of controllers for force and position. Consequently, to facilitate simulation, rule-of-thumb position and force controllers are introduced in this section as preliminary options. These controllers serve as a starting point for assessing the viability of the parallel force control architecture within a simulation environment.

4.7.1. Position Controller Design

$$\frac{Z(s) - Z_c(s)}{F_r} = \frac{1}{m_A s^2 + c_e s + k_e} \cdot e^{-t_d s} \quad (4.1)$$

(4.1) depicts the transfer function from input force to the measured output displacement, encompassing the transport delay t_d which is of 0.25 ms, as discussed in the preceding sections after contact has occurred. As the transfer function for the input force to output position

4. Control Architecture Implementation in Simulink

changes when contact occurs, in order to ensure that the controller is able to handle both scenarios, the position controller is designed based on the transfer function after contact.

$$C_p(s) = k_{pp} \cdot \frac{\frac{1}{w_{dp}}s + 1}{\frac{1}{w_{tp}}s + 1} \quad (4.2)$$

(4.2) represents a tamed PD controller, as discussed in the parallel force control section. The position controller has been designed for a bandwidth of 100 Hz in agreement with ASMPT experts. Using rule-of-thumb calculations [24], the parameter values can be determined as follows:

- $w_{dp} = 100/3 \approx 33$ Hz
- $w_{tp} = 100*3 = 300$ Hz
- To achieve the required cross-over frequency the value $k_{pp} = 14e3$

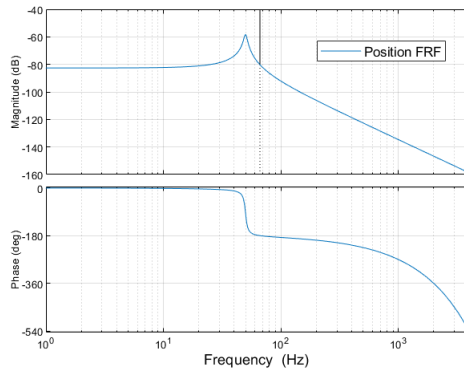


Figure 4.12.: Position FRF.

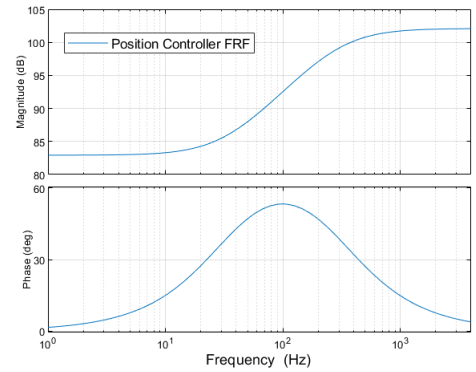


Figure 4.13.: C_p - Bandwidth 100 Hz.

Figures 4.12 and 4.13 illustrate the FRF of the position transfer function and the designed position controller, while figure 4.14 displays the open-loop position output which is the position transfer function multiplied by the position controller. The designed position controller is stable and provides a phase margin of 46.3 degrees at 100 Hz, demonstrating its suitability for this specific use case.

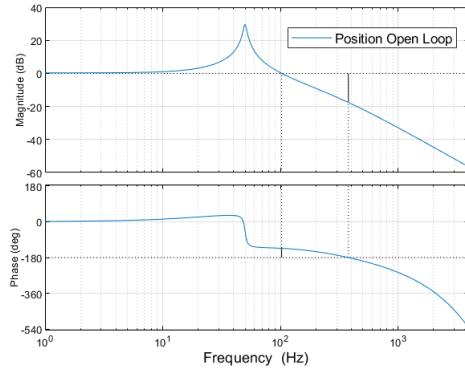


Figure 4.14.: Position open loop.

4.7.2. Force Controller Design

$$\frac{F_m}{F_r} = \frac{c_e s + k_e}{m_A s^2 + c_e s + k_e} \cdot e^{-t_d s} \quad (4.3)$$

(4.3) showcases the transfer function for the input force to the output measured force, including the sensor delay t_d after contact has occurred. Using this transfer function, the Bode plot for the FRF can be obtained, as shown in Fig. 4.15.

$$C_f = k_{pf} \cdot \frac{s + w_{if}}{s} \cdot \frac{\frac{1}{w_{df}} s + 1}{\frac{1}{w_{tf}} s + 1} \quad (4.4)$$

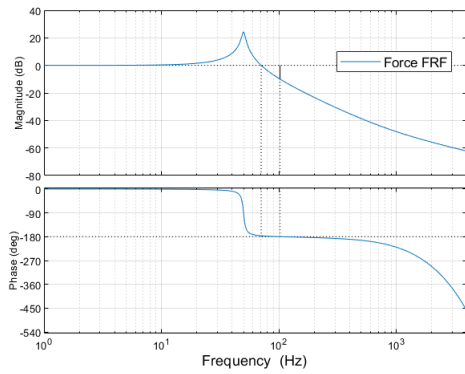
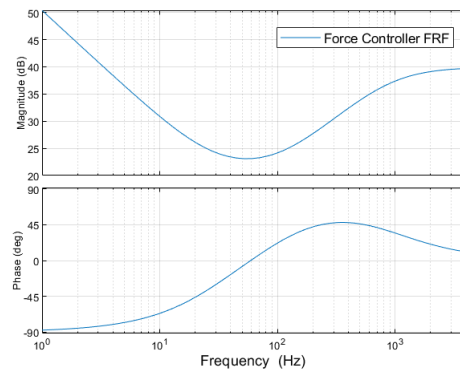


Figure 4.15.: Force FRF.


 Figure 4.16.: C_f - Bandwidth 300 Hz.

(4.4) and Fig. 4.16 present the tamed PID controller for the force loop, as discussed in the parallel force control section. Considering a bandwidth of 300 Hz based on input from

4. Control Architecture Implementation in Simulink

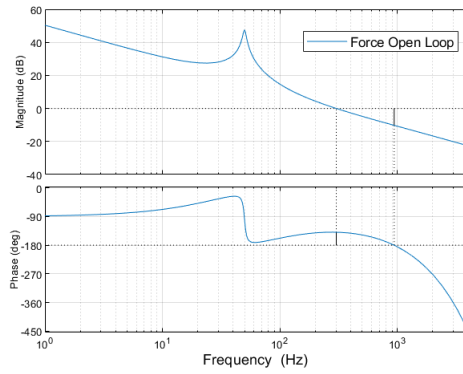


Figure 4.17.: Force open loop.

ASMPT and using rule-of-thumb calculations [24], the parameter values for the controller are as follows:

- $w_{if} = 300/10 = 30$ Hz
- $w_{df} = 300/3 = 100$ Hz
- $w_{tf} = 300*3 = 900$ Hz
- To achieve the required cross-over frequency the value $k_{pf} = 11$

Fig. 4.17 showcases the open loop force output which is the force transfer function multiplied by the force controller. The designed force controller is stable and provides a phase margin of 40.8 degrees at 300 Hz, demonstrating its suitability for this specific use case.

5. Results And Discussion

This chapter presents the simulation outputs obtained after performing simulations on the Simscape model created in chapter 3 using the control architecture implemented in Simulink in chapter 4. The simulation outputs are then analyzed to observe changes based on different values used for the input variables.

5.1. Variables used for simulation

In order to better understand the simulation outputs, two main variables have been selected for the simulations,

5.1.1. Search Velocity

The first variable for the model simulations is the search/approach velocity of the bondhead. In literature, a number of different velocities have been tested in the application of wire bonding [16; 31]. These range from 5 mm s^{-1} to 20 mm s^{-1} . For the purpose of this thesis and under the guidance of experts from ASMPT, the search velocities that have been tested in the simulation are 5, 8, 10, 12, and 15 mm s^{-1} .

5.1.2. Contact Detection Time

The second variable for the simulations is contact detection confidence. As mentioned in section 4.2, contact detection is a very important aspect of the control architecture implementation. The detection of the time of contact also determines the changes in both the position and the force inputs. In the section for contact detection, it was mentioned that for this thesis, it was assumed that a contact detection algorithm was present that can confirm the contact detection in 1.25 ms after the initial impact occurs. A number of contact detection algorithms have been designed in the literature. Kim et al. have designed and tested a contact detection algorithm and a touch detection algorithm to achieve a contact detection time of 3.05 ms [15; 14]. Another contact detection algorithm proposed by Lee et al. [13] was able to achieve the contact detection time of 2.02 ms. For the purpose of testing the control architecture based on input from ASMPT, four different contact detection times were tested, 0, 1.25, 1.8, and 2.5 ms. The contact detection time of 0 ms was selected to showcase the scenario when the contact is detected instantaneously.

5.2. Simulation Results and Discussions

Figures 5.1 to 5.5 depict the simulation results obtained by varying the search velocity V_{search} while maintaining a constant contact detection time. In this scenario, the contact detection time is set to 1.25 ms.

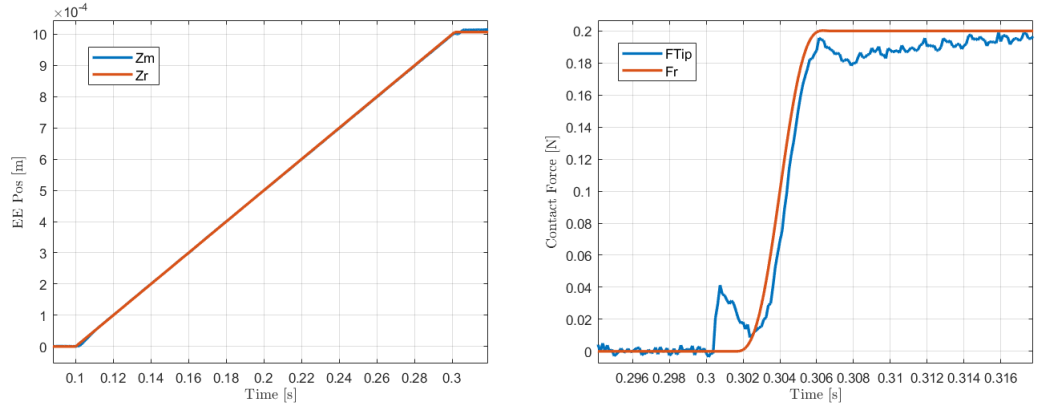


Figure 5.1.: $V_{search} = 5 \text{ mm s}^{-1}$ and Contact Detection time = 1.25 ms

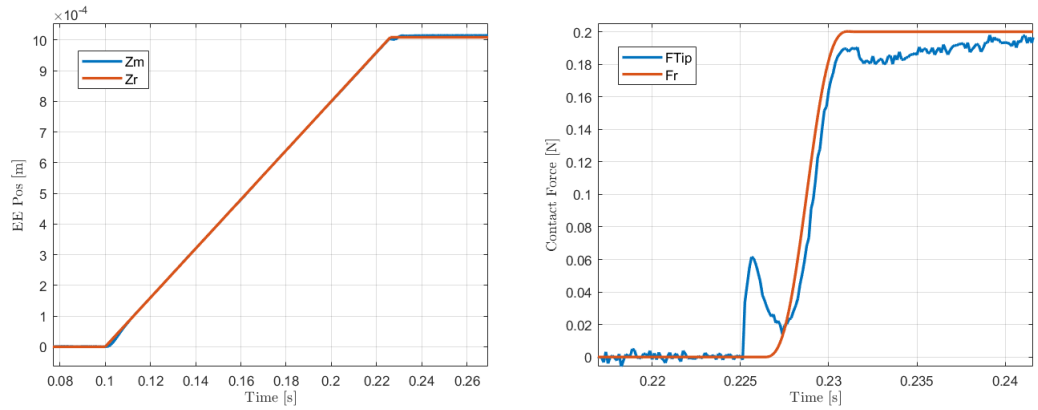


Figure 5.2.: $V_{search} = 8 \text{ mm s}^{-1}$ and Contact Detection time = 1.25 ms

The horizontal axis of these plots represents time in seconds. For the left plots, the vertical axis denotes the position of the bondhead POI in meters, while for the right plots, the vertical axis represents the contact force in Newtons.

From all the position plots shown on the left figures, it can be observed that changing the approach velocity does not affect the actual impact response of the position. All the plots show a similar initial bounce, and when the force setpoint activates, they maintain a constant position to sustain the force across the FAB spring and damper. At higher velocities, it can be observed that the position reference is slightly higher than the actual position and is positioned farther than the intermass distance. This discrepancy arises because, with an increase in velocity, the reference can increment more rapidly at the same contact detection

5.2. Simulation Results and Discussions

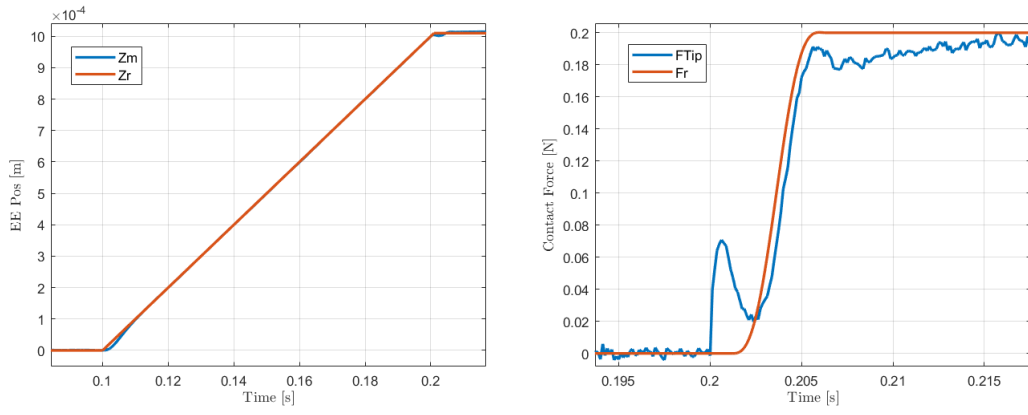


Figure 5.3.: $V_{search} = 10 \text{ mm s}^{-1}$ and Contact Detection time = 1.25 ms

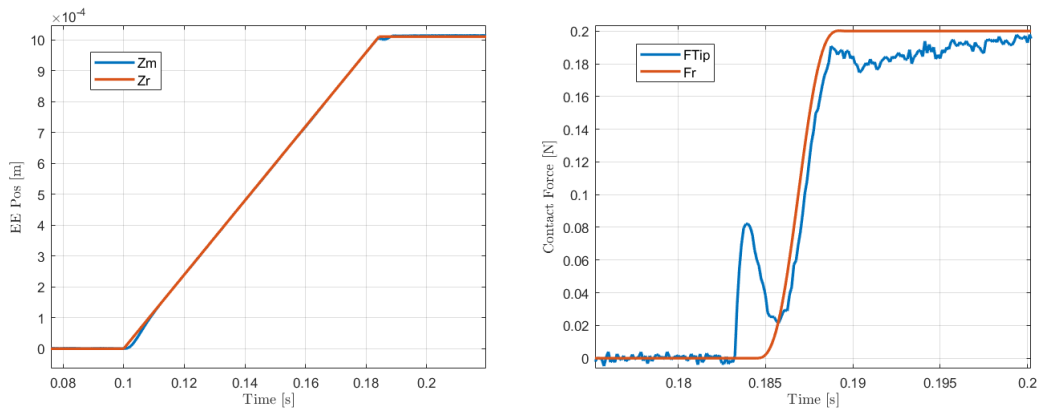


Figure 5.4.: $V_{search} = 12 \text{ mm s}^{-1}$ and Contact Detection time = 1.25 ms

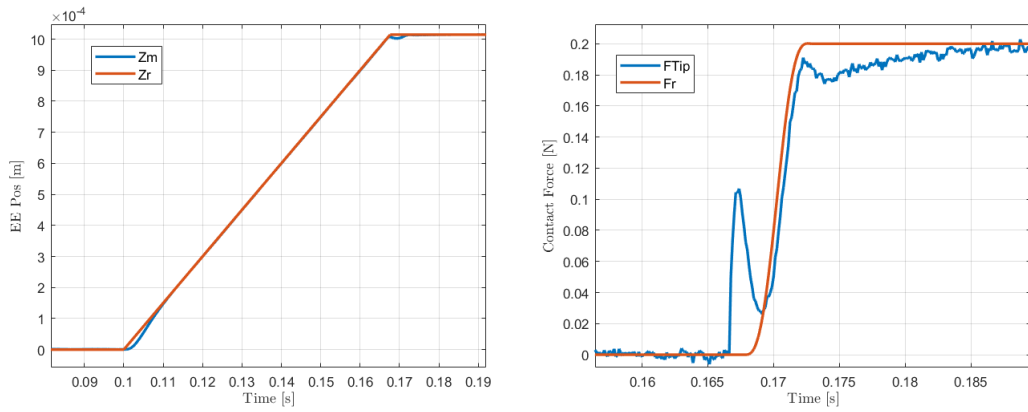


Figure 5.5.: $V_{search} = 15 \text{ mm s}^{-1}$ and Contact Detection time = 1.25 ms

5. Results And Discussion

time. As the position setpoint generator relies on contact detection to establish its final value, increased velocity allows it to cover more distance within the same number of samples.

The force plots on the right in figures 5.1 to 5.5 exhibit significant differences in the initial impact force with an increase in velocity, which is intuitive. The impact force increases from 0.04 N to 0.11 N as the velocity increases from 5 to 15 mm s⁻¹. However, this is still within a safe range, as it does not exceed the desired output force range. All plots demonstrate similar performance after the initial impact, with a settling time of 8 ms.

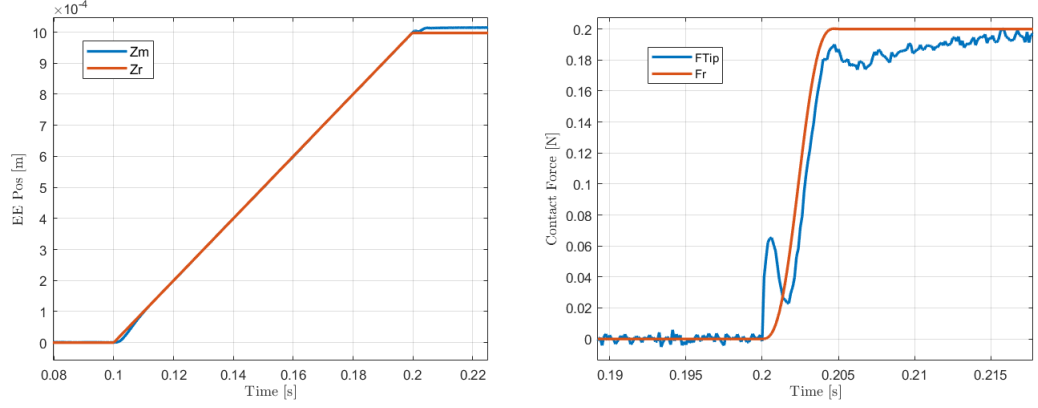


Figure 5.6.: $V_{search} = 10 \text{ mm s}^{-1}$ and Contact Detection time = 0 ms

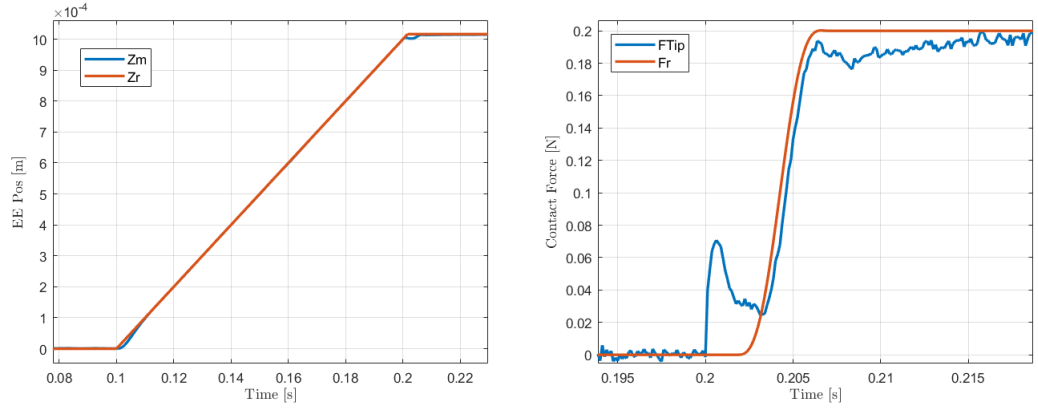


Figure 5.7.: $V_{search} = 10 \text{ mm s}^{-1}$ and Contact Detection time = 1.8 ms

Figures 5.6 to 5.8 present the simulation output when the contact detection time is varied while maintaining a constant search velocity. In this case, the velocity was held at 10 mm s⁻¹, and the contact detection varied from 0 to 2.5 ms.

Similar to the previous scenario, for the position plots, as the contact detection time increases, the position reference gains a larger magnitude for the same velocity. This is because it has more time before the contact signal can halt the position ramp. Nevertheless, the controllers can effectively manage this deviation in position reference and maintain contact.

From the force plots displayed on the right in figures 5.6 to 5.8, it is evident that the only

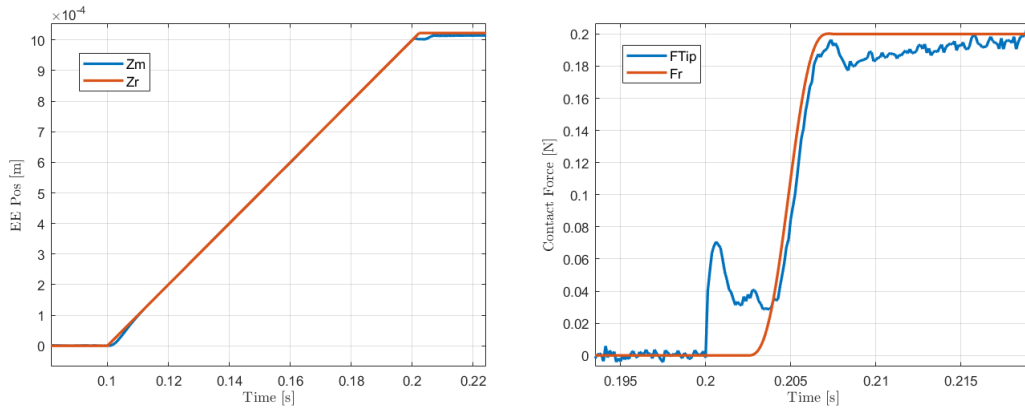


Figure 5.8.: $V_{search} = 10 \text{ mm s}^{-1}$ and Contact Detection time = 2.5 ms

difference is the timing of the smooth step setpoint initiation. Increasing the contact detection time delays the appearance of the setpoint. Consequently, small-magnitude oscillations occur in the output, as the force controller attempts to reach zero while contending with the position controller. The position controller aims to reach a setpoint beyond the intermass distance, thus generating a minor contact force on its own. This issue resolves once the force setpoint comes into play, and the force control reasserts priority, benefiting from its *PID* nature, which includes an integrator. All plots exhibit similar responses after the force smooth step trajectory initiation.

Fig. 5.6 illustrates the ideal scenarios where contact is detected instantaneously. The position plot indicates that the position reference halts precisely at the moment of contact, leading to a slight bounce in the measured position, as the position controller strives to minimize the error. Concurrently, the force controller activates and takes precedence.

These plots underscore the effectiveness of the parallel force control architecture when applied to the wire bonding process, effectively eliminating the force overshoot, even in the presence of stable yet suboptimal rule-of-thumb controllers. All the force plots exhibit a minor dip before reaching the final setpoint value. This can be potentially mitigated with the utilization of more refined controllers.

6. Conclusions and Future Recommendations

6.1. Conclusions

This thesis has successfully achieved the objectives discussed in Chapter 1, which focused on the design of a simulation model for the Z-axis bondhead-environment interaction of industrial wire bonders, as well as the implementation of a framework for simulating and testing different force and position controllers.

To fulfill these objectives, a comprehensive literature survey was conducted to examine various types of force and position controllers. These different methods and algorithms were subsequently evaluated based on critical criteria that were specifically selected for the particular use case of single DOF wire bonding.

A Simscape model representing a single mass, which simulates the bondhead in contact with the environment, was designed. Model parameters were meticulously chosen in accordance with the behavior of the wire bonding machine, under the guidance of ASMPT. The model's validity was confirmed through system identification techniques. Building on the understanding of the contact behavior of the model, an extended version of the model was developed in Simscape to include the initial impact and transition to the contact phase.

The Parallel Force Control architecture was successfully implemented in Simulink, allowing for simultaneous control of both the contact force across the FAB and the POI position of the bondhead. Setpoint generators were introduced to furnish reference positions and force inputs to the system, based on the timing of contact detection.

To enhance the model's real-world applicability and complexity, additional parameters, such as sensor delays and measurement noise, were incorporated into the system. This augmented model was then subjected to simulation and testing, utilizing stable rule-of-thumb position and force controllers, to assess the effectiveness of Parallel Force Control in regulating position and contact force during wire bonding.

By selecting a range of simulation variables across diverse scenarios and approach velocities, the simulation results clearly demonstrated that even with suboptimal rule-of-thumb controllers, force overshoot can be effectively mitigated. This achievement significantly minimizes the risk of damage to the substrate surface by the bondhead, thereby ensuring the reliability of the wire bonding process.

Overall, this thesis successfully addressed the research objectives and contributes valuable insights and methodologies to enhance the performance and control of wire bonding operations in the semiconductor industry.

6.2. Future Recommendations

To further enhance the simulation model's accuracy and its alignment with the actual wire bonding process, the following recommendations for future work are presented:

6.2.1. Modeling Refinement

In Chapter 3, an assumption was made that the dynamics of the bondhead with the wire bonding machine can be disregarded due to the compliance provided to the bondhead in the Z-direction. This simplified the bondhead's behavior before contact as that of a free mass. To refine the model further, an additional set of spring and damper elements could be introduced above the bondhead. This addition would better simulate the dynamics between the bondhead and the wire bonder frame.

6.2.2. Exploring Contact Detection Algorithms

This thesis assumed the availability of a contact detection algorithm to identify the initial impact during the bonding process, as discussed in Chapter 5. A range of contact detection algorithms have been explored in literature for wire bonding. In future work, these algorithms could be investigated and integrated into the simulations, increasing the model's complexity and aligning it more closely with real-world application scenarios.

6.2.3. Model Discretization

The existing simulation model employs continuous-time controllers. However, in real-world scenarios, sensor data reading and control algorithm execution occur in the discrete-time domain. To bridge the gap between simulation and practical application, it's important to incorporate discretization of the controller and introduce additional discretization delays. This adjustment would make the simulation more representative of the real-world usage scenario, where the continuous plant interacts with discrete-time sensors and control algorithms.

6.2.4. Experimental FRF Data

Lastly, the position and force FRF utilized during modeling and simulation were derived from an approximated dynamic model of the system. To achieve a more accurate system response, obtaining the actual FRF from a physical wire bonding setup would be valuable. This approach would facilitate the development of more precise controllers, allowing for the assessment of their real-world impact before moving on to practical experimentation and eventual implementation on an actual wire bonding machine.

A. Appendix A - Single Mass Contact

Model implementation and validation

This appendix presents the Simscape implementation as well as the system identification of the single mass contact model as discussed in chapter 3.

A.0.1. Single mass contact model representation in Simscape

The single mass contact model was implemented in Simscape Multibody, as depicted in figure A.1. The system comprises two key masses: Mass m_A , representing the Bondhead mass, and the surface mass, signifying the substrate engaged during the wire bonding process. This surface mass is attached directly to the world frame as shown in Fig. 3.1. The mass value for the surface block is inconsequential as it is directly connected to the world frame and has been provided to aid the visual interpretation of the model as seen in the graphical output of the simulation. Not including the surface block would result in the simulation output just showcasing a single mass in free space with no reference to where the environment is located with respect to the bondhead.

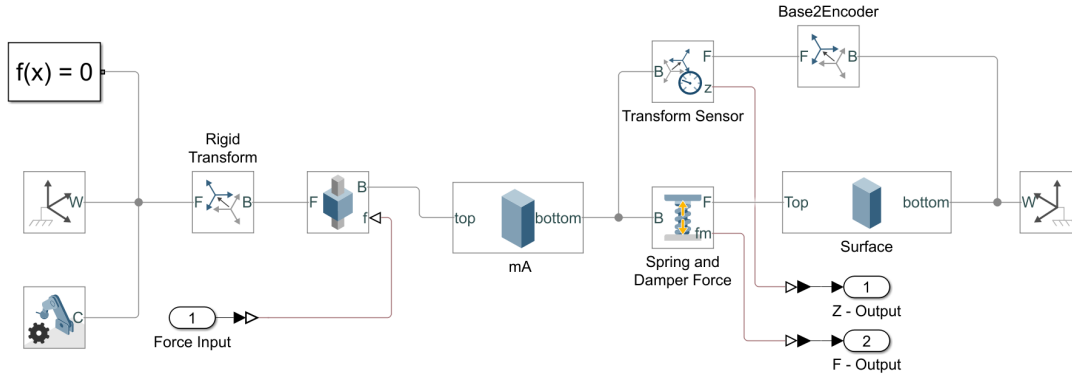


Figure A.1.: Single mass contact model in Simscape.

The interplay between these two masses in contact is portrayed by the spring and damper force block within Simscape which is connected to both the masses as visible from Fig. A.1. This block emulates the spring coefficient k_e and the damper coefficient c_e . The natural length of the spring is set to match the distance between the two masses, preventing the generation of extra forces along the spring. If the inter-mass distance is smaller than the spring's natural length, an additional outward force is exerted on the block upon simulation initiation. Conversely, an inward force is applied if the inter-mass distance exceeds the natural length. Although the intermass distance's value is inconsequential for subsequent system identification, it plays a pivotal role in the impact transition model.

The upper segment of the mass m_A is connected to a prismatic joint, designed without stiffness or damping, and is used for providing the model with the input force F_r as shown in Fig. 3.1. This module serves as the system's input force provider, as depicted in Fig. A.1. As the surface mass is connected to the world frame, a rigid transform block is used to define the distance of the bondhead from the world frame shown on the left-hand side of Fig. A.1.

A combination of two blocks is utilized to measure the output displacement Z of the bondhead POI. These are the transform sensor and another rigid transform block which was named (Base2Encoder). The rigid transform block is used to define the distance of the transform sensor from the world frame while the transform sensor block ensures the accurate capture of the actual displacement of m_A relative to the world frame.

Two outputs are garnered from the system: firstly, the displacement of the POI of bondhead mass m_A , and secondly, the contact force across the spring and damper force block.

A.1. System Identification

The model developed in Simscape Multibody can be validated through the process of system identification. This section outlines the setup employed for identifying and validating the system.

A.1.1. Transfer functions for single mass contact model

$$\frac{Z(s)}{F_r} = \frac{1}{m_A s^2 + c_e s + k_e} \quad (\text{A.1})$$

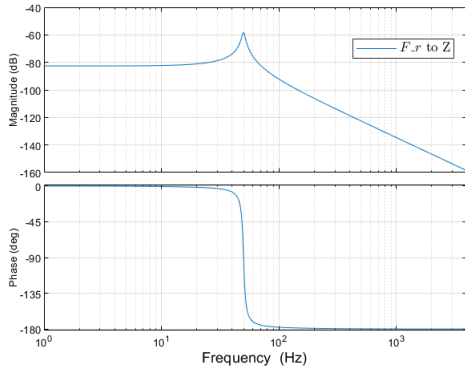
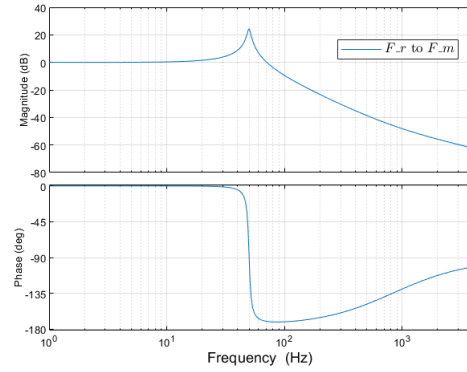
$$\frac{F_m}{Z(s)} = c_e s + k_e \quad (\text{A.2})$$

$$\frac{F_m}{F_r} = \frac{c_e s + k_e}{m_A s^2 + c_e s + k_e} \quad (\text{A.3})$$

(A.1), (A.2) and (A.3) represent the three transfer functions for the single mass contact model. (A.1) expresses the transfer function from the input force to the output displacement of the system, where $Z(s)$ signifies the output displacement, and the input reference force is denoted as F_r .

Similarly, (A.2) represents the transfer function from the output displacement to the measured contact force across the spring and damper, with F_m representing the measured force.

To obtain the transfer function from the input reference force to the output measured force, you can multiply (A.1) and (A.2), effectively eliminating the variable $Z(s)$ and yielding (A.3) which gives the transfer function from input force to the output measured contact force. The bode plots for (A.1) and (A.3) are shown in figures A.2 and A.3 respectively.


Figure A.2.: FRF F_r to Z .

Figure A.3.: FRF F_r to F_m .

A.1.2. Simulink and Matlab Setup for System Identification

Figure A.4 presents the schematic depiction of the system identification process. The white noise input enters the plant as F_r . The plant yields two outputs: Z and F_m . Following the simulation, the data extracted from these three variables is transmitted to the MATLAB workspace for system identification.

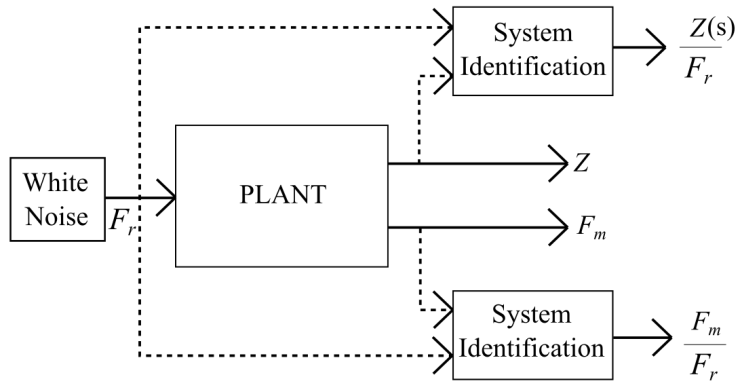


Figure A.4.: System identification schematic diagram.

Fig. A.5 illustrates the Simulink implementation of the system identification process. In the center of the figure, the single mass contact model functions as the plant for the identification procedure. As previously discussed in the preceding section, the input pertains to the reference force, while the system yields both bondhead displacement and measured contact force as outputs.

Within simulink, band-limited white noise was provided as an input to the system. For conducting system identification, the 'tfestimate' function from MATLAB was employed to estimate the transfer function from the input signal, F_r , to the output signals, Z and F_m . To ensure accurate and meaningful results, specific settings were configured during the estimation process.

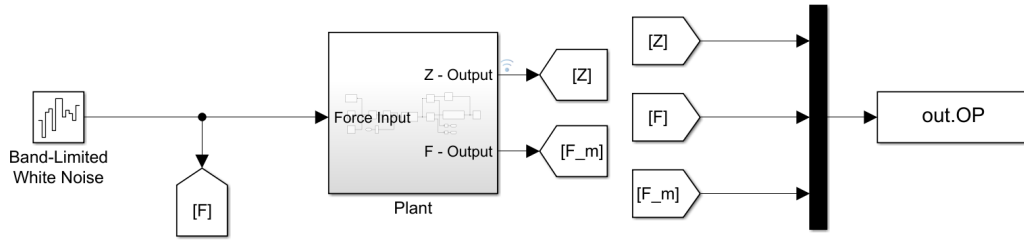


Figure A.5.: Simulink implementation of system identification.

To prepare the input signals for analysis, a Hann window was applied. This windowing technique tapered the data points at the signal edges, mitigating the potential for spectral leakage during subsequent frequency analysis. The utilization of the Hann window aimed to enhance the precision of the transfer function estimation.

In configuring the 'tfestimate' function, several key parameters were adjusted to achieve optimal results. The segment length, overlap, and frequency resolution were crucial aspects of the analysis. The segment length, set at approximately one-tenth of the signal's total length, aimed to balance the need for capturing detailed segment information with the necessity for robust averaging to yield reliable transfer function estimates. A 50% overlap between segments was chosen to ensure smooth transitions, reducing abrupt shifts that could lead to distorted outcomes.

Additionally, the frequency resolution was controlled by determining the length of the Fast Fourier Transform (FFT) window. Opting for a window length equivalent to 10% of the total signal length struck a balance between capturing frequency components in detail and maintaining computational efficiency.

A.1.3. System Identification Output

Figure A.6 displays the FRF from input force (F_r) to output displacement (Z), comparing the analytical results to those obtained from system identification. Similarly, Figure A.7 illustrates the FRF from input force to measured force (F_m). The orange curves denote the FRF acquired analytically, while the blue curves represent those deduced through system identification.

It is evident from the figures that the blue and orange curves closely parallel each other, diverging only at higher frequencies. This behavior can be attributed to the phase lag that comes into play at these higher frequencies. The phase lag phenomenon is a result of the implicit discretization and zero-order hold applied during the simulation in Simulink. These techniques inherently introduce delays in the system's response, particularly pronounced at higher frequencies. Consequently, this leads to the observed slight divergence between the analytical and identified FRFs at these frequencies. As the identified FRF is able to fully capture the analytical behavior before and at the resonance peaks which are of main concern as the stiffness and damping values govern the dynamics of impact, the created Simscape model can be considered a valid representation of the aforementioned single mass-environment contact scenario.

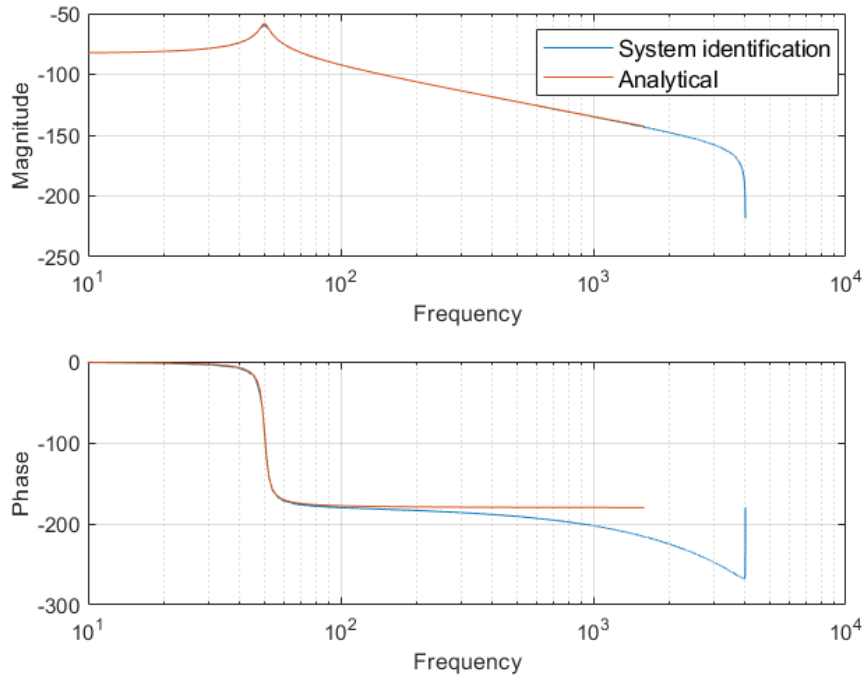


Figure A.6.: FRF F_r to Z - Analytical and System identification output.

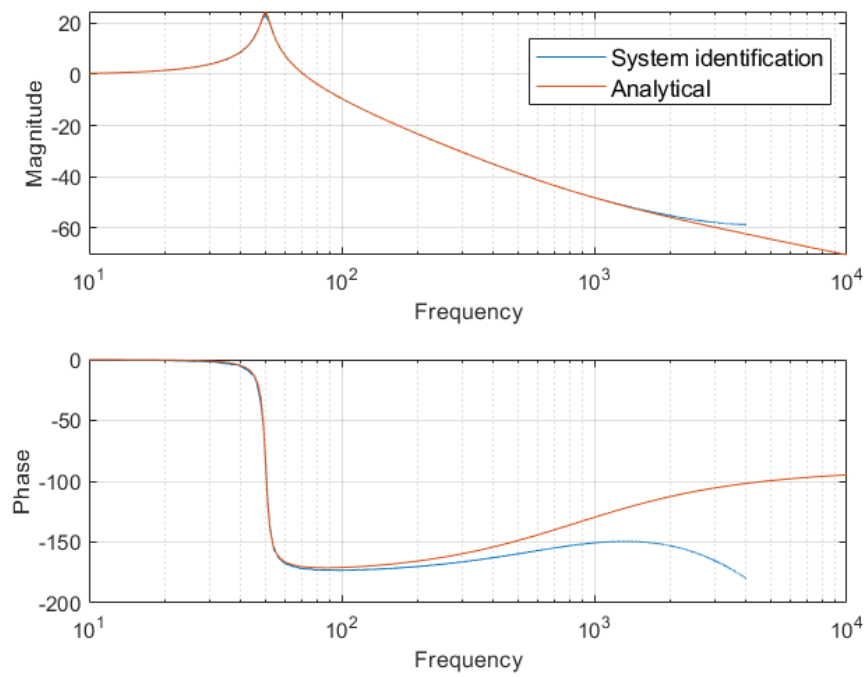


Figure A.7.: FRF F_r to F_m - Analytical and System identification output.

B. Appendix B - MATLAB Code - Position Setpoint Generation

This appendix presents the MATLAB code responsible for generating the initial position setpoint and for transmitting it to Simulink.

```
% Simulation Variables
V_A = 10e-3; % [m/s] search velocity of bondhead

% Position Setpoint Generation and transfer to Simulink

% Define the increment and number of values
increment = Ts; % Sampling Time
numValues = Tsim/Ts+1; % Tsim is the total simulation time

% Generate the T array
Time_ramp = 0:increment:increment*(numValues-1);

% Generate the Z array with holdValue
Z_input = zeros(size(Time_ramp)); % Initialize Z_input with zeros

% Find the index where the ramp should start (0.1 seconds)
start_index = find(Time_ramp >= 0.1, 1);

% Assign the ramp values to Z_input
Z_input(start_index:end) = (Time_ramp(start_index:end) - 0.1) * V_A;

figure();
plot(Time_ramp, Z_input, 'b', 'LineWidth', 2)
grid on;
xlim([0 Tsim]);
title('Initial Position setpoint profile');

Z_I.time = Time_ramp';
Z_I.signals.values = Z_input';
Z_I.signals.dimensions= 1;
```


C. Appendix C - Contact Time Detection

This appendix presents the MATLAB function code for calculating the contact detection time. The inputs to the function are the contact signal and the Simulink clock, and the output is the time at which the contact detection signal transitions from 0 to 1.

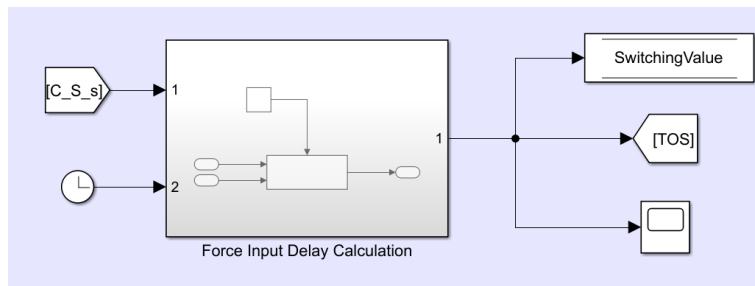


Figure C.1.: Local variable - Contact time detection.

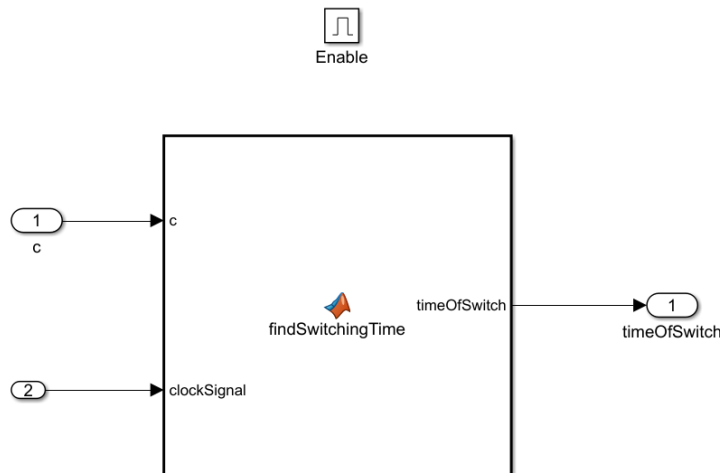


Figure C.2.: Enabled subsystem - Contact time detection.

Figures C.1 and C.2 illustrate the key blocks necessary for calculating the contact detection time. In Figure C.1, an internal subsystem is depicted, encompassing two inputs and one output. The inputs are the contact detection signal and a Simulink clock, with the output playing a pivotal role in force setpoint generation.

C. Appendix C - Contact Time Detection

Within this subsystem, an enabled subsystem, depicted in Fig. C.2, takes the same inputs and produces identical outputs as the external subsystem. The distinguishing feature is the enclosed MATLAB function and the presence of an "Enable" block at the top of the diagram. This block, with a value of 1, ensures that the subsystem executes the MATLAB function during each sample period.

Inside this function, a persistent variable 'switchTime' is initialized to 0. During the first sample, when the subsystem runs, 'switchTime' is established with this initial value. During subsequent samples, the code continuously checks the contact detection variable CS_s . When CS_s becomes 1 and 'switchTime' remains 0, 'switchTime' assumes the value of the Simulink clock. Moreover, this value is transmitted to another block containing the local variable 'SwitchingValue'. This 'Data store memory - Write' block generates an in-simulation variable, storing the contact detection time value.

The following is the MATLAB function code for the contact detection time calculation:

```
function timeOfSwitch = findSwitchingTime(c, clockSignal)
    % c: Single value representing contact (0 or 1)
    % clockSignal: Clock signal from the Simulink model (
        simulation time)

    % Check if the input c is a valid value (0 or 1)
    if ~ismember(c, [0, 1])
        error('The input c must be either 0 or 1.');
```

```
    end

    % Persistent variable to keep track of the time of the
        switch
    persistent switchTime;

    % Check if the persistent variable has been initialized
    if isempty(switchTime)
        % If not initialized, set it to -1 to indicate no switch
            yet
        switchTime = 0;
    end

    % Check if the contact occurred (i.e., c changes from 0 to
        1)
    if c == 1 && switchTime == 0
        % If the contact has occurred and the switch hasn't been
            detected yet,
        % set the time of the switch using the clock signal
        switchTime = clockSignal;
        Simulink.Bus.createMATLABStruct = switchTime;
    end

    % Set the output to the time of the switch (0.2011) if the
        switch has occurred, else -1
    timeOfSwitch = switchTime;
end
```

Bibliography

- [1] R. Anderson and M. Spong. Hybrid impedance control of robotic manipulators. *IEEE Journal on Robotics and Automation*, 4(5):549–556, 1988.
- [2] S. Chiaverini and L. Sciavicco. The parallel approach to force/position control of robotic manipulators. *IEEE Transactions on Robotics and Automation*, 9(4):361–373, 1993.
- [3] S. Chiaverini, B. Siciliano, and L. Villani. A survey of robot interaction control schemes with experimental comparison. *IEEE/ASME Transactions on Mechatronics*, 4(3):273–285, 1999.
- [4] J. De Schutter, H. Bruyninckx, W.-H. Zhu, and M. W. Spong. Force control: A bird’s eye view. In B. Siciliano and K. P. Valavanis, editors, *Control Problems in Robotics and Automation*, pages 1–17, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg.
- [5] D. M. Gorinevsky, A. M. Formal'sky, and A. Y. Schneider. *Force control of robotics systems*. Boca Raton, FL: CRC Press, 1997.
- [6] N. Hogan. Impedance control - an approach to manipulation. i - theory. ii - implementation. iii - applications. *ASME Journal of Dynamic Systems and Measurement Control B*, 107:1–24, Mar. 1985.
- [7] N. Hogan. Stable execution of contact tasks using impedance control. In *Proceedings. 1987 IEEE International Conference on Robotics and Automation*, volume 4, pages 1047–1054, 1987.
- [8] M. Hosseinzadeh, P. Aghabalaie, H. A. Talebi, and M. Shafie. Adaptive hybrid impedance control of robotic manipulators. In *IECON 2010 - 36th Annual Conference on IEEE Industrial Electronics Society*, pages 1442–1446, 2010.
- [9] D.-J. Jeong and S. Jung. Comparison studies of two major force control algorithms for a single axis force control of a robot manipulator. In *2021 21st International Conference on Control, Automation and Systems (ICCAS)*, pages 2243–2246, 2021.
- [10] S. Jung and J. W. Lee. Hybrid impedance force tracking control of a robot manipulator under non-model dynamics and nonlinear uncertainties. In *2022 13th Asian Control Conference (ASCC)*, pages 1108–1113, 2022.
- [11] R. Kelly, R. Carelli, M. Amestegui, and R. Ortega. On adaptive impedance control of robot manipulators. In *Proceedings, 1989 International Conference on Robotics and Automation*, pages 572–577 vol.1, 1989.
- [12] O. Khatib and B. Siciliano. *Springer handbook of robotics*. Springer International Publishing, 2016.
- [13] J.-H. Kim, J.-H. Cho, and S.-K. Lee. A design of hybrid contact detection algorithm for wire bonder machine. In *IECON'03. 29th Annual Conference of the IEEE Industrial Electronics Society*, volume 3, pages 2963–2967. IEEE, 2003.

- [14] J.-H. Kim and J.-H. Oh. A design of touch detection algorithm for bonding tip of wire bonder machine. In *IECON'01. 27th Annual Conference of the IEEE Industrial Electronics Society*, volume 1, pages 621–625. IEEE, 2001.
- [15] J.-H. Kim and H.-J. Park. A contact detection algorithm of the z-axis of a wire bonder. *Control Engineering Practice*, 14(9):1035–1043, 2006.
- [16] J.-H. Kim and C.-H. Yim. An impact force compensation algorithm based on a piezo force sensor for wire bonding processes. *Control Engineering Practice*, 16(6):685–696, 2008. Special Section on Large Scale Systems.
- [17] J. Li, L. Liu, Y. Wang, and W. Liang. Adaptive hybrid impedance control of robot manipulators with robustness against environment's uncertainties. In *2015 IEEE International Conference on Mechatronics and Automation (ICMA)*, pages 1846–1851, 2015.
- [18] D. Liu, Y. Chao, and C. Wang. Study of wire bonding looping formation in the electronic packaging process using the three dimensional finite element method. *Finite Elements in Analysis and Design*, 40:263–286, 2004.
- [19] L. Love and W. Book. Environment estimation for enhanced impedance control. In *Proceedings of 1995 IEEE International Conference on Robotics and Automation*, pages 1854–1859, 1995.
- [20] S. K. Prasad. *Advanced Wirebond Interconnection Technology*. Springer New York, NY, 2004.
- [21] H. Qian and J. De Schutter. Introducing active linear and nonlinear damping to enable stable high gain force control in case of stiff contact. In *Proceedings 1992 IEEE International Conference on Robotics and Automation*, pages 1374–1379 vol.2, 1992.
- [22] D. T. Rooney, L. Gullo, D. Xie, N. T. Castello, and D. Shangguan. Evaluation of reliability and metallurgical integrity of wire bonds and lead free solder joints on flexible printed circuit board sample modules. *Microelectronics Reliability*, 47(12):2152–2160, 2007.
- [23] R. Schaller. Moore's law: past, present and future. *IEEE Spectrum*, 34:52–59, 1997.
- [24] R. M. Schmidt, G. Schitter, and A. Rankers. *The design of high performance mechatronics: high-Tech functionality by multidisciplinary system integration*. Ios Press, 2020.
- [25] R. Volpe and P. Khosla. A theoretical and experimental investigation of explicit force control strategies for manipulators. *IEEE Transactions on Automatic Control*, 38(11):1634–1650, 1993.
- [26] R. Volpe and P. Khosla. The equivalence of second-order impedance control and proportional gain explicit force control. *The International journal of robotics research*, 14(6):574–589, 1995.
- [27] R. Volpe and P. K. Khosla. A theoretical and experimental investigation of impact control for manipulators. *Int. J. Robotics Res.*, 12(4):351–365, 1993.
- [28] Y. Wu, T.-J. Tarn, and N. Xi. Force and transition control with environmental uncertainties. In *Proceedings of 1995 IEEE International Conference on Robotics and Automation*, pages 899–904, 1995.

- [29] Y. Xu, D. Ma, and J. Hollerbach. Nonlinear proportional and derivative control for high disturbance rejection and high gain force control. In *[1993] Proceedings IEEE International Conference on Robotics and Automation*, pages 752–759 vol.1, 1993.
- [30] K. Youcef-Toumi and D. A. Gutz. Impact and force control. In *Proceedings of the 1989 IEEE International Conference on Robotics and Automation, Scottsdale, Arizona, USA, May 14-19, 1989*, pages 410–416. IEEE Computer Society, 1989.
- [31] L. Zhou, J. Li, L. Fu, and Z. Li. Application of a force sensor in wire bonding process. *Sensors & Transducers*, 159(11):345, 2013.

