

**Integrated Trajectory Planning
and Vibration Suppression of Transfer Robots**

January, 2015

DOCTOR OF ENGINEERING

WISNU ARIBOWO

Toyohashi University of Technology

Integrated Trajectory Planning and Vibration Suppression of Transfer Robots

Wisnu Aribowo

*Department of Electronic and Information Engineering,
Toyohashi University of Technology
May, 2015*

ABSTRACT

Automatic transfer systems are utilized to a great extent in various type of modern industry. Such systems can transfer industrial work pieces in high speed with relatively consistent performance. The cycle time reduction gained by utilization of such systems can increase throughput of the factory. However, high speed operation often results in residual vibration as side effect, usually considered undesirable in many industrial applications that require high precision such as semiconductor wafer pick-and-place system, where a multi-link robot arm is used to automatically pick up, transfer, and place semiconductor wafers. Vibration of the arm tip that occurs after the robot motion causes the robot to wait for a moment until the vibration settles to an acceptable level before performing next operation. This settling time should be reduced as much as possible in order to increase production rate in semiconductor factory.

Other type of vibration problem caused by motion of the robot arm is sloshing, or the vibration of liquid surface. Sloshing is not caused by the vibration of the mechanical structure of the robot. Instead, it is caused by the motion in the task space of the container held by the robot arm. Sloshing may cause the liquid to spill out of the container, and in the case of molten metal too much sloshing may affect the physical properties of the liquid. Moreover, other task, e.g. pouring, cannot be performed when residual sloshing still occurs, thereby potentially reducing the system productivity.

The main challenge in this kind of problem is to generate trajectories with concurrent consideration of motion time minimization and vibration (or sloshing) suppression for point

to point motion in a three dimensional working space where static obstacles may exist. Minimizing motion time means that the trajectory should be designed in the joint space, where the main constraints of robot motion exist, for example the kinematic and dynamic constraints of the joint actuators. On the other hand, the existence of obstacle or working boundary of the robot means that the solution has to consider the task space constraints as well. In addition, the vibration suppression should not increase the motion time too much.

This research proposes an integrated framework of trajectory planning and vibration suppression of multi joint robot arm to generate trajectories with quick motion time and low vibration. The framework uses two building blocks: cubic spline trajectory and input shaping. The cubic splines trajectory has a few favorable properties, for example the limited jerk, which provides smoother profile than acceleration-limited trajectories, and the quick computation. Nonetheless, it is relatively fast compared to other higher degree polynomials. A cubic spline trajectory is composed of several curve segments, each is represented by one cubic function. The segments are bounded by knots. The shape of each cubic curve is determined by four parameters of the cubic function. The trajectory has to be continuous in position, velocity, and acceleration throughout the whole time. By simplification, a system of linear equations involving the knot values, the segment times, and the accelerations can be obtained.

Input shaping method is a simple yet effective method to reduce system vibration. Using value of natural frequency and damping ratio of the vibrating system, the command input signal is decomposed into two. The magnitude and time between those two signals are set such that the second signal can compensate vibration coming from the first signal, thus effectively eliminate subsequent vibration. The input shaping has found practice in many real applications due to the combination of its simplicity and good performance. From the basic principle of input shaping, researchers have come with numerous developments and modifications, resulting in many types of input shapers for different kind of applications.

The framework is formulated as a non linear optimization model with motion time as the main objective function. The constraints consist of the joint constraints (e.g. velocity and torque limit) and task space constraints (e.g. working boundary and obstacle avoidance). The decision variables are both the segment times and the knot values. This gives the

framework enough flexibility to achieve quick motion time and to avoid obstacles at the same time.

The integrated framework is applied for two practical cases: a semiconductor wafer transfer robot arm and a liquid container transfer robot arm. In the first case, a three-link planar robot arm works in small bounded working space. In some area, the working space is so small such that the motion of the last link is constrained to a straight line only. In the second case, the system uses a seven degree-of-freedom robot arm that works in three dimensional space, where obstacle may also exists.

Through simulation and experiments, the effectiveness of the proposed solution in generating quick motion while keeping vibration (or sloshing) low is demonstrated. With only small increase in motion time, the vibration can be suppressed to a substantially smaller amount. The calculation time of trajectory generation without vibration consideration is quick. If integrated with vibration suppression, the calculation takes somewhat longer time, but is still acceptable for offline calculation.

Table of Contents

CHAPTER 1 Introduction	1
1.1 Research Background.....	1
1.2 Review of Past Researches.....	3
1.2.1 Trajectory Planning.....	3
1.2.2 Vibration Control	5
1.2.3 Sloshing Control	7
1.3 Research Objectives	8
1.4 Thesis Outline	9
CHAPTER 2 Experimental Devices and System Construction	11
2.1 Introduction	11
2.2 Semiconductor Wafer Transfer Robot Arm.....	12
2.3 Liquid Container Transfer Robot Arm	16
2.4 Summary	20
CHAPTER 3 Cubic Spline Optimization for Trajectory Generation of Multi-Joint Robot Arm	21
3.1 Introduction	21
3.2 Basic Principle of Cubic Spline Trajectory	22
3.3 Cubic Spline Trajectory for Multi Joint Robot Arm	24
3.4 Summary	32
CHAPTER 4 Vibration Suppression of Semiconductor Wafer Transfer Robot Arm	35
4.1 Introduction	35
4.2 Principles of Input Shaping	36
4.3 Identification of Multi-Mode Vibration Parameters.....	39
4.4 Implementation of Multi-Mode Input Shaping	45
4.4.1 Implementation procedure	45
4.4.2 Experiment.....	46
4.5 Integration of Trajectory Planning and Vibration Control	50
4.5.1 Optimization Model.....	50
4.5.2 Experiment Results and Discussion.....	53
4.6 Summary	56

CHAPTER 5 Trajectory Generation and Sloshing Suppression of Liquid Container Transfer Robot Arm	59
5.1 Introduction	59
5.2 Sloshing Model of Cylindrical Container	60
5.2.1 Theoretical Model.....	60
5.2.2 Equivalent Pendulum Model.....	60
5.3 Trajectory Planning and Sloshing Suppression.....	62
5.3.1 Translation Motion Planning	64
5.3.2 Rotation Motion Planning.....	70
5.3.3 Combining the Translation and Rotation Motion	71
5.4 Numerical simulation of sloshing	73
5.5 Experiment Results and Discussion	76
5.6 Summary	79
CHAPTER 6 Concluding Remarks and Recommendations for Future Work	81
6.1 Concluding Remarks	81
6.2 Recommendations for Future Work	82
References	85
Acknowledgments	92
List of Publications	94

CHAPTER 1

Introduction

1.1 Research Background

Robots are utilized to a great extent in our modern society. Historically, robots have been predominantly utilized to do industrial tasks in factories. Nowadays, they are also increasingly used in places that used to be foreign to them, such as hospital, home and office, and outdoor areas. Over time, research and developments in robotics and related fields have made them faster, smarter, and more dexterous. Robots come in various shapes and sizes, and take over many human tasks, ranging from simple repetitive works until highly difficult tasks that cannot be done otherwise. In the future, robots will become even more ubiquitous and dwell in every corner of our life domain.

In industrial factories, robots are important components of production automation for handling work pieces and performing other value-added operations. One of the common tasks is to convey work pieces from and to work centers and storage facilities automatically. The automatic transfer systems are used in various types of modern industry, most commonly in mass production factories. Such systems can transfer industrial work pieces in high speed with relatively consistent performance. The cycle time reduction gained by utilization of such systems are expected to increase the throughput of the factory.

However, high speed transfer often results in residual vibration as side effect, which is usually considered undesirable in many industrial applications that require high precision, such as semiconductor wafer pick-and-place system. In such system, a multi-link robot arm is used to pick up, transfer, and place semiconductor wafers automatically. Vibration of arm tip that occurs after movement necessitates the robot to wait for a moment until the vibration settles to an acceptable level before performing next operations. This settling time should be reduced as much as possible in order to increase the production rate in semiconductor factories.

Another type of vibration problem caused by the motion of robot arm is sloshing, or the vibration of liquid surface, in a liquid container transfer system. This system uses robot arm to transfer liquid from and to arbitrary locations in its three dimensional work space. Some example application cases for this setting are transfer of molten metal in casting industries and liquid containers handling by service robots. The sloshing is not caused by the vibration of the mechanical structure of the robot, but instead, it is caused by the motion in the task space of the container held by the robot arm. If the container moves faster, the amount of sloshing tends to be higher. Sloshing may cause the liquid to spill out of the container, and in the case of molten metal too much sloshing may affect the physical properties of the liquid. Moreover, other task, e.g. pouring, cannot be performed when residual sloshing still occurs, thereby potentially reducing the system productivity.

The two conflicting objectives – high speed and low vibration – are often considered independently, or at least consecutively. In the trajectory planning stage of a point to point motion, most of the times the main objective is to minimize the motion time with no explicit consideration of vibration suppression. On the other hand, in vibration control, of course the main goal is to suppress the vibration. However, often the action of controlling the vibration results in smoother trajectory profile, and as a result the motion becomes slower. Therefore, this suggests the need to deal with the two objectives of high speed and low vibration concurrently.

This research proposes an integrated framework of trajectory planning and vibration suppression of multi joint robot arm to generate trajectories with quick motion time and low vibration. The motion type of concern is point-to-point motion. In this motion type, the

robot may take any path between predefined start point and end point as long as the motion satisfies existing constraints. The constraints include intrinsic constraints of the robot joints (e.g. torque constraint and velocity constraint) and task space constraints (e.g. task space boundaries and obstacle avoidance).

1.2 Review of Past Researches

There are two main subsystems in the framework to be developed: trajectory planning and vibration control. Therefore, in this section, we outline review of past researches which are related to those two fields. Moreover, although sloshing is closely related with vibration, its models and control techniques may be specific, and thus it is discussed in its own subsection.

1.2.1 Trajectory Planning

The speed of a robot arm in a point-to-point motion is mainly constrained by its kinematic and dynamic limit of the joints. A robot joint moves quickest when it continuously utilize its critical constraint, resulting in bang-bang type of trajectory profile. For example, in the trapezoidal velocity profile, the acceleration at any time is either maximum, minimum, or zero. The jerk limited profile or S-curve is one order higher, where the jerk is either maximum, minimum, or zero. The generation method of the profiles is either procedural [1, 2] or by using FIR filters [3, 4, 5, 6]. However, such profiles may become difficult to calculate when there are many joints to consider and also multiple non trivial constraints exist in the system, for example torque constraints and obstacle avoidance.

Planning the trajectory of multi joint robot arm is a classic case of optimal control [7] with the joints angle and its derivatives as the states. Besides the derivative constraints, other constraints may also exists, such as torque constraints and task space constraints. There are numerous ways to solve the optimal control problem. One of them is to discretize the state and control variables and then pose the problem as a general finite non linear optimization problem, which in turn can be solved by using any of the numerous available

non linear optimization techniques [8, 9, 10].

However, in general, solving the optimal control problem directly is tough. The problem and the initial solution have to be carefully constructed so as to allow the solver to get to the proximity of the global optimum without being stuck to the many local optima. In order to reduce the optimization variables, and hence shorten the calculation time, the trajectory can be discretized into relatively few segments and then parameterized the segment in some known form or structure, such as polynomial functions [11, 12, 13], trigonometric functions [14], or B-spline function [15]. This way, the optimization would deal with relatively few variables when compared with the full discretization. For example, in case of polynomial functions, the variables would be the parameters of the respective polynomial function: a cubic function has four parameters, a quintic function has six parameters, and so on.

Of the many possible polynomial functions, the cubic (3rd order polynomial) function is particularly attractive because it is the lowest polynomial function that allows continuous acceleration in a piecewise polynomial trajectory. In a cubic trajectory, the velocity is quadratic, the acceleration is linear, and the jerk is constant. The limited jerk helps in reducing the residual vibration as well as long-term wear of the actuator structure. Higher order polynomials would generate even smoother trajectories, but they are slower in motion time. On the other hand, lower order polynomial would generate faster motion, but at the same time does not make jerk limiting easy. By using the cubic function as the parameterization, a C^2 smooth trajectory with fast motion time can be achieved.

Numerous cubic spline algorithms and optimization schemes have been proposed [12, 16, 17, 18, 19, 20, 21]. Most of them try to optimize either the motion time or the location of knots, depending on the application needs. Some works proposed time minimization by explicitly including the segment time in the objective function and leaving the knots location fixed. An early example is Lin et al. [12]. On the other hand, other works let the segment times fixed, and instead use the knots location as the optimization variables. For instance, Kolter and Ng [19] developed an approach to optimize the location of waypoints for obstacle avoidance. Similarly, Demeulenaere et al. [20, 21] proposed a convex programming framework with the knots location as the optimization variables.

Researches on cubic spline that addressed concurrent minimization of time and knots

location are relatively few, for example the work of Chettibi et al. [18]. This work is probably the closest to the objective of our current research, in that the trajectory is parameterized as cubic splines and it considered time minimization while allowing the knots location to be free (not fixed). However, the work did not consider vibration suppression and obstacle avoidance in the task space.

1.2.2 Vibration Control

The limited jerk of cubic spline trajectory prevents wear out of the structure and reduces the response of the system flexible modes [22]. It induces less vibration, even without explicit knowledge of the system flexible modes. For applications that require high vibration suppression, however, limited jerk alone may be not enough. There are other approaches to further reduce the vibration by utilizing knowledge of the flexible modes.

One way to reduce mechanical vibration of the robot is by physical modification of the robot structure, i.e. changing either mass, stiffness, or damping property of the system [23]. This method is known as passive vibration control, because vibration reduction is achieved by introducing passive elements to the system.

On the other hand, active vibration control method exists, in which some sort of control technique is used to reduce the vibration [24, 25]. The close loop feedback approach uses measurement value of system vibration in real time, thus it is inherently robust toward disturbances and inaccuracy in parameter estimation. The drawback is that sensors have to be always installed while the system is operational, which can be costly and difficult to apply in some practical use. Open loop approach, on the other hand, does not need sensors to give feedback to controller. Instead, it works by altering original input reference in such a way so as the vibration can be reduced. However, it requires some knowledge about the plant's model.

Among the many methods developed for vibration suppression in mechanical system, input shaping proves to be a simple yet effective technique. Using value of the natural frequency and damping ratio of the vibrating system, the command input signal is decomposed into two signals. The magnitude and time between those two signals are set

such that the second signal can compensate vibration coming from the first signal, thus effectively eliminate subsequent vibration. Posicast control [26] was an early proposal of such shaping method. Later improvements addressed the robustness issues [27, 28], making the shaping technique widely practiced in many different applications. Later, many other researches extended the input shaping method (e.g. [29]). Input shaping techniques have been applied to various vibrating systems, such as industrial crane [30, 31], coordinate measuring machine [32], and sloshing on open container [33].

When compared with other feedforward filtering techniques, for example notch filter, low-pass filter, or band-pass filter, input shaping is often found to be favorable. Input shaping were shown to have significant advantages in terms of rise time and vibration reduction, as well as robustness, especially when large modelling errors exist [34, 35]. In addition, given the same design criteria, input shaping always has larger possible solution space that includes the solution space of low pass filters and notch filters. The consequence is that low pass filters and notch filters can never be shorter in duration than input shapers [36].

In practice, input shaping can also combined with closed-loop feedback control, for example with proportional control [37], with Time Delay Control (a robust feedback control law) [38], or with GA-tuned PID control [39]. The combinations are generally found to perform better than input shaping alone or feedback control alone. Although input shaping is primarily developed and installed outside the loop, it is also beneficial when used inside the feedback loop [40].

With regard to the transfer device, there are several other researches that specifically targeted the same or similar semiconductor wafer transfer robot arm. Tao et al. [41] used a combination of a digital acceleration filter applied to a third order trajectory and PID control to suppress residual vibration of a SCARA robot arm. Hosek and Moura [42] proposed several techniques, including iterative learning control and neural-network based perturbation estimation for similar robot arm devices. Kawamura, et al. [23] proposed a structural modification method to shift the center of gravity of wafer robot links in order to suppress its residual vibration after motion.

Lately, Uchiyama, et al. [43] developed an optimization based technique for essentially

the same experimental device as the one used in this research. The algorithm models each of the last two joints as two entities separated by a rotational spring, which introduces flexibility to the end effectors. At one side is the motor that provides needed torque, and at the other side is the driven link. In the optimization formulation, the time domain is discretized into many small sampling periods. Sequential quadratic programming iterative method is then used to find the optimal trajectory that minimizes the motion time and the vibration at the robot end effectors.

Another closely related paper is the work of Yamashita, et al. [44], which also addressed vibration control in semiconductor wafer transfer robot arm. The paper focuses on the application of input shaping to the robot tip trajectory, as well as its hybrid combination with joint space input shaping. In contrast, this thesis focuses on the integration of input shaping and trajectory planning, as well as the automatic identification of vibration parameters.

1.2.3 Sloshing Control

Sloshing is a special type of vibration which occurs in liquid body. As such, it exhibits a lot of similar phenomena as vibration does. There have been numerous researches dealing with sloshing suppression in liquid container transfer system. Feedback control based approaches present good and relatively robust sloshing control ability [44, 45]. However, in some practical cases, sensor measurements cannot always be reliable, or even is difficult to perform, for example in the case of controlling sloshing of high-temperature molten metal. In these cases, feedforward approaches [33, 46, 47, 48, 49, 50], which do not rely on the feedback of sloshing states, are more useful. Based on sloshing models, the behavior of the system can be predicted, and thus the trajectory or motion path can be designed accordingly. Examples include infinite impulse response (IIR) filter [46], input shaping filter [47, 48, 49], and Hybrid Shape Approach filters [50].

With respect to the motion path, traditionally sloshing is analyzed for one dimensional straight motion, where the sloshing phenomena is often modeled as a simple pendulum. The motion planning and sloshing suppression is relatively simple and straightforward for

this kind of problem [49]. Researches that address suppression of sloshing or pendulum sway in higher dimensional space are relatively few. Tzamtzi, et al. [51] modeled the sloshing as simple pendulum in 2D planar vertical motion. Williams, et al. [52] used a dynamic programming approach to solve the sway-free motion of suspended spherical pendulum in 2D planar horizontal motion. Yano and Terashima [50] proposed the Hybrid Shape Approach filters to control sloshing in three degree of freedom Cartesian liquid container transfer system.

1.3 Research Objectives

From the description of the problems and the literature review of past researches, we define the objective of this research as follows. This research aims to develop an integrated framework of trajectory generation and vibration suppression for point-to-point motion of multi-joint robot arm. The solution has to respect the joint constraints, e.g. angle, velocity, torque limits. It also has to consider the task-space based constraints, such as obstacle avoidance. While the main criteria of the trajectory generation is minimizing motion time, it needs to keep the motion-induced vibration (or sloshing) to a low level. Finally, it has to be provable either by simulation, experiment, or preferably both.

In order to realize the framework, we have identified two main building blocks: *piecewise cubic splines* for the trajectory generation and *input shaping* for the vibration suppression. The cubic splines trajectory has a few favorable properties, for example the limited jerk, which provides smoother profile than acceleration-limited trajectories, and the quick computation. Nonetheless, it is relatively fast compared to other higher degree polynomials. We use an implementation of cubic spline optimization in which some floating (free) via points are inserted between fixed via points. The location of those additional points are included as optimization variables along with the segment times, thus we aim for simultaneous optimization of both the points location and the motion time. The addition of the free points serves to improve the velocity profile of the trajectory and reduce the motion time, as well as to provide enough flexibility for the motion to avoid obstacles

Meanwhile, the input shaping offers good vibration suppression with only small increase

in motion time. It has found practice in many real applications for its combination of simplicity and good performance. From the basic principle of input shaping, researchers have come with numerous developments and modifications, resulting in many types of input shapers for different kind of applications.

1.4 Thesis Outline

This thesis is organized as follows:

Chapter 2 outlines the experimental devices that are used in this research. There are two application cases: semiconductor wafer transfer robot arm and liquid container transfer robot arm. In addition to the description of the physical system configuration, the chapter also explains the frame assignments and kinematic relations.

Chapter 3 describes the development of the trajectory planning framework based on cubic splines optimization. While the framework is meant to be generic – it means the framework should be applicable, or at least easily extendable, to any case of multi joint robot arm, the application case shown in the chapter is the semiconductor wafer transfer robot arm. The main consideration of the framework is minimal motion time, but without neglecting the smoothness of the joint trajectories.

Chapter 4 deals with the vibration aspect of trajectory planning. The chapter starts with a review of multi-mode vibration identification method and a direct application of input shaping to prebuilt trajectories. But following after that is the gist of the chapter, which is the integration of input shaping principles into the cubic spline based trajectory planning framework, to realize simultaneous consideration of motion time and vibration suppression.

Chapter 5 describes the trajectory planning and sloshing suppression for an automatic liquid transfer system. The integrated framework is an extension of the cubic spline based framework described in Chapter 2 and Chapter 3, at least in two points: the input shaping is applied in task space and the obstacle avoidance is explicit.

Chapter 6 summarizes the research as a whole and describes the important points concluded from it. It also identifies a few aspects that can be improved as a guidance for probable future works.

The general connections among those chapters are shown in the diagram of Fig. 1.1.

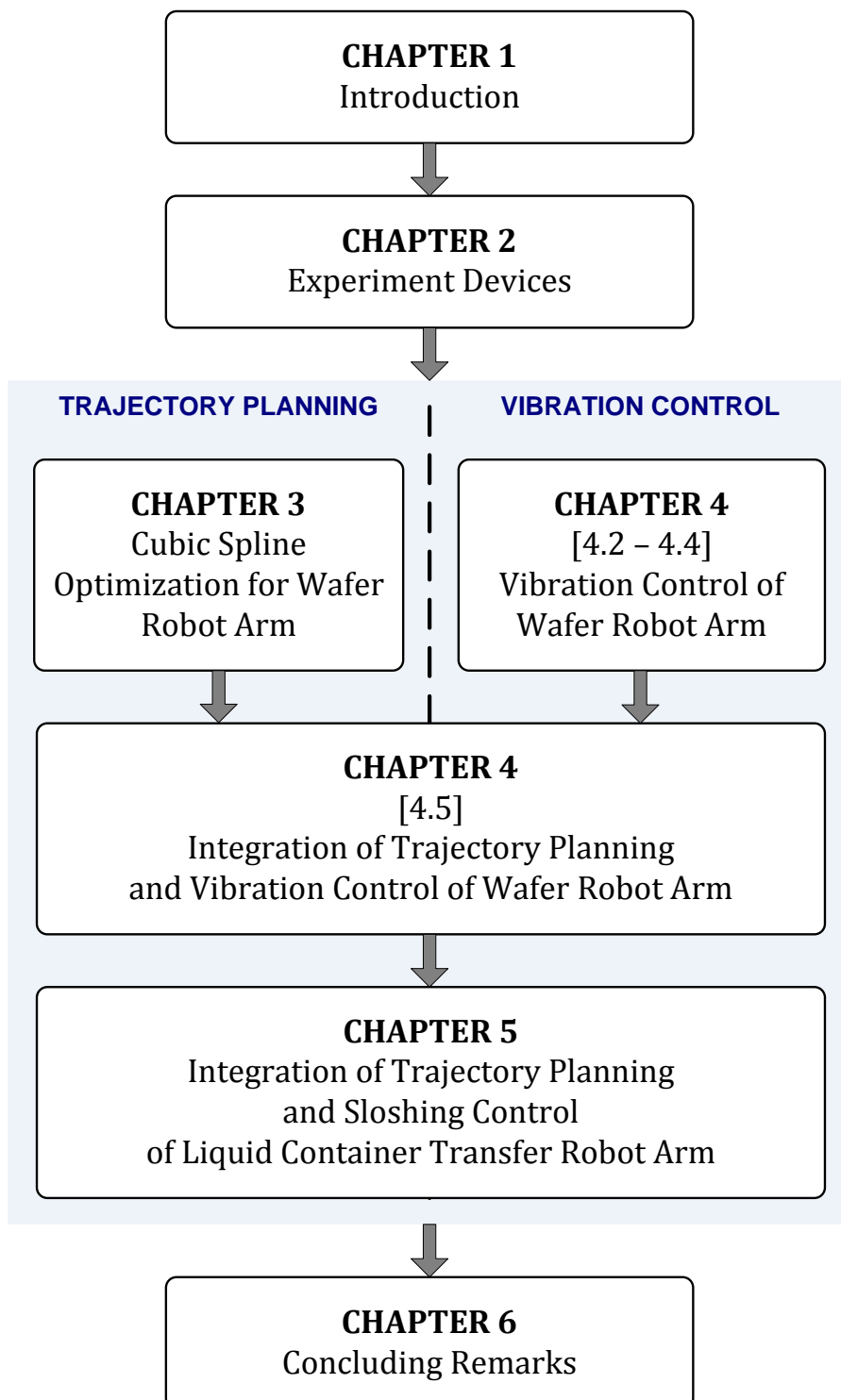


Fig. 1.1 Outline of the thesis

CHAPTER 2

Experimental Devices and System Construction

2.1 Introduction

This research focuses on two application cases. The first is a semiconductor wafer transfer robot arm, which is typically used in a highly automated semiconductor production facilities. The second is a robot arm to transfer liquid-filled container in a three dimensional workspace. This chapter outlines the description of those two application cases, including the physical parameters and the kinematics relationships of the system. The forward kinematics are the transformation of the coordinates in the joints space to the Cartesian task space. On the contrary, the inverse kinematics are to transform from the task space to the joint space. Both the forward and inverse kinematics will be used in the trajectory planning optimization, where constraints may be represented in either joint or task space. The semiconductor wafer robot arm is explained in Section 2.2, and following after that, Section 2.3 explains the liquid container transfer robot arm.

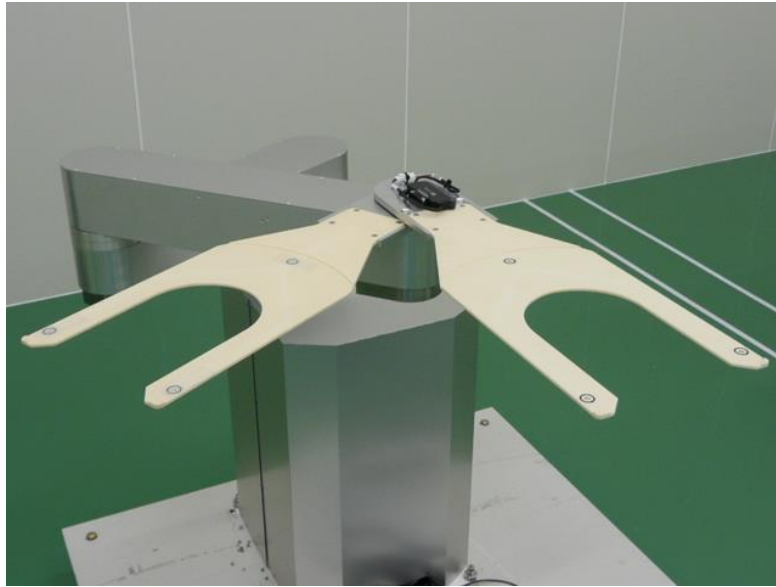


Fig. 2.1 The semiconductor wafer transfer robot arm

2.2 Semiconductor Wafer Transfer Robot Arm

Semiconductor wafer transfer robot arms are used in semiconductor industries to transfer semiconductor wafers from and to processing devices in a production cell. The robot we use, shown in Fig. 2.1, is a SCARA type robot with four links that operate in horizontal X-Y plane and one link that handles motion in vertical Z direction. For our current application, we consider only the horizontal planar links (links 1-4), where each link is driven by a rotational motor joint: T, R, H1, and H2, respectively, as shown in Fig. 2.2. Moreover, the last two links overlap each other and thus share the same task space in X-Y plane, but with different height in Z direction. In this research, the trajectories of those two last links are always the same, and therefore we will refer them as one link (link 3), and refer the actuating joint as joint H. Table 2.1 shows the physical parameters of the semiconductor wafer transfer robot arm.

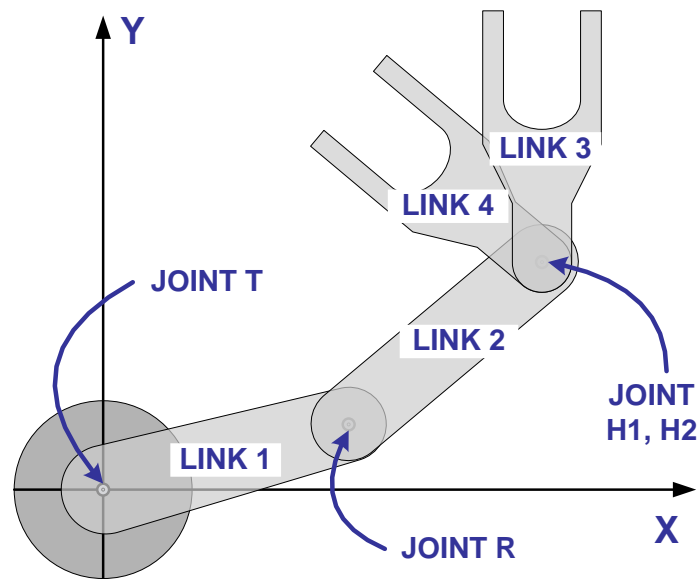


Fig. 2.2 The links and joints of the wafer robot arm

Table 2.1 Parameters of the semiconductor robot

Parameter	Nominal value			
	Link 1	Link 2	Link 3	Link 4
Length of link [m]	0.45	0.45	0.35	0.35
Distance of mass center [m]	0.208	0.244	0.095	0.100
Mass of link [kg]	8.505	4.319	1.327	1.292
Inertia of link [kg/m ²]	0.333	0.089	0.024	0.024
Maximum torque [Nm]	0.6	0.298	0.075	0.075
Gear ratio	1/133	1/82	1/41	1/41
Maximum velocity [rad/s]	2.362	3.831	7.662	7.662

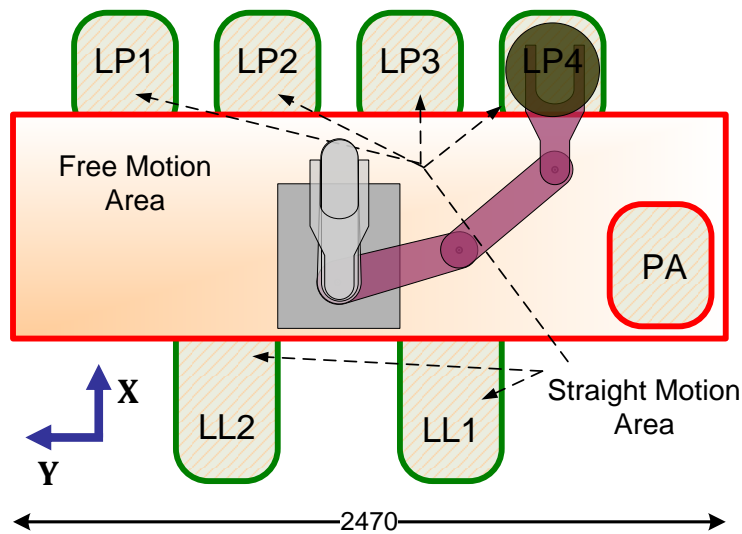


Fig. 2.4 Layout of the semiconductor production cell

The robot works inside a production cell, as shown in Fig. 2.4. The figure also shows the 'home' configuration, where the robot arm links are fully folded, and one example configuration where the arm is stretched to reach inside port LP4 while holding the wafer. The wafer is thin circular shaped plate with diameter 300 mm. The working area is mainly inside the rectangular 'Free Motion Area', where the robot arm is free to move as long as it does not collide with the boundary. Other than that, there are also several 'Straight Motion

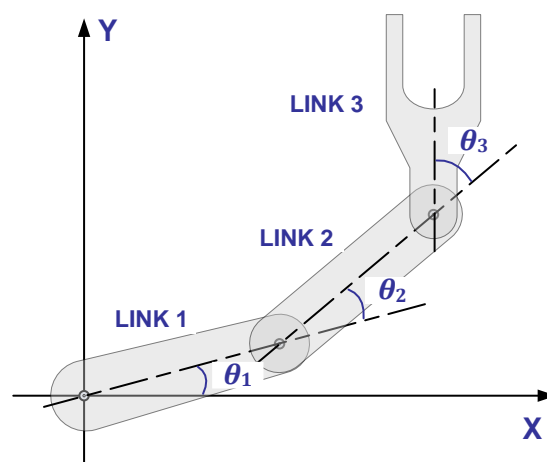


Fig. 2.3 Joint angles in the wafer robot arm1

Area', where the ports (LP1, LP2, LP3, LP4, LL1, LL2) are located. The ports are narrow, so that the last link has to enter the area in straight direction parallel to the X axis. Table 2.2 shows the location of the ports in the Cartesian space.

The forward kinematic relations between the position of tip of links in the Cartesian space and the joint configurations are:

$$x_j = \sum_{r=1}^j \left(l_r \cos \sum_{k=1}^r \theta_k \right) \quad (2.1)$$

$$y_j = \sum_{r=1}^j \left(l_r \sin \sum_{k=1}^r \theta_k \right) \quad (2.2)$$

$$\varphi_j = \sum_{k=1}^j \theta_k \quad (2.3)$$

where l_j is the length of link j , x_j and y_j are the position of link tip in X and Y axis, respectively, and φ_j is the orientation of link j relative to the base frame. Fig. 2.3 shows the joint angles definition of the wafer robot arm, where θ_j is the angle value of joint j . The frame of the first link is the same as the base frame.

Table 2.2 Location of ports in Cartesian space

Port	X (home)	X (gate)	X (port)	Y
Load Port 1 (LP1)	0.350	0.450	0.800	0.7575
Load Port 2 (LP2)	0.350	0.450	0.800	0.2525
Load Port 3 (LP3)	0.350	0.450	0.800	-0.2525
Load Port 4 (LP4)	0.350	0.450	0.800	-0.7575
Load Lock 1 (LL1)	-	0.070	-0.436	0.3950
Load Lock 2 (LL2)	-	0.070	-0.436	-0.3950
Pre-aligner (PA)	-	-	0.090	-1.0725

The inverse kinematics relationships are as follows:

$$\theta_1 = \text{atan2}(y_2, x_2) \pm \text{acos} \left(\frac{\left(L_1^2 - L_2^2 + (x_2^2 + y_2^2) \right)}{2L_1A} \right) \quad (2.4)$$

$$\theta_2 = \pi \mp \text{acos} \left(\frac{\left(L_1^2 + L_2^2 - (x_2^2 + y_2^2) \right)}{2L_1L_2} \right) \quad (2.5)$$

$$\theta_3 = \varphi - \theta_1 - \theta_2 \quad (2.6)$$

For measurement of vibration, two sensors are used. The main measurement uses a laser displacement sensor, which is placed near the end point of the robot motion path. When the robot stops at the end of motion, the residual vibration that is generated will be picked up by the sensor. An accelerometer is fixed on the last robot link, thus it moves with the robot and captures the vibration while the robot is moving. It is also used to determine the time when the robot starts and stops moving.

2.3 Liquid Container Transfer Robot Arm

The liquid container transfer system is used to transfer a liquid-filled container from and to different locations in three dimensional space. The system uses the 7 degree-of-freedom Mitsubishi PA 10-7C robot arm. The frames naming and placement, as well as links

Table 2.3 Limit value of joint angle and angular velocity

Limit	Joint						
	S1	S2	S3	E1	E2	W1	W2
V^{MAX} [rad/s]	1	1	2	2	2π	2π	2π
q^{UB} [degree]	177	91	174	137	255	165	360

dimension of the robot arm are shown in Fig. 2.5. Table 2.3 lists the limit value of joint angle and angular velocity of each robot joint.

A cylindrical liquid container is attached to an extension at the robot tip. The diameter and height of the liquid container are 0.15 m and 0.25 m, respectively. An electric-resistance level sensor is attached to the container, placed on one side of the cylinder to measure the height of liquid over time. It works by detecting changes in the resistance between two electrode probes. Sloshing in the container can be measured from fluctuation of the resistance.

The distance between the robot tip and the center of the liquid container is 0.12 m, thus the length of the last link is practically 0.2 m. For our current application, one of the arm joints, the S3 joint, is locked so that only six joints are actively used in this simulation study. Table 2.4 lists the Denavit-Hartenberg parameters of the robot arm configuration.

The transformation matrix of a frame relative to the preceding frame is as follows:

$$A_i = \begin{bmatrix} {}^{i-1}R_i & {}^{i-1}p_i \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\cos \theta_i \sin \alpha_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.7)$$

where θ_i is the angle value of joint i . The position and orientation of a link with respect to

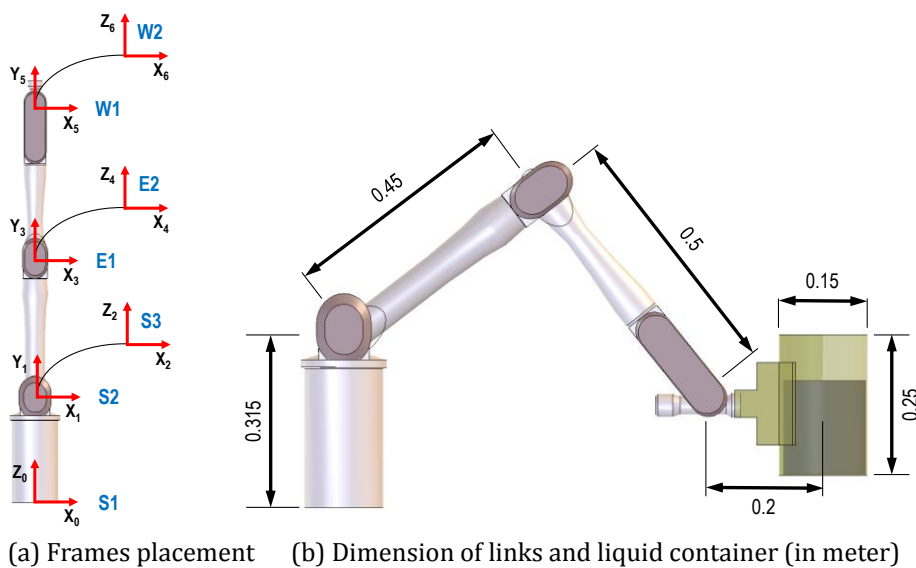


Fig. 2.5 Frames assignment and dimension of system

Table 2.4 Denavit-Hartenberg parameters

Parameter	Joint						
	S1	S2	S3	E1	E2	W1	W2
α	$\pi / 2$	$-\pi / 2$	$\pi / 2$	$-\pi / 2$	$\pi / 2$	$-\pi / 2$	0
d	0.315	0	0.45	0	0.5	0	0.2
a	0	0	0	0	0	0	0

other link can be obtained by chain product of the transformation matrices:

$${}^i T_j = A_{i+1} A_{i+2} \cdots A_j = \begin{bmatrix} {}^i R_j & {}^i p_j \\ 0 & 1 \end{bmatrix} \quad (2.8)$$

The upper left 3×3 rotation matrix ${}^i R_j$ describes the link orientation, while the vector ${}^i p_j$ represents the position. For example, the orientation and location of the tip of the last link with respect to the global frame can be obtained from the matrix ${}^0 T_7$.

The joints E2, W1, and W2 form a spherical wrist, where the three rotational joint axes intersect at a common point. This robot configuration simplifies the inverse kinematic calculation because it can be decoupled into two simpler subsystems: inverse kinematics for position and inverse kinematics for orientation. The first three rotational joints S1, S2, and E1 provides the positioning in three dimension space, while the spherical wrist adjusts the orientation of the end effector.

In the inverse kinematics for position, the position of wrist locus in the task space (i.e. x_w , y_w , and z_w) is known, and the configuration of S1, S2, and E1 joints (i.e. θ_1 , θ_2 , and θ_4) are to be sought. The values can be calculated geometrically as follows

$$\theta_1 = \text{atan2}(y_w, x_w) \quad (2.9)$$

$$\theta_4 = \text{atan2}(\sqrt{1 - D^2}, D) \quad (2.10)$$

$$\theta_2 = \text{atan2}\left(z_w - L_1, \sqrt{x_w^2 + y_w^2}\right) - \text{atan2}(L_3 \sin \theta_4, L_2 + L_3 \cos \theta_4) - \frac{\mu}{2} \quad (2.11)$$

where

$$D = \frac{x_w^2 + y_w^2 + (z_w - L_1)^2 - L_2^2 - L_3^2}{2L_2L_3} \quad (2.12)$$

and L_i is the length of link i .

Meanwhile, the inverse kinematics for orientation calculates the configuration of E2, W1, and W2 joints (i.e. θ_5 , θ_6 , and θ_7) when the desired orientation of end effector relative to the base frame (0R_7) is known. First, from the inverse kinematics for position, the rotation matrix describing the orientation of the third link can be obtained (0R_4). Then, the rotation operation of the wrist joints which is required to realize the end effector posture is calculated as:

$${}^4R_7 = ({}^0R_4^T)({}^0R_7) \quad (2.13)$$

The three Euler angles can be calculated from the matrix, and finally we can obtain the configuration of the wrist joints:

$$\theta_5 = \text{atan2}({}^4R_7[2,3], {}^4R_7[1,3]) \pm \pi \quad (2.14)$$

$$\theta_6 = \text{atan2}\left(\sqrt{1 - {}^4R_7[3,3]^2}, {}^4R_7[3,3]\right) \quad (2.15)$$

$$\theta_7 = \text{atan2}({}^4R_7[3,2], {}^4R_7[3,1]) \pm \pi \quad (2.16)$$

where ${}^4R_7[r, c]$ is the element of the matrix 4R_7 at row r and column c . When there are more than one possibilities of angle values, choose the one suitable for the application.

2.4 Summary

In this chapter, the configuration of the experiment devices is explained. Besides the basic physical parameters and limits, the forward and inverse kinematics of the devices are also derived. Those data and kinematic calculation will be used in the development of the trajectory planning framework.

CHAPTER 3

Cubic Spline Optimization for Trajectory Generation of Multi-Joint Robot Arm

3.1 Introduction

This chapter mainly deals with the development of cubic spline based optimization to generate trajectory. The main objective of the planner is reducing the motion time. However, smoothness of the trajectory, which ultimately relates with vibration, is also preserved by inserting jerk limit constraint to the optimization. This is where cubic splines is favorable, because among all jerk-limited profiles, cubic splines results in the fastest motion.

Section 3.2 explains the basic principle of the cubic spline trajectory. In the end of the section, the relation between joint angle, joint acceleration, and segment time is derived. The following section, Section 3.3, explains the development of the optimization framework in the context of a multi-joint robot arm, where the semiconductor wafer transfer robot arm is used as the application case.

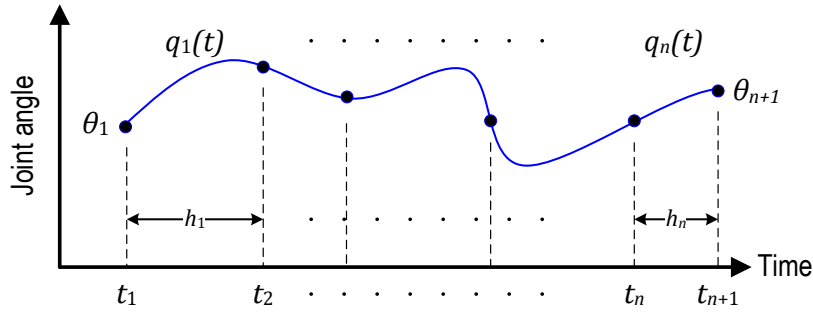


Fig. 3.1 Cubic spline trajectory

3.2 Basic Principle of Cubic Spline Trajectory

A cubic spline trajectory of a robot joint is composed of several curve segments. Fig. 3.1 illustrates such joint trajectory, where each segment is represented by a cubic polynomial function of time. The general form of the cubic function in a segment i is:

$$q_i(t) = a_i(t - t_i)^3 + b_i(t - t_i)^2 + c_i(t - t_i) + d_i, \tag{3.1}$$

where t denotes the time, which ranges from t_i to t_{i+1} . The shape of the cubic curve is determined by those four parameters: a , b , c , and d . From the function of joint trajectory, the derivative functions (joint acceleration, joint velocity, and joint jerk) can be obtained by differentiation.

Curve segments are connected by knots. In a trajectory that consists of n curve segments, there are $n + 1$ prespecified joint angle values θ_i ($i = 1..n + 1$), which consist of a start point, an end point, and $n - 1$ knots between them. One condition to fulfill is that the cubic segments run through all those points. In other words, the value of the cubic function at time $t = t_i$ has to equal θ_i :

$$\begin{aligned} \theta_i &= q_i(t_i) = a_i(t_i - t_i)^3 + b_i(t_i - t_i)^2 + c_i(t_i - t_i) + d_i \\ &= d_i \end{aligned} \tag{3.2}$$

Meanwhile, the value at time $t = t_{i+1}$ has to equal θ_{i+1} :

$$\begin{aligned}\theta_{i+1} &= q_i(t_{i+1}) = a_i(t_{i+1} - t_i)^3 + b_i(t_{i+1} - t_i)^2 + c_i(t_{i+1} - t_i) + d_i \\ &= a_i h_i^3 + b_i h_i^2 + c_i h_i + d_i\end{aligned}\quad (3.3)$$

where $h_i = (t_{i+1} - t_i)$ is the time interval of the segment. Other requirements are continuity in velocity and acceleration at every knot. At the knot between segments i and $i + 1$, the velocity $\dot{\theta}_{i+1}$ set by segment i has to be equal to that set by segment $i + 1$:

$$\begin{aligned}\dot{q}_i(t_{i+1}) &= \dot{q}_{i+1}(t_{i+1}) \\ 3a_i h_i^2 + 2b_i h_i + c_i &= c_{i+1}\end{aligned}\quad (3.4)$$

and similarly for the acceleration:

$$\begin{aligned}\ddot{q}_i(t_{i+1}) &= \ddot{q}_{i+1}(t_{i+1}) \\ 6a_i h_i + 2b_i &= 2b_{i+1}\end{aligned}\quad (3.5)$$

In a trajectory with n segments, there are $4n$ unknown parameters ($a_i, b_i, c_i, d_i: i \in I$), where I is the set of all segment indexes in the curve. However, the number of parameters can be reduced by formulating the problem in a more simple form with less unknowns. From Eq. (3.3), we can get:

$$c_i = \frac{\theta_{i+1} - \theta_i}{h_i} - a_i h_i^2 - b_i h_i \quad (3.6)$$

Putting Eqs. (3.5) and (3.6) into Eq. (3.4), we obtain:

$$b_i h_i + 2b_{i+1}(h_i + h_{i+1}) + b_{i+2} h_{i+1} = 3 \left(\frac{\theta_{i+2} - \theta_{i+1}}{h_{i+1}} - \frac{\theta_{i+1} - \theta_i}{h_i} \right) \quad (3.7)$$

Then, because $\ddot{\theta}_i = \ddot{q}_i(t_i) = 2b_i$, in the end we can obtain the relationship between joint angle θ_i , joint acceleration $\ddot{\theta}_i$, and time interval h_i :

$$\ddot{\theta}_i h_i + 2\ddot{\theta}_{i+1}(h_i + h_{i+1}) + \ddot{\theta}_{i+2} h_{i+1} = 6 \left(\frac{\theta_{i+2} - \theta_{i+1}}{h_{i+1}} - \frac{\theta_{i+1} - \theta_i}{h_i} \right) \quad (3.8)$$

In a joint trajectory with n segments (or $n + 1$ points), there are $n - 1$ such equations. Together with the boundary requirements of acceleration $\ddot{\theta}_1 = \ddot{\theta}_{n+1} = 0$, we have a total

because they use the same value of segment times h_i . Henceforth $q_{ij}(t)$ denotes the trajectory of joint j in segment i . Likewise, θ_{ij} , $\dot{\theta}_{ij}$, $\ddot{\theta}_{ij}$, and $\dddot{\theta}_{ij}$ respectively denotes the angle, velocity, acceleration, and jerk of joint j at knot i .

There are several constraints to consider in the trajectory generation: robot position in the task space, joint velocity, motor torque, and joint jerk.

Position constraints

The robot works inside a production cell, as shown in Fig. 2.4. It is important for the links not to exceed the boundary of the production cell. As an example of the robot motion, the motion from home position to Load Port (LP) consists of free motion from the home position until the LP gate and straight motion from the LP gate until the LP position where the wafer will be unloaded and processed. During the free motion, the body of every link has to be always inside the rectangle. Because all links are simple straight links, it is sufficient to calculate only the position of the tip of each link by forward kinematics equations. For our planar robot, the X and Y position of the tip of link j in segment i are as follows:

$$x_{ij}(t) = \sum_{r=1}^j \left(l_r \cos \sum_{k=1}^r q_{ik}(t) \right) \quad (3.13)$$

$$y_{ij}(t) = \sum_{r=1}^j \left(l_r \sin \sum_{k=1}^r q_{ik}(t) \right) \quad (3.14)$$

where l_j is the length of link j . The link tip positions in each segment reach extreme when the time derivatives of Eqs. (3.13) and (3.14) are zero. The link tip positions must be between the maximum and minimum limits in X and Y direction:

$$x^{\text{MIN}} \leq x_{ij}^*(t) \leq x^{\text{MAX}}, \quad i \in F, j \in J \quad (3.15)$$

$$y^{\text{MIN}} \leq y_{ij}^*(t) \leq y^{\text{MAX}}, \quad i \in F, j \in J \quad (3.16)$$

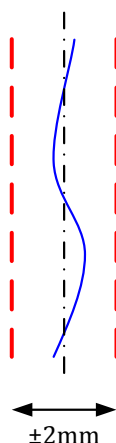


Fig. 3.2 Position bound on the straight motion area

where F is the set of segments that belong to the free motion area and J is the set of all joint indexes ($J = \{1, 2, 3\}$). The values of x^{MIN} , x^{MAX} , y^{MIN} , and y^{MAX} are defined according to the real workspace, as explained in Section 2.2.

In the straight motion area, the motion path should be a straight line in X direction, or very close to it, in order to avoid contact of the robot structure with obstacles. During that motion, the tip position of link 2 and 3 has to follow the straight line as much as possible. It also implicitly means maintaining the absolute orientation of the link parallel with the straight line. Checking should be done against the extremum Cartesian positions of the link in Y direction. This time, the positions should be within some small distance from the perfect straight line:

$$|y_{ij}^* - y^{\text{STRAIGHT}}| \leq \delta, \quad i \in S, j \in \{2, 3\} \quad (3.17)$$

Here, y^{STRAIGHT} denotes the Y coordinate value of the straight line, S is the set of segment indexes that belong to the straight motion area, and δ is the tolerance of distance to the nominal straight line. For the current application, δ equals 1 mm. Fig. 3.2 illustrates the position bound on the straight motion area where the solid line is the motion path of the link tip, the two dashed lines form the virtual boundary, and the dash-dotted line represents the nominal straight line in the area. The path is considered straight as long as it is inside the virtual boundary.

Velocity constraints

The joint velocities have to be less than some specified maximum velocity. In cubic spline the velocity is quadratic, and therefore could have one extrema value inside its time interval. The extrema velocity of a segment is the velocity when its respective acceleration is zero. The time is thus calculated as follows:

$$t_{ij}^* = t_{ij} + \frac{\ddot{\theta}_{ij} h_i}{\ddot{\theta}_{ij} - \ddot{\theta}_{i+1,j}} \quad (3.18)$$

If the t_{ij}^* is within the time interval of the segment, or in other words $t_{ij} < t_{ij}^* < t_{ij} + h_i$, then the extreme joint velocity v_{ij}^* equals $\dot{q}_{ij}(t_{ij}^*)$. Otherwise, it simply equals the velocity at knot $\dot{q}_{ij}(t_{ij})$. The velocity constraint is as follows:

$$|v_{ij}^*| \leq v_j^{\text{MAX}}, \quad i \in I, j \in J \quad (3.19)$$

where $I = F \cup S$, or in other words the set I contains all segment indexes in the curves. The values of v_j^{MAX} are given for each joint in Table 2.1.

Torque constraints

Other than velocity limit, the motors used to drive the robot have specified torque limit. Motor torques are calculated using the dynamic equations of the robot arm. The torque constraint is:

$$|M(\theta)\ddot{\theta} + c(\theta, \dot{\theta})| \leq \tau^{\text{MAX}} \quad (3.20)$$

where $M(\theta)$ is the inertia matrix, $c(\theta, \dot{\theta})$ is the vector of centrifugal and Coriolis forces, and τ^{MAX} is the vector of joints torque limit. For current application, the joints torque limit is defined as 80% of the maximum motor torques, which is listed in Table 2.1.

Jerk constraints

Jerk constraints are enforced in order to prevent too much jerk:

$$|\ddot{\theta}_{ij}| \leq J^{MAX}, \quad i \in I, j \in J \quad (3.21)$$

where J^{MAX} is the maximum jerk. It is set equal to 250 rad/s³.

We formulate the above trajectory generation problem as an optimization problem with the motion time, which is the time required to move from start point to end point, as the objective function. According to the system of linear equations in Eq. (3.9), there are three kinds of variables involved: segment time, joint acceleration, and joint angle at knots. We let both the segment time and the joint angles (knots location) as the optimization variables. The optimization problem is as follows:

$$\min_p F(p) = \sum_{i \in I} h_i \quad (3.22)$$

subject to (3.15)(3.16)(3.17)(3.19)(3.20)(3.21)

where p is the decision variable which contains the segment times as well as the joint angles excluding the fixed start point, end point, and the two virtual points:

$$p = (h_1, \dots, h_n, \theta_{3j}, \dots, \theta_{(n-1)m}) \quad (3.23)$$

Here, all internal knots are defined as part of the decision variables, hence free knots. However, if the planned motion is required to pass certain fixed points on the way, they will be added as equality constraints. Fig. 3.3 illustrates the concept of free knots. In addition to be allowed to change in horizontal direction (thus changing the h_i value), a free knot can also change in vertical direction (thus changing the θ_{ij} value, which in turn will change the motion path in the task space).

We use the MATLAB Optimization Toolbox implementation of Sequential Quadratic Programming (SQP) [54] for solving the above constrained nonlinear trajectory planning optimization problem. The SQP is one of the most used methods in solving general constrained nonlinear optimization problems. The method solves the problem by iteratively solving a quadratic subproblem, which is the quadratic approximation of the Lagrange

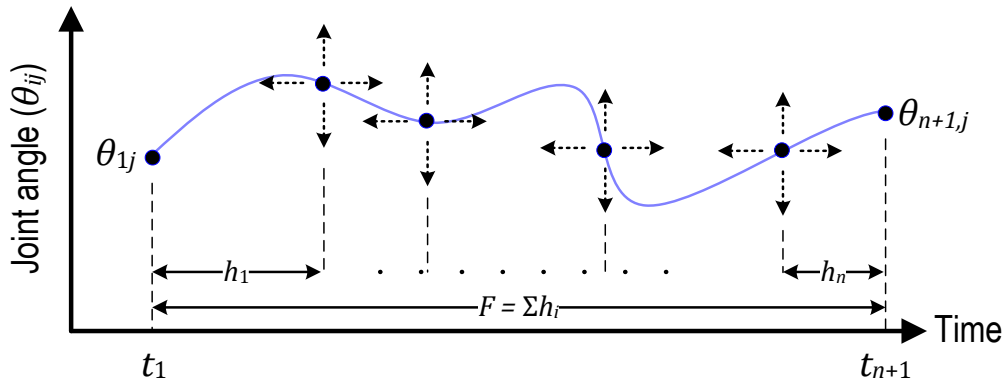


Fig. 3.3 The free knots in cubic trajectory

function of the original nonlinear problem. The solution of the quadratic subproblem is then used to form a new set of optimization variables for the next iteration of the original problem. In each optimization iteration, the following procedures are performed:

- (1) Referring to Eq. (3.9), form the left hand matrix involving h_i variables and the right hand matrix involving θ_{ij} variables, and then calculate joint accelerations $\ddot{\theta}_{ij}$ of all knots by using the tridiagonal matrix algorithm (TDMA). Note that the calculation is to be performed for each joint using the same value of h_i .
- (2) Using the joint acceleration values, calculate the joint velocities and jerks. Furthermore, torques can be calculated using the dynamics equations of the robot arm.
- (3) Calculate all the constraints: position constraints in Eqs. (3.15), (3.16), (3.17), velocity constraints in Eq. (3.19), torque constraints in Eq. (3.20), and jerk constraints in Eq. (3.21).
- (4) Calculate the total motion time as the objective function.

Fig. 3.4 shows an example trajectory generated by the optimization for motion LP1 to LP3. The motion curve starts from ‘home’ position of LP1, moving to the ‘gate’ position of LP3, and finally moving straight until the ‘port’ position of LP3. As such, actually the ‘gate’ of LP3 (0.45, -0.2525) is the only via point that needs to be passed by the robot tip. However, putting only that fixed via point into the optimization problem would result in very poor

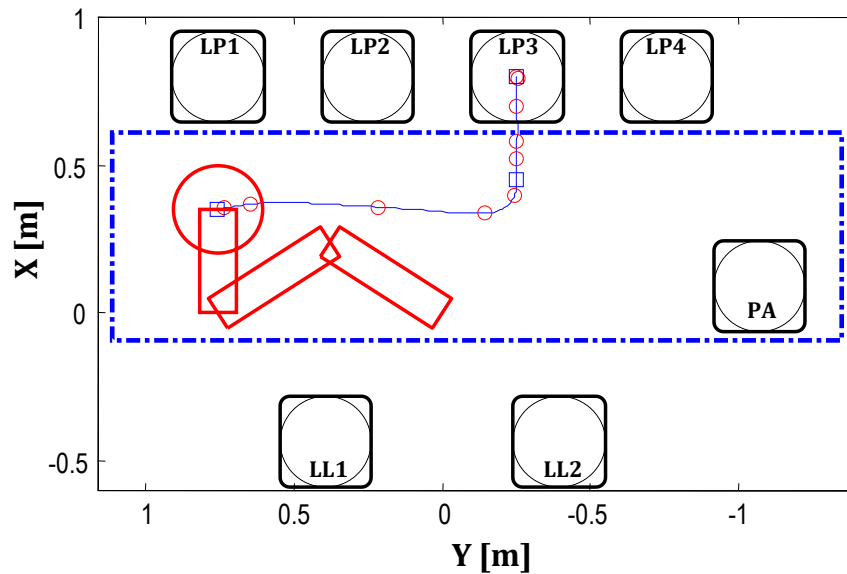


Fig. 3.4 Generated path of example LP1-LP3 motion

motion time. The reason is that with cubic spline algorithm, kinematic states within a segment are governed by one single polynomial. For example, the velocity in a segment is governed by one quadratic equation, which profile is limited (e.g. only one extremum). The acceleration is either constant or linearly increasing or decreasing. Therefore, too few knots would not give enough room to minimize the motion time.

For the above reasons, in addition to the fixed gate point, four more free points are added, equally spaced on the vertical line between the gate and end point to guide in the straight motion area. Similarly, four free points are also specified on the horizontal line between the gate and start point. Table 3.1 lists the initial value of the points. Other than the fixed points (start point, gate point, and end point) and the free points mentioned above, there are also two virtual points. As explained in Section 3.2, those virtual points are added to enable using the velocity boundary conditions. In Fig. 3.4, fixed points are denoted by small rectangles, while other points are denoted by circles. The total motion time obtained for the example is 1.28 second.

Table 3.1 Initial value of via points

Via point	Task space values (X,Y,orientation)	Joint space angle (T,R,H)	Note
1	(0.35, 0.7575, 0)	(1.00, 1.14, -2.14)	Fixed
2	-	-	Virtual
3	(0.35, 0.505, 0)	(0.60, 1.95, -2.55)	Free
4	(0.35, 0.2525, 0)	(0.28, 2.57, -2.86)	Free
5	(0.35, 0, 0)	(0, 3.14, -3.14)	Free
6	(0.45, -0.2525, 0)	(0.07, 3.75, -3.83)	Fixed
7	(0.5375, -0.2525, 0)	(0.28, 3.86, -4.13)	Free
8	(0.6250, -0.2525, 0)	(0.40, 4.00, -4.40)	Free
9	(0.7125, -0.2525, 0)	(0.45, 4.17, -4.17)	Free
10	-	-	Virtual
11	(0.8, -0.2525, 0)	(0.45, 4.36, -4.81)	Fixed

Fig. 3.5 shows the angle, velocity, acceleration and jerk values of the obtained optimization result. From Fig. 3.5(b), it can be understood that joint R determines the motion time as it hits the velocity limit the most. The velocity profile of joint R in the free motion area (the first 5 segments) can achieve the ideal jerk-limited velocity profile where it ramps up and then cruise in maximum velocity, then finally ramps down.

We note that if all constraints other than velocity constraint are neglected, the velocity profile would cruise longer until near end. The profile would be similar to that obtained by point to point jerk-limited algorithms [3]. But this would have violated constraints and made it unusable. Please note also that such profile could not be achieved if we do not use knots at all, as the velocity would be one big quadratic curve extending from start to the end, if at all possible.

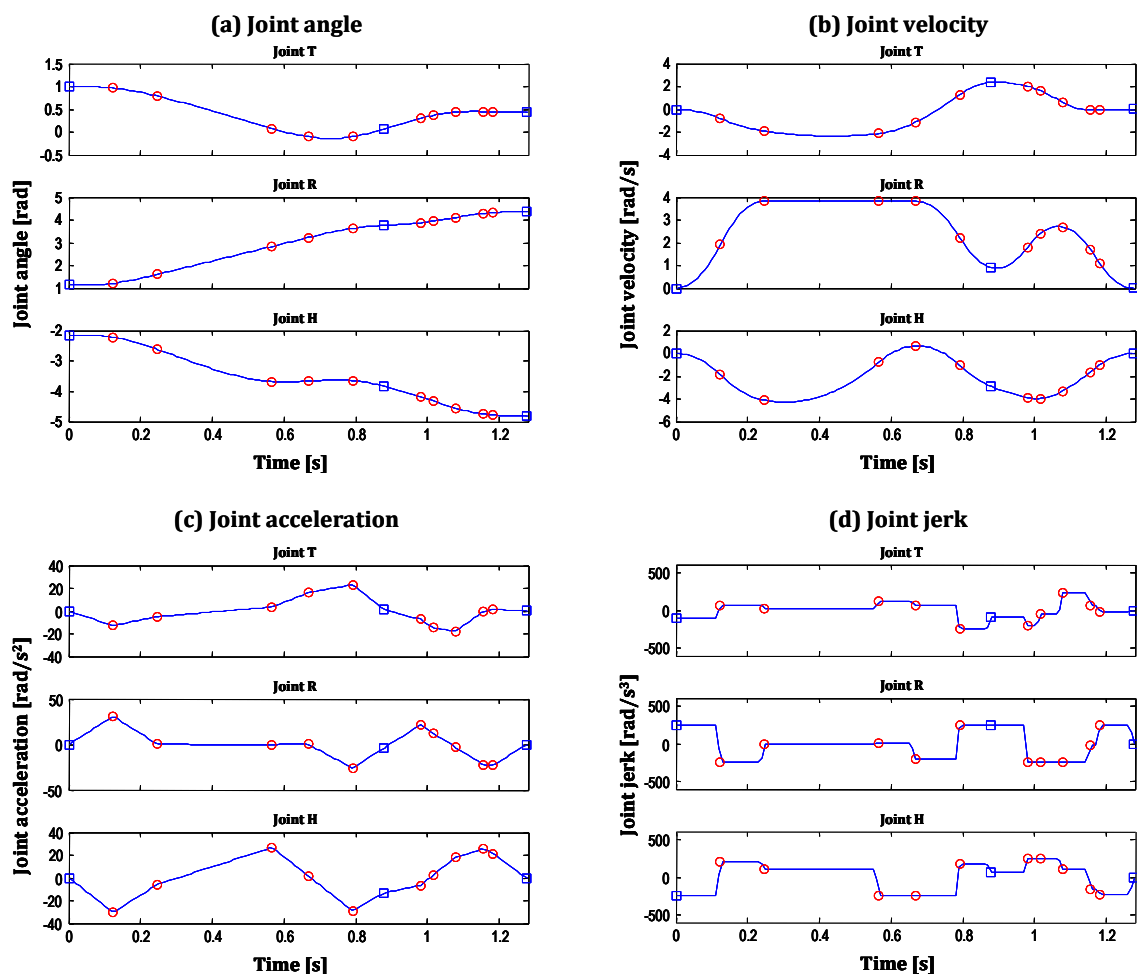


Fig. 3.5 Generated trajectory of LP1-LP3 motion: (a) joint angle, (b) joint velocity, (c) joint acceleration, (d) joint jerk

3.4 Summary

We have developed a trajectory planning framework based on piecewise cubic spline optimization for multi-joint robot arm with the semiconductor wafer transfer robot arm as the application case. The constraints include the joint intrinsic constraints (velocity and torque), task space constraints (workspace boundaries), and jerk constraints.

An implementation of ‘floating’ knots, by including the location of the knots (via points) as part of the decision variables along with the segment times, is proposed. It has been

shown that the approach is better because it gives the optimizer a greater space to improve the velocity profile in the generated trajectory, and as a result the motion time can become even shorter. In addition, the ‘floating’ approach allows better obstacle avoidance because not only the curve in the segments but also the location of knots themselves are adjustable. The trajectory generation framework has been applied to plan various motion paths of a semiconductor wafer transfer robot arm.

CHAPTER 4

Vibration Suppression of Semiconductor Wafer Transfer Robot Arm

4.1 Introduction

This chapter deals with the vibration aspect of trajectory planning. The chapter starts with a review of the principles of input shaping in Section 4.2. For the detail description of input shaping technique, we refer interested readers to [27, 28, 53]. The input shaping technique needs the value of natural frequency and damping ratio of the vibrating system. Therefore, we first develop a multi-mode vibration identification method in Section 4.3. It is followed in Section 4.4 by the direct application of input shaping to prebuilt trajectories. This demonstrates the versatility of input shaping in that it can be directly applied to any kind of input reference simply by convolving the trajectory with an appropriate input shaper.

Following after that, in Section 4.5, is the gist of the chapter, which is the integration of input shaping principles into the cubic spline based trajectory planning framework, to realize simultaneous consideration of motion time and vibration suppression. The

capability of the framework to suppress the vibration while keeping motion time short is demonstrated by experiment.

4.2 Principles of Input Shaping

An uncoupled, linear, vibratory system of any order can be specified as a cascaded set of second-order poles with the decaying sinusoidal response:

$$y(t) = \left[A \frac{\omega_n}{\sqrt{1-\zeta^2}} e^{-\zeta\omega_n(t-t_0)} \right] \sin\left(\omega_n\sqrt{1-\zeta^2}(t-t_0)\right) \quad (4.1)$$

where A is the amplitude of the impulse, ω_n is the undamped natural frequency of the plant, ζ is the damping ratio of the plant, t is time, and t_0 is time of the impulse input. For a system with only one mode of vibration, given impulse of unity amplitude at time $t_0 = 0$, the response is:

$$y_1(t) = \left[\frac{\omega_n}{\sqrt{1-\zeta^2}} e^{-\zeta\omega_n t} \right] \sin\left(\omega_n\sqrt{1-\zeta^2}t\right) \quad (4.2)$$

In order to suppress the vibration by the above first impulse, we add a second impulse at time ΔT such that the response of the two impulses cancel out each other. The response of the second impulse is thus as follows:

$$y_2(t) = \left[K \frac{\omega_n}{\sqrt{1-\zeta^2}} e^{-\zeta\omega_n(t-\Delta T)} \right] \sin\left(\omega_n\sqrt{1-\zeta^2}(t-\Delta T)\right) \quad (4.3)$$

where K is the amplitude of second impulse.

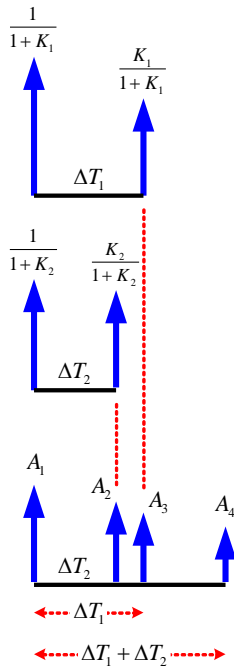
By setting the sum of $y_1(t)$ and $y_2(t)$ to zero for all time after ΔT , we obtain the magnitude (K) and time (ΔT) of the second impulse as follows:

$$K = e^{-\zeta\pi/\sqrt{1-\zeta^2}} \quad (4.4)$$

$$\Delta T = \frac{\mu}{\omega_n \sqrt{1 - \zeta^2}} \quad (4.5)$$

Adjustment should be made so that the sum of the two impulses equals one (thus the command does not go beyond maximum value), finally giving the values $1 / (1 + K)$ and $K / (1 + K)$ as the magnitude of the first and second impulse, respectively.

This set of impulses is called an input shaper, and the two-impulse input shaper above is known as Zero Vibration (ZV) shaper. The above formula for single mode system can be generalized to multiple mode system [54]. When there are more than one vibration mode in a system, the impulse sequence built independently for each mode should be convolved with each other to form an impulse sequence that targets multi-mode system vibration. Just as in the single-mode case, the impulse sequence in higher modes should be normalized. In this multi-mode case, the length of the input shaper is the sum of each individual shaper's length. For example, two-mode ZV input shaper can be derived as follows:



$$A_1 = \frac{1}{(1 + K_1)(1 + K_2)}, \quad A_2 = \frac{K_2}{(1 + K_1)(1 + K_2)}$$

$$A_3 = \frac{K_1}{(1 + K_1)(1 + K_2)}, \quad A_4 = \frac{K_1 K_2}{(1 + K_1)(1 + K_2)} \quad (4.6)$$

$$t_1 = 0, \quad t_2 = \Delta T_2$$

$$t_3 = \Delta T_1, \quad t_4 = \Delta T_1 + \Delta T_2 \quad (4.7)$$

First, the input shaper for mode 1 and 2 are formed individually. It results in two individual input shapers with different length according to the vibration parameters of each mode. Then the two input shapers are convolved with each other. The result is an input shaper

with four impulses with amplitudes and timing as shown in Eq. (4.6) and Eq. (4.7), respectively. The length of the two-mode input shaper equals the length sum of the two original input shapers.

A more robust input shaper type can be obtained by convolving two ZV input shapers together, resulting in three-impulse input shaper, known as Zero Vibration and Derivative (ZVD) input shaper. Thus the calculation of ZVD input shaper [27] is the same as calculation of two-mode input shaper where the two modes are identical. The amplitudes of the three impulses are as follows:

$$A_1^{ZVD} = \frac{1}{(1+K)(1+K)} \quad (4.8)$$

$$A_2^{ZVD} = \frac{2K}{(1+K)(1+K)} \quad (4.9)$$

$$A_3^{ZVD} = \frac{K^2}{(1+K)(1+K)} \quad (4.10)$$

Other than those basic input shapers, there are still many more input shapers developed with different characteristics and capabilities to tackle different type of vibration problems.

In a linear system, any signal can be represented as a collection of scaled and shifted impulses. Therefore the system response to a signal is the linear sum of the responses of each component impulse. It follows that the input shaping can be applied by convolving arbitrary system input signal with the desired input shaper, as shown in Fig. 4.1. As a result,

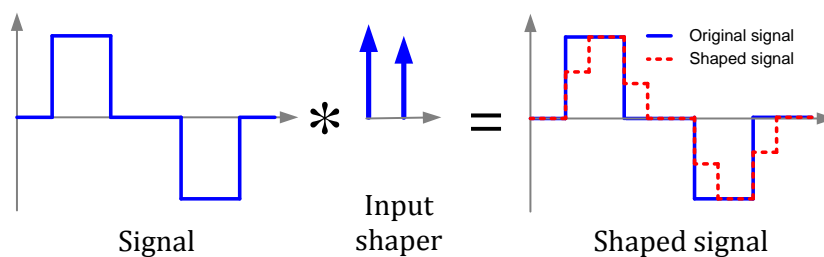


Fig. 4.1 Shaping arbitrary signal by convolution

the motion time becomes longer, as much as the length of the input shaper. For example, applying ZV input shaper increases the motion time by half damped period of vibration, while with ZVD input shaper the motion time becomes one damped period longer. For general planar horizontal motion, the input signal may be in the form of acceleration or velocity reference of the motion.

For systems whose motion is commanded by rotational commands, for example rotary crane and other rotating mechanical systems, regular input shapers do not work well for suppressing motion induced vibration. Those problems exhibit different characteristics in that the angular command impulse excites sway in radial as well as tangential directions. As the angular velocity increases, the natural frequency bifurcates from the nominal natural frequency, resulting in two sway modes in the system. For this kind of problem, Lawrence and Singhose [55] developed ZV2lin shaper. The input shaper consists of three impulses, where the amplitude and timing are determined by solving an optimization problem which seeks minimization of vibration at the end of impulse train. Those are determined not only by natural frequency and damping ratio, but also by the nominal angular velocity and rotation radius. The input shaper length is approximately equal to one damped period, more or less comparable to the length of ZVD shaper.

4.3 Identification of Multi-Mode Vibration Parameters

Input shaping implementation requires values of natural frequency and damping ratio to be known in advance. Those two vibration parameters are identified from decaying residual vibration data obtained from displacement sensor. If more than one vibration mode is present, parameters have to be identified for each mode. Accuracy of the estimation is an important factor that determines effectiveness of the input shaping method.

Identification of natural frequency is carried out in frequency domain. First, residual vibration data is transformed to frequency domain using Fast Fourier Transform (FFT) to get power spectrum. Damped natural frequency ω_d is identified as the frequency which has significant peak power on the power spectrum. In general, the number of detected significant peaks corresponds to the number of vibration modes in the system. Next, we

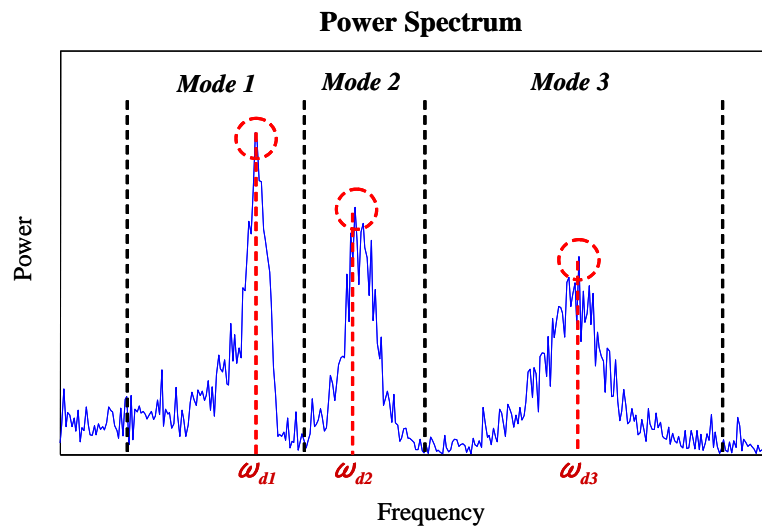


Fig. 4.2 Determination of natural frequencies and decomposition of vibration modes

find minimum power value between two adjacent modes. Those minimum values become boundary of vibration modes. Finally, separation of each mode is obtained by filtering the power spectrum with its boundaries as the filter limits of band-pass filter. The procedure of finding natural frequencies and decomposition of vibration modes is shown in Fig. 4.2, in which three damped natural frequencies (ω_{d1} , ω_{d2} , ω_{d3}) corresponding to three vibration modes are identified.

Estimation of damping ratio is carried out in time domain. Each mode i identified through the band-pass filter, as shown in Fig. 4.2, is inverse-transformed back to time domain for analysis. In time domain, damping determines how much vibration decays. Logarithmic decrement method is one of the easiest method to find damping ratio of a vibrating system in time domain. The principle can be explained in Fig. 4.3. By observing magnitudes of any two successive peaks Y_n and Y_{n+1} , one can calculate value of the logarithmic decrement δ_i , and eventually damping ratio ζ_i according to the following formula:

$$\delta_i = \ln \frac{Y_n}{Y_{n+1}} \quad (4.11)$$

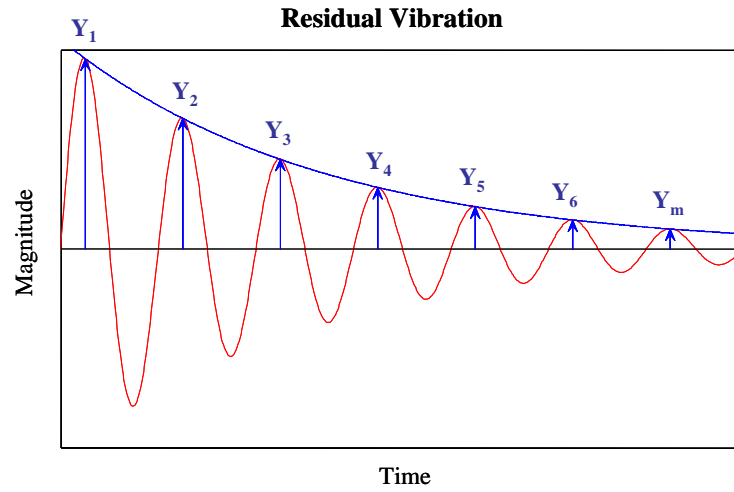


Fig. 4.3 Estimation of damping ratio

$$\zeta_i = \frac{\delta_i}{\sqrt{(2\mu)^2 + \delta_i^2}} \quad (4.12)$$

In our algorithm implementation, we extend the above principle to cover not only two, but many data points, to increase the estimation accuracy. Notice that the decaying sinusoidal signal of each mode follows Eq. (4.1). First, we identify many peaks on the signal, as shown in Fig. 4.3 as Y_1 until Y_m . In other words, we pick all values where the sine term on Eq. (4.1) equals one. Therefore, smooth line connecting those maxima points decays exponentially, following:

$$y_i(t) = A_i \frac{\omega_{ni}}{\sqrt{1 - \zeta_i^2}} e^{-\zeta \omega_{ni} t} \sin \left(\omega_{ni} \sqrt{1 - \zeta_i^2} t \right) \quad (4.13)$$

where i is the mode number and N is the total number of modes.

Thus if we take natural logarithm of those values, we will get a linear decreasing function of time:

$$\ln(y_i(t)) = -\zeta_i \omega_{ni} t + \ln \left(A_i \frac{\omega_{ni}}{\sqrt{1 - \zeta_i^2}} \right) \quad (4.14)$$

The slope of the above linear function is $a_i = -\zeta_i \omega_{ni}$, which is a function of damping ratio and undamped natural frequency. Because residual vibration decays over time, value of the slope a_i should be negative. The value can be identified from the points by least square linear method. Having value of the slope, we can estimate damping ratio using the following formula:

$$\zeta_i = \frac{-a_i}{\sqrt{\omega_{di}^2 + a_i^2}} \quad (4.15)$$

where ω_{di} is the damped natural frequency, whose relationship with the undamped natural frequency is expressed by:

$$\omega_{ni} = \frac{\omega_{di}}{\sqrt{1 - \zeta_i^2}} \quad (4.16)$$

In addition, we can also identify value of impulse amplitude A_i . If the parameters ω_{ni} and ζ_i are already identified, amplitude A_i is the only one left in Eq. (4.14), which can be easily calculated. For complete reconstruction of original sinusoidal signal, actually we need to identify also the signal phase. However, for the input shaping purpose, the phase value of residual vibration signal is not needed.

Although it is possible to detect any number of modes in the system, not all detected modes have significant contribution to the system vibration. The algorithm decides whether a mode is significant or not by checking each separated mode in time domain. If the vibration range is more than acceptable level, then it is significant. Otherwise, it is not significant. Only significant modes will be taken into account in the input shaping calculation. Fig. 4.4 shows the flow chart of the overall parameter identification procedure and flow chart of the estimation of damping ratio.

We will demonstrate the above procedure on a simple numerical simulation. One

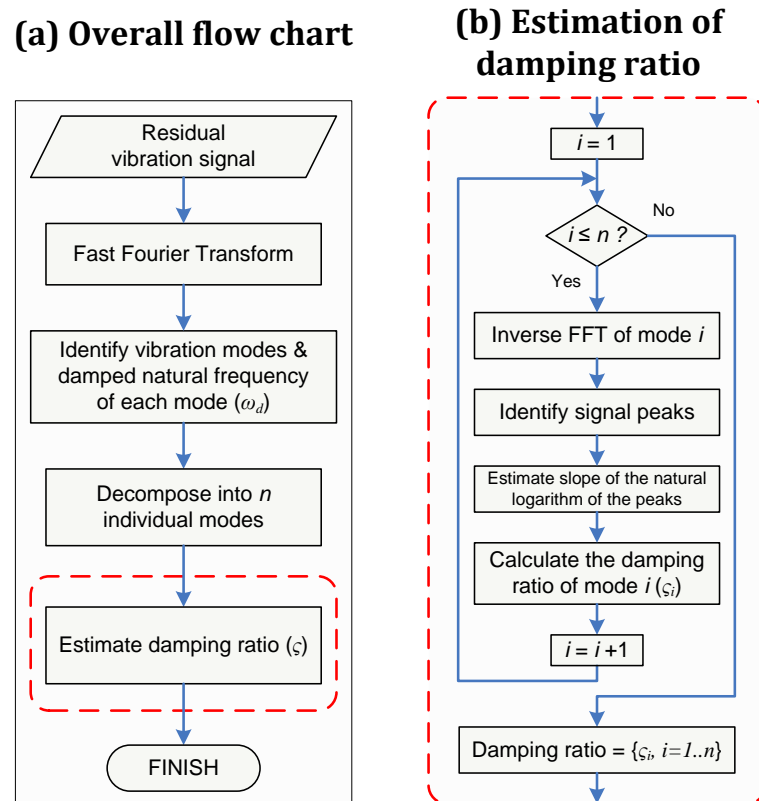


Fig. 4.4 Flow chart of the vibration parameters identification algorithm:

(a) Overall flow chart, (b) Estimation of damping ratio

hypothetical residual vibration signal is composed of three independent decaying sinusoidal signals, with natural frequency 10, 20, and 25 Hz, respectively. Damping ratios are 0.02, 0.04, and 0.01, while impulse amplitudes are 0.5, 0.3, and 0.2, respectively. In addition, random noise is introduced to the signal to resemble typical noise of laser sensor data. Data are sampled at 1000 Hz.

The power spectrum of the signal, shown in Fig. 4.6, is obtained by using FFT. From the graph, it can be seen easily that there are three modes composing the signal, with peaks identified at around the nominal natural frequencies. In general, frequency resolution of a power spectrum is determined by measured signal length, which is three seconds in this case. Higher accuracy in identification of natural frequency, if needed, can be obtained by applying interpolation.

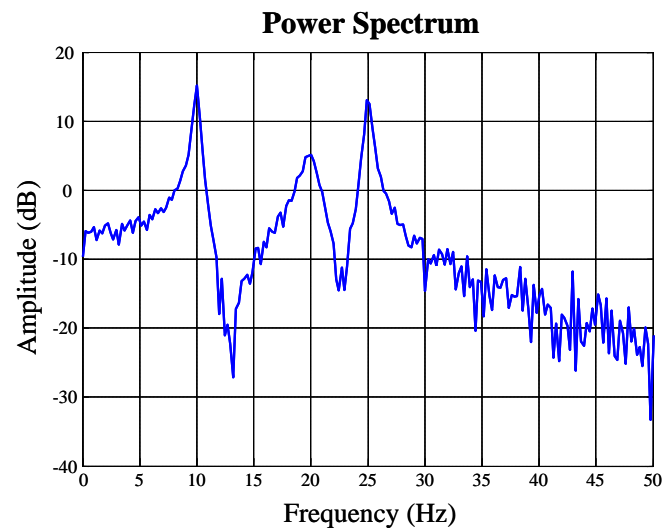


Fig. 4.6 Power spectrum of the numerical example

Three damped natural frequencies, corresponding to the three modes can be identified using the above algorithm: 9.996, 19.915, and 24.987 Hz. The next step is to decompose the spectrum into separate modes by band-pass filtering the spectrum according to the

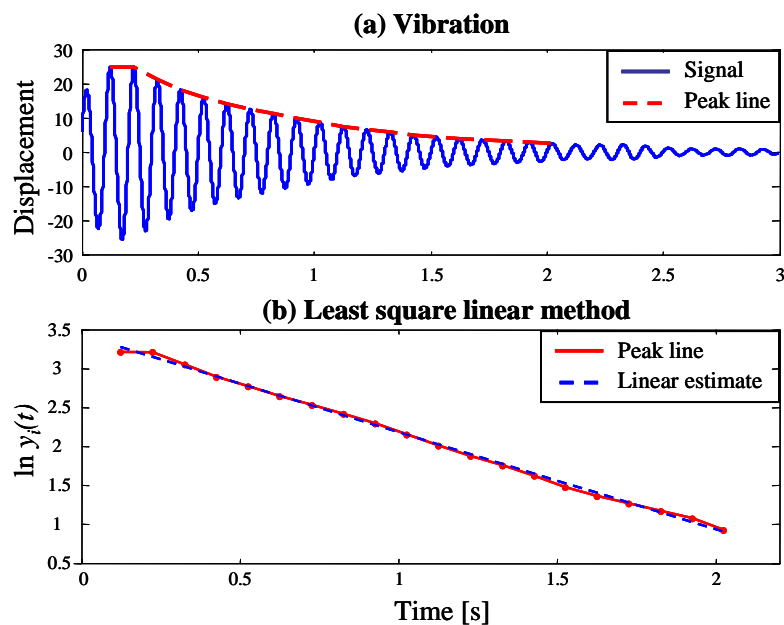


Fig. 4.5 Damping ratio estimation of the numerical example

Table 4.1 Identified parameter values of the numerical example

Mode	Natural frequency ω_{ni} (Hz)		Damping ratio ζ_i	
	Original	Identified	Original	Identified
1	10	9.998	0.02	0.0198
2	20	19.932	0.04	0.0410
3	25	24.988	0.01	0.0101

identified modes and convert it back to time domain where identification of damping ratio is carried out. Fig. 4.5(a) shows the first vibration mode in time domain representation. In that figure, the identified peaks are connected by dashed line. The natural logarithm values are shown connected by solid line in Fig. 4.5(b). In the figure, the dashed line is the best linear estimate of the values, as obtained by the linear least square method. Gradient of the line slope is -1.2467, giving damping ratio value of 0.0198 according to Eq. (4.15). Damping ratio of the rest two modes are obtained by the same procedure. The undamped natural frequencies are calculated using Eq. (4.16). Table 4.1 shows all identified parameter values, as compared to the original values. It shows that the developed procedure can estimate vibration parameters of the signal quite well.

Fig. 4.7 plots the signal constructed from the identified parameters against the original signal. The general shape of the original signal can be well estimated by the algorithm. Calculation time through all procedure is less than 1 second, and therefore there is no problem in industrial application.

4.4 Implementation of Multi-Mode Input Shaping

4.4.1 Implementation procedure

The developed tool takes input reference and vibration data as the input. In our semiconductor wafer experiment case, input reference is velocity pulse command.

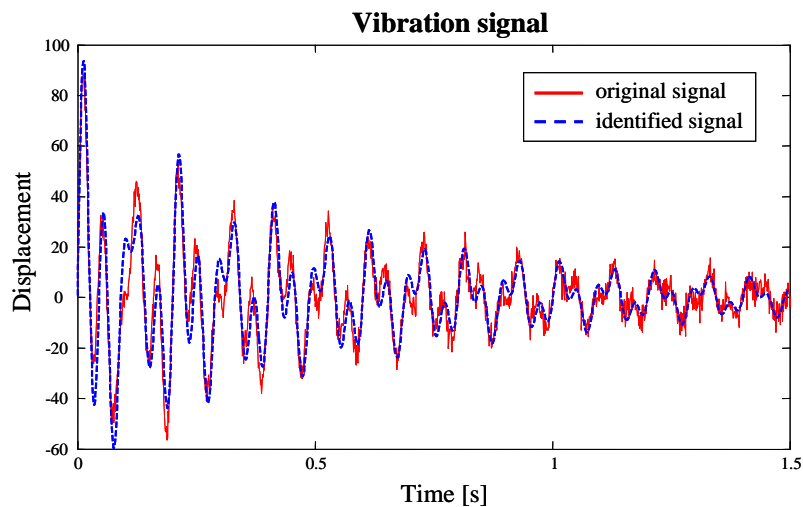


Fig. 4.7 Comparison of original signal and identified signal of the numerical example

Vibration data is the measured value of residual vibration as the result of actual operation of the robot. Algorithms are implemented in MATLAB development environment. Typical program execution time, from vibration parameters identification until final trajectory generation, is well under one second.

The function of the transfer robot is to pick up, transfer, and place semiconductor wafers during production activity of semiconductor factory. It is placed vertically on ground at the center of a production cell which comprises of several production facilities and storage. First, raw (unshaped) input reference is used to move the robot while taking measurement of residual vibration. The acquired vibration data is then analyzed off-line by the vibration parameters identification part to determine natural frequency and damping ratio for each mode. Based on those values, input shaper is designed. The result is amplitude and timing of the shaper impulses. And finally, the tool creates new shaped input reference by convolving raw input with shaper impulses.

4.4.2 Experiment

Fig. 4.8 shows residual vibration signal obtained from laser sensor on a simple straight path movement of the robot. The start time of the residual vibration is the time when the

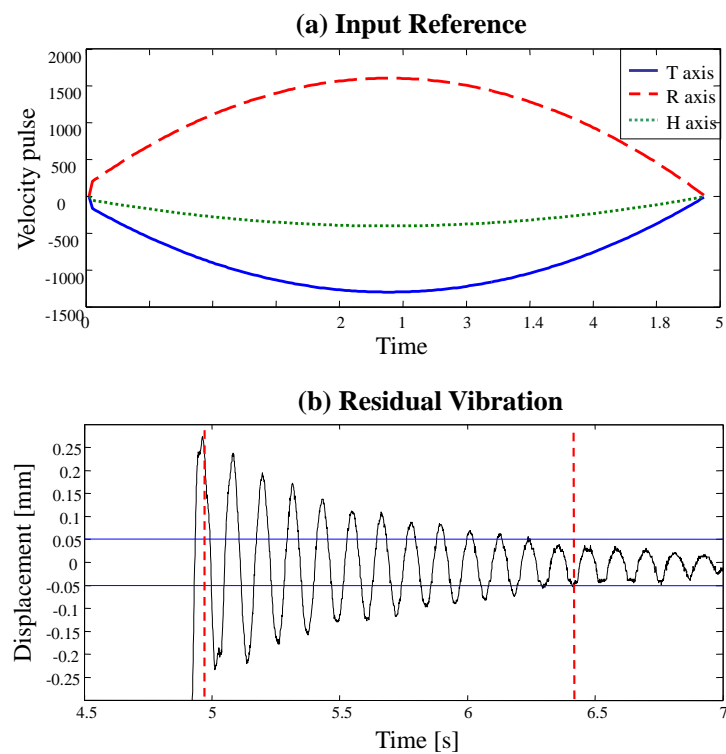


Fig. 4.8 Experiment on straight path with shaped reference:
(a) input reference (b) residual vibration.

robot has finished the commanded movement. The finish time of the vibration is defined as the latest time when the robot arm tip displacement is outside the acceptable vibration range, which is $\pm 50 \mu\text{m}$ from the steady state value. By that definition, the residual vibration starts at time 4.961 second and finishes at time 6.416 second. The start and finish time are marked by two dashed vertical red lines. Time span between those values is the settling time. In this case, vibration settling time is 1.455 second.

FFT analysis of the residual vibration by the algorithm reveals that only one mode is significant. The natural frequency is 8.6691 Hz, while the damping ratio is 0.0252. For those values, the amplitude of the designed input shaper impulses are 0.5198 and 0.4802, respectively, while the impulse time is 0.0577 second. Because command pulse sampling time is 10 ms, the impulse is extrapolated proportionally to the nearest sampling times. The result obtained by applying the shaped input reference to the robot is shown in Fig. 4.9. It can be noticed that the magnitude of the residual vibration has been reduced. The settling

Table 4.2 Experiment result of straight line motion

Command input	Movement time [s]	Settling time [s]	Total time [s]
Original	1.95	1.455	3.405
Shaped	2.01	0.788	2.798

time decreases to 0.788 second, which is approximately 45% reduction from the original settling time. However, the robot movement time is 2.01 second, which increases slightly from 0.95 second of the original movement time, as a consequence of the input shaper length. Table 4.2 summarizes the result.

In practice, robot trajectories are seldom as simple as straight path. In that case, many

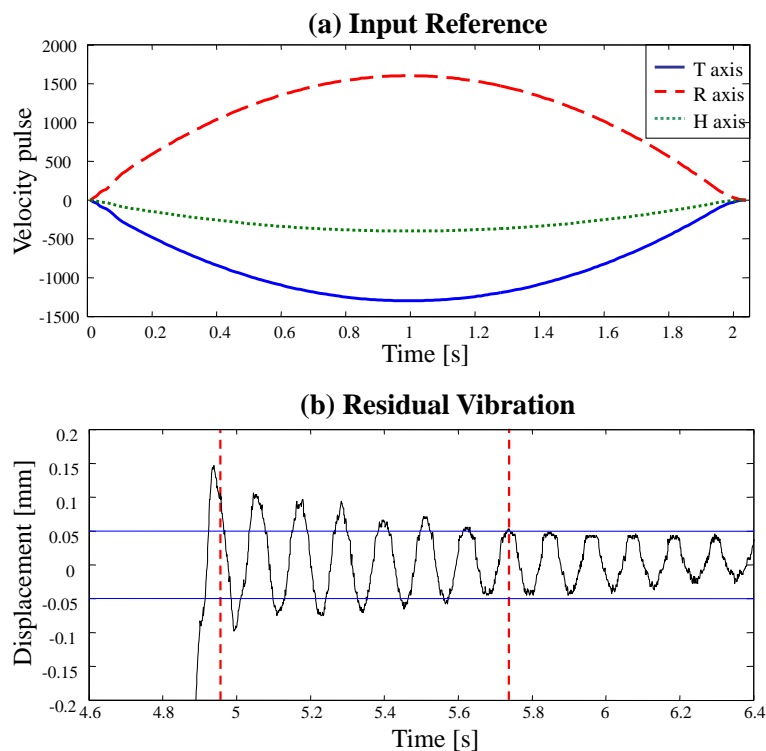


Fig. 4.9 Experiment on straight path with shaped reference:
(a) input reference (b) residual vibration.

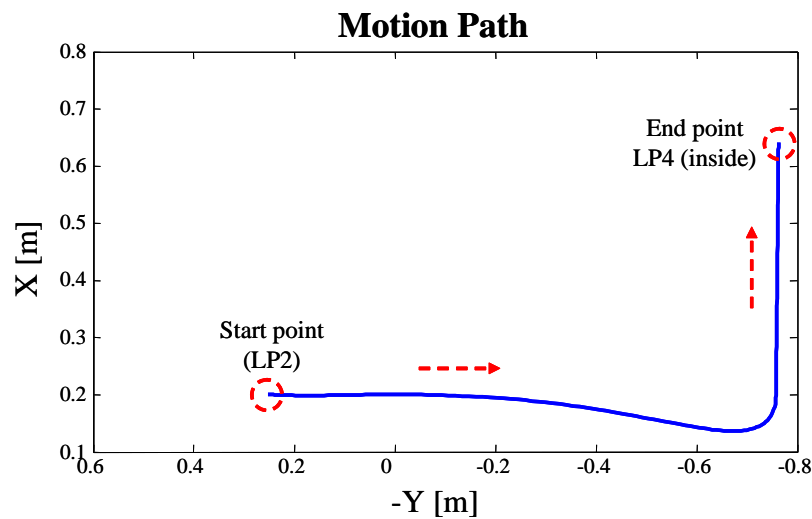


Fig. 4.10 Motion path LP2 \rightarrow LP4

vibration modes can be excited. Another experimental case is movement from port LP2 to port LP4. It starts from the ‘home’ position of port LP2, moving towards LP4, and finally approaching inside LP4. Fig. 4.10 shows the motion path.

First, the robot arm is operated using the raw (unshaped) input reference. The residual vibration data, measured on X direction and sampled at 1000 Hz, is then used as input for the system identification algorithm, because the final motion at the end time is in X direction. Five vibration modes, along with their respective natural frequency and damping ratio, can be detected by the parameter identification algorithm.

In order to show the effect of each mode, trials are conducted several times for the same path but with different input reference. The first trial uses original input reference, while the next three uses the shaped reference of the first three, four, and five modes, respectively. In each trial, vibration measurement was done for both X and Y direction to evaluate the residual vibration. Table 4.3 summarizes the result. The movement time gradually increases because of longer input shaper used by each additional mode that is taken into account. On the other hand, the settling time, as well as the total time, gradually decreases. The best result is obtained by the input shaper that takes all five modes into account. The above is true for vibration measured in both X and Y direction, where the vibration of robot arm tip

Table 4.3 Motion time, settling time, and total time of LP2-LP4 movement

(a) measured in X direction

	Original	Shaped (with partial modes)		
		1,2,3	1,2,3,4	1,2,3,4,5
Movement time (s)	1.71	1.85	1.87	1.9
Settling time (s)	4.504	1.181	1.106	0.87
Total time (s)	6.214	3.031	2.976	2.77

(b) measured in Y direction

	Original	Shaped (with partial modes)		
		1,2,3	1,2,3,4	1,2,3,4,5
Movement time (s)	1.71	1.85	1.87	1.9
Settling time (s)	3.27	1.492	1.362	0.958
Total time (s)	4.98	3.342	3.232	2.858

settles in 0.87 and 0.958 second, respectively. The big difference between the vibration settling times and the total time of shaped trajectory and those of original trajectory shows the effectiveness of our developed solution.

4.5 Integration of Trajectory Planning and Vibration Control

4.5.1 Optimization Model

With its limited jerk, the smooth trajectory obtained by the cubic spline optimization should induce less vibration compared with non-smooth trajectory. In the optimization, the maximum jerk can be lowered even more to reduce the vibration. Nevertheless, vibration still remains, and reducing the jerk would increase the motion time. In order to further suppress the remaining vibration, we employ the input shaping approach, utilizing information of the system flexible modes.

In a single-mode flexible system with natural frequency ω_n and damping ratio ζ , the

transfer function between the tracking error and its acceleration reference trajectory is often modeled as a second order harmonic oscillator as follows:

$$G(s) = \frac{E(s)}{A_{REF}(s)} = \frac{1}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (4.17)$$

The sinusoidal response of such system in time domain when the input is a single pulse of amplitude a at time t_0 with sufficiently small pulse width τ is:

$$x(t) = \frac{\tau a}{\omega_d} e^{-\zeta\omega_n(t-t_0)} \sin \omega_d(t - t_0) \quad (4.18)$$

where ω_d is the damped frequency. Any arbitrary forcing function can be expressed as sum of pulse functions, and therefore the response is the sum of its individual pulse responses.

The simplest implementation of input shaping in an undamped single mode vibrating system is to split the original reference input into two inputs having half of the original amplitude. One of them is delayed by half of natural period, thus cancels the vibration response of the first one. This can be done by convolution of the trajectory with an input shaper function, which consists of two impulses with certain amplitudes and timing. For a vibrating system with damping, adjustment is made to take into account the damping effect of damping.

By applying input shaping, the trajectory is altered. As a result, the motion path in task space is also changed and may violate the position constraints. That is why the input shaping cannot be implemented by simple convolution with the trajectory given by the optimization. Instead, we must implement the input shaping inside the optimization iterations. The cubic spline optimization problem now becomes: *find a time-minimizing variables (segment times and knot values) where the shaped trajectory satisfy all constraints*. At each iteration, the trajectory is shaped before checking for constraints. This integration of input shaping and cubic splines trajectory generation guarantees that position constraints are always observed while at the same time exploiting available allowances to minimize motion time.

To implement the input shaping, we found it easier to work with jerk impulses. We define jerk impulse as the difference between jerks of subsequent knots:

$$p_{ij} = \ddot{\theta}_{ij} - \ddot{\theta}_{i-1,j} \quad (4.19)$$

and t_i as the time where those impulses are applied:

$$t_i = t_{i-1} + h_{i-1} \quad (4.20)$$

where $t_1 = 0$. As a result of the convolution of the jerk impulses with input shaper, the number of impulses doubles (in case of ZV input shaper). We denote the new jerk impulse variable as p'_{ij} , where after convolution i runs from 1 to $2n$. From p'_{ij} , by integration we can derive the associated joint jerk, acceleration, velocity, and joint angle at knots. Then from those values we can obtain the segment trajectory $q_{ij}(t)$ at every time point by cubic interpolation.

The overall procedure of the integrated cubic spline trajectory planning and input shaping is similar to the one in Section 3.3, with the following modification to step (2):

- (2.a) Using the joint acceleration values, calculate the joint velocities and jerks. Furthermore, torques can be calculated using the dynamics equations of the robot arm.
- (2.b) Calculate the jerk impulses p_{ij} . Apply input shaping by convolving the jerk impulses with the desired input shaper, resulting in the shaped jerk impulses p'_{ij} . Obtain the new joint jerk, acceleration, velocity and angle by integration. Further, $q_{ij}(t)$ can be obtained by cubic interpolation.

By prior analysis of residual vibration signal of the experimental system, we estimated beforehand that there are two dominant flexible modes at the frequencies $\omega_n = [10.58, 9.14]$ Hz with damping ratio $\zeta = [0.0185, 0.0285]$. The vibration signal was measured by a displacement sensor placed near the end point of the motion path.

Fig. 4.11 shows the generated cubic spline trajectory with ZV input shaping for the example LP1-LP3 motion used in previous section. As before, fixed points are denoted by small rectangles, while other points are denoted by circles. By comparing Fig. 4.11 with Fig. 3.5, we can readily see that the number of points are now quadrupled because of the ZV input shaping for two vibration modes. In this case, one impulse in the original

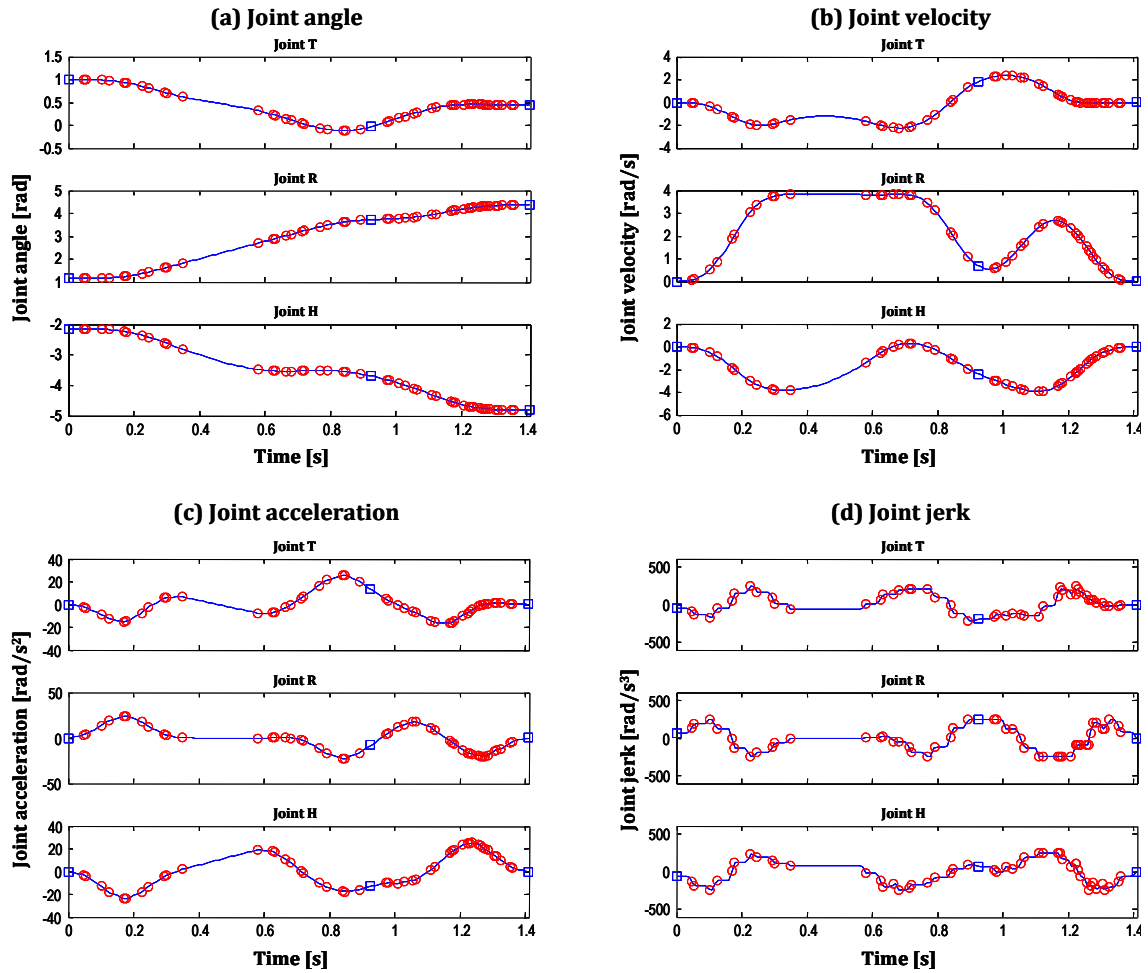


Fig. 4.11 Generated trajectory of LP1-LP3 motion with ZV input shaping:

(a) joint angle, (b) joint velocity, (c) joint acceleration, (d) joint jerk

trajectory will be eventually split into four impulses. The total motion time of the ZV shaped trajectory is 1.41 second, or 0.13 second longer than the unshaped trajectory. The time increase is approximately equal to the input shaper length, which is 0.102 second. The difference comes from the fact that the optimization has to ensure the position constraints are observed.

4.5.2 Experiment Results and Discussion

Experiments are conducted in order to verify the trajectory as well as to measure the real residual vibration induced by the motion. We consider three motion paths: LP1-LP3, LP2-

LP3, and LP4-LP3. For each motion path, we generate the cubic spline trajectory and the shaped trajectories with ZV and ZVD input shaping. Those trajectories are used as reference input to move the real robot arm.

Residual vibrations are measured by using a displacement sensor which is placed near the end of the motion path. It captures the movement of the end link in X and Y directions after the commanded motion is finished. Table 4.4 lists the motion time, vibration time, maximum residual vibration amplitude, and the calculation time of each trajectory. Motion time is the time required to execute the motion from start point to end point. Vibration time is defined as the time required until the residual vibration settles within some tolerated vibration range, which is $\pm 50\mu\text{m}$ for our current application. Maximum vibration is the biggest difference between the maximum and minimum displacement in the residual vibration signal. Because vibration is measured in both X and Y direction, highest values are used. Calculation time is the time required to execute the optimization algorithm until solution is obtained. For each trajectory, the optimization is executed 20 times, and the value shown in the table is the average calculation time. The processor used for calculation is Intel Core2 Duo running at 3 Ghz.

Our optimization framework can generate trajectories with short motion time in reasonably short calculation time. For comparison purpose, we used another approach [43]

Table 4.4 Experiment results

Motion path	Trajectory type	Motion time [s]	Vibration time [s]	Maximum vibration [mm]	Calculation time [s]
LP1 - LP3	Cubic	1.28	1.381	0.669	13.65
	Cubic + ZV	1.41	0.495	0.139	49.53
	Cubic + ZVD	1.48	0	0.075	120.91
LP2 - LP3	Cubic	0.91	1.387	0.693	9.85
	Cubic + ZV	1.02	0.492	0.13	29.18
	Cubic + ZVD	1.07	0	0.084	77.93
LP4 - LP3	Cubic	0.97	1.404	1.043	8.36
	Cubic + ZV	1.1	0.29	0.115	27.78
	Cubic + ZVD	1.17	0.05	0.101	70.94

to generate trajectories for those motion paths. The algorithm models each of the last two joints as two entities separated by a rotational spring, which introduces flexibility to the end effectors. At one side is the motor that provides needed torque, and at the other side is the driven link. Stiffness coefficient of the spring is identified experimentally. Sequential quadratic programming iterative method is then used to find the optimal trajectory that minimizes the motion time and the vibration at the robot end effectors. The approach discretizes the time domain to many small sampling periods. The obtained motion times are 1.95 second, 1.81 second, and 2 second, for LP1-LP3, LP2-LP3, and LP4-LP3 motion path respectively. Those are significantly longer than the ones generated by our cubic spline framework even when the input shaping is applied, i.e. 1.48 second, 1.07 second, and 1.17 second, respectively.

From Table 4.4, we can observe that the trajectories obtained by the cubic spline optimization consistently results in the longest vibration time and the highest maximum vibration in all motion paths, as compared to the shaped ones (cubic+ZV and cubic+ZVD). This is not unexpected because there is no explicit consideration of vibration suppression in the trajectory generation without input shaping. The vibration time and maximum vibration are substantially smaller in the trajectories obtained by integrated input shaping and cubic spline optimization. The lowest vibration is obtained by the integrated ZVD shaped trajectories where in two instances (LP1-LP3 and LP2-LP3) the residual vibration is contained within tolerable limit and thus zero vibration time, and in another instance (LP4-LP3) the vibration time is very small. The integration of input shaping lowers the frequency content in the modeled vibration modes, therefore the vibration in those frequency ranges could be suppressed. Fig. 4.12 shows the comparison of the residual vibration of the three trajectories in LP1-LP3 motion path as measured in Y direction. It is clear that the vibration generated by the ZV and ZVD integrated trajectories are significantly smaller than that in the cubic spline trajectory. In case of ZVD, the residual vibration is always inside the tolerable limit of $\pm 50\mu\text{m}$.

In all cases, the motion time of the ZV shaped trajectories are longer than that of the non-shaped ones, and ZVD shaped trajectories are the longest. However, the increase in motion time is always less than the decrease in vibration time. That makes the motion time

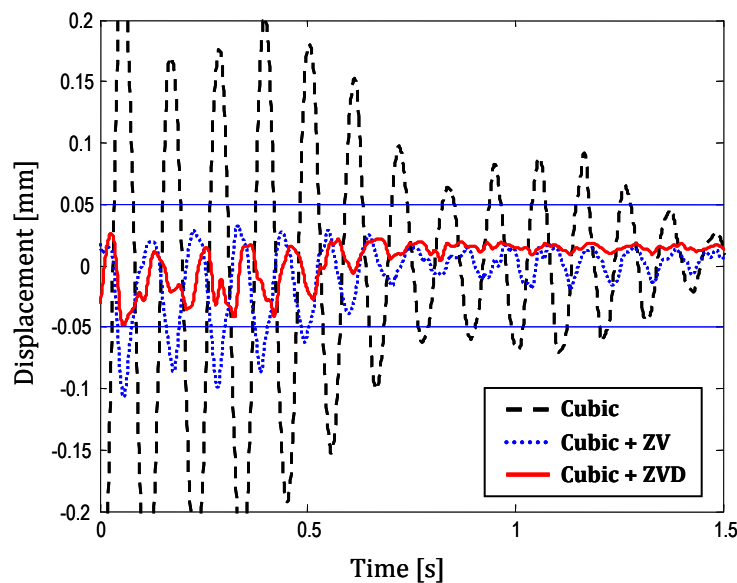


Fig. 4.12 Comparison of residual vibration of three trajectories in LP1-LP3 motion (cubic, cubic + ZV, cubic + ZVD), measured in Y direction

increase justifiable when we consider the total time as the sum of motion time and vibration time. In real practice, the robot arm is not allowed to do any operation while the residual vibration exists.

4.6 Summary

Input shaping is a simple yet effective technique to suppress vibration. For the technique to work, the natural frequency and damping ratio of the vibrating system have to be identified. For that purpose, we have developed an identification tool, which is later integrated with the input shaping itself to realize practical use of the input shaping principle in various applications. In this chapter, it is applied to suppress the residual vibration of a semiconductor wafer transfer robot arm. The parameter identification tool works by decomposing the vibration signal into several signal which each represents one vibration mode. This approach is easier to comprehend (compared to concurrent fitting approach [54]) because it retains the physical meaning of the vibration modes.

Input shaping can be applied to any arbitrary trajectory. However, the convolution of

trajectory with an input shaper changes the motion path. While it may not cause problem in many applications, in this particular semiconductor wafer transfer robot arm, the change in the motion path may cause the robot to collide with the workspace boundary. In order to avoid that, we proposed an integrated algorithm of trajectory planning and input shaping based vibration suppression. The effectiveness of the approach to generate trajectories with short motion time and low vibration has been shown through simulations and experiments. In addition, the calculation time to solve the optimization problem is still practical and acceptable for offline calculation.

CHAPTER 5

Trajectory Generation and Sloshing Suppression of Liquid Container Transfer Robot Arm

5.1 Introduction

This chapter describes the trajectory planning and sloshing suppression for an automatic liquid transfer system. The integrated framework is an extension of the cubic spline based framework described in Chapter 3 and Chapter 4, at least in two points: the input shaping is applied in task space and the obstacle avoidance is explicit. The input shaping is done in task space because sloshing occur because of the motion of the liquid container. In addition, the framework also must be able to handle the rotation of the robot hand, which allows the transfer system to control the orientation of the liquid container in addition to the three translation directions.

Section 5.2 describes the analytical sloshing model and its equivalent pendulum model. The equivalent pendulum model is used to simulate the effect of liquid container motion. The development of the trajectory planning and sloshing control is explained in Section 5.3.

To analyze the effect of vertical motion to the sloshing, a numerical simulation of sloshing is built in Section 5.4. Finally, experiments are performed, explained in Section 5.5, to assess the performance in real motion cases.

5.2 Sloshing Model of Cylindrical Container

As explained in Chapter 2, the liquid transfer system uses a cylindrical container attached to the end effector of the robot arm. Although the theoretical model exists for such shape, for control analysis and simulation, the phenomenon of sloshing is often modeled by its simpler mechanical equivalents, for example pendulum and mass-spring-damper system.

5.2.1 Theoretical Model

The theoretical natural frequency of sloshing inside an upright cylindrical container can be derived from Navier-Stokes equations as [56]:

$$\omega_n^2 = \left(g \xi_n / R \right) \tanh \left(\xi_n h / R \right) \quad (5.1)$$

where g is the gravity constant, R is the container radius, h is the liquid height, and ξ_n is the root of the derivative of Bessel function of the first kind where n is the sloshing mode. For the fundamental first mode, ξ_1 equals 1.841.

5.2.2 Equivalent Pendulum Model

In the pendulum model, the sway of the pendulum corresponds to the elevation of the liquid surface. A simple pendulum system, as depicted in Fig. 5.1 (a), consists of a point mass suspended by a massless rigid cable of length L to a friction-less movable support. For the case where the support moves in one dimension line, it can be described by the following equation of motion:

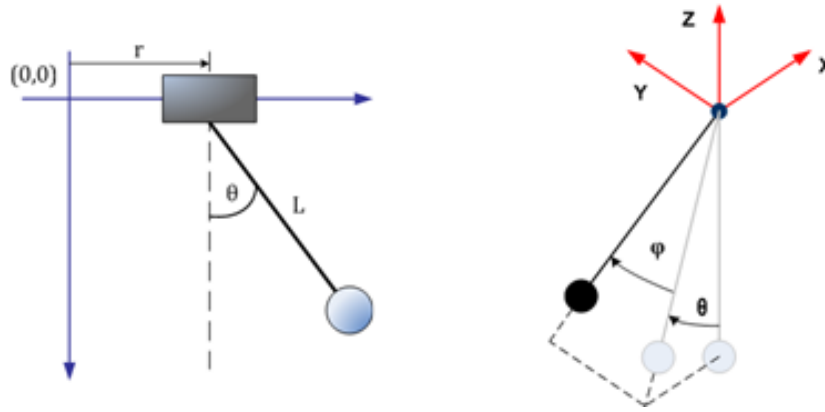


Fig. 5.1 (a) Simple pendulum, and (b) spherical pendulum

$$L\ddot{\theta} + g \sin \theta = -\ddot{x} \cos \theta \tag{5.2}$$

where x is the distance of the support from origin.

In cases where the support moves in two-dimensional horizontal space, the system can be modeled as a spherical pendulum. Fig. 5.1 (b) illustrates the spherical pendulum model, where the pendulum bob can freely swing in two degrees of freedom spherical space as response of movement of the support in horizontal planar space. Angle θ is the angle between plane ZY and the mass, and φ is the angle in the perpendicular direction.

The position of the pendulum bob in Cartesian coordinate system is:

$$x_p = -L \sin \theta \cos \varphi \tag{5.3}$$

$$y_p = L \sin \theta \tag{5.4}$$

$$z_p = -L \cos \theta \cos \varphi \tag{5.5}$$

The equations of motion of the spherical pendulum can be derived as follows [52]:

$$\ddot{\theta} = 2\dot{\theta}\dot{\varphi} \tan \varphi - \frac{g \sin \theta}{L \cos \varphi} + \frac{1 \cos \theta}{L \cos \varphi} \ddot{x} \tag{5.6}$$

$$\ddot{\varphi} = -\dot{\theta}^2 \dot{\varphi} \tan \varphi - \frac{g}{L} \cos \theta \sin \varphi - \frac{1}{L} (\ddot{x} \sin \theta \sin \varphi + \ddot{y} \cos \varphi) \quad (5.7)$$

where x and y is the position of the support. Indeed, when there is no acceleration in Y direction, and initially $\varphi = 0$, the above equations of motion will reduce to the simple pendulum case in (5.2).

5.3 Trajectory Planning and Sloshing Suppression

The liquid transfer system concerned uses the Mitsubishi PA 10-7C robot arm, which is explained in detail in Section 2.3. The links E2, W1, and W2 form a spherical wrist, where the three rotational joint axes intersect at a common point. This robot configuration simplifies the inverse kinematic calculation because it can be decoupled into two simpler

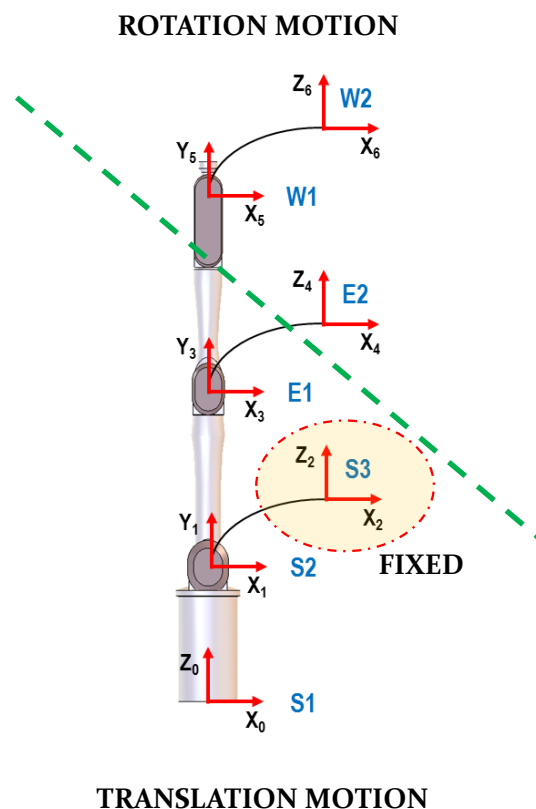


Fig. 5.2 Decoupling the kinematics into translation and rotation subsystems

subsystems, as shown in Fig. 5.2. The first three rotational links S1, S2, and E1 provides the positioning in three dimension space, while the spherical wrist adjusts the orientation of end effector.

Accordingly, the trajectory planning of the liquid container transfer system is decoupled into planning of the translation of the robot wrist position and rotation of the robot hand. The translation part is designed as joint space cubic spline trajectory, while the hand rotation is designed in task space to maintain upright orientation of the liquid container. Upright orientation means the container has only one degree of freedom: the rotation along the vertical Z axis of fixed coordinate system. Those translation and rotation trajectories are then combined to make the final quick slosh-free trajectory solution. The trajectory planning steps are shown in Fig. 5.3, which clearly depicting whether each step is done in joint space or task space. Transformation between joint and task space is carried out using direct and inverse kinematics.

Fig. 5.3 (a) shows the overview of planning the translation motion. It starts with specifying the start point, end point, as well as several knots between them in task space, which are then transformed to joint space as the initial solution to the cubic spline optimization. The optimization step encompasses joint space as well as task space. Although the trajectory is generated in joint space, it is possible to define task space constraints in the optimization, for example obstacle avoidance. The obtained solution is joint space trajectory, which is transformed back to task space, where the command shaping takes place. The solution of the translation motion is obtained as joint trajectories of links S1, S2, and E1.

Fig. 5.3 (b) shows the steps of planning the rotation motion of the robot hand. The input is the angles of the container along Z axis relative to XZ vertical plane at the start and the end of motion. The shaped rotation trajectory is then transformed to the joint space as the solution of the rotation part of the motion in form of joint trajectories of links E2, W1, and W2. Although the rotation trajectory generation is independent to the translation trajectory generation, it has to be performed after because the former needs the orientation history of the translation motion as well as the total motion time in order to coordinate the overall motion.

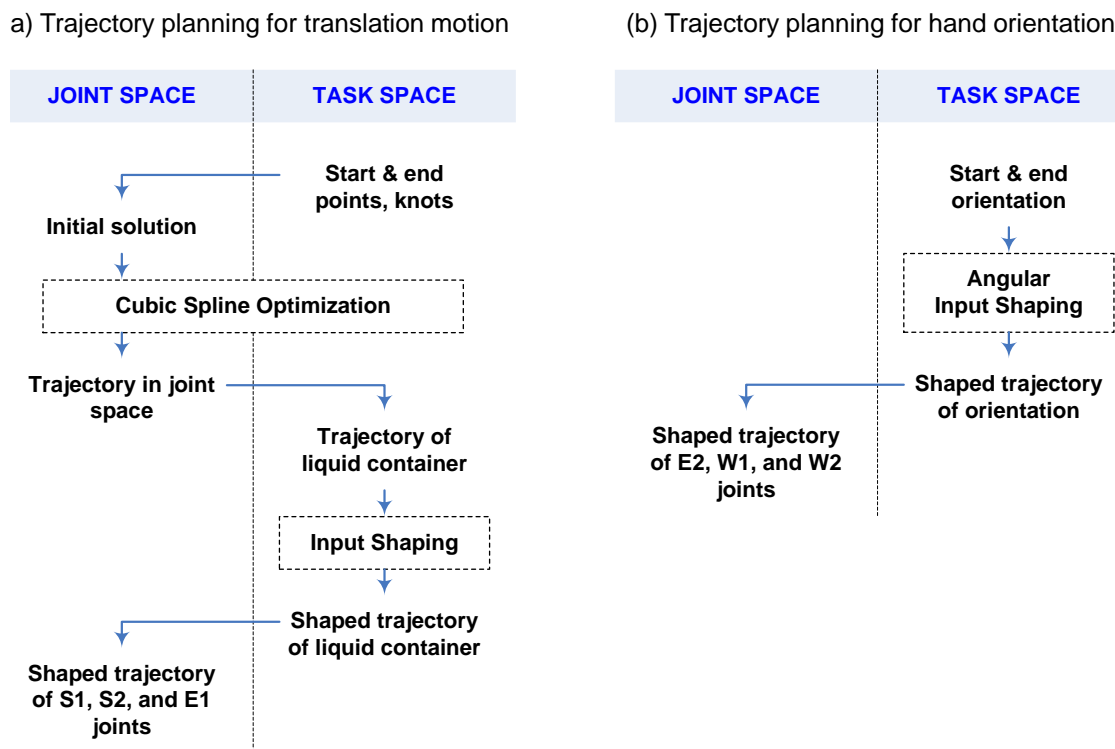


Fig. 5.3 Trajectory planning steps for (a) translation motion, and (b) hand orientation

5.3.1 Translation Motion Planning

The trajectory planning for translation motion builds up from the cubic spline framework as already explained in Section 3.3. The trajectories of S1, S2, and E1 joints are each designed as piecewise cubic splines. The joint trajectories are coupled to each other by using the same value of time interval variables h_i . We formulate the trajectory generation problem as a nonlinear optimization problem. The objective function is combination of motion time and squared acceleration. There are several constraints to consider in the trajectory generation: robot position in the task space, joint velocity, and joint jerk. The framework uses the ‘floating’ approach, which means both the segment time and the joint angles (knots location) are included as the optimization variables. The optimization problem is thus as follows:

$$\min_p F = \sum_{i=1}^n h_i + w_a \int \ddot{q}_i^2 dt$$

subject to

$$|q_{ij}^*| \leq q_j^{\text{MAX}}; \quad i = 1..n, \quad j = 1..m \quad (5.8)$$

$$|v_{ij}^*| \leq v_j^{\text{MAX}}; \quad i = 1..n, \quad j = 1..m$$

$$|\ddot{\theta}| \leq J^{\text{MAX}}; \quad i = 1..n + 1, \quad j = 1..m$$

$$d_i(L_j, O) \geq r_{L_j} + r_O; \quad i = 1..n, \quad j = 1..m$$

where p is the optimization variables which contains the segment times as well as the joint angles excluding the fixed start point, end point, and the two virtual points:

$$p = (h_1, \dots, h_n, \theta_{ij}, \dots, \theta_{nm}); \quad i = 3..n - 1, \quad j = 1..m \quad (5.9)$$

Here, all internal knots are part of the optimization variables, instead of pre-defined points. This provides flexibility for the optimization to generate quick motion as well as obstacle avoidance at the same time. If the planned motion is required to pass certain fixed points on the way, they can be added as additional equality constraints. The first and second constraints are joint angle and velocity constraints, according to the physical joint limits of the robot. The third is jerk constraint, to prevent excessive value of joint jerk. The maximum jerk of each joint is set equal to 15 m/s³.

The last constraint is for obstacle avoidance, where the obstacle and robot links are modeled as rigid capsules with radius r_O and r_{L_j} , respectively. The function $d_i(L_j, O)$ is the minimum distance between link L_j and static obstacle O in segment i . The obstacle and links' axis are discretized into several representative points. The distance between a link and obstacle is defined as the minimum distance between the points in the obstacle and the points in the link.

We use the MATLAB Optimization Toolbox implementation of Sequential Quadratic Programming (SQP) for solving the above constrained nonlinear trajectory planning

optimization problem. The calculation time differs depending on the computer used for calculation; it is around 15 seconds for a typical problem size in our Core2 Duo computer.

Fig. 5.4 shows an example motion path of a cubic trajectory generated by the above method in perspective view and top view. The PA-10 robot arm is stationed at origin. The robot has to move its wrist from start point $[0.3, -0.8, 0.2]$ to end point $[0.5, 0.5, 0.2]$ in the task space, which correspond to $\theta_{\text{START}} = [-1.21, -1.25, -0.87]$ rad and $\theta_{\text{END}} = [0.79, -0.97, -1.44]$ rad in the joint space. The robot posture shown in the figure is the configuration at start point. A cylindrical obstacle with radius 0.05 m lies vertically in the workspace at $[0.5, -0.3]$. For this problem, four initial via points are specified randomly between the start and end points. The line going from the start point and the end point is the generated motion path in the task space, which successfully avoids the obstacle. The small spheres on the line denote the location of output via points. The total motion time is 2.7 second. Fig. 5.5 shows the joint trajectories of the example case: the piecewise cubic

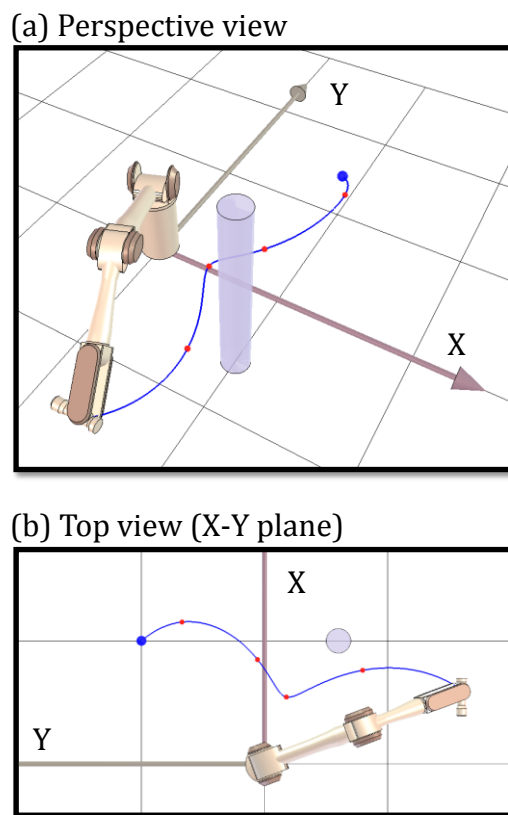


Fig. 5.4 Motion path of the example case of translation motion

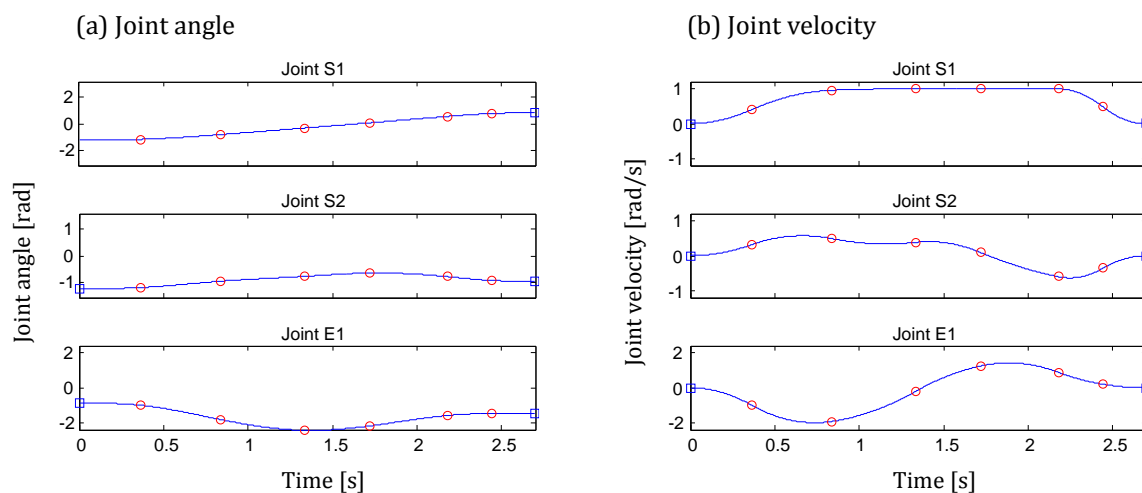


Fig. 5.5 Trajectory of the example case of translation motion:

(a) joint angle, (b) joint velocity

joint angles and the piecewise quadratic joint velocities. The small circles in the figure denote the respective values at the knots, including at the two virtual knots. From the velocity figure, we can understand that the motion time is limited by joint S1, where the velocity hits its limit most of the time.

The next step in the translation motion planning is to suppress the sloshing generated by the above obtained trajectory. For analysis, we use the pendulum model in Section 5.2.2 to represent the sloshing, where the pendulum support is attached at the robot wrist locus and the pendulum sways freely as the robot moves. For the case where the pendulum support can move in a horizontal planar space, we first decompose the input signal to its projection into orthogonal Cartesian axes and then apply input shaping independently to each input signal. By doing this, the spherical pendulum is considered as two independent simple pendulums which act on orthogonal vertical planes such that the sway behavior in one subsystem is not affected by the states in the other subsystem. This simplified approach works well in small angle regions, and indeed linearization of Equations (5.6) and (5.7) by small angle approximations and removing quadratic and higher terms results in two independent equations of simple pendulum.

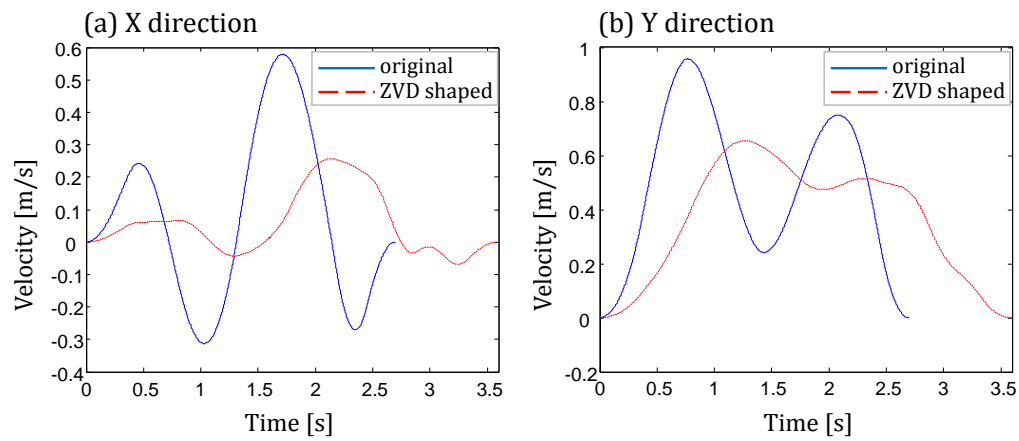


Fig. 5.6 The original and ZVD shaped velocity in task space:

(a) X direction, (b) Y direction

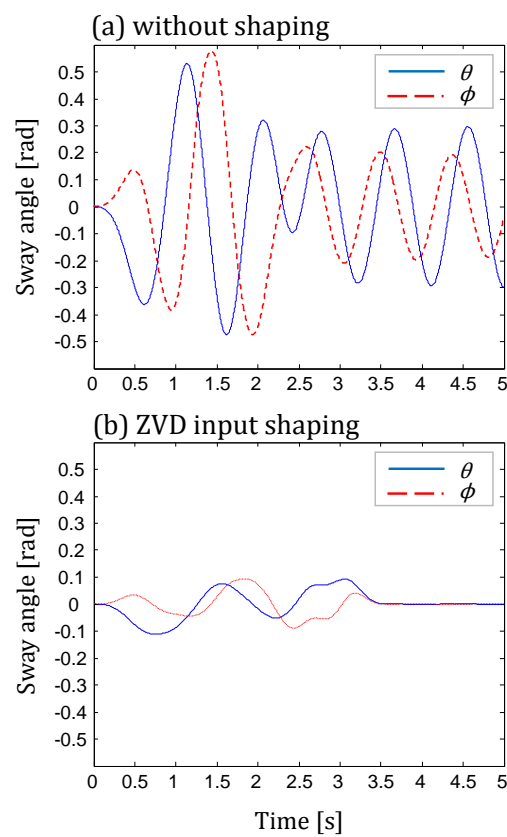


Fig. 5.7 Pendulum sway angles:

(a) without shaping, (b) with ZVD shaping

We use the ZVD input shaping for the sloshing suppression. For that, the velocity trajectory is decomposed into two orthogonal velocities parallel to X and Y axis, respectively. Each velocity trajectory is then convolved individually with ZVD input shaper. The shaped results are then added back and integrated to produce the shaped trajectory in task space. Joint space solution can be obtained by inverse kinematics.

As an example, we simulate a spherical pendulum of length 0.2 meter moving according to the trajectory obtained above. The natural frequency and the vibration period of the pendulum model are 7.0 rad/sec and 0.8973 second, respectively. In sloshing problems, the damping is usually very small, therefore here we assume zero damping in the pendulum model. Fig. 5.6 shows the original and ZVD shaped Cartesian velocities in X and Y directions. The motion time is 2.7 second and 3.6 second in the unshaped and ZVD shaped case, respectively. Fig. 5.7 shows the sway angles in the spherical coordinate system. At the end of motion, sway is much suppressed in the shaped trajectory. Fig. 5.8 compares the projection of the pendulum position at horizontal plane for the two cases: original and shaped trajectories. It is clear from the figure that the shaped trajectory generates significantly less sway than its unshaped counterpart.

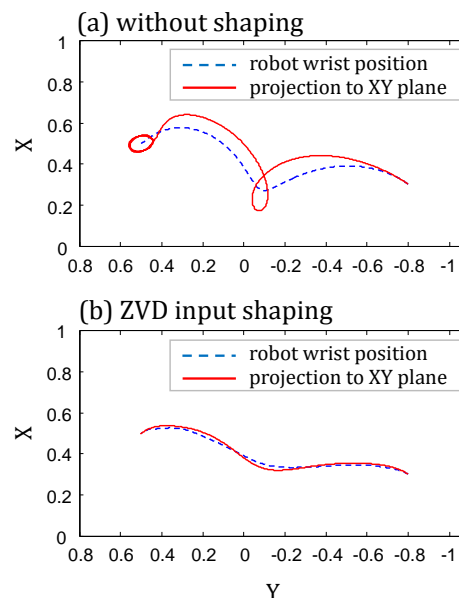


Fig. 5.8 Projection of pendulum bob position in XY plane:
 (a) without shaping, (b) with ZVD shaping

5.3.2 Rotation Motion Planning

The liquid container has to be maintained upright all time. This means there is only one task space DOF: the rotation along the vertical Z axis. Thus the hand rotation motion can be simply parameterized by the angle along Z axis relative to the XZ plane. Here, robot hand means any parts after the wrist, including the last robot link, the container holder, and the liquid container itself. The length of the robot hand is the distance between the wrist locus and center of container, which equals 0.2 meter. The trajectory is generated by applying ZV2lin input shaping to a rectangular velocity profile, in which the maximum angular velocity is:

$$\dot{s} = \frac{\gamma_f - \gamma_s}{T} \tag{5.10}$$

where T is the time of the unshaped translation motion, and γ_s and γ_f are the desired start and finish angle value, respectively. This is to coordinate the translation and rotation motions, so that both motions ends at approximately the same time.

As an example, we use the same spherical pendulum model to simulate the motion of the pendulum support on a circular trajectory of length $\pi/2$ radian, where the robot hand

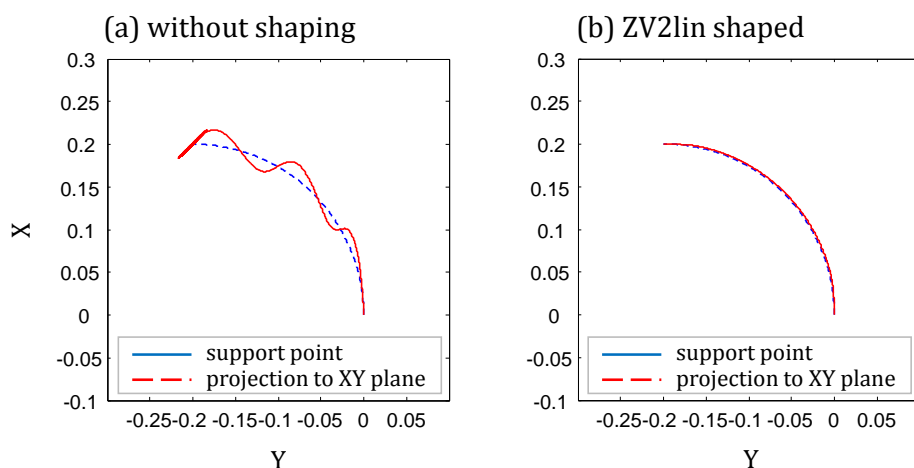


Fig. 5.9 Pendulum position in XY plane of circular motion for (a) without shaping, (b) with shaping

moves from angle $-\pi/2$ radian to 0 (relative to XZ plane). The unshaped translation motion time is 2.7 second, thus the maximum angular velocity for the rotation motion equals 0.58 rad/sec. Upon applying ZV2lin input shaper, the rotation motion time becomes 3.47 second.

Fig. 5.9 shows the comparison of the projections of the pendulum bob on XY plane of the unshaped trajectory and the trajectory shaped by ZV2lin input shaper. The circular motion starts from the origin and stops at (0.2, 0.2). In the shaped case, the pendulum sway is much suppressed, both while the support is moving as well as in the residual sway. This shows the effectiveness of the shaper in suppressing vibration for motions generated by angular command.

The trajectories of the three spherical wrist links can be obtained using the shaped angle value and the orientation history of the translation motion part. The rotation matrix of the required transformation can be obtained from the following relationship:

$${}^4R_7 = ({}^0R_4^T)({}^0R_7) \quad (5.11)$$

The rotation matrix 0R_4 is the orientation of link 4 relative to the base frame which is obtained from the trajectory of the translation motion. The rotation matrix 0R_7 is the desired orientation of the liquid container:

$${}^0R_7 = \begin{bmatrix} \cos \gamma & -\sin \gamma & 0 \\ \sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.12)$$

where γ is the angle about Z axis. Joint angle value of joints E2, W1, and W2 can be obtained next from the rotation matrix 4R_7 .

5.3.3 Combining the Translation and Rotation Motion

The translation and rotation motions are independent as each is handled by different set of robot joints. Because of that, the final trajectory can be obtained simply by combining the joint trajectories of the translation and rotation motions. Once again, we simulate cases where a spherical pendulum is attached to the robot tip, where the robot wrist locus moves from location (0.3, -0.8, 0.2) to (0.5, 0.5, 0.2) while at the same time the robot hand rotates

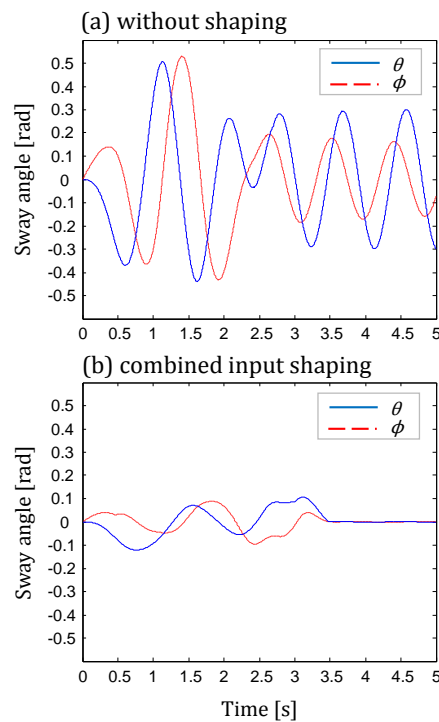


Fig. 5.10 Pendulum sway angles: (a) without shaping, (b) combined input shaping

with respect to the wrist from angle $-\pi/2$ rad to 0 rad. In short, the trajectory is the combination of the simulated trajectories at Section 5.3.1 and 5.3.2. Fig. 5.10 compares the sway angles of the case without shaping and the shaped case. The shaping is done individually: the translation motion is shaped using ZVD input shaper, the rotation motion is shaped using ZV2lin input shaping. The motion time is the same as in Section 5.3.1: 2.7 second and 3.6 second for the unshaped and shaped case, respectively. Fig. 5.11 shows the motion path of the robot tip and the projection of the pendulum bob in XY plane. From those two figures, we can understand that the strategy of combining individually shaped trajectories result in much reduced sway of the pendulum. In that way, for different conditions (in this case, translation and rotation motion), we can use suitable input shaping for each condition and then combined the generated trajectories.

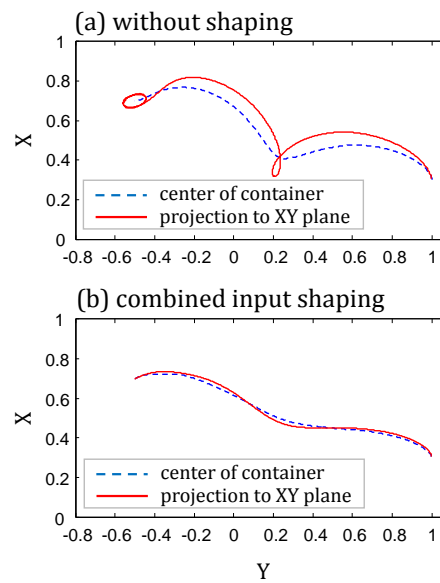


Fig. 5.11 Projection of pendulum bob position in XY plane:
(a) without shaping, (b) combined input shaping

5.4 Numerical simulation of sloshing

In order to assess the effectiveness of the whole system of trajectory planning and the input shaping, we built a numerical simulation of sloshing inside a moving liquid container based on exact distributed parameter Computational Fluid Dynamics (CFD) model, not the simple pendulum model. The liquid container is cylindrical, with 150 mm diameter and 250 mm in height. The depth of liquid is 170 mm. The simulation model is developed in the open source OpenFOAM software package. The liquid oscillation is measured during simulation time with measurement sampling time 0.01 second by two virtual probes that record the liquid height near the container wall at the direction of X and Y axis, respectively. By using the measured sloshing data, we identified the dominant fundamental mode of the sloshing at frequency 2.3 Hz.

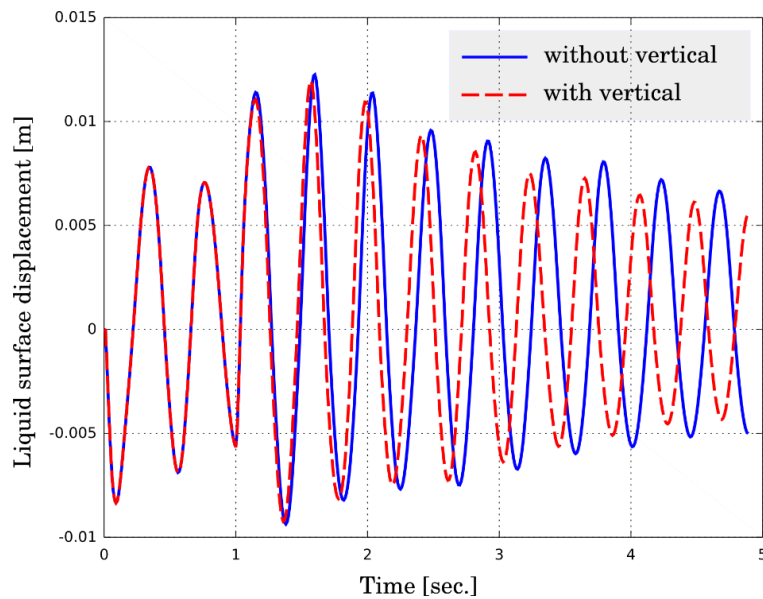


Fig. 5.12 Comparison of the simulated sloshing under vertical acceleration

As explained in Section 5.3.1, input shaping for translation motion is implemented by decomposing the trajectory into two Cartesian components (X and Y axis) and then apply input shaping to those two trajectory components individually. The shaped trajectories are then combined as the final shaped trajectory. By using the spherical pendulum model, it has been shown that the approach works well in suppressing the pendulum sway. However, when we consider movement in 3D space, the motion component in Z axis also has to be taken into consideration. To examine the effect of acceleration in Z axis toward sloshing, we run a simulation comparing trajectories with and without vertical acceleration. In both trajectories, the liquid container moves horizontally for the first one second to generate some sloshing. Then, in trajectory A, we let the container stay still, while in trajectory B, the liquid container moves upward. Fig. 5.12 shows the comparison of the sloshing generated by both trajectories. From second one, the sloshing frequency differs slightly, where it is a bit higher when the liquid container accelerates upward. Fig. 5.13 shows the comparison in frequency domain. This simulation result agrees with the theoretical model, which states that natural frequency of sloshing is directly proportional to square root of vertical acceleration (Eq. (5.1)). In the usual case of lateral movement, it consists of only

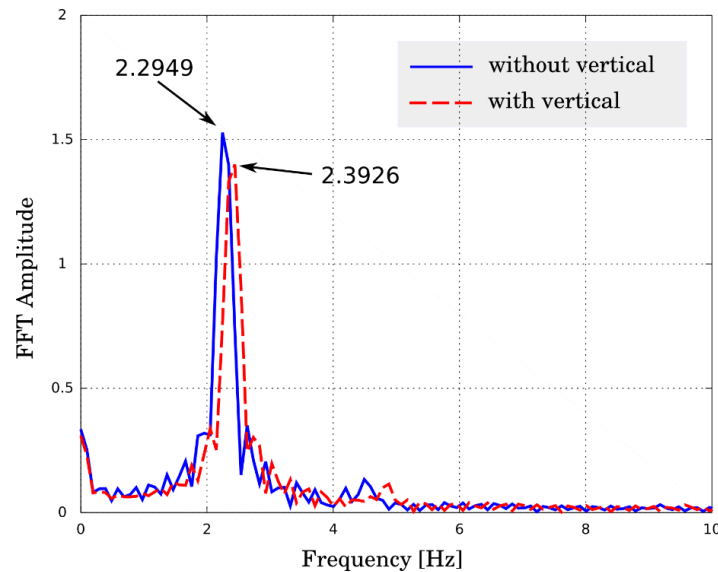


Fig. 5.13 Comparison of the sloshing frequency under vertical acceleration

the constant gravitational acceleration. But when it accelerates upward, the vertical acceleration would be higher than g , and as a result natural frequency becomes higher. The theoretical natural frequency under constant vertical acceleration of 1 m/s^2 equals 2.4 Hz , which is approximately equal with the simulation result.

The consequence of this to the sloshing suppression is that we have to consider the frequency shift in the input shaper calculation. From simulation, we understand that the range of the frequency shift is not much. With current setup of the robot arm, the vertical acceleration of a typical motion is under $\pm 4 \text{ m/s}^2$, which approximately corresponds to maximum frequency shift of $\pm 0.4 \text{ Hz}$. To deal with this small shift, we use the Zero Vibration and Damping (ZVD) input shaper, which is a more robust version of the input shaper, instead of the basic ZV shaper.

To examine the sloshing in a vertical dominant motion trajectory, we setup another simulation case where liquid container is moved from location $(0.2, 0.6, 0.2)$ to location $(0.3, 0.2, 1.1)$. In this trajectory, vertical movement is relatively dominant. Thus, we expect the natural frequency to shift along the trajectory. Fig. 5.14 shows the sloshing comparison. Consistent with the previous example case, the shaped trajectories results in less vibration. In addition, the robust ZVD shaped trajectory improves the sloshing suppression. The

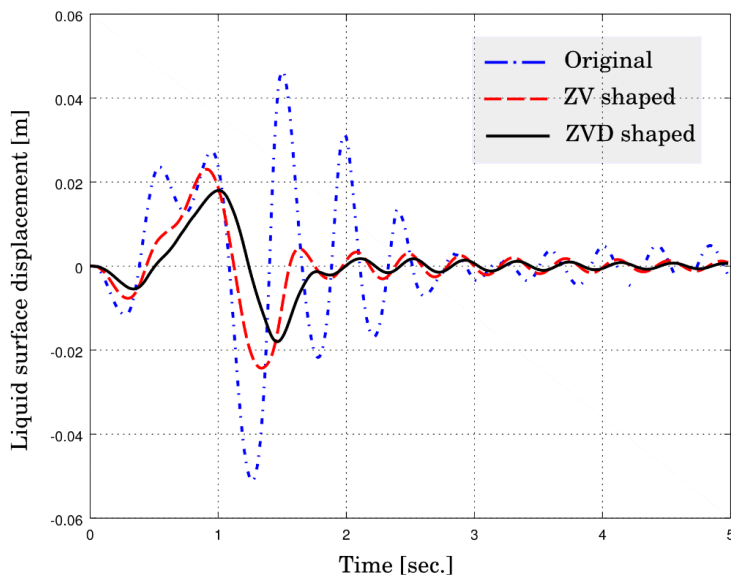


Fig. 5.14 Comparison of sloshing in vertically dominant case

maximum residual sloshing are 46.22 mm, 4.04 mm, and 1.76 mm for original, ZV-shaped, and ZVD-shaped trajectories, respectively.

5.5 Experiment Results and Discussion

In order to assess the effectiveness of the overall proposed system, including the trajectory planning for translation and rotation motion as well as the sloshing, we setup a few experiment cases. Table 5.1 lists the start and end position and orientation. The start points and end points are the position of the center of the container liquid. The angles are

Table 5.1 Start and end configuration of numerical simulation cases

Case	Point		Angle	
	Start	End	Start	End
1	(0.3, -0.8, 0.2)	(0.8, 0.1, 0.3)	$-\pi / 2$	0
2	(0.8, 0.1, 0.3)	(0.1, 0.9, 0.2)	0	$\pi / 2$
3	(0.1, 0.9, 0.2)	(0.5, 0.6, 0.9)	$\pi / 2$	0

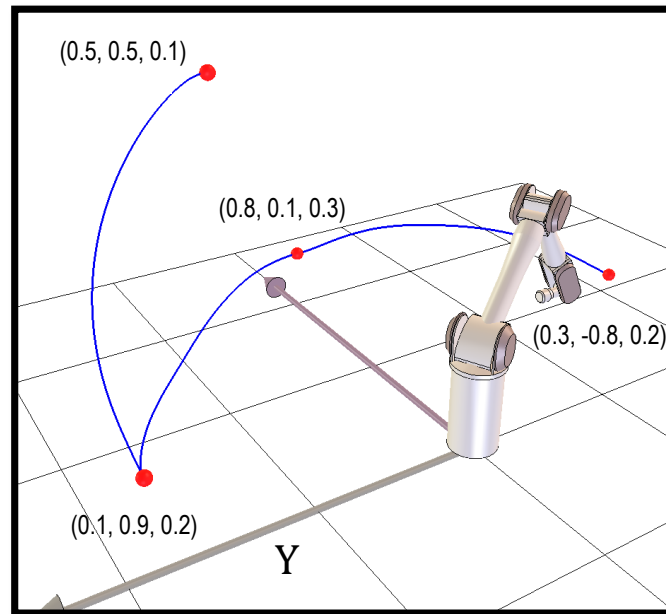


Fig. 5.15 Motion path of experiment cases 1, 2, and 3

measured relative to X-Z plane. Fig. 5.15 illustrates the motion path of the three simulation cases, where the start and end points of each case are shown by small spheres. The robot configuration in the figure is the start position of case 1. From initial experiment for identification, the natural frequency of the dominant mode is identified as 15.506 rad/s or 2.468 Hz.

The generated sloshing, as measured by level sensor, is shown in Fig. 5.16. It is measured for ten seconds from the start of motion. The motion time, residual vibration, and

Table 5.2 Result of experiments

Case	Motion time [s]		Max residual sloshing [mm]		Sloshing reduction [%]
	Unshaped	ZVD	Unshaped	ZVD	
1	1.791	2.197	25.735	6.123	76.207
2	1.783	2.189	22.1	5.929	73.172
3	1.156	1.562	45.997	9.921	78.431

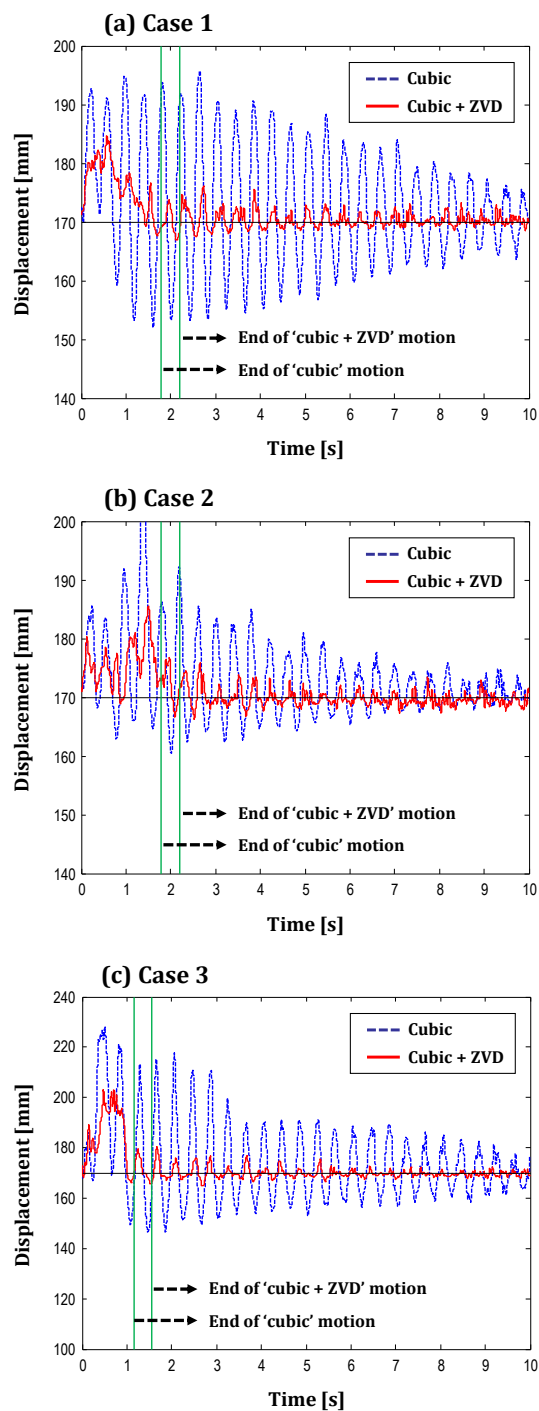


Fig. 5.16 Sloshing measurement of (a) Case 1, (b) Case 2, (c) Case 3

vibration reduction of the three cases are shown in Table 5.2. The motion time is the total time required to move from the start point to end point. The maximum residual sloshing for both unshaped and ZVD shaped trajectories is measured from the ZVD shaped motion time

until the end of measurement, so as to make fair comparison. In general, we can see that the shaped trajectories generate much less sloshing at expense of longer motion times. The additional motion time is 0.406 second for all cases. That amount is equivalent to the length of the ZVD input shaper, which equals to one vibration period. The sloshing reduction is more or less the same in all cases, where the largest is as much as 78.431% in Case 3. The motion path of Case 3 is mostly in upward direction. This shows that the solution does not have problem in handling trajectory that contains vertical motion.

5.6 Summary

This chapter shows the application of the integrated framework of trajectory planning and vibration control for the liquid container transfer problem. Different from the semiconductor wafer transfer robot arm, the sloshing is not caused by vibration of the mechanical structure of the robot, but by the motion of the container instead. For that reason, the input shaping mechanism is changed to task-space input shaping. It means the shaping is done in the task space, which later converted back to the joint space. The shaping is performed in two orthogonal direction, X and Y.

In designing the framework, pendulum model is used for quick simulation. In addition, numerical simulation of distributed CFD model is also used to analyze the effect of vertical motion to the sloshing. By the simulation we can show that the container motion in vertical direction results in slight changes in the natural frequency of sloshing. The small shift justifies the use of the Zero Vibration and Derivative (ZVD) input shaping, which is more robust than the default ZV input shaping.

Finally, experiments are performed to assess the performance in real cases of motion in three dimensional space. The motions contain not only translation motion in X, Y, and Z directions, but also rotation of the robot hand. From the experiment results, it can be concluded that the proposed integration of trajectory planning and vibration control is effective, shown by its ability to reduce the residual sloshing with only small addition in motion time.

CHAPTER 6

Concluding Remarks and Recommendations for Future Work

6.1 Concluding Remarks

This research has developed an integrated framework of trajectory generation and vibration suppression for multi joint robot arm using the piecewise cubic spline and input shaping as the building blocks. The optimization framework is formulated as a general non-linear optimization problem, which is then solved by Sequential quadratic programming algorithm. The solution is a set of joint trajectories in form of piecewise cubic splines. The framework is general enough in that it can be applied with little modifications to two different application cases of transfer robot: a semiconductor wafer transfer robot arm and a liquid container transfer robot arm. In both cases, the effectiveness of the framework to generate trajectories with quick motion time and low vibration can be demonstrated.

A few notable findings in the research are as follows:

- (1) The use of ‘floating’ knots in the trajectory planning approach is valuable. Although it increases the problem size, which in turn increases calculation time, but it also

gives the optimizer greater space to search for good solutions. As a result, the velocity profile of the generated trajectory can be improved, which means the motion time becomes shorter. In addition, the ability to avoid obstacle is improved because not only the curve inside the segments but also the location of knots themselves are adjustable.

- (2) The action of convolving a trajectory with an input shaper causes the shaped motion path to shift from the original one. This would cause problem if the working space is limited, as is the case of the wafer robot. In the proposed framework, the input shaper is integrated inside the framework. It means the input shaper convolution is performed inside each optimization iteration instead of after the optimization. Further, it means that the shaped motion path is always checked against position constraints and thus the problem can be avoided.
- (3) In the sloshing case, the whole kinematic of the robot arm is decomposed into position kinematics and orientation kinematics. This decomposition is very useful because the trajectory planning problem can be greatly simplified without any cost. As a result, the vibration suppression action can also be designed independently for the positioning motion and the orienting motion.

6.2 Recommendations for Future Work

Throughout the lengthy research process, we have identified several pointers that we can recommend as possible directions for future researches in this particular topic, as follows:

- (1) Decrease the calculation time by formulating the problem in other way, for example in convex optimization [59, 20]. Convex optimization problems are known to be solvable quickly and the global optimality of the solution is guaranteed. The quick computation time is particularly interesting for online trajectory planning because decision has to be taken quickly in real-time. However, the difficulty may lie in how to accommodate constraints like obstacle avoidance and dynamic constraints.

- (2) In the current implementation of the cubic spline optimization framework, the decision variables include the segment times and the knot locations. Meanwhile, the number of assigned knots in the trajectory is more or less the judgment of the designer. For cases where there are no active torque, jerk, or task space constraints, a minimum number of four knots are adequate. But that cannot be generalized to other cases because more knots could be needed when constraints are tight to make good trajectory. Now, if that decision making can be automated and built in the algorithm, it would be beneficial.
- (3) In the vibration parameter identification tool, one of the decision to make is whether a vibration mode is significant or not. Further work could be directed on this, in order to make it very robust and practical.
- (4) In the sloshing case, although the robot arm used in the experiment has seven degree-of-freedom (provided by seven rotational joints), redundancy has not been considered because one of the joint is fixed. In the future, the redundancy of the robot arm could be explored more to provide better maneuverability for avoiding obstacle in the task space.
- (5) In the sloshing case, now only one task-space DOF is used in the hand rotation, which is the rotation along the Z axis of the base frame. In fact, the other two DOFs (rotation along the X and Y axes) could be utilized, for example to realize motion with zero sloshing (relative to the opening of the container). Other possible use, in the application context, is of course in controlling pouring of the liquid.

References

- [1] K. D. Nguyen, T.-C. Ng and I.-M. Chen, "On Algorithms for Planning S-curve Motion Profiles," *International Journal of Advanced Robotic Systems*, vol. 5, no. 1, pp. 99-106, 2008.
- [2] S. Y. Jeong, Y. J. Choi, P. G. Park and S. G. Choi, "Jerk limited velocity profile generation for high speed industrial robot trajectories," in *Proceedings of the 16th IFAC World Congress*, Prague, 2005.
- [3] L. Biagiotti and C. Melchiorri, "FIR Filters for Online Trajectory Planning with Time- and Frequency-Domain Specifications," *Control Engineering Practice*, vol. 20, no. 12, pp. 1385-1399, 2012.
- [4] R. Bearee and A. Olabi, "Dissociated jerk-limited trajectory applied to time-varying vibration reduction," *Robotics and Computer-Integrated Manufacturing*, vol. 29, pp. 444-453, 2013.
- [5] M. Muenchhof and T. Singh, "Jerk Limited Time Optimal Control of Flexible Structures," *Journal of Dynamic Systems, Measurement, and Control*, vol. 125, pp. 139-142, 2003.
- [6] G. Lee, J. Kim and Y. Choi, "Convolution-Based Trajectory Generation Methods Using Physical System Limits," *Journal of Dynamic Systems, Measurement, and Control*, vol. 135, p. 011001, 2013.
- [7] O. v. Stryk and R. Bulirsch, "Direct and Indirect Methods for Trajectory Optimization," *Annals of Operations Research*, vol. 37, pp. 357-373, 1992.
- [8] J. T. Betts, "Survey of Numerical Methods for Trajectory Optimization," *Journal of Guidance, Control, and Dynamics*, vol. 21, no. 2, pp. 193-207, 1998.

- [9] A. V. Rao, "A survey of numerical methods for optimal control," *Advances in the Astronautical Sciences*, vol. 135, no. 1, pp. 497-528, 2009.
- [10] B. A. Conway, "A Survey of Methods Available for the Numerical Optimization of Continuous Dynamic Systems," *Journal of Optimization Theory and Applications*, vol. 152, no. 2, pp. 271-306, 2012.
- [11] K. Erkorkmaz and Y. Altintas, "High speed CNC system design. Part I: jerk limited trajectory generation and quintic spline interpolation," *International Journal of Machine Tools & Manufacture*, vol. 41, p. 1323–1345, 2001.
- [12] C. S. Lin, P. R. Chang and J. Y. Luh, "Formulation and Optimization of Cubic Polynomial Joint Trajectories for Industrial Robots," *IEEE Transactions on Automatic Control*, Vols. AC-28, no. 12, pp. 1066-1074, 1983.
- [13] C.-S. Lin and P.-R. Chang, "Joint Trajectories of Mechanical Manipulators for Cartesian Path Approximation," *IEEE Transactions on Systems, Man, and Cybernetics*, Vols. SMC-13, no. 6, pp. 1094-1102, 1983.
- [14] D. Simon and C. Isik, "Optimal trigonometric robot joint trajectories," *Robotica*, vol. 9, pp. 379-386, 1991.
- [15] L. Biagiotti and C. Melchiorri, "B-Spline Based Filters for Multi-Point Trajectories Planning," in *2010 IEEE International Conference on Robotics and Automation*, Anchorage, 2010.
- [16] A. Piazzzi and A. Visioli, "Global Minimum-Jerk Trajectory Planning for Robot Manipulators," *IEEE Transactions on Industrial Electronics*, vol. 47, no. 1, pp. 140-149, 2000.
- [17] A. Gasparetto and V. Zanotto, "A Technique for Time-Jerk Optimal Planning of Robot Trajectories," *Robotics and Computer Integrated Manufacturing*, vol. 24, pp. 415-426, 2008.
- [18] T. Chettibi, H. Lehtihet, M. Haddad and S. Hanchi, "Minimum Cost Trajectory Planning for Industrial Robots," *European Journal of Mechanics A/Solids*, vol. 23, pp. 703-715, 2004.

- [19] J. Kolter and A. Ng, "Task-space trajectories via cubic spline optimization," in *Proceedings of IEEE International Conference on Robotics and Automation*, Kobe, 2009.
- [20] B. Demeulenaere, G. Pipeleers, J. De Caigny, J. Swevers, J. De Schutter and L. Vandenbergh, "Optimal Splines for Rigid Motion Systems: A Convex Programming Framework," *Journal of Mechanical Design*, vol. 131, no. 10, pp. 101004-101004-11, 2009.
- [21] B. Demeulenaere, J. De Caigny, G. Pipeleers, J. De Schutter and J. Swevers, "Optimal Splines for Rigid Motion Systems: Benchmarking and Extensions," *Journal of Mechanical Design*, vol. 131, no. 10, pp. 101005-101005-13, 2009.
- [22] R. Bearee, P.-J. Barre and J.-P. Hautier, "Vibration reduction abilities of some jerk-controlled movement laws for industrial machines," in *Proceedings of the 16th IFAC World Congress*, Prague, 2005.
- [23] S. Kawamura, Y. Iwamoto, H. Minamoto, T. Kamigaki, Y. Taniyama and H. Kawamura, "Structural Design Optimization for A Two-link Robot to Suppress Undesirable Vibration," *Journal of Advanced Mechanical Design, Systems, and Manufacturing*, vol. 3, no. 4, pp. 289-298, 2009.
- [24] R. Lozano and B. Brogliato, "Adaptive Control of Robot Manipulators with Flexible Joints," *IEEE Transactions on Automatic Control*, vol. 37, no. 2, pp. 174-181, 1992.
- [25] S. E. Talole, J. P. Kolhe and S. B. Phadke, "Extended-State-Observer-Based Control of Flexible-Joint System With Experimental Validation," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 4, pp. 1411-1419, 2010.
- [26] O. Smith, "Posicast control of damped oscillatory systems," *Proceedings of the IRE*, vol. 45, no. 9, pp. 1249-1255, 1957.
- [27] N. C. Singer and W. P. Seering, "Preshaping command inputs to reduce system vibration," *Journal of Dynamic Systems, Measurement, and Control*, vol. 112, pp. 76-82, 1990.
- [28] T. Singh and S. Vadali, "Robust time-delay control," *Journal of Dynamic Systems*,

- Measurement, and Control*, vol. 115, pp. 303-306, 1993.
- [29] K. L. Sorensen and W. E. Singhose, "Command-induced vibration analysis using input shaping principles," *Automatica*, vol. 44, pp. 2392-2397, 2008.
- [30] N. Singer, W. Singhose and E. Kriikku, "An input shaping controller enabling cranes to move without sway," in *ANS 7th topical meeting on robotics and remote systems*, 1997.
- [31] K. L. Sorensen, W. Singhose and S. Dickerson, "A controller enabling precise positioning and sway reduction in bridge and gantry cranes," *Control Engineering Practice*, vol. 15, p. 825-837, 2007.
- [32] S. D. Jones and A. G. Ulsoy, "An approach to control input shaping with application to coordinate measuring machines," *Journal of Dynamic Systems, Measurement and Control*, vol. 121, no. 2, pp. 242-247, 1999.
- [33] K. Terashima and K. Yano, "Sloshing Analysis and Suppression Control of Tilting-Type Automatic Pouring Machine," *Control Engineering Practice*, vol. 9, pp. 607-620, 2001.
- [34] N. Singer, W. Singhose and W. Seering, "Comparison of Filtering Methods for Reducing Residual Vibration," *European Journal of Control*, vol. 5, pp. 208-218, 1999.
- [35] Z. Mohamed and M. Tokhi, "Command shaping techniques for vibration control of a flexible robot manipulator," *Mechatronics*, vol. 14, pp. 69-90, 2004.
- [36] W. Singhose and J. Vaughan, "Reducing Vibration by Digital Filtering and Input Shaping," *IEEE Transactions on Control Systems Technology*, vol. 19, no. 6, pp. 1410-1420, 2011.
- [37] K. L. Hillsley and S. Yurkovich, "Vibration Control of a Two-Link Flexible Robot Arm," *Dynamics and Control*, vol. 3, pp. 261-280, 1993.
- [38] P. H. Chang and J. Park, "A concurrent design of input shaping technique and a robust control for high-speed/high-precision control of a chip mouter," *Control*

- Engineering Practice* , vol. 9, p. 1279–1285, 2001.
- [39] F. M. Aldebrez, M. S. Alam and M. O. Tokhi, "Input-shaping with GA-tuned PID for target tracking and vibration reduction," in *Proceedings of the 13th Mediterranean Conference on Control and Automation*, Limassol, 2005.
- [40] J. R. Huey, K. L. Sorensen and W. E. Singhose, "Useful applications of closed-loop signal shaping controllers," *Control Engineering Practice* , vol. 16, pp. 836-846, 2008.
- [41] W. M. Tao, M. Zhang, M. Liu and X. Yun, "Residual Vibration Analysis and Suppression for SCARA Robot Arm in Semiconductor Manufacturing," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robot and Systems*, Beijing, 2006.
- [42] M. Hosek and J. T. Moura, "Advanced Control Techniques for Semiconductor and Flat Panel Display Substrate Handling Robots," in *Proceedings of the 17th IFAC World Congress*, Seoul, 2008.
- [43] N. Uchiyama, K. Mori, K. Terashima, T. Saeki, T. Kamigaki and H. Kawamura, "Optimal Motion Trajectory Generation and Real-Time Trajectory Modification for an Industrial Robot Working in Rectangular Space," *Journal of System Design and Dynamics*, vol. 7, no. 3, pp. 278-292, 2013.
- [44] T. Yamashita, W. Aribowo, N. Uchiyama, K. Terashima, Y. Masui and T. Saeki, "Application of Preshaping Control to a Semiconductor Wafer Transfer Robot and Experimental Verification," *Transactions of the Society of Instrument and Control Engineers*, vol. 50, no. 4, pp. 328-334, 2014. (in Japanese)
- [45] K. Yano and K. Terashima, "Robust Liquid Container Transfer Control for Complete Sloshing Suppression," *IEEE Transactions on Control Systems Technology*, vol. 9, no. 3, pp. 483-493, 2001.
- [46] M. Reyhanoglu and J. R. Hervas, "Nonlinear modeling and control of slosh in liquid container transfer via a PPR robot," *Communications in Nonlinear Science and Numerical Simulation*, vol. 18, pp. 1481-1490, 2013.

- [47] J. Feddema, C. Dohrmann and G. Parker, "A Comparison of Maneuver Optimization and Input Shaping Filters for Robotically Controlled Slosh-Free Motion of an Open Container of Liquid," in *Proceedings of the American Control Conference*, Albuquerque, 1997.
- [48] M. Hamaguchi and T. Taniguchi, "Transfer Control and Curved Path Design for Cylindrical Liquid Container," in *Proceedings of 15th IFAC World Congress*, Barcelona, 2002.
- [49] A. Aboel-Hassan, M. Arafa and A. Nassef, "Design and Optimization of Input Shapers for Liquid Slosh Suppression," *Journal of Sound and Vibration*, vol. 320, no. 1-2, pp. 1-15, 2009.
- [50] W. Aribowo, T. Yamashita and K. Terashima, "Input Shaping Control to Suppress Sloshing on Liquid Container Transfer using Multi-Joint Robot Arm," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taiwan, 2010.
- [51] K. Yano and K. Terashima, "Sloshing Suppression Control of Liquid Transfer Systems Considering a 3-D Transfer Path," *IEEE/ASME Transactions on Mechatronics*, vol. 10, no. 1, pp. 8-16, 2005.
- [52] M. Tzamtzi, F. Koumboulis and N. Kouvakas, "A Two Stage Robot Control for Liquid Transfer," in *Proceedings of IEEE Conference on Emerging Technologies and Factory Automation*, Patras, 2007.
- [53] C. Williams, G. Starr, J. Wood and R. Lumia, "Curvilinear Transport of Suspended Payloads," in *Proceedings of IEEE International Conference on Robotics and Automation*, Roma, 2007.
- [54] J. Nocedal and S. Wright, *Numerical Optimization*, 2nd edition, vol. Springer Series in Operations Research, : Springer Verlag, 2006.
- [55] T. Singh and W. Singhose, "Tutorial on Input Shaping/Time Delay Control of Maneuvering Flexible Structures," in *American Control Conference*, Anchorage, 2002.
- [56] M. D. Duong, K. Terashima, T. Kamigaki and H. Kawamura, "Development of a

- Vibration Suppression GUI Tool Based on Input Preshaping and its Application to Semiconductor Wafer Transfer Robot," *International Journal of Automation Technology*, vol. 2, no. 6, pp. 479-485, 2008.
- [57] J. Lawrence and W. Singhose, "Command Shaping Slewing Motions for Tower Cranes," *Journal of Vibration and Acoustics*, vol. 132, pp. 011002-1-11, 2010.
- [58] R. Ibrahim, "Liquid Sloshing Dynamics, Theory and Applications", New York: Cambridge University Press, 2005.
- [59] J. Swevers, G. Pipeleers, M. Diehl and J. De Schutter, "Pushing motion control systems to their limits using convex optimization techniques," in *The 11th IEEE International Workshop on Advanced Motion Control*, Nagaoka, 2010.
- [60] W. Aribowo, T. Yamashita, K. Terashima, Y. Masui, T. Saeki, T. Kamigaki and H. Kawamura, "Vibration Control of Semiconductor Wafer Transfer Robot by Building an Integrated Tool of Parameter Identification and Input Shaping," in *IFAC World Congress*, Milan, 2011.
- [61] W. Aribowo and K. Terashima, "Cubic Spline Trajectory Planning and Vibration Suppression of Semiconductor Wafer Transfer Robot Arm," *International Journal of Automation Technology*, vol. 8, no. 2, pp. 265-274, 2014.
- [62] W. Aribowo and K. Terashima, "An Implementation of Cubic Spline Optimization with Floating Via Points and Vibration Suppression for Trajectory Planning of Industrial Robot Arm," in *SICE Annual Conference*, Nagoya, 2013.
- [63] R. F. Hartl, S. P. Sethi and R. G. Vickson, "A Survey of the Maximum Principles for Optimal Control Problems with State Constraints," *SIAM Review*, vol. 37, no. 2, pp. 181-218, 1995.
- [64] J. E. Bobrow, "Optimal Robot Path Planning Using the Minimum-Time Criterion," *IEEE Journal of Robotics and Automation*, vol. 4, no. 4, pp. 443-450, 1988.

Acknowledgments

Undertaking this doctoral study is truly an experience, a worthy one. It turns out to be a long arduous process, but along the way it rewards with new understanding and self awareness. This thesis marks the completion of the process, and hopefully at the same time marks a beginning of a new journey in society. During this study, I have received support and help from many people, whom here I would like to acknowledge.

First, I would like to express my sincere gratitude to my supervisor, Professor Kazuhiko Terashima, who has given continuous support and encouragement throughout the years. I really appreciate his guidance and advice in the technical aspect related with study as well as in various other aspects.

I would also like to thank Professor Zhang, Professor Miyoshi, and Professor Uchiyama, for reviewing the thesis and providing valuable comments and inputs to the improvement of the thesis.

For the experiments in this study, Sinfonia Technology Japan had provided the semiconductor wafer transfer robot arm equipment. The laboratory of Professor Kawamura and Uchiyama had also allowed me to carry out experiments in their laboratories. During part of my study, I had received scholarship from the GCOE program of Toyohashi University of Technology, as well as research funding from Terashima laboratory.

I would like to express my gratitude to Mr. Tetsutaro Hoshi, who initially had arranged my arrival to study in Japan and who has helped me in very many occasions throughout my living in Japan. I would also like to thank late Mr. Tokushi Nishijima and Mr. Yutaka Nishijima, the former and current president of Nishijima Corporation, for providing various support and cooperation. I am also thankful for the support and encouragement from fellow teaching staffs in Industrial Engineering Department of Bandung Institute of Technology.

I also would like to thank all colleagues and members of Terashima laboratory. Professor

Terashima and Professor Miyoshi had allowed me to roam around the laboratory and use the facilities. During the experiments I had also received much help from fellow students in the laboratory, especially from fellow Arm and Hand Robot group members, and especially from Mr. Yamashita, with whom I did many experiments together. I would also like to thank fellow doctoral students who had taken study together, as well as all other and former students in the laboratory.

Last but not least, I am truly grateful for the continuous support and prayer from everyone in my family: my parents, brothers, and sisters. And I am feeling especially blessed to have my wonderful wife, Amaliatun Saleha, and my son, Raka Pandu Praditya Aribowo. I could not have reached this far without their presence around me.

List of Publications

Journal Papers

- W. Aribowo and K. Terashima, "Cubic Spline Trajectory Planning and Vibration Suppression of Semiconductor Wafer Transfer Robot Arm," *International Journal of Automation Technology*, vol. 8, no. 2, pp. 265-274, 2014.
- 山下貴仁, W. Aribowo, 内山直樹, 寺嶋一彦, 増井陽二, 佐伯亨, “半導体ウェハ搬送ロボットへの Preshaping 制御の応用と実験検証,” 計測自動制御学会論文集, vol. 50, no. 4, pp. 328-334, 2014.
- W. Aribowo, T. Yamashita and K. Terashima, "Integrated Trajectory Planning and Sloshing Suppression for Three-Dimensional Motion of Liquid Container Transfer Robot Arm," *Journal of Robotics*, vol. 2015, article ID 279460, 2015.

Conference Proceedings Papers (refereed)

- W. Aribowo, T. Yamashita and K. Terashima, "Input Shaping Control to Suppress Sloshing on Liquid Container Transfer using Multi-Joint Robot Arm," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taiwan, 2010.
- W. Aribowo, T. Yamashita, K. Terashima, Y. Masui, T. Saeki, T. Kamigaki and H. Kawamura, "Vibration Control of Semiconductor Wafer Transfer Robot by Building an Integrated Tool of Parameter Identification and Input Shaping," in *IFAC World Congress*, Milan, 2011.
- 山下貴仁, W. Aribowo, 内山直樹, 増井陽二, 佐伯亨, 神垣敏雄, 河村博年, 寺嶋一彦: 残留振動の同定手法の構築と Input Shaping による半導体ウェハ搬送ロボットアームの制振制御, 第 54 回自動制御連合講演会, 2011.

- K. Matsuo, Y. Ochi, W. Aribowo, R. Tasaki, K. Terashima, T. Yamamoto, T. Sakakibara, "Tracking Control System for Walking Assist Using Intention Estimation," in *15th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines*, Baltimore, 2012.
- W. Aribowo and K. Terashima, "An Implementation of Cubic Spline Optimization with Floating Via Points and Vibration Suppression for Trajectory Planning of Industrial Robot Arm," in *SICE Annual Conference*, Nagoya, 2013.

Awards

- CLAWAR 2012 Technical paper award (runner-up): Tracking control system for walking assist using intention estimation (2012).
- 計測自動制御学会中部支部第45期技術賞・産学連携賞：“振動抑制と衝突回避を考慮した半導体ウェハ搬送ロボットの最適制御に関する研究” (2015)