

Research Article

Integrated Trajectory Planning and Sloshing Suppression for Three-Dimensional Motion of Liquid Container Transfer Robot Arm

Wisnu Aribowo,^{1,2} Takahito Yamashita,¹ and Kazuhiko Terashima¹

¹Department of Mechanical Engineering, Toyohashi University of Technology, Toyohashi 441-8580, Japan

²Department of Industrial Engineering, Bandung Institute of Technology, Bandung 40132, Indonesia

Correspondence should be addressed to Wisnu Aribowo; wisnu@mail.ti.itb.ac.id

Received 17 November 2014; Accepted 14 April 2015

Academic Editor: Rene V. Mayorga

Copyright © 2015 Wisnu Aribowo et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

For liquid transfer system in three-dimensional space, the use of multijoint robot arm provides much flexibility. To realize quick point-to-point motion with minimal sloshing in such system, we propose an integrated framework of trajectory planning and sloshing suppression. The robot motion is decomposed into translational motion of the robot wrist and rotational motion of the robot hand to ensure the upright orientation of the liquid container. The trajectory planning for the translational motion is based on cubic spline optimization with free via points that produces smooth trajectory in joint space while it still allows obstacle avoidance in task space. Input shaping technique is applied in the task space to suppress the motion induced sloshing, which is modeled as spherical pendulum with moving support. It has been found through simulations and experiments that the proposed approach is effective in generating quick motion with low amount of sloshing.

1. Introduction

In this research, we address a case where a robot arm is used to move a liquid filled container in three-dimensional space. Some example application cases for this setting are transfer of molten metal in casting industries and liquid containers handling by service robots. The use of articulated robot arm with high degree of freedom for these purposes provides more dexterity and flexibility when compared with the more dedicated type of liquid transfer devices. There are two main performance characteristics to consider in this system: quick motion and minimal sloshing. Those two criteria are conflicting, as quick motion tends to generate much sloshing of the contained liquid. 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, for example, pouring, cannot be performed when residual sloshing still occurs, thereby potentially reducing the system productivity. Thus the challenge in this problem is to generate trajectories with concurrent consideration of motion time minimization and

sloshing suppression for point to point motion in a three-dimensional working space where static obstacles may exist.

The motion speed of robot arm is mainly constrained by its kinematic and dynamic limit of the joints. One method to generate trajectories for such system is by posing the optimal control problem as a nonlinear optimization problem where the system states are discretized in some way. To reduce the large solution space and its associated computation time, the trajectory is often parameterized into a known structure, for example, polynomial function. Among the polynomial splines, cubic spline is the lowest order polynomial trajectory that guarantees continuation in acceleration and velocity as well as limited jerk. Higher order polynomials would result in smoother profile but longer motion time, while lower order ones would result in unlimited jerk.

Numerous cubic spline algorithms and optimization schemes have been proposed in the literature [1–5]. Most of them try to optimize either the motion time or the location of knots, depending on the application needs. Relatively few addressed the optimization of time and location of knots concurrently [5]. In the latter approach, the location of

knots is not fixed, which is suitable for trajectory planning problem with obstacle avoidance. In addition, it often results in a quicker motion because there is less restriction for the optimization algorithm to reach optimal velocity profile.

The smooth cubic spline trajectories induce less vibration when compared to nonsmooth or lower order trajectories. However, in cases where minimal sloshing is required, smooth trajectory alone is not enough, and it needs to be supplemented with explicit sloshing control strategy. 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 [6, 7]. However, in some practical cases, sensor measurements cannot always be reliable or even are difficult to perform, for example, in the case of controlling sloshing of high-temperature molten metal. In these cases, feedforward approaches [8–14] 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 generalized bang-bang control [9], infinite impulse response (IIR) filter [8], input shaping filter [10–13], and Hybrid Shape Approach filters [14]. Among them, the input shaping approach is particularly interesting because of its simplicity and effectiveness in practice. It works by decomposing the command input signal into two or more, using only the knowledge of natural frequency and damping ratio of the system [15, 16]. Over time, various type of input shapers with different characteristics have been proposed, and they have been successfully applied in practice for many kind of vibration problems.

With respect to the motion path, traditionally sloshing is analyzed for one-dimensional straight motion, where the sloshing is often modeled as a simple pendulum. The motion planning and sloshing suppression is relatively simple and straightforward for this kind of problem [9, 13]. Researches that address suppression of sloshing or pendulum sway in higher dimensional space are relatively few. Tzamtzi et al. [17] modeled the sloshing as simple pendulum in 2D planar vertical motion. Williams et al. [18] used a dynamic programming approach to solve the sway-free motion of suspended spherical pendulum in 2D planar horizontal motion. Yano and Terashima [14] proposed HSA approach to control sloshing in three degree of freedom Cartesian liquid container transfer system.

In this paper, we propose the integration of the trajectory planning of seven degrees of freedom liquid container transfer robot arm and the sloshing suppression of the contained liquid. We use the Mitsubishi PA 10-7C robot arm as the experiment device. The container is allowed to rotate about vertical axis while the pitch and roll angles are regulated to maintain upright orientation of the liquid container. Such system setup is different from that used in other related papers (e.g., [14, 17, 18]); in that translational motion in three-dimensional space plus orientation of the container needs to be controlled. The output of the proposed method is joint trajectories, to be used as a reference of the robot motion, thus obviating the need to change the closed-loop controller. Consequently, the proposed solution is easier to be applied to different kind of systems.

Because the robot arm has spherical wrist, the motion can be conveniently decoupled into translational motion of the wrist locus and rotational motion of robot hand. This decomposition is useful in our application case, because for sloshing suppression we want to take full control of the container orientation. For the trajectory generation of the wrist translational motion, we define the problem as a nonlinear optimization model where each joint trajectory is parameterized as cubic spline. The locations of knots are part of decision variables, thus not fixed, to allow obstacle avoidance in task space and quick motion. It is an extension to a more basic cubic spline optimization reported in our previous paper [5] in two points: motion in three-dimensional space is addressed by using multijoint robot arm, and obstacle avoidance constraints are added. As for the sloshing suppression, we use a new strategy of applying input shaping. The decoupled translational and rotational motions are each shaped by using its respective suitable input shaper and then combined as the final trajectory. We use the spherical pendulum model to simulate the response of the trajectory. In addition, a numerical sloshing simulation model is built for the purpose of examining the sloshing behavior in response to vertical motion of the liquid container in 3D space. Finally, experiments are carried out for a few example cases.

2. Sloshing Model and Suppression

2.1. Equivalent Sloshing Model. The theoretical natural frequency of sloshing inside an upright cylindrical container can be derived from Navier-Stokes equations as [19]

$$\omega_n^2 = \left(\frac{g\xi_n}{R} \right) \tanh \left(\frac{\xi_n h}{R} \right), \quad (1)$$

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.

For control analysis, the phenomenon of sloshing is often modeled by its simpler mechanical equivalents, for example, pendulum and mass-spring-damper system. Here, we model the sloshing as a simple pendulum, where the sway of the pendulum corresponds to the elevation of the liquid surface. A simple pendulum system, as depicted in Figure 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 straight line, it can be described by the following equation of motion:

$$L\ddot{\theta} + g \sin \theta = -\dot{x} \cos \theta, \quad (2)$$

where g is the gravity constant and 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. Figure 1(b) illustrates the spherical pendulum model, where the pendulum bob can freely swing in two degrees of freedom spherical space as response to movement of the support in horizontal planar space. Angle θ is the angle

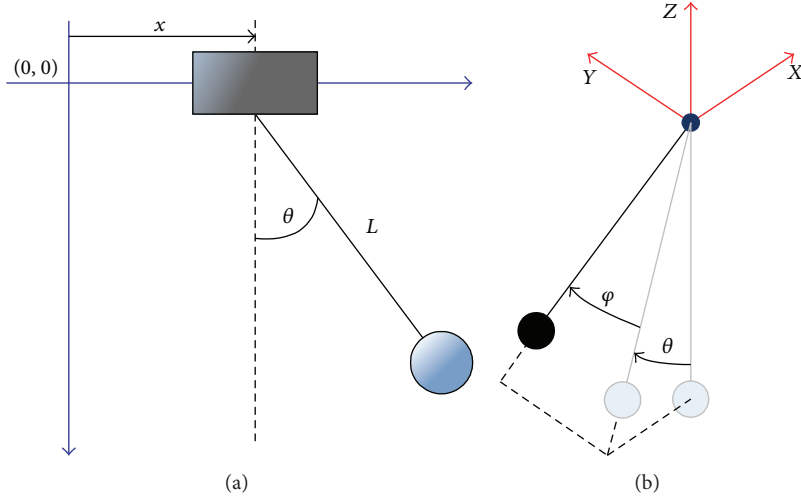


FIGURE 1: (a) Simple pendulum and (b) spherical pendulum.

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

$$\begin{aligned} x_p &= -L \sin \theta \cos \varphi, \\ y_p &= L \sin \theta, \\ z_p &= -L \cos \theta \cos \varphi. \end{aligned} \quad (3)$$

The equations of motion of the spherical pendulum can be derived as follows [18]:

$$\begin{aligned} \ddot{\theta} &= 2\dot{\theta}\dot{\varphi} \tan \varphi - \frac{g \sin \theta}{L \cos \varphi} + \frac{1 \cos \theta}{L \cos \varphi} \ddot{x}, \\ \ddot{\varphi} &= -\dot{\theta}^2 \dot{\varphi} \tan \varphi - \frac{g}{L} \cos \theta \sin \varphi \\ &\quad - \frac{1}{L} (\ddot{x} \sin \theta \sin \varphi + \ddot{y} \cos \varphi), \end{aligned} \quad (4)$$

where x and y are 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 (2).

2.2. Input Shaping. In the simple pendulum case, the natural frequency and damping ratio can be obtained by linearization of the equation of motion around equilibrium as

$$\begin{aligned} \omega_0 &= \sqrt{\frac{g}{L}}, \\ \zeta &= \frac{c}{2m} \sqrt{\frac{L}{g}}. \end{aligned} \quad (5)$$

When excited by an impulse, the sway response of the pendulum is decaying sinusoidal:

$$\theta(t) = A \frac{\omega_0}{\sqrt{1-\zeta^2}} e^{-\zeta\omega_0 t} \sin\left(\omega_0 t \sqrt{1-\zeta^2}\right). \quad (6)$$

Such sway can be suppressed by shaping the impulse input such that the sum of the response is zero. It is performed by splitting the original impulse into two or more impulses and delays them such that their responses cancel each other [15, 16]. For example, in the simplest case of splitting into two impulses, the amplitudes of the first and second impulses are $K/1+K$ and $1/1+K$, respectively, where

$$K = e^{-\zeta\pi/\sqrt{1-\zeta^2}}, \quad (7)$$

and time between impulse equals one-half damped period of vibration:

$$\Delta T = \frac{\pi}{\omega_0 \sqrt{1-\zeta^2}}. \quad (8)$$

This set of impulses is called an input shaper, and the two-impulse input shaper above is known as zero vibration (ZV) shaper. 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. 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 the system input signal with the desired input shaper, as shown in Figure 2. As a result, the motion time becomes longer, as much as the length of the input shaper. For example, applying ZV input shaper

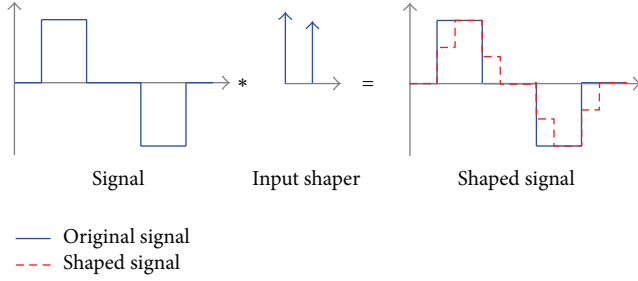


FIGURE 2: Shaping arbitrary signal by convolution.

TABLE 1: Limit values of joint velocities and angles.

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

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 [20] 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.

3. Trajectory Planning

The liquid transfer system concerned uses the Mitsubishi PA 10-7C robot arm. The frames placement and link dimension of the robot arm are shown in Figure 3. Table 1 lists the limit value of each joint angle and angular velocity. A cylindrical liquid container is attached to an extension at the robot tip. The diameter and height of the liquid container are 0.15 mm and 0.25 m, respectively. The distance between robot tip and the center of the liquid container is 0.12 meter; thus the length of the last link is practically 0.2 meter. For our current application, one of the arm joints, the S3 joint, is locked so that only six joints are actively used. Table 2 lists the Denavit-Hartenberg parameters of the robot arm configuration. The

TABLE 2: Denavit-Hartenberg parameters.

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

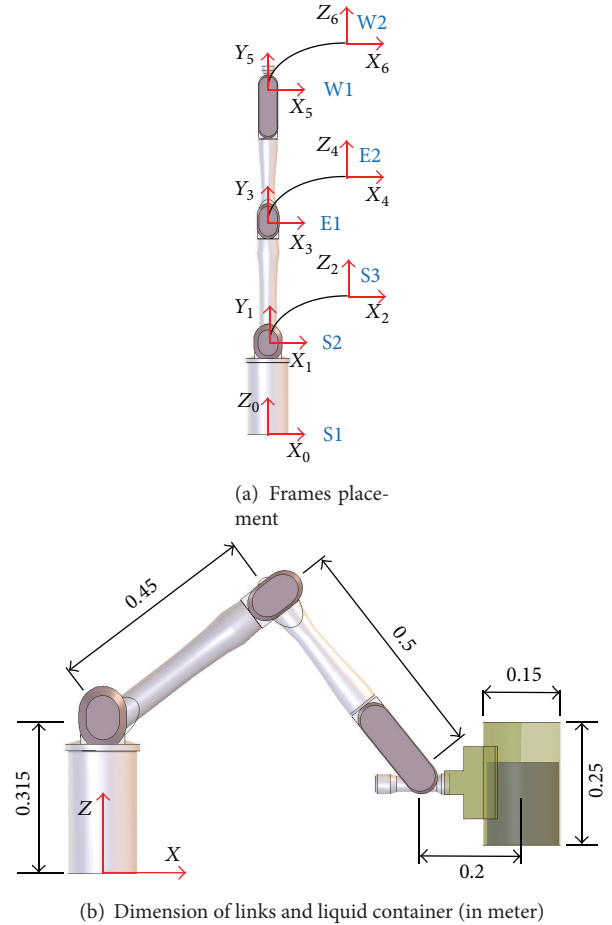


FIGURE 3: (a) Frames placement and (b) dimension of the liquid container system and the base frame.

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

$$\begin{aligned}
 A_i &= \begin{bmatrix} {}^{i-1}R_i & {}^{i-1}p_i \\ 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\cos \theta_i \sin \alpha_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (9)
 \end{aligned}$$

The position and orientation of a link with respect to other link can be obtained by chain product of the transformation matrices:

$${}^i T_j = A_{i+1} A_{i+2} \cdots A_j = \begin{bmatrix} {}^i R_j & {}^i P_j \\ 0 & 1 \end{bmatrix}. \quad (10)$$

The upper left 3×3 rotation matrix ${}^i R_j$ describes the link orientation, while the vector ${}^i P_j$ represents the position.

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 subsystems. The first three rotational links S1, S2, and E1 provide the positioning in three-dimensional space, while the spherical wrist adjusts the orientation of 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 associated 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:

$$\begin{aligned} \theta_1 &= \text{atan2}(y_w, x_w), \\ \theta_4 &= \text{atan2}\left(\sqrt{1 - D^2}, D\right), \\ \theta_2 &= \text{atan2}\left(z_w - L_1, \sqrt{x_w^2 + y_w^2}\right) \\ &\quad - \text{atan2}(L_3 \sin \theta_4, L_2 + L_3 \cos \theta_4) - \frac{\mu}{2}, \end{aligned} \quad (11)$$

where L_i is the length of link i , atan2 is the arc tangent function with two arguments, and

$$D = \frac{x_w^2 + y_w^2 + (z_w - L_1)^2 - L_2^2 - L_3^2}{2L_2 L_3}. \quad (12)$$

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 (${}^0 R_7$) is known. First, from the inverse kinematics for position, the rotation matrix describing the orientation of the third link can be obtained (${}^0 R_4$). Then, the rotation operation of the wrist joints required to realize the end effector posture is calculated as

$${}^4 R_7 = ({}^0 R_4)^T ({}^0 R_7). \quad (13)$$

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

$$\theta_5 = \text{atan2}\left({}^4 R_7 [2, 3], {}^4 R_7 [1, 3]\right) \pm \pi, \quad (14)$$

$$\theta_6 = \text{atan2}\left(\sqrt{1 - {}^4 R_7 [3, 3]^2}, {}^4 R_7 [3, 3]\right), \quad (15)$$

$$\theta_7 = \text{atan2}\left({}^4 R_7 [3, 2], {}^4 R_7 [3, 1]\right) \pm \pi, \quad (16)$$

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

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 that 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 Figure 4, which clearly depicts 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.

Figure 4(a) shows the overview of planning the translational motion. It starts with specifying the start point and 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 translational motion is obtained as joint trajectories of links S1, S2, and E1.

Figure 4(b) shows the steps of planning the rotational motion of the robot hand. The input is the start and end angle of the container along z -axis relative to XZ vertical plane. The shaped rotation trajectory is then transformed to joint space as the solution of the rotation part of the motion in form of joint trajectories of links E2, W1, and W2. Although rotation trajectory generation is independent of translation trajectory generation, it has to be performed after because the former needs the orientation history of the translational motion as well as the total motion time in order to coordinate the overall motion.

3.1. Translational Motion Planning. In this section, we outline the cubic spline optimization approach for trajectory planning. It builds from the same basic approach as in a previous paper [5] for two-dimensional problem and then adapted to the current application of motion in three-dimensional space and is complemented with input shaping in the task space.

A cubic spline trajectory of a robot joint is comprised of several curve segments, as illustrated in Figure 5, where each segment is represented by a cubic polynomial function of time. The general form of the cubic function in a segment i is

$$q_i(t) = a_i(t - t_i)^3 + b_i(t - t_i)^2 + c_i(t - t_i) + d_i, \quad (17)$$

where t denotes the time, which ranges from t_i to t_{i+1} .

The curve segments are connected by knots. Thus for a trajectory that consists of n curve segments, there are $n + 1$ prespecified joint angle values θ_i , ($i = 1 \cdots n + 1$) which consist of a start point, an end point, and $n - 1$ knots between

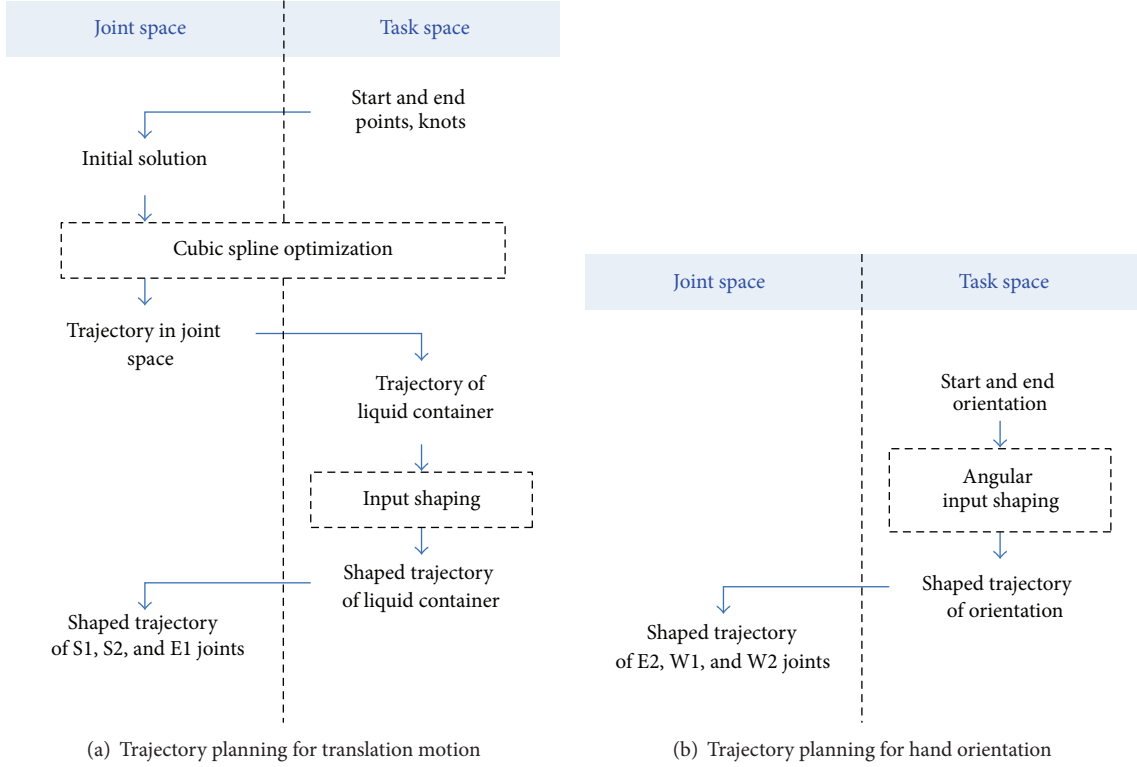


FIGURE 4: Trajectory planning steps for (a) translational motion and (b) hand orientation.

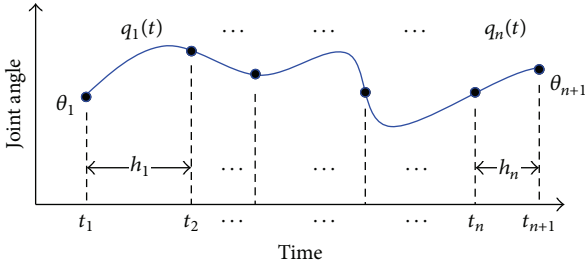


FIGURE 5: Cubic spline trajectory.

them. The trajectory is required to pass through all those points. Other requirements are continuity in velocity and acceleration at every knot. The relationship between joint angle θ_i , joint velocity $\dot{\theta}_i$, joint acceleration $\ddot{\theta}_i$, and time interval $h_i = (t_{i+1} - t_i)$ is as follows:

$$\begin{aligned} \ddot{\theta}_i h_i + 2\ddot{\theta}_{i+1} (h_i + h_{i+1}) + \ddot{\theta}_{i+2} h_{i+1} \\ = 6 \left(\frac{\theta_{i+2} - \theta_{i+1}}{h_{i+1}} - \frac{\theta_{i+1} - \theta_i}{h_i} \right). \end{aligned} \quad (18)$$

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 of $n + 1$ equations. The robot motion usually also requires that velocities at start and end points equal zero. To accommodate this, two more unknowns are added by inserting two virtual points: one after the start point and another one just before

the end point. Furthermore, because $\ddot{\theta}_1$ and $\ddot{\theta}_{n+1}$ are already known as zero, they can be removed from the equations. Joint angles θ_i are defined beforehand by the trajectory designer. The knots may be specified as points in Cartesian task space and later transformed into joint space by inverse kinematics. The above system of linear equations forms a symmetric tridiagonal system, which can be solved efficiently using the tridiagonal matrix algorithm (TDMA) to find acceleration at each knot.

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 weighted squared acceleration. There are several constraints to consider in the trajectory generation: robot position in the task space, joint velocity, and joint jerk. According to (18), 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 thus as follows:

$$\begin{aligned} \min_{\mathbf{x}} \quad & F = \sum_{i=1}^n h_i + w_a \int \ddot{q}_i^2 dt \\ \text{subject to} \quad & |q_{ij}^*| \leq q_j^{\text{MAX}}, \quad i = 1 \cdots n, \quad j = 1 \cdots m \\ & |v_{ij}^*| \leq v_j^{\text{MAX}}, \quad i = 1 \cdots n, \quad j = 1 \cdots m \end{aligned}$$

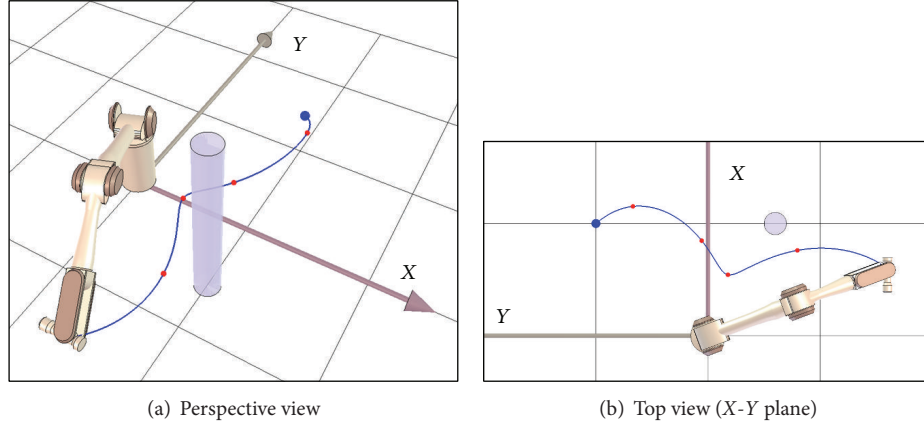


FIGURE 6: Motion path of example case of translation motion.

$$\begin{aligned}
 |\ddot{\theta}| &\leq J^{\text{MAX}}; \\
 i &= 1 \cdots n + 1, \quad j = 1 \cdots m \\
 d_i(L_j, O) &\geq r_{L_j} + r_O; \\
 i &= 1 \cdots n, \quad j = 1 \cdots m,
 \end{aligned} \tag{19}$$

where \mathbf{x} is the vector of 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:

$$\begin{aligned}
 \mathbf{x} &= (h_1, \dots, h_n, \theta_{ij}, \dots, \theta_{nm}); \\
 i &= 3 \cdots n - 1, \quad j = 1 \cdots m.
 \end{aligned} \tag{20}$$

Here, all internal knots are part of the optimization variables instead of predefined 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 term of the objective function is to minimize the total motion time, which is the main objective. The second term is used to avoid unnecessary motion of robot joints, where w_a is the weight parameter. In our subsequent simulations and experiments, the parameter value equals $1e^{-4}$.

The first and second constraints are joint angle and velocity constraints, according to the physical joint limits in Table 1. The third is jerk constraint, to prevent excessive value of joint jerk. The maximum jerk of each joint is set equal to 15 m/s^3 . 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 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 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. 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.

Figure 6 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 of the world frame. 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 seconds. Figure 7 shows the joint trajectories of the example case: the piecewise cubic 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 translational motion planning is to suppress the sloshing generated by the trajectory obtained above. For analysis, we use the pendulum model in Section 2.1 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

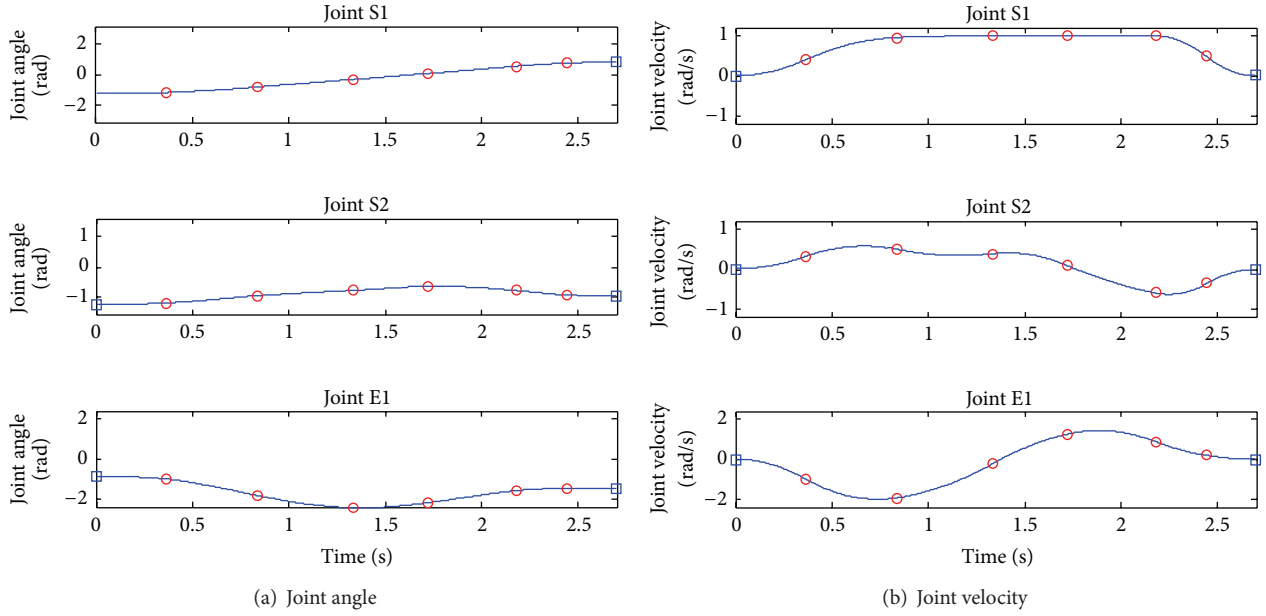


FIGURE 7: Trajectory of example case of translational motion: (a) joint angle (b) joint velocity.

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 (4) by small angle approximations and removing quadratic and higher terms results in two independent equations of simple pendulum.

We use the ZVD input shaping for sloshing suppression. For that, the velocity trajectory is decomposed into two orthogonal velocities parallel to x -axis 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 seconds, respectively. In sloshing problems, the damping is usually very small; therefore here we assume zero damping in the pendulum model. Figure 8 shows the original and ZVD shaped Cartesian velocities in X and Y directions. The motion time is 2.7 seconds and 3.6 seconds in the unshaped and ZVD shaped case, respectively. Figure 9 shows the sway angles in the spherical coordinate system. At the end of motion, sway is much suppressed in the shaped trajectory. Figure 10 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.

3.2. Rotational Motion Planning. The liquid container has to be maintained upright all time. This means that there is only

one task space DOF: the rotation along the vertical z -axis. Thus the hand rotational 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 the 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

$$\dot{s} = \frac{\gamma_f - \gamma_s}{T}, \quad (21)$$

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 translational and rotational motions, so that both motions end 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 moves from angle $-\pi/2$ radian to 0 (relative to XZ plane). The unshaped translation motion time is 2.7 seconds; thus the maximum angular velocity for the rotational motion equals 0.58 rad/sec. Upon applying ZV2lin input shaper, the rotational motion time becomes 3.47 seconds.

Figure 11 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 the sway while the support is moving and the residual sway after the support stops. This shows the effectiveness of the shaper in suppressing vibration for motions generated by angular command.

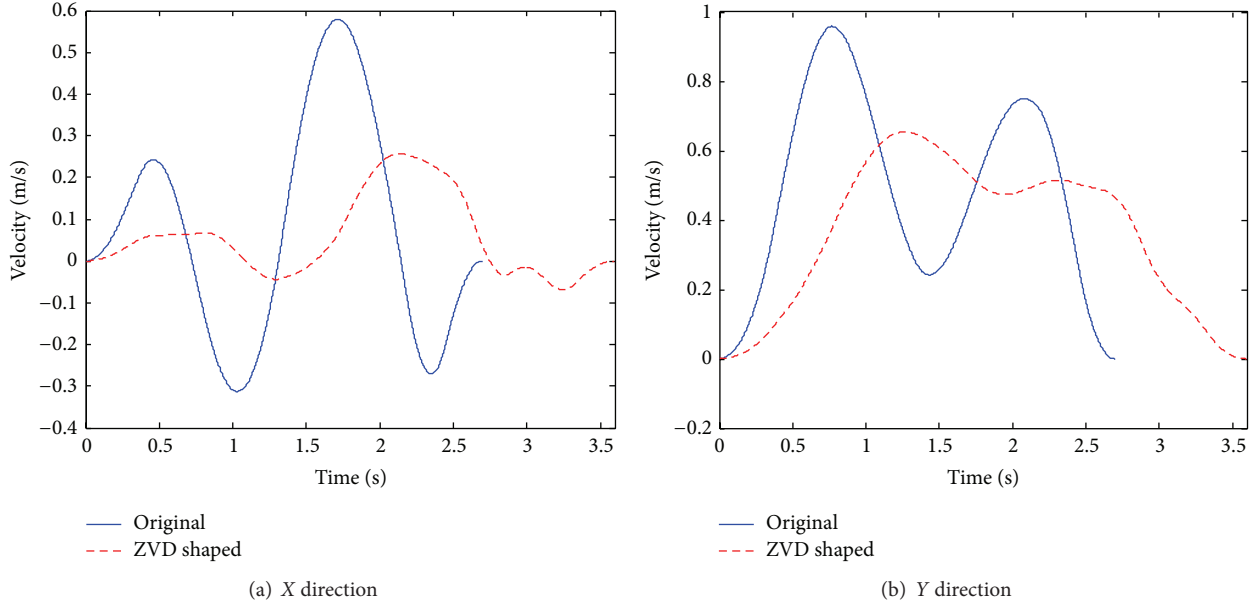


FIGURE 8: Original and ZVD shaped velocity in task space: (a) X direction (b) Y direction.

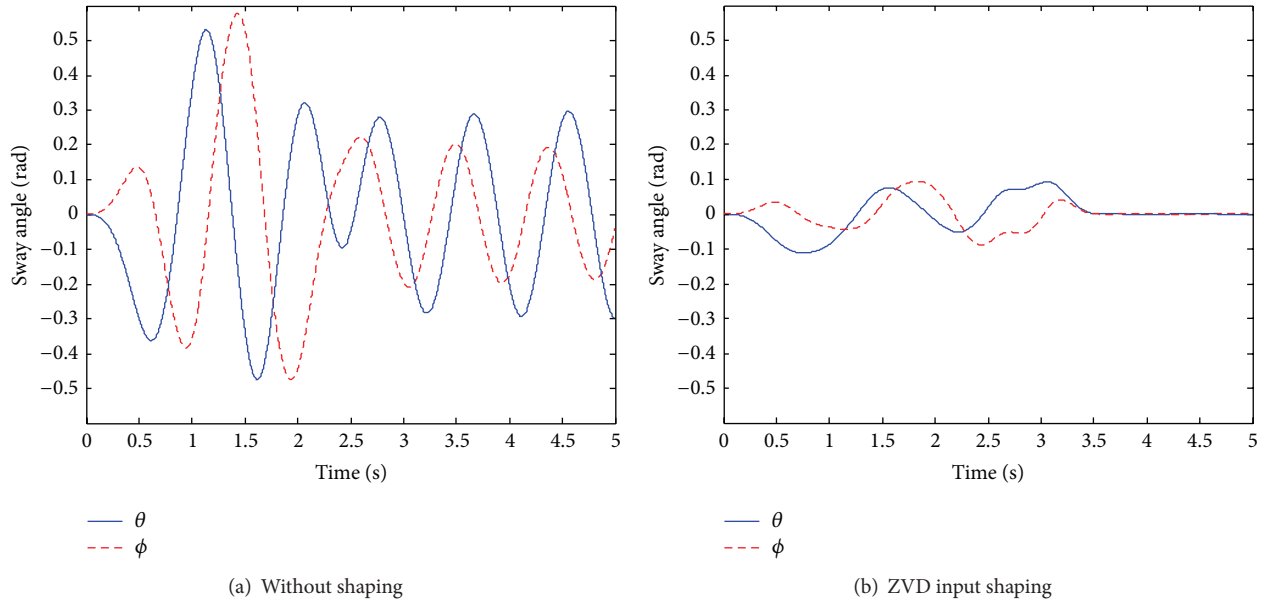


FIGURE 9: Pendulum sway angles: (a) without shaping (b) with ZVD shaping.

The trajectories of the three spherical wrist links can be obtained using the inverse kinematic relations in (13)–(16), the shaped angle value, and the orientation history of the translation motion part. The rotation matrix 0R_7 is the desired orientation of the liquid container:

$${}^0R_7 = \begin{bmatrix} \cos \gamma & -\sin \gamma & 0 \\ \sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (22)$$

where γ is the angle about z -axis.

The translational and rotational 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 translational and rotational motions. Once again, we simulate cases where 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 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 Sections 3.1 and 3.2. Figure 12 compares the sway angles of the case without shaping and the shaped case. The shaping

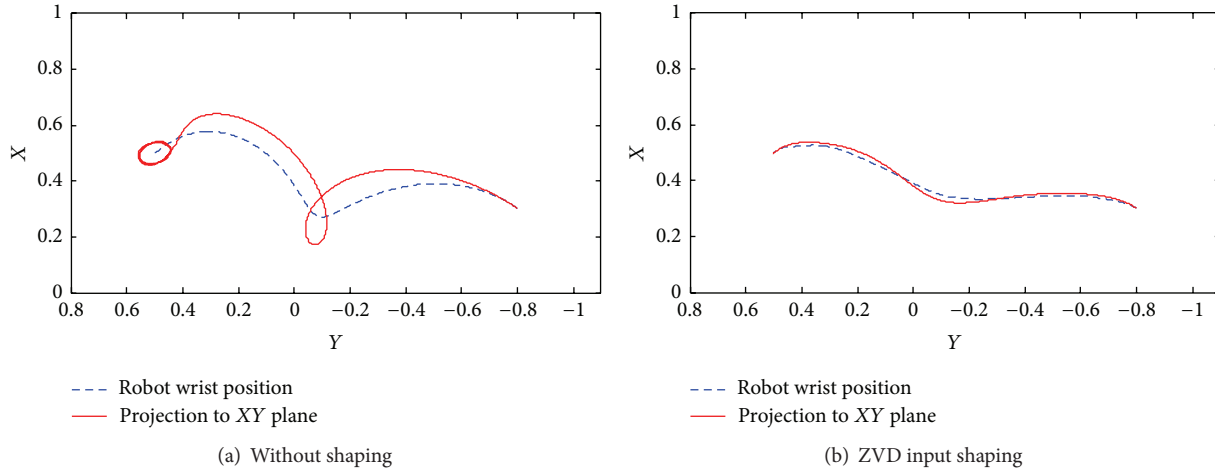


FIGURE 10: Projection of pendulum bob position in XY plane: (a) without shaping (b) with ZVD shaping.

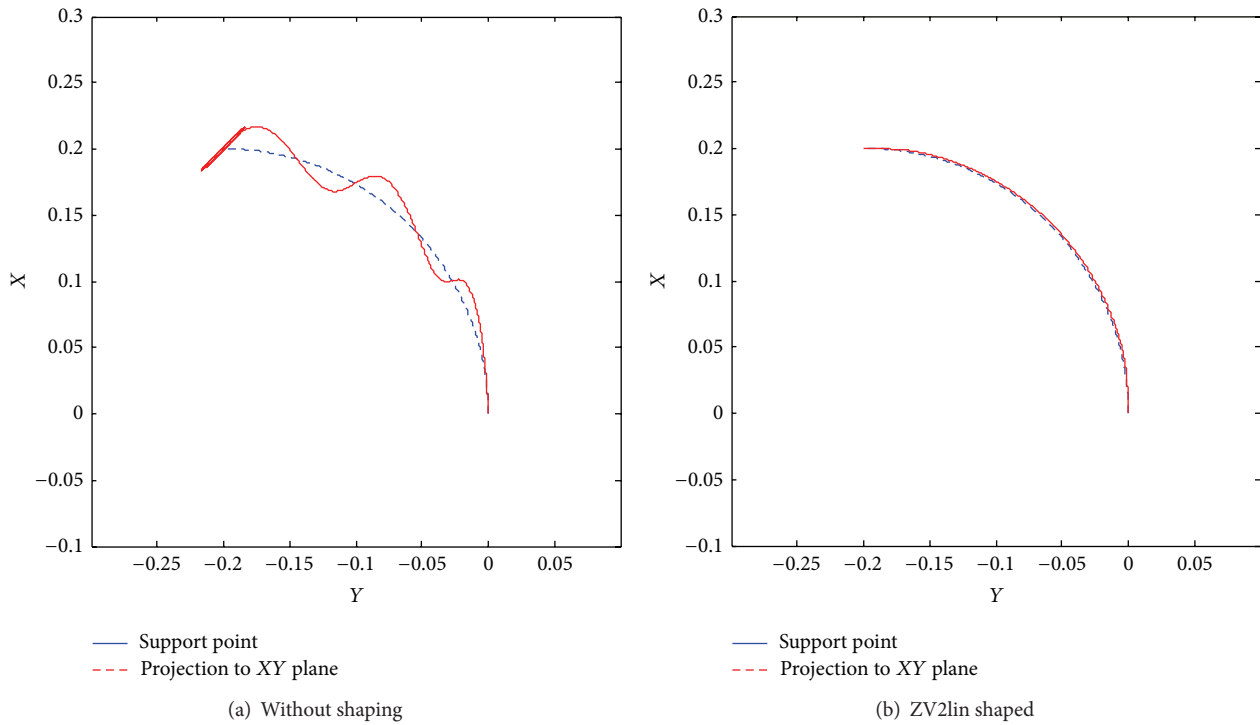


FIGURE 11: Pendulum position in XY plane of circular motion for (a) without shaping (b) with shaping.

is done individually: the translation motion is shaped using ZVD input shaper and the rotational motion is shaped using ZV2lin input shaping. The motion time is the same as in Section 3.1: 2.7 seconds and 3.6 seconds for the unshaped and shaped case, respectively. Figure 13 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, translational and rotational motion), we can use suitable input shaping for each condition and then combined the generated trajectories.

4. Numerical Simulation of Sloshing

In order to analyze the effect of vertical motion to sloshing, 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 seconds by two virtual probes that record the

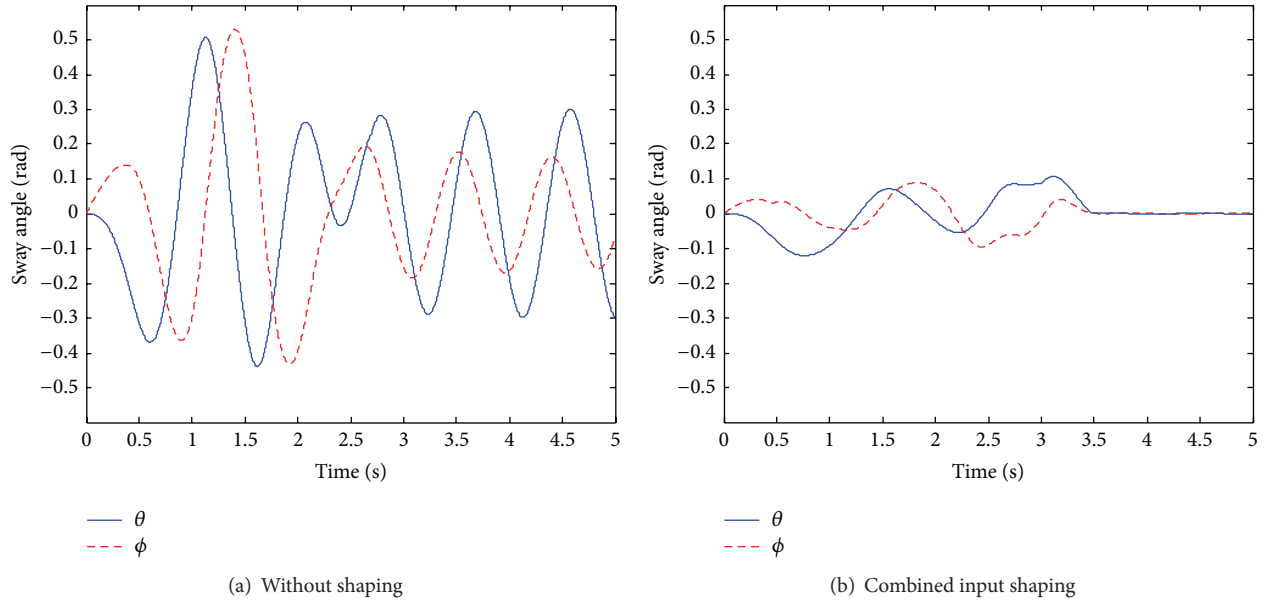


FIGURE 12: Pendulum sway angles: (a) without shaping (b) combined input shaping.

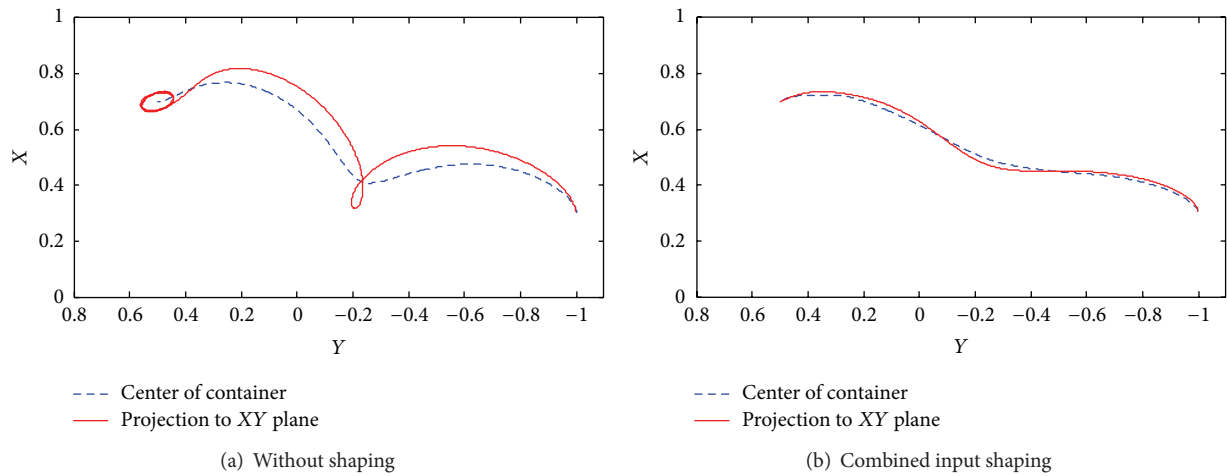


FIGURE 13: Projection of pendulum bob position in XY plane: (a) without shaping (b) combined input shaping.

liquid height near the container wall at the direction of x -axis and y -axis, respectively. By using the measured sloshing data, we identified the dominant fundamental mode of the sloshing at frequency 2.3 Hz.

As explained in Section 3.1, input shaping for translational motion is implemented by decomposing the trajectory into two Cartesian components (x -axis and y -axis) and then applying 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 motion 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. Figure 14 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. Figure 15 shows the comparison in frequency domain. This simulation result agrees with the theory that states natural frequency of sloshing is directly proportional to square root of vertical acceleration (1). In the usual case of lateral movement, it consists of only 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

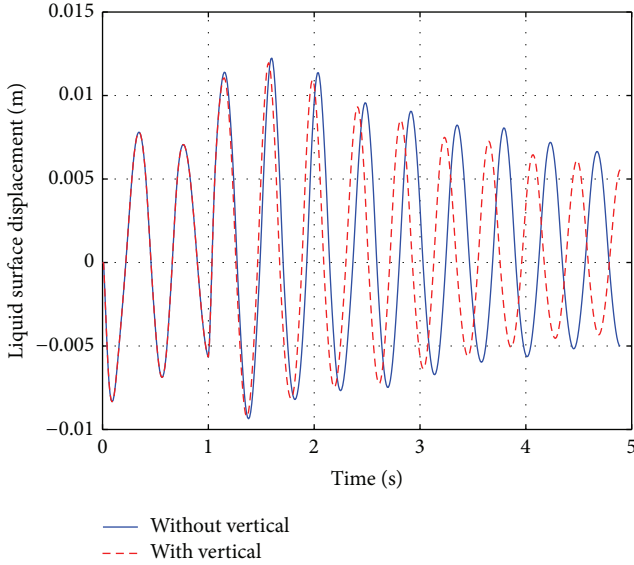


FIGURE 14: Comparison of the simulated sloshing under vertical acceleration.

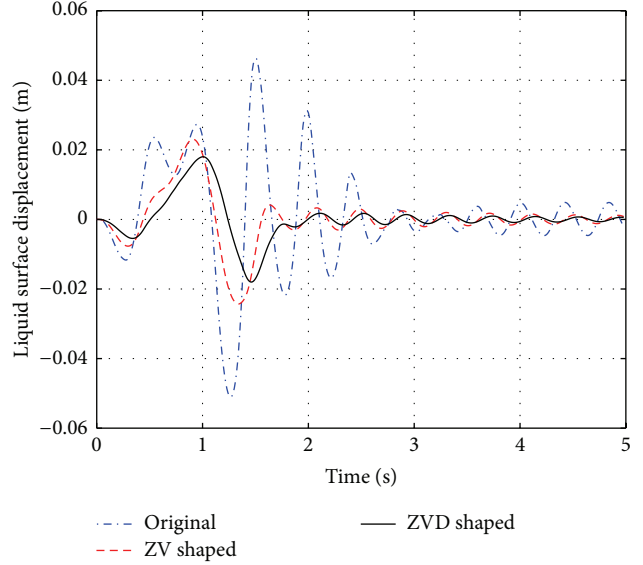


FIGURE 16: Comparison of sloshing in vertically dominant case.

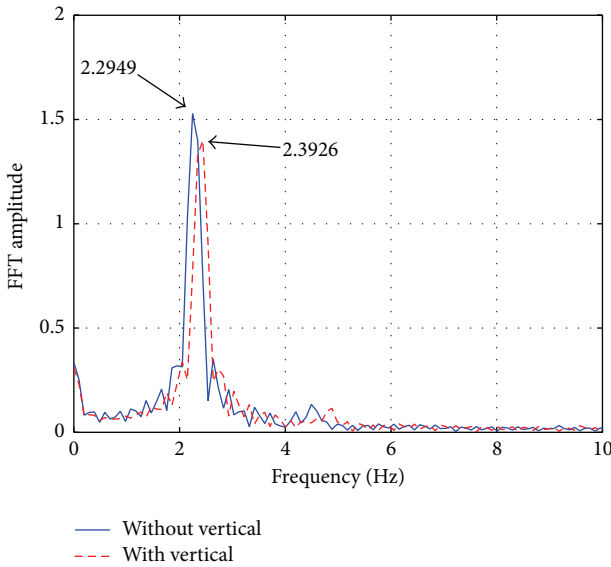


FIGURE 15: Comparison of the sloshing frequency under vertical acceleration.

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 wide. 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 zero vibration and damping (ZVD) input shaper, which is a more robust version of the input shaper, instead of the basic ZV shaper.

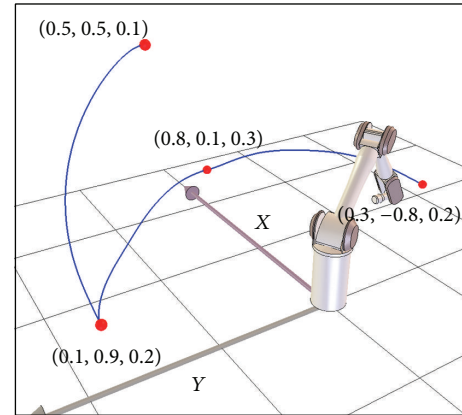


FIGURE 17: Motion path of the experiment cases.

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. Figure 16 shows the sloshing comparison. Consistent with the previous example case, the shaped trajectories result in less vibration. In addition, the robust ZVD shaped trajectory improves the sloshing suppression. The maximum residual sloshing is 46.22 mm , 4.04 mm , and 1.76 mm for original, ZV shaped, and ZVD shaped trajectories, respectively. From this simulation study, we understand that vertical motion affects the sloshing in that small shift of natural frequency occurs. 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, in our proposed solution.

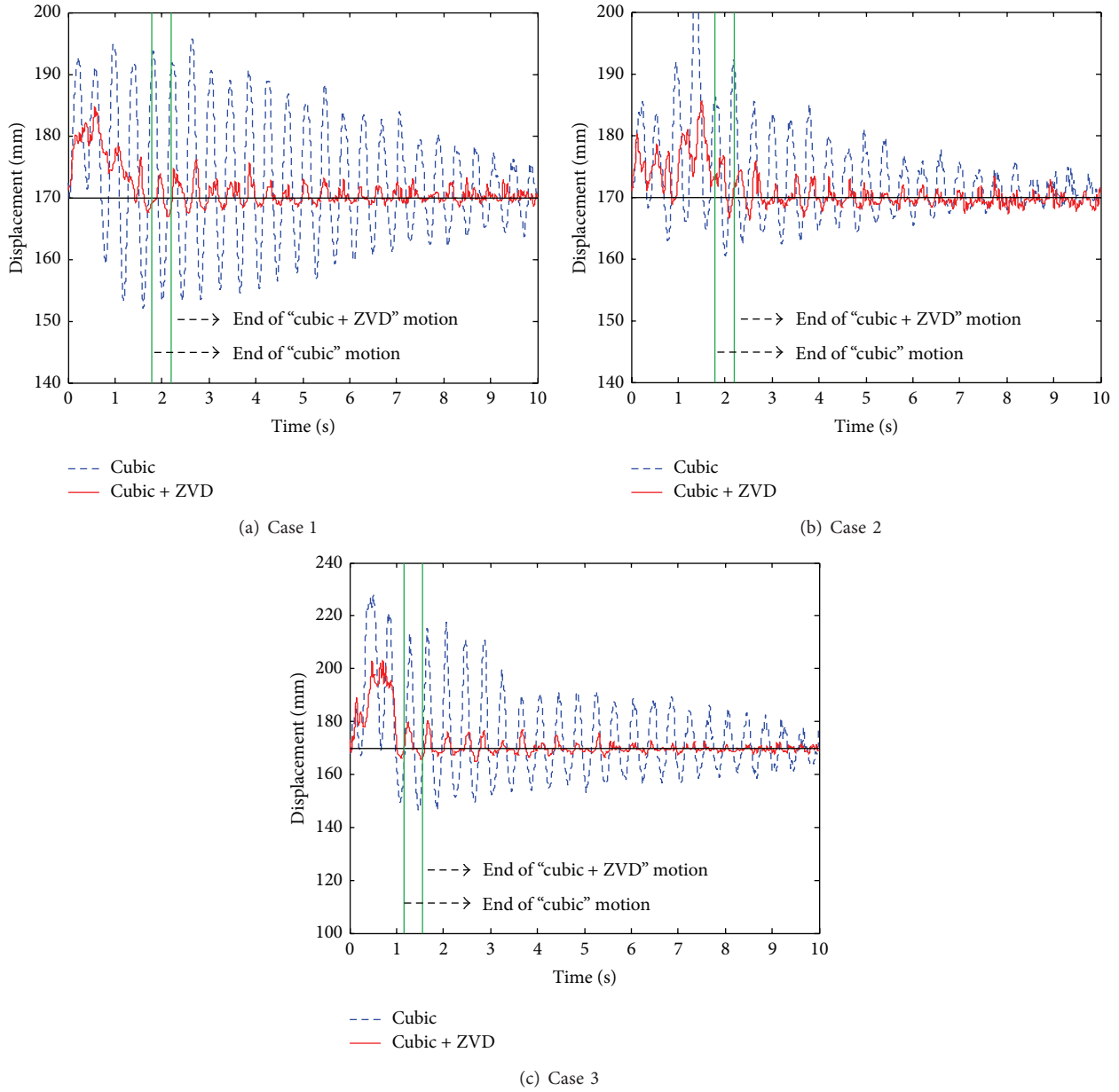


FIGURE 18: Sloshing measurement of (a) Case 1, (b) Case 2, and (c) Case 3.

5. Experiment Results and Discussion

In order to assess the effectiveness of the overall proposed system, including the trajectory planning for translational and rotational motion as well as the sloshing, we setup a few experiment cases. Table 3 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 measured relative to XZ plane. Figure 17 illustrates the motion path of the three experiment cases, where the start and end points of each case are shown by small spheres.

The generated sloshing, as measured by level sensor is shown in Figure 18. It is measured for ten seconds from the start of motion. The motion time, residual vibration, and

TABLE 3: The start and end configuration of the experiment 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

vibration reduction of the three cases are shown in Table 4. 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

TABLE 4: 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

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 seconds for all cases. That amount is equivalent to the length of the ZVD input shaper, which equals 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.

6. Conclusion and Further Work

This paper has examined the case of liquid container transfer system using multi joint robot arm where quick motion and minimal sloshing are concurrently addressed. The proposed framework integrates trajectory planning in three-dimensional space and input shaping for sloshing suppression. The trajectory planning algorithm, based on cubic spline optimization, has been built to generate trajectories for translation motion in three-dimensional space with joint kinematic constraints and obstacle avoidance. The sloshing suppression is built into the algorithm, based on principle of input shaping and decoupling of the robot motion into translation and hand rotation subsystems. It has been shown using simulations and experiments that the generated trajectories induce much less sloshing, thus demonstrating the benefit of the suppression strategy. Further work will mainly be directed to improve the current framework by incorporating the dynamics consideration into the algorithm. Another planned improvement is to provide redundancy by utilizing one additional joint of the robot arm for better maneuverability.

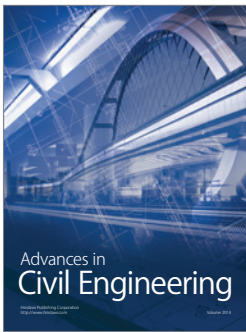
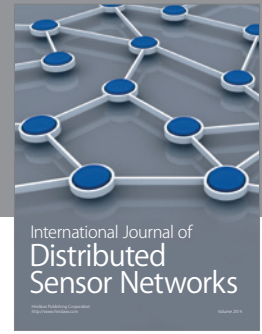
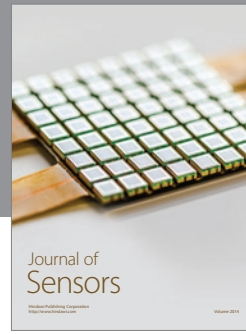
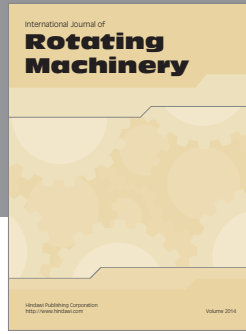
Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

- [1] C.-S. Lin, P.-R. Chang, and J. Y. S. Luh, "Formulation and optimization of cubic polynomial joint trajectories for industrial robots," *IEEE Transactions on Automatic Control*, vol. 28, no. 12, pp. 1066–1074, 1983.
- [2] A. Piazzi and A. Visioli, "Global minimum-jerk trajectory planning of robot manipulators," *IEEE Transactions on Industrial Electronics*, vol. 47, no. 1, pp. 140–149, 2000.
- [3] T. Chettibi, H. E. Lehtihet, M. Haddad, and S. Hanchi, "Minimum cost trajectory planning for industrial robots," *European Journal of Mechanics—A/Solids*, vol. 23, no. 4, pp. 703–715, 2004.
- [4] A. Gasparetto and V. Zanotto, "A technique for time-jerk optimal planning of robot trajectories," *Robotics and Computer-Integrated Manufacturing*, vol. 24, no. 3, pp. 415–426, 2008.
- [5] 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.
- [6] 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.
- [7] M. Reyhanoglu and J. Rubio Hervas, "Nonlinear modeling and control of slosh in liquid container transfer via a PPR robot," *Communications in Nonlinear Science and Numerical Simulation*, vol. 18, no. 6, pp. 1481–1490, 2013.
- [8] 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, NM, USA, 1997.
- [9] L. Consolini, A. Costalunga, A. Piazzzi, and M. Vezzosi, "Minimum-time feedforward control of an open liquid container," in *Proceedings of the 39th Annual Conference of the IEEE Industrial Electronics Society (IECON '13)*, pp. 3592–3597, IEEE, Vienna, Austria, November 2013.
- [10] K. Terashima and K. Yano, "Sloshing analysis and suppression control of tilting-type automatic pouring machine," *Control Engineering Practice*, vol. 9, no. 6, pp. 607–620, 2001.
- [11] M. Hamaguchi and T. Taniguchi, "Transfer control and curved path design for cylindrical liquid container," in *Proceedings of 15th IFAC World Congress*, pp. 1479–1479, Barcelona, Spain, 2002.
- [12] 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.
- [13] W. Aribowo, T. Yamashita, K. Terashima, and H. Kitagawa, "Input shaping control to suppress sloshing on liquid container transfer using multi-joint robot arm," in *Proceedings of the 23rd IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems (IROS '10)*, pp. 3489–3494, Taipei, Taiwan, October 2010.
- [14] 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.
- [15] N. C. Singer and W. P. Seering, "Preshaping command inputs to reduce system vibration," *Journal of Dynamic Systems, Measurement, and Control*, vol. 112, no. 1, pp. 76–82, 1990.

- [16] T. Singh and S. R. Vadali, "Robust time-delay control," *Journal of Dynamic Systems, Measurement and Control*, vol. 115, no. 2A, pp. 303–306, 1993.
- [17] M. P. Tzamtzi, F. N. Koumboulis, and N. D. Kouvakas, "A two stage robot control for liquid transfer," in *Proceedings of the 12th IEEE International Conference on Emerging Technologies and Factory Automation*, pp. 1324–1333, Patras, Greece, September 2007.
- [18] C. Williams, G. Starr, J. Wood, and R. Lumia, "Curvilinear transport of suspended payloads," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '07)*, pp. 4537–4543, IEEE, Roma, Italy, April 2007.
- [19] R. Ibrahim, *Liquid Sloshing Dynamics, Theory and Applications*, Cambridge University Press, New York, NY, USA, 2005.
- [20] J. Lawrence and W. Singhose, "Command shaping slewing motions for tower cranes," *Journal of Vibration and Acoustics*, vol. 132, no. 1, Article ID 011002, 11 pages, 2010.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

