SOFT COMPUTING APPROACHES FOR EFFECTIVE UTILIZATION OF ELASTIC ENERGY IN FLEXIBLE MANIPULATOR

DOCTORAL DISSERTATION

YIZHI GAI

DIVISION OF HUMAM MECHANICAL SYSTEMS AND DESIGN

GRADUATE SCHOOL OF ENGINEERING

HOKKAIDO UNIVERSITY

January, 2016

TABLE OF CONTENTS

Dissertation Abstract

学　位　論　文　内　容　の　要　旨

博士の専攻分野の名称　　博士（工学）　　氏名　蓋　軼之 (Yizhi GAI)

学　位　論　文　題　名

Soft Computing Approaches for Effective Utilization of Elastic Energy in Flexible Manipulator

(柔軟マニピュレータの弾性エネルギー活用のためのソフトコンピューティング手法)

The concept of Soft Computing was developed in the year around 1990. Soft Computing techniques are designed to model and exploit the solutions to real world problems. As the opposite concept of Soft Computing, Hard Computing requires precise model with exact input data and just produces precise answers. In engineering researches, the system with complexity, disturbances, nonlinearity, or time variation commonly requires the computing techniques that provide a certain tolerance to uncertainty, imprecision, and partial truth. Soft Computing can keep robustness, low solution cost, and flexibility to tackle such engineering problems.

Flexible manipulator own many outstanding characteristics comparing with the normal rigid manipulator such as light weight, low production cost, easy miniaturization, the characteristic of elastic energy storage, and better energy utilization efficiency. For

these advantages, the manipulator with flexible structure have drawn extensive attention on many aspects and applications such as handling radiation infected waste of nuclear power plant, precision measure instrument, and manipulator equipped on space shuttle. Fast motion control of the flexible arms is more difficult than that of rigid arms for involving some problems on position trajectory and vibration suppression. And there are challenging tasks for control of flexible manipulator comparing with that of the existing rigid manipulator. Amount of researches focus on the aspect which is commonly to control the performance of the flexible manipulator; however this research mainly pays attention for utilizing the flexible manipulator's elastic energy more effectively through the Soft Computing strategy. Then it is necessary to design the proper control strategy for better elastic energy utilizing performance.

In order to research the elastic energy of the flexible manipulator, this thesis investigates a ball throwing robot design with one rigid link and one flexible link. This manipulator is made for throwing a ball with motor torques and at the same time by utilizing the elastic energy of the flexible manipulator. Then for utilizing the elastic energy of this flexible manipulator more efficiently, this dissertation consists of the four parts as follows: Firstly, the frame structure and model of the ball throwing manipulator are invented with considering the bending deformation of the flexible link. The system model is built up on the design requirements. Then the proper control torques are designed and tested through the proposed simulation models. And the profile pattern of each torque is determined through Soft Computing approaches. Thirdly, the residual vibration is generated after ball has been thrown from manipulator. The vibration of the flexible manipulator needs to be suppressed with some control strategies. Finally, the simulations and experiments were operated to validate the theoretical derivation of the model, performance of the designed control inputs, and also the effect of the residual vibration suppression. In order to solve these problems, this thesis is organized in the following five chapters:

Chapter 1: The research background about the topic for this thesis was introduced in the first chapter. The contributions of this research were presented and the outlines of this thesis were also provided in this chapter.

Chapter 2: In order to realize the throwing motion with utilizing the elastic energy of the manipulator, the design of the manipulator with a proper structure is needed in the following chapter. In this chapter, a two link manipulator with one flexible link was designed and built up for utilizing the elastic energy. The structure of the manipulator was analyzed and the mathematical model of this flexible manipulator was derived through the Hamilton's principle. For the simulation of the whole motion, not only the system model while the manipulator holds the ball, but also the model after releasing the ball is necessary to be derived. Also in order to decrease the operation time of the program and code complexity for the Soft Computing approach, the linearized model is derived for the further simulation.

Chapter 3: In order to throw the ball with less vibration, the control parameters of the torque inputs should be determined properly. In this chapter, a kind of composite sinusoidal form control torque was proposed as the input for realizing the swing-throw motion and smooth releasing motion of the flexible manipulator. Then one of the swarm intelligence computations which belong to the Soft Computing approach was introduced to determine the control inputs of this ball throwing manipulator. The object functions for the swarm intelligence were designed to determine the control parameters of the input torques according to the dynamic characteristics of the flexible manipulator.

Chapter 4: After ball throwing, the residual vibration occurred on the flexible link. So in this chapter, a control strategy for residual vibration suppression was presented. For this manipulator, the control torque was designed before manipulator operation in the former chapter. This means during the throwing motion, the controller doesn't need to get any feedback signal from the beam structure. The input shaper was formed by a sequence of impulses with proper amplitudes and time locations and then the shaper was convolved with the desired command input. As the result, the system response of the shaped command input showed few or no vibration comparing with the unshaped one.

Chapter 5: In this chapter, some experiments were designed and implemented to check and evaluate the proposed method and simulation results based on the two link manipulator experimental device. The reliability of the model was confirmed on the experiment. And then, the performance of the Soft Computing strategy discussed in the third chapter was verified through contrast experiments. Also the effectiveness of the input shaper designed for the residual vibration suppression was evaluated through experiment in this chapter.

Chapter 6: Conclusions of the whole research are presented and the further studies are described in the final chapter.

Chapter 1 Introduction

1.1 Research Background

So many robots are commonly used all over the world, nowadays. Especially in the industrial field, many robotic manipulators are working in the tedious, repetitive, and dangerous positions instead of human beings' activities. But among these manipulators, most of them are designed with heavy materials and massy elements in order to suppose the vibration of the manipulators caused by their own flexibility. For its big mass and heavy inertial, this kind of rigid manipulator requires a large amount of energy to be operated and is hard to achieve a rapid motion. In order to overcome these weak points, flexible manipulator appears on the stage of this research field.

For flexible robotic manipulator, it owns many advantages such as better payload weight ratio, less operation power requirement, and higher motion speed comparing with the rigid one. Many researchers pay much attention in this field for these valued points over the past research. A pioneer researcher Book [1-2] presented the history, status, and future direction of the flexible system in his papers. Dwivedy and Eberhard [3] also states a literature review on the dynamics analysis of the flexible system. However, for the distributed nature of the governing equations which describe dynamics of the flexible systems, the control of such systems has traditionally involved complex processes [4-6]. Moreover, to compensate for flexural effects and thus yield robust control, the design focuses primarily on non-collocated controllers [7-8]. This research focuses on throwing motion of human being. In order to realize the throwing motion more humanlike or even exceed human beings, a manipulator equipped with the rapid motion function is necessary. The flexible manipulator is a better choice than others for its advantage to simulate fast action more human like or even exceed the athletic ability of human beings.

Research on flexible manipulator systems varies from a single-link manipulator rotating about a fixed axis [9] to three-dimensional multi-link arms [10]. However, experimental work, in general, is almost exclusively limited to single-link systems for the complexity of multi-link manipulator systems. This results from more degrees of freedom and the increased interactions between gross and deformed motions. It is important for control purposes to recognize the flexible nature of the manipulator system and to build a suitable mathematical framework for modeling of the system. Forward dynamics, inverse dynamics, and controller design are the three parts for using the dynamic models of the flexible manipulator systems. Flexible manipulators are distributed parameter systems with rigid body as well as flexible movements. Comparing the rigid manipulator systems, there are two physical limitations for the flexible systems. One is that the control torque of the flexible system can only be applied at the joint. The other is that only a finite number of sensors of bounded bandwidth can be applied and at restricted locations along the length of the manipulator.

The first experimental single-link flexible manipulator systems were developed in the early 1980s [7]. Although research for the control of single-link setups continues, much attention has now changed to multi-link flexible manipulator systems. A wide range of multi-link flexible manipulator systems exist in this field, starting form two-link experimental set-ups to seven degrees of freedom remote manipulator system for assembling work of space station. In Hu [11] research, a good survey of initial efforts in flexible manipulator research is given. But many of the manipulators discussed in Hu's paper are now decommissioned.

Typical flexible manipulator system's structures have not changed too much since the early experimental systems [7]. The current commercial availability of sensor and actuator hardware has made it much easier to build an experimental flexible manipulator system. A typical flexible manipulator system owns a flexible link, an actuator-gear mechanism to rotate the link, an optical encoder to measure joint rotation, accelerometers, and strain gauges to sense flexible motion, an optical arrangement to measure the end-point position, and an occasional force sensor attached to the end-point. However variations in configuration of different flexible manipulator exist among them. Some setups have directly driven d.c. motors and others are equipped with harmonic drive gears with d.c. motors; some only have accelerometers or strain gauges to measure the deflections of the flexible link while others own cameras; some have semi-rigid flexible links while others have very flexible links. The researchers can make a decision to meet the needs of their experimental equipment in a wide variety by selecting different sensor and actuator hardware.

The single-link flexible manipulator systems illustrated in Nagarkatti and Li's paper [12-13] use the camera to obtain the end-point position. The system used in Su's paper [14] has a photodiode. A camera lens focuses light on the photodiode and infrared emitting diode to sense end-point deflection of flexible link. In Landau's research [15], a very flexible arm is made of two aluminum strips coupled by ten regularly spaced rigid frames. A d.c. motor is used as actuator and encoder with a one-turn potentiometer records the angular rotation data. The sensor to measure the end-point deflection is unique. It combines a mirror-detector at the hub with an LED at the end-point. The mirror is placed on the center of the hub axis and can be rotated by a d.c. motor. Another single-link flexible manipulator in Goh's research [16] is constructed to move in both horizontal and vertical planes. It allows any coupling between the motions to be studies. The flexible link of this manipulator is a homogeneous, cylindrical, aluminum rob with constant properties along its length. Unlike many other experimental setups, the link motion is not artificially constrained in any plane within the safe working envelope. It is symmetrical about its longitudinal axis, and the axis intersects horizontal and vertical axes at the hub. The end-point displacements of the flexible link on horizontal and vertical direction are calculated through the following measurements. At the beginning, a precision potentiometer is used to measure the end-point position of the link in relation to the hub axis. Then an incremental optical encoder is used to measure the angle of the hub in relation to horizontal or vertical axis. Finally these data are utilized for implementing the control algorithms. In Alam and Tokhi's research [105-106], the vibration control of a single-link flexible manipulator operated on a horizontal plane was developed. And the optimization strategy were applied in his research for designing the optimal control for the manipulator.

Most two-link flexible manipulators are constrained to the horizontal plane. Stanford University built one of the earliest two-link flexible manipulator for experimental research to demonstrate non-collocated control [17]. The manipulator's links are actuated by directly connected d.c. motors in a horizontal plane. The rotary variable differential transformers measure the joint rotations. The end-point position is measured by a CCD camera. The links of the manipulator are made by very flexible

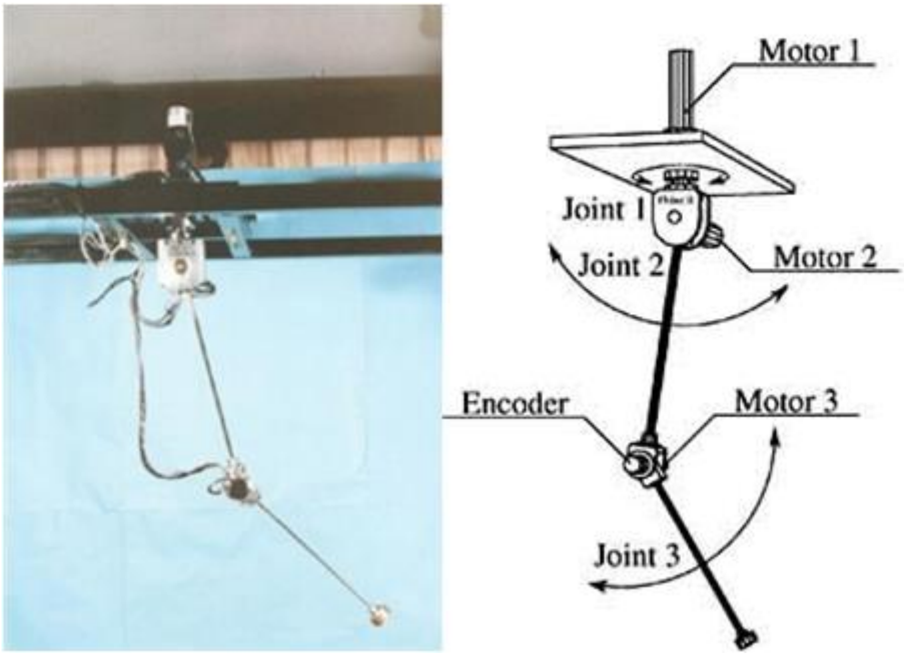


Fig. 1-1 Two-link flexible manipulator [104]

material. University of Washington builds the two-link flexible manipulator which motion is limited to the horizontal plane for position and force control research [18]. It is actuated by two direct drive d.c. motors, joint angles are measured through optical encoders, end-point position is measured by a three-dimensional infrared tracking system, and it has a six DOF torque sensor attached to its end-point. The two-link flexible manipulator built as the experimental space robot simulator contains two flexible links in Romano's research [19]. Both links are actuated by d.c. motors with 50 times gear ratio harmonic drives. Four air pads on a granite table support the manipulator. The optical encoders mounted on motor shafts are used to obtain the joint angular position. End-point position is measured through a monochrome camera. Another two-link flexible manipulator built by Keio University [20] drives first link and second link with the directly driven d.c. motor and harmonic geared motor separately. And a position sensitive detector mounted above the two-link flexible manipulator is used to measure the end-point position. Most flexible manipulator devices use external optical sensing facility to measure the end-point position. But many experimental facilities use strain gauges for measuring the flexible deformation [21-24]. A two-link flexible manipulator system named as FLEBOT II is shown as Fig. 1-1. It owns two flexible links and three joints. It is designed for the space robotics application research. Its operation space includes both vertical and horizontal plane which setup is different from other manipulators. Hoshino [107] in Hokkaido University developed a two-link golf-swing like Fig. 1-2. It has a rigid link with a golf shaft as the flexible link. This manipulator focuses on vibration control problem with the vibration excited after hitting the golf ball. Also Yin [108] from Hokkaido University had a two-link manipulator research as Fig. 1-3. This research tried to suppress the vibration occurred on flexible link after a fast motion on a three dimensional space.

For multi-link flexible manipulator, Tohoku University constructs an experimental aerospace dual-arm flexible manipulator [25]. It has two arms with seven joints and two flexible links in each arm. Contact torque is measured by the force sensors attached to the end-effectors. Link deflections are measured by strain gauges. The displacement between the end-effector and the surface of the object are measured by laser displacement sensors. It is driven by hydraulic actuators for handing heavy payloads. Ruhr-University builds a four-link manipulator with each link composite of rigid and flexible part [26]. It has four joints and all of them are driven by d.c. motors with harmonic gear drives. Elastic deflections of the links are measured by strain gauges.

There are some researches on simulating human being’s throwing or batting motion, such as a round plate throwing robot using a rigid link in Fig. 1-4 [27] and multi-joint batting manipulator in Fig. 1-5 [28]. However, they are all rigid link manipulators comparing with the flexible one.

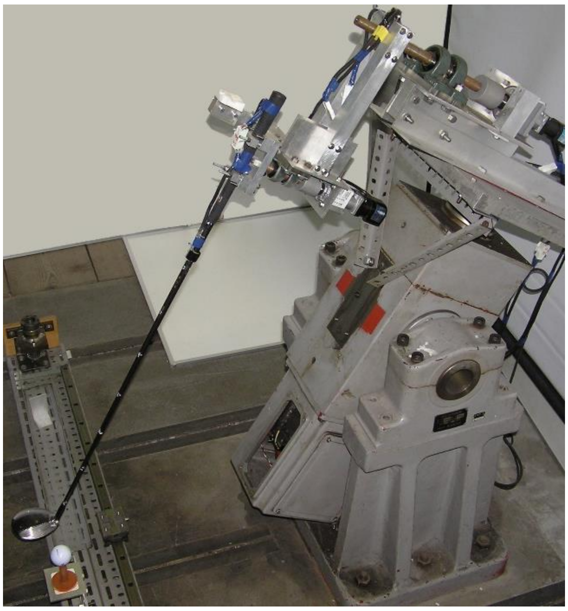


Fig. 1-2 Golf-swing robot [107]

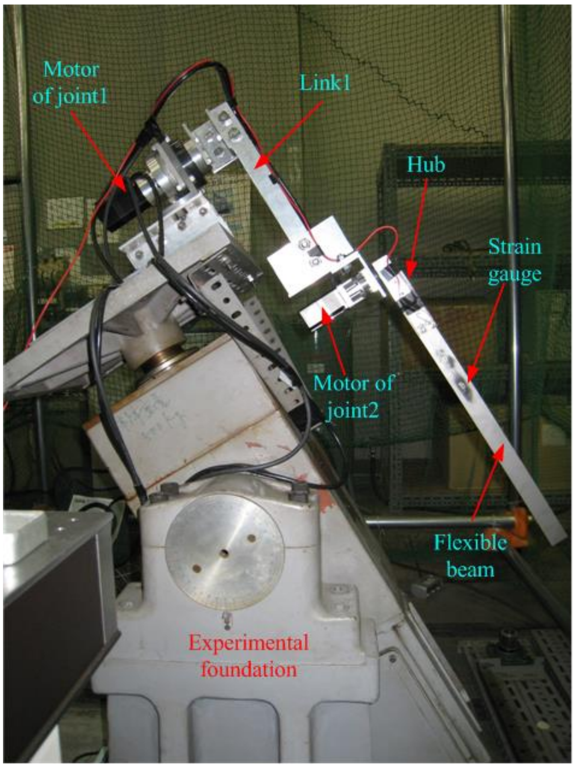


Fig. 1-3 two link fast motion robot [108]

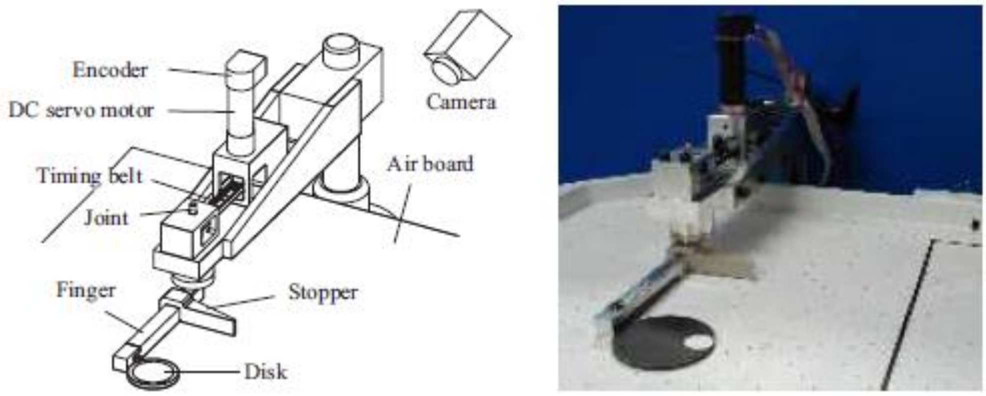


Fig. 1-4 Round plate throwing robot [27]

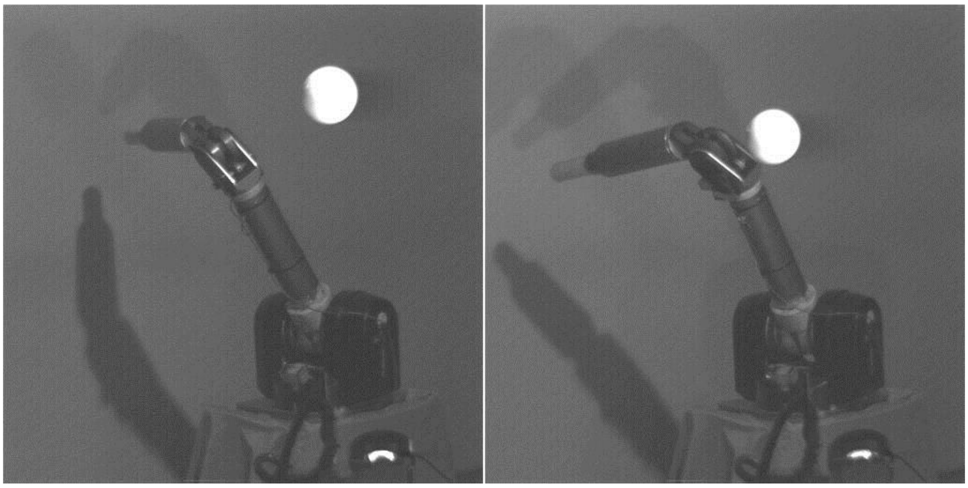


Fig. 1-5 Batting manipulator [28]

1.2 Task and Outline of This Research

This paper's main purpose is to apply soft computing technique to research and develop a flexible throwing manipulator which can utilize elastic energy efficiently. On the basis of the research background for this throwing robot, this thesis proposes three tasks listed as follows:

Firstly, the model of this ball throwing robot manipulator is investigated with considering the bending mechanism of the flexible beam. A mechanical structure of this manipulator needs to be designed for fulfilling the throwing task. Secondly, the proper control torques is required for realizing the throwing motion. Thirdly, the residual vibration of the flexible link of the manipulator should be reduced. Then the vibration suppression scheme is proposed to reduce the residual vibration. Finally, the experiment is carried out to validate the theoretical models and scheme which are proposed.

In order to fulfill the above tasks, this paper consists of five chapters and is organized in the following chapters:

Chapter 1 presents the research background about the ball throwing manipulator. The tasks of this research are introduced and the brief outlines of this paper are provided in this chapter.

Chapter 2 shows the structure and model of the throwing manipulator. And the Hamilton's principle is introduced to derive the dynamic model of this flexible throwing manipulator. For the flexible structure, the mode shapes are needed to be obtained. So the model constraint method is introduced to calculate them.

Chapter 3 introduces the PSO (Particle Swarm Optimization) algorithm which is a kind of Soft Computing technique for optimization operation. In order to obtain proper control torque, the profile of the control torque is designed and optimized through PSO and advanced PSO algorithm.

Chapter 4 investigates the feedforward strategy for suppressing the residual vibration of the flexible link. In this section, the input shaping technical scheme is used. Two methods are applied here. One is the Zero Vibration Derivative (ZVD) shaper and the other one is Infinite Impulse Response (IIR) filter.

Chapter 5 presents the experiments which are carried out to validate the theories and performances of the control strategies listed on the above chapters.

Chapter 6 concludes the whole thesis and presents the further works based on this research.

Chapter 2 Theoretical Analysis of Two-link Manipulator

2.1 Introduction

The research of the robotic manipulator is always an active research field. On the technological aspect, the robotic manipulator arms have been found that they were highly practical on the ground and aerospace applications. But on the theoretical aspect, their systems show the complex dynamics for analyzing. The introduction of the flexible structure for the manipulator adds additional challenges beyond the only rigid link manipulator. The low stiffness allows the flexible manipulator to vibrate owing to elastic deformation. And this pushes the dynamic analysis of the flexible manipulator further complex.

However in order to control the flexible manipulator, firstly, a mathematical model of the flexible manipulator is needed to be derived. Different models for flexible manipulators have been developed according to various principles. For example, Lagrange equation and modal expansion (or assumed mode method) [29-31], Lagrange equation and finite elements method [32-34], Euler-Newton equation and model expansion [35-36], Euler-Newton equation and finite element method [37-39], singular perturbation [40], and frequency-domain techniques [41-42].

For Lagrange equation and modal expansion method, the deformation of the flexible link of the manipulator can be calculated as the summation of all modes. For each mode, it can be represented through the product of two functions. One is the function that dependent on time in a generalized coordinate. The other one is the function dependent on the coordinate of length through the link of manipulator. Theoretically, the summation should include the infinite number modes. However in practical condition, a finite number of modes can be used in calculation. For Lagrange equation and finite elements method, it is similar to the above method, and only the generalized coordinates here is replaced by the displacement and slope on the specific points along the link of manipulator.

For the Euler-Newton's method, they are more direct method for calculating the system dynamics of the manipulator. Instead of Lagrange equation, Newton's second law is applied to balance the terms of change rate of linear and angular momentum through the applied forces [43]. In forward dynamics, the forces from actuators are known conditions while the manipulator's linear and angular momentums are unknown conditions. Euler-Newton equation and model expansion approach and divide the manipulator into a number of elements, and then it balances the dynamics of each element. This method is much easier to take the non-linear effects into consideration without complicating the system model. However the process will be so tedious for a model with a large number of elements.

For the singular perturbation method, the system modes can be divided into two kinds of modes. One is the low frequency mode and the other can be viewed as high frequency mode. So the system dynamics can be separated as slow subsystem and fast subsystem. For the flexible system, the rigid body part is the low frequency mode or the slow subsystem which can be viewed as the same order as the equivalent rigid manipulator. The flexible body part is the high frequency mode or the fast subsystem and the variables of the slow subsystem can be considered as the constant values comparing with that of fast subsystem.

For frequency-domain technique, it is a method modeling the manipulator in the time domain space which is based on the frequency domain analysis. And a uniform Euler-Bernoulli beam theory is used to build up a transfer matrix model. But this method has some disadvantages. For example, the manipulator's gross motion and flexible dynamics can't be considered easily in this method and only an approximate model for the manipulator can be obtained.

In this section, the system model of the throwing manipulator is characterized through a set of infinite natural frequency modes, and then state equation based on state-space is developed.

2.2 Mathematical Modal of Two-link Manipulator

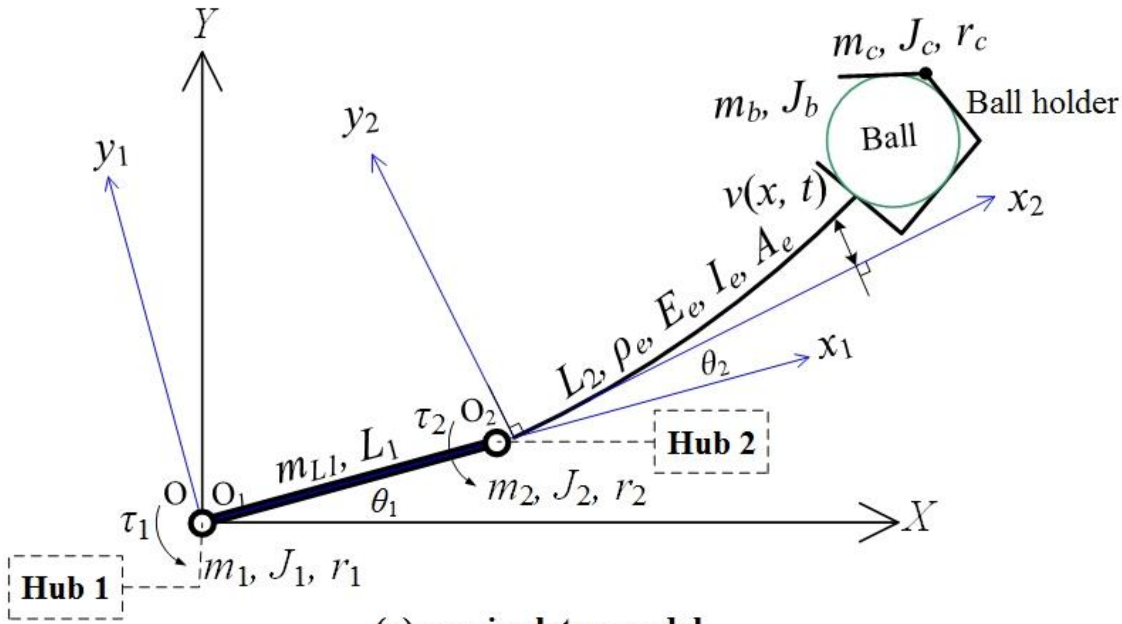
2.2.1 Modeling of Two-link Manipulator

For this kind of robotic manipulator, it was designed to throw the ball fast with less energy consumption. So through the flexible structure, the weight of the manipulator structure and utilize its deformation energy could be reduced at the same time. So a two-link flexible manipulator which is able to perform the action of throwing a ball was designed.

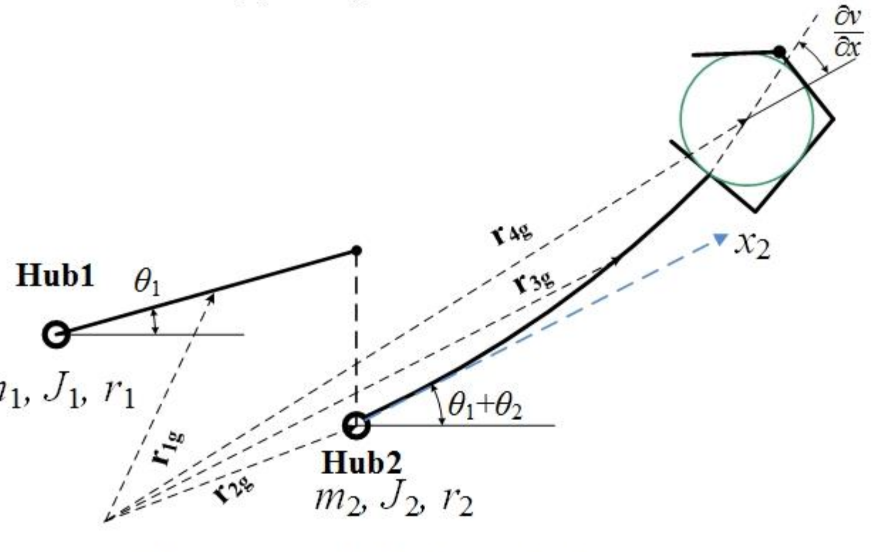
Figure 2-1 shows the model of the flexible manipulator with a ball discussed in this paper. The model consists of a rigid link fixed on a hub where rotational coordinates O_1-x_1y_1 is fixed on the hub 1, and its mass, inertia, and radius is represented by m_1, J_1, and r_1, respectively. Then a flexible beam fixed on hub 2 where rotational coordinates O_2-x_2y_2 is fixed on the hub 2. Similarly, m_2, J_2, and r_2 represents the mass, inertia, and radius of hub 2, respectively. A ball holder which is used to hold a ball is fixed on extreme end of the flexible part of the manipulator. For the ball holder, m_c, J_c, and r_c are the mass, inertia, and equivalent radius for it. The global coordinates O-XY denotes the inertial reference frame. For the first rigid link, the link mass is m_{L1} and L_1 is the link length. The angles θ_1 and θ_2 represents the rotational angles of the rigid link relative to the inertial reference frame and the flexible link relative to the coordinate O_1-x_1y_1 under the control torques τ_1 and τ_2, respectively. On both sides of the flexible beam which length is represented by L_2, it is reinforced through the FRP sheets. Then ρ_e, E_e, I_e, and

A_e are the equivalent density, Young's modulus, inertia of cross section, and cross section for the flexible link. In this model, $v(x,t)$ represents the displacement due to a deflection at an arbitrary point x of the flexible link. The mass and inertia of the ball grabbed in the ball holder is m_b and J_b. In Figure 2-1 (b), it shows the position vectors of the manipulator system. Here, $\mathbf{r}_{1g}$, $\mathbf{r}_{2g}$ and $\mathbf{r}_{4g}$ represent the position vectors of the gravity center for the rigid link, the second hub, and ball holder, respectively. Similarity, $\mathbf{r}_{3g}(x)$ is the position vector of the infinitesimal element at position x of the flexible link.

Figure 2-1 represents only the condition of holding a ball. Indeed the whole process for a ball throwing motion is shown as Figure 2-2. Besides model 1 (Figure 2-2 (a)) is same as in Figure 2-1, and model 2 shown in Figure 2-2 (b) is the model after throwing motion. Figure 2-3 shows the model of the flexible manipulator after throwing the ball discussed in this paper.



(a) manipulator model



(b) vector model of the manipulator

Fig. 2-1 Model of a ball throwing robot (with a ball)

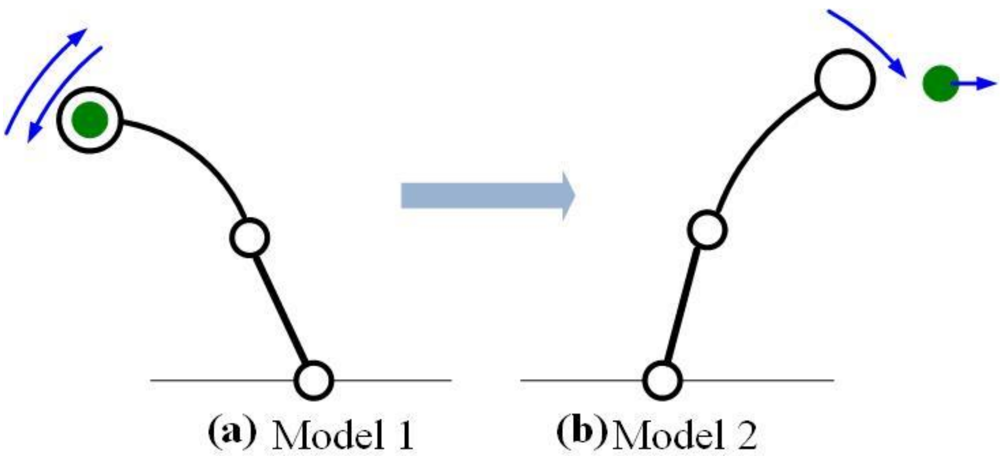


Fig. 2-2 Throwing motion of manipulator

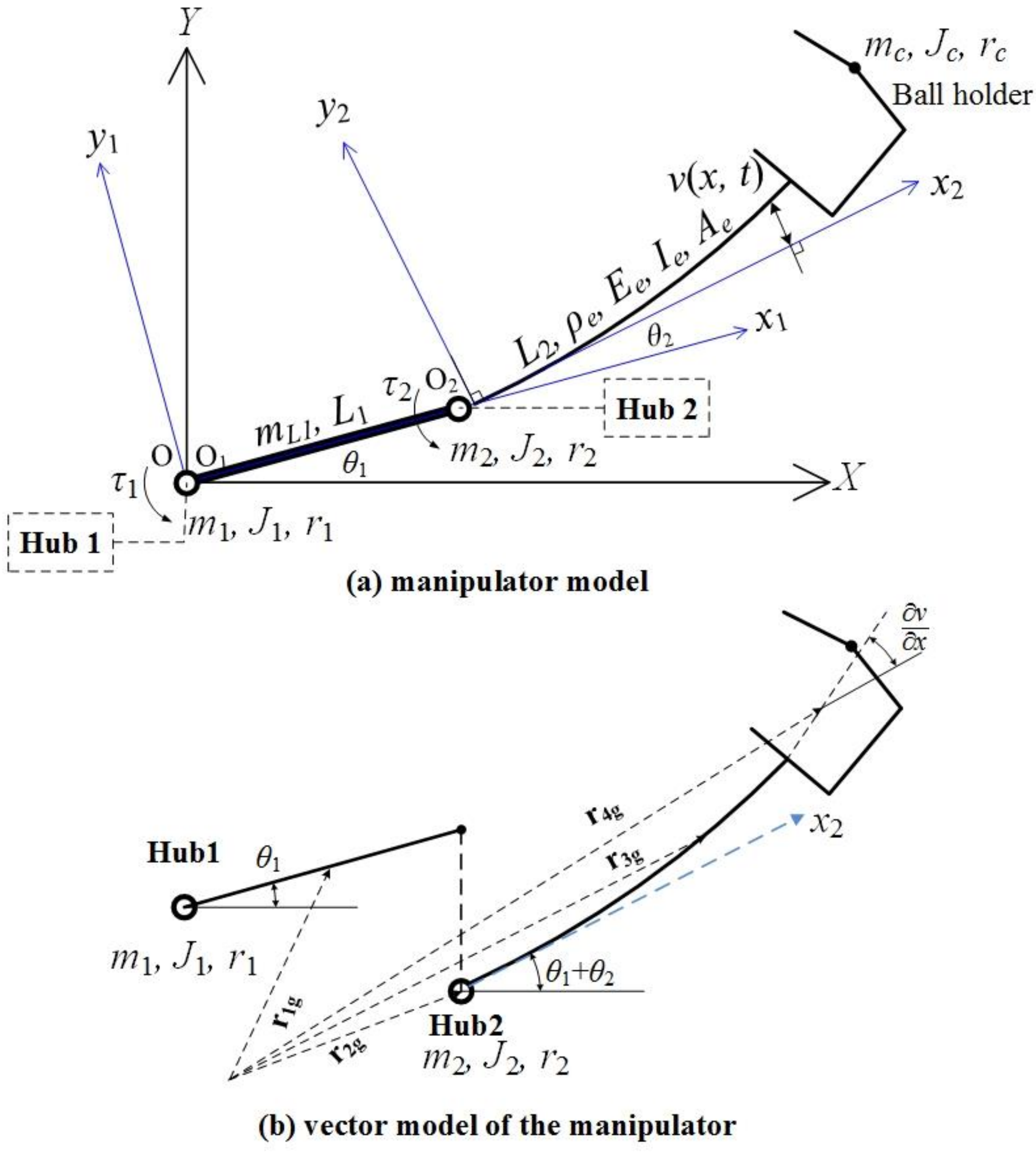


Fig. 2-3 Model of a ball throwing robot (without a ball)

For the model in Fig. 2.1, the manipulator is assumed as an Euler-Bernoulli beam model; Comparing with the length of the beam, the thickness is small enough to be neglected; the manipulator is operated on the horizontal plane, so the influence of gravity can be neglected. Relationship between the rotational coordinate O$_1$-x_1y_1 which is fixed on the hub 1 and global coordinate O-XY can be expressed as

$$\begin{Bmatrix} X \\ Y \\ 1 \end{Bmatrix} = \mathbf{T}_{\mathrm{O}_1\to\mathrm{O}} \begin{Bmatrix} x_1 \\ y_1 \\ 1 \end{Bmatrix} = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} x_1 \\ y_1 \\ 1 \end{Bmatrix}. \tag{2.1}$$

Here $\mathbf{T}_{\mathrm{O}_1\to\mathrm{O}}$ is the transform matrix from the rotational coordinate O$_1$-x_1y_1 to the global coordinate O-XY.

Relationship between the rotational coordinate O$_2$-x_2y_2 which is fixed on the hub 1 and global coordinate O$_1$-x_1y_1 can be expressed as

$$\begin{Bmatrix} x_1 \\ y_1 \\ 1 \end{Bmatrix} = \mathbf{T}_{\mathrm{O}_2\to\mathrm{O}_1} \begin{Bmatrix} x_2 \\ y_2 \\ 1 \end{Bmatrix} = \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & r_2\cos\theta_2 + L_1 + r_1 \\ \sin\theta_2 & \cos\theta_2 & r_2\sin\theta_2 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} x_2 \\ y_2 \\ 1 \end{Bmatrix}. \tag{2.2}$$

Here $\mathbf{T}_{\mathrm{O}_2\to\mathrm{O}_1}$ is the transform matrix from the rotational coordinate O$_2$-x_2y_2 to the global coordinate O$_1$-x_1y_1.

The position vector for the gravity center of the first rigid link $\mathbf{r}_{1g}$, the second hub $\mathbf{r}_{2g}$, infinitesimal element at position x of the flexible link $\mathbf{r}_{3g}(x)$, and ball holder $\mathbf{r}_{4g}$ can be expressed as

$$\mathbf{r}_{1\mathrm{g}} = \mathbf{T}_{\mathrm{O}_1 \to \mathrm{O}} \begin{Bmatrix} 0.5L_1 + r1 \\ 0 \\ 1 \end{Bmatrix},$$

$$\mathbf{r}_{2\mathrm{g}} = \mathbf{T}_{\mathrm{O}_1 \to \mathrm{O}} \begin{Bmatrix} L_1 + r_1 + r_2 \\ 0 \\ 1 \end{Bmatrix},$$

$$\mathbf{r}_{3\mathrm{g}}\left(x\right) = \mathbf{T}_{\mathrm{O}_1 \to \mathrm{O}} \mathbf{T}_{\mathrm{O}_2 \to \mathrm{O}_1} \begin{Bmatrix} x \\ v\left(x,t\right) \\ 1 \end{Bmatrix}, \tag{2.3}$$

$$\mathbf{r}_{4\mathrm{g}} = \mathbf{T}_{\mathrm{O}_1 \to \mathrm{O}} \mathbf{T}_{\mathrm{O}_2 \to \mathrm{O}_1} \begin{Bmatrix} L_2 + r_c \\ v_e\left(t\right) \\ 1 \end{Bmatrix}.$$

Here v_e is the gravity center of the ball holder, and it can be approximated by

$$v_e\left(t\right) = v\left(L_2, t\right) + r_c \frac{\partial v\left(L_2, t\right)}{\partial x}. \tag{2.4}$$

For the equation of motion of the system, the kinetic energy of the whole system, R, can be expressed as

$$R = R_{\mathrm{hub1}} + R_{\mathrm{rigid}} + R_{\mathrm{hub2}} + R_{\mathrm{flexible}} + R_{\mathrm{holder}} + R_{\mathrm{ball}}, \tag{2.5}$$

where the kinetic energies for each part, R_{hub1}, R_{rigid}, R_{hub2}, R_{flexible}, R_{holder}, and R_{ball} can be obtained as:

Hub 1

$$R_{\mathrm{hub1}} = \frac{1}{2} J_1 \dot{\theta}_1^2 \tag{2.6}$$

First rigid link

$$R_{\text{rigid}} = \frac{1}{2} m_{L1} \dot{\mathbf{r}}_{1g}^T \dot{\mathbf{r}}_{1g} \tag{2.7}$$

Hub 2

$$R_{\text{hub2}} = \frac{1}{2} m_2 \dot{\mathbf{r}}_{2g}^T \dot{\mathbf{r}}_{2g} + \frac{1}{2} J_2 \left(\dot{\theta}_1 + \dot{\theta}_2 \right)^2 \tag{2.8}$$

Second flexible link

$$R_{\text{flexible}} = \int_0^{L_2} \frac{1}{2} \rho_e A_e \dot{\mathbf{r}}_{3g}^T (x) \dot{\mathbf{r}}_{3g} (x) dx \tag{2.9}$$

Ball holder

$$R_{\text{holder}} = \frac{1}{2} m_c \dot{\mathbf{r}}_{4g}^T \dot{\mathbf{r}}_{4g} + \frac{1}{2} J_c \left\{ \dot{\theta}_1 + \dot{\theta}_2 + \frac{\partial v(L_2, t)}{\partial x} \right\}^2 \tag{2.10}$$

Ball

$$R_{\text{ball}} = \frac{1}{2} m_b \dot{\mathbf{r}}_{4g}^T \dot{\mathbf{r}}_{4g} + \frac{1}{2} J_b \left\{ \dot{\theta}_1 + \dot{\theta}_2 + \frac{\partial v(L_2, t)}{\partial x} \right\}^2 \tag{2.11}$$

Here ($\dot{}$) is the time differential operator. For the assumption mentioned at the former part of this section, the manipulator is operated on a horizontal plane, and then the gravity effect for this manipulator can be neglected here. So the potential energy of the flexible link, V, can be expressed as

$$V = \frac{1}{2}\int_0^{L_2} E_e I_e \left\{ \frac{\partial^2 v(x,t)}{\partial x^2} \right\}^2 dx. \tag{2.12}$$

The external work W generated by τ_1 and τ_2 is obtained from:

$$W = \tau_1 \theta_1 + \tau_2 \theta_2. \tag{2.13}$$

2.2.2 Equations of Motion for the Manipulator

In order to obtain the equation of motion of the system with boundary conditions, Hamilton's principle is introduced as:

$$\delta \int_{t_1}^{t_2} (R - V + W)\, dt \equiv 0, \tag{2.14}$$

based on Eq. (2.5), (2.12), and (2.13) with Hamilton's principle above and variation method, the governing equations of the system with the boundary conditions can be derived as:

Equations of motion for $\delta\theta_1$:

$$
\begin{aligned}
&\ddot{\theta}_1 \Big\{ J_1 + J_2 + J_c + J_b + m_{L_1}\left(0.5L_1 + r_1\right)^2 + m_2 L_r^2 + \left(m_c + m_b\right) \\
&\Big(\underline{2L_f L_r \cos\theta_2 - 2L_r \upsilon_e \sin\theta_2} + L_{rf}^2 + r_2^2 + L_r^2 + 2L_{rf} r_2 + \underline{v_e^2} \Big) + \\
&\rho_e A_e \int_0^{L_2} \Big[\underline{2L_r\left(x + L_r\right)\cos\theta_2 - 2L_r \upsilon \sin\theta_2} + L_r^2 + \left(r + x\right)^2 + \underline{v^2} \Big] dx \Big\} + \\
&\ddot{\theta}_2 \Big\{ J_2 + J_c + J_b + \left(m_c + m_b\right) \\
&\Big[\underline{L_f L_r \cos\theta_2 - L_r v_e \sin\theta_2} + \left(L_{rf} + r\right)^2 + \underline{v_e^2} \Big] \\
&+ \rho_e A_e \int_0^{L_2} \Big[\underline{L_r\left(x + r_2\right)\cos\theta_2 - L_r \upsilon \sin\theta_2} + \left(r_2 + x\right)^2 + \underline{v^2} \Big] dx \Big\} + \\
&\ddot{v}_e \left(m_c + m_b\right)\Big(\underline{L_r \cos\theta_2} + L_f \Big) \\
&+ \rho_e A_e \int_0^{L_2} \Big(\underline{L_r \cos\theta_2} + x + r_2 \Big) \ddot{v}\, dx + \left(J_c + J_b\right)\ddot{v}'\left(L_2\right) \\
&- \underline{\left(m_c + m_b\right)\Big[2L_f \dot{\theta}_1 \dot{\theta}_2 + L_f \dot{\theta}_2^2 + 2\left(\dot{\theta}_1 + \dot{\theta}_2\right)\dot{v}_e \Big] L_r \sin\theta_2} \\
&- \underline{\left(m_c + m_b\right)\left(2\dot{\theta}_1 \dot{\theta}_2 v_e + \dot{\theta}_2^2 v_e\right) L_r \cos\theta_2} + 2\left(m_c + m_b\right) \\
&\underline{\left(\dot{\theta}_1 + \dot{\theta}_2\right) v_e \dot{v}_e + \rho_e A_e \int_0^{L_2} \Big\{ \Big[2\left(x + r_2\right)\dot{\theta}_1 \dot{\theta}_2 - \left(x + r_2\right)\dot{\theta}_2^2} \\
&\underline{+ 2\left(\dot{\theta}_1 + \dot{\theta}_2\right)\dot{v} \Big] L_r \sin\theta_2 - \left(2\dot{\theta}_1 \dot{\theta}_2 v + \dot{\theta}_2^2 v\right) L_r \cos\theta_2} \\
&+ \underline{2\left(\dot{\theta}_1 + \dot{\theta}_2\right) v \dot{v}} \Big\} dx = \tau_1,
\end{aligned}
\tag{2.15}
$$

Equations of motion for $\delta\theta_2$:

$$
\begin{aligned}
&\ddot{\theta}_1 \Big\{ J_2 + J_c + J_b + \left(m_c + m_b\right) \\
&\Big[\underline{L_f L_r \cos\theta_2 - L_r v_e \sin\theta_2} + \left(L_{rf} + r_2\right)^2 + \underline{v_e^2} \Big] + \\
&\rho_e A_e \int_0^{L_2} \Big[\underline{L_r\left(x + r_2\right)\cos\theta_2} - \underline{L_r v \sin\theta_2 + v^2} + \left(x + r_2\right)^2 \Big] dx \Big\} + \\
&\ddot{\theta}_2 \Big\{ J_2 + J_c + J_b + \left(m_c + m_b\right)\Big[\left(L_{rf} + r_2\right)^2 + \underline{v_e^2} \Big] + \\
&\rho_e A_e \int_0^{L_2} \Big[\left(r_2 + x\right)^2 + \underline{v^2} \Big] dx \Big\} + \\
&\ddot{v}_e \left(m_c + m_b\right) L_f + \\
&\rho_e A_e \int_0^{L_2} \left(x + r_2\right)\ddot{v} dx + \left(J_c + J_b\right)\ddot{v}'\left(L_2\right) + \underline{L_r \dot{\theta}_1^2 \left(m_c + m_b\right)} \\
&\Big(\underline{L_f \sin\theta_2 + v_e \cos\theta_2} \Big) + \underline{2\left(m_c + m_b\right)\left(\dot{\theta}_1 + \dot{\theta}_2\right) v_e \dot{v}_e} \\
&+ \underline{\rho_e A_e \int_0^{L_2} L_r \dot{\theta}_1^2 \Big[\left(x + r_2\right)\sin\theta_2 + v \cos\theta_2 \Big] + 2\left(\dot{\theta}_1 + \dot{\theta}_2\right) v \dot{v} \Big\} dx} \\
&= \tau_2,
\end{aligned}
\tag{2.16}
$$

Equations of motion for δv :

$$\begin{aligned}&\rho_e A_e \left(\underline{L_r \cos\theta_2} + r_2 + x \right) \ddot{\theta}_1 + \rho_e A_e \left(r_2 + x \right) \ddot{\theta}_2 + \rho_e A_e \ddot{v} \\ &+ E_e I_e v'''' + C E_e I \dot{v}'''' + \underline{\rho_e A_e L_r \dot{\theta}_1^2 \sin\theta_2 - \rho_e A_e v \left(\dot{\theta}_1 + \dot{\theta}_2 \right)^2} = 0,\end{aligned} \tag{2.17}$$

with the boundary conditions:

at x=0

$$v(0) = 0, \tag{2.18}$$

$$v'(0) = 0, \tag{2.19}$$

at x=L_2

$$\begin{aligned}&\left(m_c + m_b \right)\left(L_f + L_r \cos\theta_2 \right) \ddot{\theta}_1 + \left(m_c + m_b \right) L_f \ddot{\theta}_2 \\ &+ \left(m_c + m_b \right) \ddot{v}_e - \left(m_c + m_b \right)\left(\dot{\theta}_1 + \dot{\theta}_2 \right)^2 v_e + \left(m_c + m_b \right) \\ &L_r \dot{\theta}_1^2 \sin\theta_2 - E_e I_e v'''\left(L_2 \right) = 0,\end{aligned} \tag{2.20}$$

$$\begin{aligned}&\left(J_c + J_b \right)\left(\ddot{\theta}_1 + \ddot{\theta}_2 \right) + \left(J_c + J_b \right) \ddot{v}'\left(L_2 \right) + E_e I_e v'''\left(L_2 \right) \\ &+ r_c E_e I_e v''\left(L_2 \right) = 0,\end{aligned} \tag{2.21}$$

where

$$\begin{aligned}L_r &= L_1 + r_1 + r_2 \\ L_{rf} &= L_2 + r_c \\ L_f &= L_2 + r_2 + r_c \\ v_e &= v\left(L_2 \right) + r_c v'\left(L_2 \right).\end{aligned}$$

Here (˙) is the time differential operator and (´) is the differential operator to *x*.

2.3 Modal Analysis for the Manipulator

For the equations of motion in the former section, the vibration displacement should be expressed through the sum of infinite vibration modes. But in practical simulation or control design, only a finite number of modes are necessary to be applied. In this section, a method called mode constraint method is used to obtain the mode shape of this flexible robot manipulator. So here the flexible arm is assumed to be fixed to a wall and can't be rotated. The orthogonality of the eignfunctions is applied to obtain the approximate solutions of the vibration displacement.

2.3.1 Mode Constraint Method

In this section, in order to use the model constraint method, the cantilever beam model shown as Figure 2-4 is used here. The equation of motion for this beam is

$$\rho_e A_e \frac{\partial^2 v(x,t)}{\partial t^2} + E_e I_e \frac{\partial^4 v(x,t)}{\partial x^4} = 0, \tag{2.22}$$

with the boundary conditions,

$$v(0,t) = \frac{\partial v(0,t)}{\partial x} = 0, \tag{2.23}$$

$$(m_c + m_b)\frac{d^2}{dt^2}\left\{v(L_2,t) + r_c \frac{\partial v(L_2,t)}{\partial x}\right\} = E_e I_e \frac{\partial^3 v(L_2,t)}{\partial x^3}, \tag{2.24}$$

$$(J_c + J_b)\frac{\partial^3 v(L_2,t)}{\partial t^2 \partial x} = -E_e I_e \left\{\frac{\partial^2 v(L_2,t)}{\partial x^2} + r_c \frac{\partial^3 v(L_2,t)}{\partial x^3}\right\}, \tag{2.25}$$

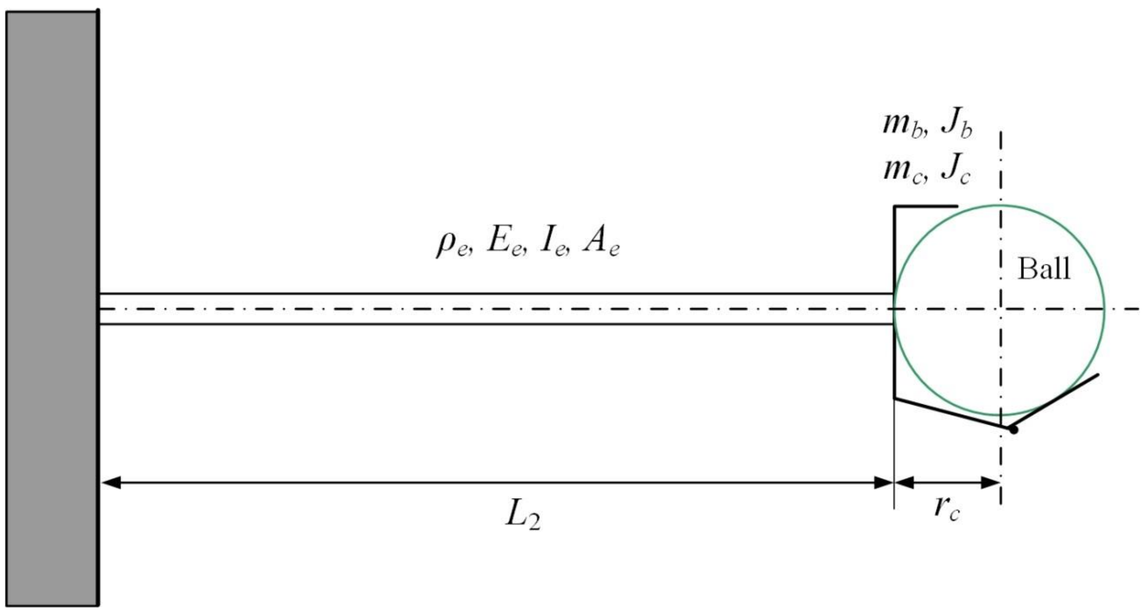


Fig. 2-4 Cantilever beam model with an end-effector

2.3.2 Equations of Mode Shape

The displacement of the vibration v can be expressed by an eigenfunction $\phi_i(x)$ obtained from the ith order vibration mode of a contilevered beam and a time function $q_i(t)$ [43].

$$v(x,t)=\sum_{i=1}^{\infty}\phi_i(x)q_i(t). \tag{2.26}$$

And the equations from Eq. (2.22) to Eq. (2.25) can be rewritten as

$$E_eI_e\phi''''(x)-\rho_eA_e\omega^2\phi(x)=0, \tag{2.27}$$

$$\phi(0)=\phi'(0)=0, \tag{2.28}$$

$$E_eI_e\phi'''(L_2)=-\omega^2(m_c+m_b)\frac{d^2}{dt^2}\left[\phi(L_2)+r_c\phi'(L_2)\right], \tag{2.29}$$

$$E_eI_e\left[\phi''(L_2)+r_c\phi'''(L_2)\right]=\omega^2(J_c+J_b)\phi'(L_2), \tag{2.30}$$

here

$$k^4=\frac{\rho_eA_e}{E_eI_e}\omega^2, \tag{2.31}$$

then Eq. (2.27) can be rewritten as

$$\phi''''(x)-k^4\phi(x)=0. \tag{2.32}$$

The general solution form of the above differential equation is given as

$$\phi(x) = a\cos kx + b\sin kx + c\cosh kx + d\sinh kx, \tag{2.33}$$

Substituting this general solution into the boundary conditions, it can get

$$a + c = 0, \quad b + d = 0, \tag{2.34}$$

$$\begin{aligned}
&\rho_e A_e \left(a\sin kL_2 - b\cos kL_2 + c\sinh kL_2 + d\cosh kL_2\right) + \\
&k\left(m_c + m_b\right)\left(a\cos kL_2 + b\sin kL_2 + c\cosh kL_2 + d\sinh kL_2\right) + \\
&r_c k^2\left(m_c + m_b\right)\left(-a\sin kL_2 + b\cos kL_2 + c\sinh kL_2 + d\cosh kL_2\right) \\
&= 0,
\end{aligned} \tag{2.35}$$

$$\begin{aligned}
&\rho_e A_e \left(-a\cos kL_2 - b\sin kL_2 + c\cosh kL_2 + d\sinh kL_2\right) + \\
&r_c k \rho_e A_e \left(a\sin kL_2 - b\cos kL_2 + c\sinh kL_2 + d\cosh kL_2\right) - \\
&k^3\left(J_c + J_b\right)\left(-a\sin kL_2 + b\cos kL_2 + c\sinh kL_2 + d\cosh kL_2\right) \\
&= 0,
\end{aligned} \tag{2.36}$$

Equation (2.34) to Eq. (2.36) can be rewritten as a matrix form like

$$\begin{bmatrix} \Delta_1 & \Delta_2 \\ \Delta_3 & \Delta_4 \end{bmatrix}\begin{bmatrix} a \\ b \end{bmatrix} = 0, \tag{2.37}$$

then the frequency mode equation can be derived as

$$\begin{aligned}
&\left[\left(J_c + J_b\right)\left(m_c + m_b\right)k^4 - \left(\rho_e A_e\right)^2\right]\cos kL_2 \cosh kL_2 + \\
&\rho_e A_e k\left[k^2\left(J_c + J_b + m_c r_c^2 + m_b r_c^2\right) - m_c - m_b\right]\cos kL_2 \sinh kL_2 + \\
&\rho_e A_e k\left[k^2\left(J_c + J_b + m_c r_c^2 + m_b r_c^2\right) - m_c - m_b\right]\sin kL_2 \cosh kL_2 + \\
&\left(J_c + J_b\right)\left(m_c + m_b\right)k^4 - \left(\rho_e A_e\right)^2 \\
&= 0.
\end{aligned} \tag{2.38}$$

There are infinite numbers of solutions k_i for the above equation.

2.3.3 Orthogonality of Eigenfunctions

In this section, the orthogonal condition between the same and different mode shapes is derived. Equation (2.32) can be rewritten as

$$\phi_i''''(x) - k_i^4 \phi_i(x) = 0. \tag{2.39}$$

For the above equation, ϕ_j is multiplied on both sides of the equation, do the integration from 0 to L_2, and apply the partial integration and boundary conditions. Then the equation can be modified as

$$\begin{aligned}
&\rho_e A_e \int_0^{L_2} \phi_i'' \phi_j'' \, dx = \\
&k_i^4 \Big\{ \int_0^{L_2} \rho_e A_e \phi_i \phi_j \, dx + \left(m_c + m_b\right) \phi_i(L_2) \phi_j(L_2) + \\
&\left(J_c + J_b + m_c r_c^2 + m_b r_c^2\right) \phi_i'(L_2) \phi_j'(L_2) + \\
&r_c \left(m_c + m_b\right) \left[\phi_i'(L_2) \phi_j(L_2) + \phi_i(L_2) \phi_j'(L_2) \right] \Big\}.
\end{aligned} \tag{2.40}$$

Exchanging i and j in the above equation and subtracting each other, a new equation can be derived as follows:

$$\begin{aligned}
&\left(k_i^4 - k_j^4\right) \Big\{ \int_0^{L_2} \rho_e A_e \phi_i \phi_j \, dx + \left(m_c + m_b\right) \phi_i(L_2) \phi_j(L_2) + \\
&\left(J_c + J_b + m_c r_c^2 + m_b r_c^2\right) \phi_i'(L_2) \phi_j'(L_2) + \\
&r_c \left(m_c + m_b\right) \left[\phi_i'(L_2) \phi_j(L_2) + \phi_i(L_2) \phi_j'(L_2) \right] \Big\} \\
&= 0.
\end{aligned} \tag{2.41}$$

In order to fulfill the above equation, the formula in the curly braces should be zero, if k_i does not equal to k_j. The equation which is derived here is an orthogonal equation which can be expressed as

$$\begin{aligned}&\int_0^{L_2} \rho_e A_e \phi_i \phi_j dx + \left(m_c + m_b\right)\phi_i(L_2)\phi_j(L_2) + \\ &\left(J_c + J_b + m_c r_c^2 + m_b r_c^2\right)\phi_i'(L_2)\phi_j'(L_2) + \\ &r_c\left(m_c + m_b\right)\left[\phi_i'(L_2)\phi_j(L_2) + \phi_i(L_2)\phi_j'(L_2)\right] \\ &= 0 \qquad\qquad\qquad\qquad \left(i \neq j\right)\end{aligned} \tag{2.42}$$

and then

$$\rho_e A_e \int_0^{L_2} \phi_i'' \phi_j'' dx = 0 \qquad \left(i \neq j\right). \tag{2.43}$$

2.3.4 Eigenfunction

The solution form of the eigenfunction ϕ_i can be expressed as

$$\phi_i\left(x\right) = \alpha_i\left(\cos k_i x + \beta_i \sin k_i x - \cosh k_i x - \beta_i \sinh k_i x\right) \tag{2.44}$$

and

$$\beta_i = \frac{A_i}{B_i}, \tag{2.45}$$

$$\begin{aligned}A_i = {}& \rho_e A_e\left(\sin k_i L_2 - \sinh k_i L_2\right) + \\ & k_i\left(m_c + m_b\right)\left[\left(\cos k_i L_2 - \cosh k_i L_2\right) - \right. \\ & \left. r_c k_i\left(\sin k_i L_2 + \sinh k_i L_2\right)\right]\end{aligned} \tag{2.46}$$

$$
\begin{aligned}
B_i = {} & \rho_e A_e \left(\cos k_i L_2 + \cosh k_i L_2\right) - \\
& k_i \left(m_c + m_b\right)\left[\left(\sin k_i L_2 - \sinh k_i L_2\right) + \right. \\
& \left. r_c k_i \left(\cos k_i L_2 - \cosh k_i L_2\right)\right]
\end{aligned}
\tag{2.47}
$$

$$
\alpha_i = \frac{1}{\mathrm{Max}\left(\cos k_i x + \beta_i \sin k_i x - \cosh k_i x - \beta_i \sinh k_i x\right)} \tag{2.48}
$$

For the eigenfunction of model 2 in Figure 2-3, it just needs to set m_b and J_b equal to zero and do the calculation again.

2.4 State Equation

2.4.1 State Equation for Nonlinear Model

In order to get the state equation of the model, here the nonlinear equation to form the state equation is derived first, and then obtain the linear equation to form the state equation for the model. The displacement of the vibration v can be expressed by the eigenfunction $\phi_i(x)$ obtained from the ith order vibration mode of a cantilevered beam and a time function $q_i(t)$ as Eq. (2.26). Then by substituting the Eq. (2.26) into Eq. (2.15) ~ Eq. (2.17) and fulfilling the orthogonal equation and the boundary conditions mentioned in the former part, the equations of motion can be rewritten as

For the equations of motion for the first rigid link:

$$A_1\ddot{\theta}_1 + B_1\ddot{\theta}_2 + \sum_{i=1}^{\infty} C_{1i}\ddot{q}_i + h_1 = \tau_1 \tag{2.49}$$

here

$$\begin{aligned} A_1 = {} & J_1 + J_2 + J_c + J_b + m_{L_1}\left(0.5L_1 + r_1\right)^2 + m_2 L_r^2 + \\ & \left(m_c + m_b\right)\left(2L_f L_r \cos\theta_2 - 2L_r\upsilon_e \sin\theta_2 + L_{rf}^2 + \right. \\ & \left. r_2^2 + L_r^2 + 2L_{rf}r_2 + v_e^2\right) + \rho_e A_e \int_0^{L_2}\left[2L_r\left(x + L_r\right)\cos\theta_2 - \right. \\ & \left. 2L_r\upsilon\sin\theta_2 + L_r^2 + \left(r + x\right)^2 + v^2\right]dx \end{aligned} \tag{2.50}$$

$$
\begin{aligned}
B_1 = {} & J_2 + J_c + J_b + \left(m_c + m_b\right) \\
& \left[L_f L_r \cos\theta_2 - L_r v_e \sin\theta_2 + \left(L_{rf} + r\right)^2 + v_e^2 \right] + \\
& \rho_e A_e \int_0^{L_2} \left[L_r \left(x + r_2\right)\cos\theta_2 - L_r \upsilon \sin\theta_2 + \left(r_2 + x\right)^2 + v^2 \right] dx
\end{aligned} \tag{2.51}
$$

$$
\begin{aligned}
C_{1i} = {} & \left(m_c + m_b\right)\left(L_r \cos\theta_2 + L_f\right)\left(\phi_i\left(L_2\right) + r_c \phi_i'\left(L_2\right)\right) + \\
& \rho_e A_e \int_0^{L_2} \left(L_r \cos\theta_2 + x + r_2\right)\phi_i\left(x\right) dx + \\
& \left(J_c + J_b\right)\phi_i'\left(L_2\right)
\end{aligned} \tag{2.52}
$$

$$
\begin{aligned}
h_1 = {} & \left(m_c + m_b\right)\left[2L_f \dot{\theta}_1\dot{\theta}_2 + L_f \dot{\theta}_2^2 + 2\left(\dot{\theta}_1 + \dot{\theta}_2\right)\dot{v}_e \right] L_r \sin\theta_2 - \\
& \left(m_c + m_b\right)\left(2\dot{\theta}_1\dot{\theta}_2 v_e + \dot{\theta}_2^2 v_e\right) L_r \cos\theta_2 + 2\left(m_c + m_b\right) \\
& \left(\dot{\theta}_1 + \dot{\theta}_2\right) v_e \dot{v}_e + \rho_e A_e \int_0^{L_2} \left\{ \left[2\left(x + r_2\right)\dot{\theta}_1\dot{\theta}_2 - \left(x + r_2\right)\dot{\theta}_2^2 + \right.\right. \\
& \left.2\left(\dot{\theta}_1 + \dot{\theta}_2\right)\dot{v} \right] L_r \sin\theta_2 - \left(2\dot{\theta}_1\dot{\theta}_2 v + \dot{\theta}_2^2 v\right) L_r \cos\theta_2 + \\
& \left. 2\left(\dot{\theta}_1 + \dot{\theta}_2\right) v\dot{v} \right\} dx
\end{aligned} \tag{2.53}
$$

For the equations of motion for the second flexible link:

$$
A_2\ddot{\theta}_1 + B_2\ddot{\theta}_2 + \sum_{i=1}^{\infty} C_{2i}\ddot{q}_i + h_2 = \tau_2 \tag{2.54}
$$

here

$$
\begin{aligned}
A_2 = {} & J_2 + J_c + J_b + \left(m_c + m_b\right) \\
& \left[L_f L_r \cos\theta_2 - L_r v_e \sin\theta_2 + \left(L_{rf} + r_2\right)^2 + v_e^2 \right] + \\
& \rho_e A_e \int_0^{L_2} \left[L_r \left(x + r_2\right)\cos\theta_2 - L_r v \sin\theta_2 + v^2 \right] dx
\end{aligned} \tag{2.55}
$$

$$
\begin{aligned}
B_2 = {} & J_2 + J_c + J_b + \left(m_c + m_b\right)\left[\left(L_{rf} + r_2\right)^2 + v_e^2 \right] + \\
& \rho_e A_e \int_0^{L_2} \left[\left(r_2 + x\right)^2 + v^2 \right] dx
\end{aligned} \tag{2.56}
$$

$$
\begin{aligned}
C_{2i} = {} & L_f\left(m_c + m_b\right)\left(\phi_i\left(L_2\right) + r_c\phi_i'\left(L_2\right)\right) + \\
& \rho_e A_e \int_0^{L_2}\left(x + r_2\right)\phi_i\left(x\right)dx + \\
& \left(J_c + J_b\right)\phi_i'\left(L_2\right)
\end{aligned}
\tag{2.57}
$$

$$
\begin{aligned}
h_2 = {} & L_r\dot{\theta}_1^2\left(m_c + m_b\right)\left(L_f \sin\theta_2 + v_e\cos\theta_2\right) + \\
& 2\left(m_c + m_b\right)\left(\dot{\theta}_1 + \dot{\theta}_2\right)v_e\dot{v}_e + \\
& \rho_e A_e \int_0^{L_2} L_r\dot{\theta}_1^2\left[\left(x + r_2\right)\sin\theta_2 + v\cos\theta_2\right] + \\
& 2\left(\dot{\theta}_1 + \dot{\theta}_2\right)v\dot{v}\}\, dx
\end{aligned}
\tag{2.58}
$$

For Eq. (2.17), after substituting the expression for eigenfunction into it, ϕ_j is multiplied on both sides of the equation, do the integration from 0 to L_2, and then apply the orthogonal condition obtained from the former section. Finally Eq. (2.17) can be rewritten as

$$
T_{i1}\ddot{\theta}_1 + T_{i2}\ddot{\theta}_2 + COD_i\ddot{q}_i + \sum_{j=1}^{\infty} R_{ij}\dot{q}_i + \omega_i^2 COD_i\dot{q}_i + h_{3i} = 0,
\tag{2.59}
$$

here

$$
\begin{aligned}
T_{i1} = {} & L_r\cos\theta_2\left(m_c + m_b\right)\left(\phi_i\left(L_2\right) + r_c\phi_i'\left(L_2\right)\right) + \\
& \rho_e A_e \int_0^{L_2}\left(L_r\cos\theta_2 + x + r_2\right)\phi_i\left(x\right)dx + \\
& \left(J_c + J_b\right)\phi_i'\left(L_2\right) + L_f\left(m_c + m_b\right)\left(\phi_i\left(L_2\right) + r_c\phi_i'\left(L_2\right)\right)
\end{aligned}
\tag{2.60}
$$

$$
\begin{aligned}
T_{i2} = {} & L_f\left(m_c + m_b\right)\left(\phi_i\left(L_2\right) + r_c\phi_i'\left(L_2\right)\right) + \\
& \rho_e A_e \int_0^{L_2}\left(x + r_2\right)\phi_i\left(x\right)dx + \\
& \left(J_c + J_b\right)\phi_i'\left(L_2\right)
\end{aligned}
\tag{2.61}
$$

$$COD_i = \int_0^{L_2} \rho_e A_e (\phi_i)^2 \, dx + (m_c + m_b)[\phi_i(L_2)]^2 + (J_c + J_b + m_c r_c^2 + m_b r_c^2)[\phi_i'(L_2)]^2 + 2r_c(m_c + m_b)\phi_i(L_2)\phi_j'(L_2) \tag{2.62}$$

$$R_{ij} = C\rho_e A_e \int_0^{L_2} \omega_j^2 \phi_i(x)\phi_j(x)\, dx \tag{2.63}$$

$$h_{3i} = L_r \dot{\theta}_1^2 \sin\theta_2 (m_c + m_b)\big(\phi_i(L_2) + r_c\phi_i'(L_2)\big) - v_e\left(\dot{\theta}_1 + \dot{\theta}_1\right)^2 (m_c + m_b)\big(\phi_i(L_2) + r_c\phi_i'(L_2)\big) + \rho_e A_e \int_0^{L_2} \left[L_r \dot{\theta}_1^2 \sin\theta_2 - v\left(\dot{\theta}_1 + \dot{\theta}_1\right)^2 \right] \phi_i(x)\, dx \tag{2.64}$$

where

$$L_r = L_1 + r_1 + r_2$$
$$L_{rf} = L_2 + r_c$$
$$L_f = L_2 + r_2 + r_c$$
$$v_e = v(L_2) + r_c v'(L_2).$$

For the torques used to actuate each link, the torque τ_1 and τ_2 can be expressed as the following equations

$$\tau_1(t) = \tau_{\mathrm{DC}}(t) - J_{\mathrm{DC}}\ddot{\theta}_1 - C_{\mathrm{DC}}\dot{\theta}_1 \tag{2.65}$$

$$\tau_2(t) = \tau_{\mathrm{Servo}}(t) - J_{\mathrm{Servo}}\ddot{\theta}_2 - C_{\mathrm{Servo}}\dot{\theta}_2 \tag{2.66}$$

here

τ_{DC} and τ_{Servo} : motor torques (N•m)

J_{DC} and J_{Serovo} : the inertia moment of the motor shaft (kg•m^2)

C_{DC} and C_{Serovo} : the viscosity coefficient of the motor shaft (N•m•s)

Equation (2.49), (2.54), and (2.59) can be transformed into the matrix form. So the equations of motion can be expressed as

$$\mathbf{M}_{\mathrm{nl}}\ddot{\mathbf{z}}+\mathbf{C}_{\mathrm{nl}}\dot{\mathbf{z}}+\mathbf{K}_{\mathrm{nl}}\mathbf{z}+\mathbf{H}_{\mathrm{nl}}=\mathbf{D}_{\mathrm{nl}}\boldsymbol{\tau} \tag{2.67}$$

where

$$\mathbf{z}=\left\{\theta_1,\theta_2,q_1,q_2,\cdots q_n\right\}^T,$$

$$\boldsymbol{\tau}(t)=\left[\tau_{\mathrm{DC}}(t)\quad \tau_{\mathrm{Servo}}(t)\right]^T.$$

The matrix $\mathbf{M}_{\mathrm{nl}}$, $\mathbf{C}_{\mathrm{nl}}$, $\mathbf{K}_{\mathrm{nl}}$, $\mathbf{H}_{\mathrm{nl}}$, and $\mathbf{D}_{\mathrm{nl}}$ can be expressed as

$$\mathbf{M}_{\mathrm{nl}}=\begin{bmatrix} A_1+J_{\mathrm{DC}} & B_1 & C_{11} & \cdots & \cdots & C_{1n} \\ A_2 & B_2+J_{\mathrm{Servo}} & C_{21} & \cdots & \cdots & C_{2n} \\ T_{11} & T_{12} & COD_1 & 0 & \cdots & 0 \\ T_{21} & T_{22} & 0 & COD_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ T_{n1} & T_{n2} & 0 & 0 & \cdots & COD_n \end{bmatrix} \tag{2.68}$$

$$\mathbf{C}_{\mathrm{nl}}=\begin{bmatrix} C_{\mathrm{DC}} & 0 & 0 & 0 & \cdots & 0 \\ 0 & C_{\mathrm{Servo}} & 0 & 0 & \cdots & 0 \\ 0 & 0 & R_{11} & 0 & \cdots & 0 \\ 0 & 0 & 0 & R_{22} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & R_{nn} \end{bmatrix} \tag{2.69}$$

$$
\mathbf{K}_{\mathrm{nl}} = \begin{bmatrix} 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & \omega_1^2 COD_1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \omega_2^2 COD_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \omega_n^2 COD_n \end{bmatrix} \tag{2.70}
$$

$$
\mathbf{H}_{\mathrm{nl}} = \begin{bmatrix} h_1 \\ h_2 \\ h_{31} \\ h_{32} \\ \vdots \\ h_{3n} \end{bmatrix} \tag{2.71}
$$

$$
\mathbf{D}_{\mathrm{nl}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{bmatrix} \tag{2.72}
$$

Then the Eq. (2.67) can be rewritten as a state equation

$$
\dot{\mathbf{x}} = \mathbf{A}_{\mathrm{nl}}\mathbf{x} + \mathbf{B}_{\mathrm{nl}}\mathbf{u} + \mathbf{W}_{\mathrm{nl}}, \tag{2.73}
$$

where

$$
\mathbf{x} = \left\{ \mathbf{z}^T, \dot{\mathbf{z}}^T \right\},
$$

$$
\mathbf{A}_{\mathrm{nl}} = \begin{bmatrix} \mathbf{0}_{(n+2)\times(n+2)} & \mathbf{I}_{(n+2)\times(n+2)} \\ -\mathbf{M}_{\mathrm{nl}}^{-1}\mathbf{K}_{\mathrm{nl}} & -\mathbf{M}_{\mathrm{nl}}^{-1}\mathbf{C}_{\mathrm{nl}} \end{bmatrix},
$$

$$\mathbf{B}_{\mathrm{nl}} = \begin{bmatrix} \mathbf{0}_{(n+2)\times 2} \\ \mathbf{M}_{\mathrm{nl}}{}^{-1}\mathbf{D}_{\mathrm{nl}} \end{bmatrix},$$

$$\mathbf{W}_{\mathrm{nl}} = \begin{bmatrix} \mathbf{0}_{(n+2)\times 2} \\ -\mathbf{M}_{\mathrm{nl}}{}^{-1}\mathbf{H}_{\mathrm{nl}} \end{bmatrix}.$$

For the system model shown in Figure 2-3, m_b and J_b are assumed to be zero.

2.4.2 State Equation for Linear Model

In this section, the nonlinear equation of motion is modified into the linear one, and then form the state equation for the linear model. By neglecting the nonlinear terms (which are shown by the underline) in Eq. (2.15) and Eq. (2.16) and modify them with the boundary condition, the equations of motion for the linear model can be obtained.

For the equations of motion of the linear model for the first rigid link:

$$V_1\ddot{\theta}_1 + V_2\ddot{\theta}_2 + \sum_{i=1}^{\infty} Q_{1i}\ddot{q}_i = \tau_1 \tag{2.74}$$

here

$$\begin{aligned} V_1 = {} & J_1 + J_2 + J_c + J_b + m_{L_1}\left(0.5L_1 + r_1\right)^2 + m_2 L_r^2 + \\ & \left(m_c + m_b\right)\left(2L_f L_r + L_{rf}^2 + r_2^2 + L_r^2 + 2L_{rf} r_2\right) + \\ & \rho_e A_e \int_0^{L_2} \left[2L_r\left(x + L_r\right) + L_r^2 + \left(r + x\right)^2 \right] dx \end{aligned} \tag{2.75}$$

$$\begin{aligned} V_2 = {} & J_2 + J_c + J_b + \left(m_c + m_b\right) \\ & \left[L_f L_r + \left(L_{rf} + r\right)^2 \right] + \\ & \rho_e A_e \int_0^{L_2} \left[L_r\left(x + r_2\right) + \left(r_2 + x\right)^2 \right] dx \end{aligned} \tag{2.76}$$

$$\begin{aligned} Q_{1i} = {} & (m_c + m_b)(L_r + L_f)(\phi_i(L_2) + r_c\phi_i'(L_2)) + \\ & \rho_e A_e \int_0^{L_2} (L_r + x + r_2)\phi_i(x)\,dx + \\ & (J_c + J_b)\phi_i'(L_2) \end{aligned} \tag{2.77}$$

For the equations of motion for the linear model for the second flexible link:

$$V_3\ddot{\theta}_1 + V_4\ddot{\theta}_2 + \sum_{i=1}^{\infty} Q_{2i}\ddot{q}_i = \tau_2 \tag{2.78}$$

here

$$\begin{aligned} V_3 = {} & J_2 + J_c + J_b + (m_c + m_b) \\ & \left[L_f L_r + (L_{rf} + r_2)^2\right] + \\ & \rho_e A_e \int_0^{L_2} L_r (x + r_2)\ dx \end{aligned} \tag{2.79}$$

$$\begin{aligned} V_4 = {} & J_2 + J_c + J_b + (m_c + m_b)(L_{rf} + r_2)^2 + \\ & \rho_e A_e \int_0^{L_2} (r_2 + x)^2 dx \end{aligned} \tag{2.80}$$

$$\begin{aligned} Q_{2i} = {} & L_f (m_c + m_b)(\phi_i(L_2) + r_c\phi_i'(L_2)) + \\ & \rho_e A_e \int_0^{L_2} (x + r_2)\phi_i(x)\,dx + \\ & (J_c + J_b)\phi_i'(L_2) \end{aligned} \tag{2.81}$$

By substituting the expression of vibration displacement into Eq. (2.17) and fulfilling the orthogonal equation, the boundary conditions mentioned in the former part, and linearize the equation, the equation of motion for governing the vibration can be rewritten as

$$U_{i1}\ddot{\theta}_1 + U_{i2}\ddot{\theta}_2 + LOD_i\ddot{q}_i + \sum_{j=1}^{\infty} RL_{ij}\dot{q}_i + \omega_i^2 LOD_i\dot{q}_i = 0 \tag{2.82}$$

here

$$
\begin{aligned}
U_{i1} = {} & L_r \left(m_c + m_b\right)\left(\phi_i\left(L_2\right) + r_c\phi_i'\left(L_2\right)\right) + \\
& \rho_e A_e \int_0^{L_2} \left(L_r + x + r_2\right)\phi_i\left(x\right)dx + \\
& \left(J_c + J_b\right)\phi_i'\left(L_2\right) + L_f\left(m_c + m_b\right)\left(\phi_i\left(L_2\right) + r_c\phi_i'\left(L_2\right)\right)
\end{aligned} \tag{2.83}
$$

$$
\begin{aligned}
U_{i2} = {} & L_f \left(m_c + m_b\right)\left(\phi_i\left(L_2\right) + r_c\phi_i'\left(L_2\right)\right) + \\
& \rho_e A_e \int_0^{L_2} \left(x + r_2\right)\phi_i\left(x\right)dx + \\
& \left(J_c + J_b\right)\phi_i'\left(L_2\right)
\end{aligned} \tag{2.84}
$$

$$
\begin{aligned}
LOD_i = {} & \int_0^{L_2} \rho_e A_e \left(\phi_i\right)^2 dx + \left(m_c + m_b\right)\left[\phi_i(L_2)\right]^2 + \\
& \left(J_c + J_b + m_c r_c^2 + m_b r_c^2\right)\left[\phi_i'(L_2)\right]^2 + \\
& 2r_c\left(m_c + m_b\right)\phi_i(L_2)\phi_j'(L_2)
\end{aligned} \tag{2.85}
$$

$$
RL_{ij} = C\rho_e A_e \int_0^{L_2} \omega_j^2 \phi_i\left(x\right)\phi_j\left(x\right)dx \tag{2.86}
$$

Also the same torques conditions expressed as Eq. (2.65) and Eq. (2.66) used to actuate each link, by considering expression for torque τ_1 and τ_2, the equations of motion for the linear model can be written as

$$
\mathbf{M}_{\mathrm{L}}\ddot{\mathbf{z}} + \mathbf{C}_{\mathrm{L}}\dot{\mathbf{z}} + \mathbf{K}_{\mathrm{L}}\mathbf{z} = \mathbf{D}_{\mathrm{L}}\boldsymbol{\tau} \tag{2.87}
$$

where

$$
\mathbf{z} = \left\{\theta_1, \theta_2, q_1, q_2, \cdots q_n\right\}^T,
$$

$$
\boldsymbol{\tau}\left(t\right) = \left[\tau_{\mathrm{DC}}\left(t\right) \quad \tau_{\mathrm{Servo}}\left(t\right)\right]^T.
$$

The matrix $\mathbf{M}_{\mathrm{L}}$, $\mathbf{C}_{\mathrm{L}}$, $\mathbf{K}_{\mathrm{L}}$, and $\mathbf{D}_{\mathrm{L}}$ can be expressed as

$$\mathbf{M}_{\mathrm{L}} = \begin{bmatrix} V_1 + J_{\mathrm{DC}} & V_2 & Q_{11} & \cdots & \cdots & Q_{1n} \\ V_3 & V_4 + J_{\mathrm{Servo}} & Q_{21} & \cdots & \cdots & Q_{2n} \\ U_{11} & U_{12} & LOD_1 & 0 & \cdots & 0 \\ U_{21} & U_{22} & 0 & LOD_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ U_{n1} & U_{n2} & 0 & 0 & \cdots & LOD_n \end{bmatrix} \tag{2.88}$$

$$\mathbf{C}_{\mathrm{L}} = \begin{bmatrix} C_{\mathrm{DC}} & 0 & 0 & 0 & \cdots & 0 \\ 0 & C_{\mathrm{Servo}} & 0 & 0 & \cdots & 0 \\ 0 & 0 & RL_{11} & 0 & \cdots & 0 \\ 0 & 0 & 0 & RL_{22} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & RL_{nn} \end{bmatrix} \tag{2.89}$$

$$\mathbf{K}_{\mathrm{L}} = \begin{bmatrix} 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & \omega_1^2 LOD_1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \omega_2^2 LOD_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \omega_n^2 LOD_n \end{bmatrix} \tag{2.90}$$

$$\mathbf{D}_{\mathrm{L}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{bmatrix} \tag{2.91}$$

Then the Eq. (2.87) can be rewritten as a state equation

$$\dot{\mathbf{x}} = \mathbf{A}_{\mathrm{L}}\mathbf{x} + \mathbf{B}_{\mathrm{L}}\mathbf{u}, \tag{2.92}$$

where

$$\mathbf{x} = \left\{ \mathbf{z}^T, \dot{\mathbf{z}}^T \right\},$$

$$\mathbf{A}_{\mathrm{L}} = \begin{bmatrix} \mathbf{0}_{(n+2)\times(n+2)} & \mathbf{I}_{(n+2)\times(n+2)} \\ -\mathbf{M}_{\mathrm{L}}{}^{-1}\mathbf{K}_{\mathrm{L}} & -\mathbf{M}_{\mathrm{L}}{}^{-1}\mathbf{C}_{\mathrm{L}} \end{bmatrix},$$

$$\mathbf{B}_{\mathrm{L}} = \begin{bmatrix} \mathbf{0}_{(n+2)\times 2} \\ \mathbf{M}_{\mathrm{L}}{}^{-1}\mathbf{D}_{\mathrm{L}} \end{bmatrix},$$

For the system model shown in Figure 2-3, m_b and J_b are assumed to be zero.

2.5 Summary

In this chapter, in order to make a ball throwing manipulator which can throw a ball in a certain speed, firstly a two-link flexible manipulator is designed. For this manipulator, the first link is a rigid link and the second link is a flexible one. On the extreme end of the flexible link, it has a ball holder which is used to catch and release the ball. After calculating the kinetic and conservation energy of each part of the manipulator, the Hamilton's principle is introduced to derive the equations of motion for the manipulator.

For the flexible manipulator, the displacement of the vibration is the important but the complex part for calculation. So in this section, a method called mode model constraint method is applied to get the mode shape of the flexible manipulator. By equations of mode shape, the vibration displacement can be calculated by the expression which contains the eigenfunction and time function. Also the orthogonal condition between the same and different mode shapes is obtained.

For simulating the manipulator through the computer, the state equation for the manipulator model is necessary. Here the state equation for the nonlinear model and linear model are derived.

In the next chapter, the control torques for realizing the throwing motion need to be designed. The control torques should make the manipulator throwing a ball in a certain speed and at the same time, using the control energy more efficiently.

Chapter 3 Control Torque and Structure Design

3.1 Introduction

In the second chapter, the manipulator is built up for throwing a ball and then derive the equations of motion and state equations for this ball throwing manipulator. In this chapter, the control torque which can realize the throwing motion is needed to be designed, and then in order to pitch a ball in a certain high speed with the lower energy consumption, to optimize the control torque which is designed here is also necessary.

Commonly speaking, optimization is to detect the optimal solution for a problem among all candidates. In optimization, the constraints set according to the problem or user's demand are commonly needed to limit the range of the candidate solutions. If a candidate solution meets all requirements of the constraints, it is a feasible solution. And the global optimization is the optimal problem to search the optimal solution among all these feasible ones. However, in many cases, the local optimization is acceptable for hard or unnecessary to detect the global optimal solution if the sub-optimal one owns the acceptable quality. If the actual problem is taken into a mathematical model with corresponding constraints and the candidate solutions are translated into numerical vector or variables, a proper function can be built for the optimal program searching which candidate solution makes the function reaching the minimum value, This built function is named as objective function and it is the key function of the global optimization problem. Also if to invert the sign of the objective function, the optimal minimum problem can be translated into the optimal maximum one equivalently. For the optimization of the actual problem, it faces some problems such as local minimum solution problem, constraint problem for the complex optimization and so on [44]. In mathematical aspect, the optimization problem can be expressed as

$$\min_{x \in A} f(x), \qquad C_i(x) \le 0, \qquad i = 1, 2, \ldots, k. \tag{3.1}$$

here A belong to a n-dimensional Euclidean space. $C_i(x)$ stands for the constraints for the problem and k is the constraint numbers (the constraints here is not restrictive and they can be expressed by other different forms). For the constraint problem, a point which belongs to the A space is named as feasible point if it can fulfill all constraint requirement conditions. The set of all feasible points above forms the feasible set. If all constraint conditions in Eq. (3.1) are removed, the problem becomes an unconstrained optimization problem and all points in the A space form the feasible set.

For the optimization problems, according to different types of objective function or constraint conditions, they can be distinguished to the following categories. The simple one is the linear optimization problem which objective function and constraints are linear ones. For the optimization problem, if it contains at least one nonlinear objective function, the optimization becomes the nonlinear optimization. The third one is the convex optimization. The objective functions and feasible sets for this optimization are the convex ones. The quadratic optimization owns the quadratic objective functions for minimization and linear constraints. The last one is the stochastic optimization. Based on statistical distributions, it introduces the noise for function evaluations or random selection for variables. And there are many studies on the above optimization problems [45-51].

Commonly, the objective function of the optimization problem is a time-unvarying single function. If the objective function changes into a time-varying one, the optimization problem becomes the dynamic optimization. For this optimization, when the global minimizer moves in the search space, it can find the position of the minimizer faster than the common static optimization. Also it can increase the level of robustness for the solutions. On other aspect, if the problems own not only a single objective function but two or multi objective functions, the optimization problems change into multi-objective optimization. In this condition, the minimizer needs to minimize all objective functions together [52-55].

If the categories of the problem from the aspect of the search space and variables of the problem are considered, the optimization problems can be divided into three categories, discrete optimization, continuous optimization, and mixed integer optimization. The variables of the objective function in the discrete optimization are the discrete values. Comparing with the discrete optimization, the values of the objective function for continuous optimization are real values. For the final one, mixed integer optimization, real and integer values are used for the objective functions [56-57].

The above optimizations need strict assumptions on mathematical aspect, but in the real applications, these assumptions are hard to be satisfied. For example, in the optimization, the objective function is required to keep convexity or differentiability. But in the real case, the objective function is obtained through the measurements of the observation data or some complex procedures, and it is difficult to check or maintain the requirements of the objective function above. So in this condition, the efficient algorithm is created for solving these problems.

The algorithm can be divided into two categories roughly. One is the deterministic algorithm and the other is the stochastic algorithm [58-59]. The methods of deterministic algorithm include grid search, covering methods, and trajectory-based methods [49]. For grid search, it doesn't search for the optimization information of the previous steps. It evaluates solution qualities which lay on a grid in the search space. So the grid density for this method is a very important factor for the final result of the algorithm. Covering method focuses on searching and excluding the pasts which don't contain the global minimizer in the search space. In the trajectory-based methods, the points which traverse trajectories are searched and these trajectories intersect the global minimizer's vicinity. The methods for stochastic algorithm contains random search, clustering, and the method based on probabilistic models of the objective function [46]. For random search method, in order to generate search points, it introduces the probability distribution. For the pure random search, the new sample of points is produced in each algorithm's iteration. If it is hoped to increase the accuracy of the algorithm, the size of sample will become so large or even approach infinity. So in real-world application, in order to obtain the acceptable results, the huge number of evaluation functions is inefficient. The pure random search shows a bad performance in real application. By introducing a local search procedure, the performance of the random search can be improved. Clustering method try to initiate a single local search for each local minimizer detected. This makes it possible to produce the samples of points and cluster them to find the neighborhoods of local minimizer. And for each cluster, a single local search can be proceeded.

Indeed in modern research field, some algorithm such as evolutionary computation and swarm intelligence appear in the center of researchers' attention.

The evolutionary computation describes the methods which intersect between global optimization and computational intelligence. The evolutionary algorithm uses some strategy such as stochasticity, adaptation, and learning to make the schemes for intelligent optimization. The basic theories or the structures of evolutionary algorithms are formed according to the Darwinian theory of Species. This algorithm assumes that the populations of the potential solutions evolve iteratively and the procedure simulates the Darwinian Theory that human and animal biologically evolve through natural selection. The first application of evolutionary algorithm based on Darwinian Theory was reported in the middle of 50^{th} century in the field of machine learning [60]. In modern research, there are three major categories on evolutionary computation.

First one is the evolutionary programming. This evolutionary method was first developed by L. J. Fogel in the middle of 60^{th} century [61]. His main purpose was to make an intelligence program. For this program, it can predict its environment and make a proper response based on its predictions. The program observes the passing symbols and then generates the output which can predict the next environment symbol accurately. The environment is explored through finite-state machines. So the each population machine can output the predicted symbol according to the input symbol. The value of a pay-off function which is constructed to evaluate the performance of each machine based on the quality of prediction is viewed as this machine's fitness value. Among these fitness values, the highest ones will be selected to form the population for the next iteration or generation of the algorithm. And the evolution procedure won't be stopped until the accuracy predictions of all symbols in one machine are generated. Some problems such as scheduling, traveling salesman problem, neural networks training, and continuous optimization [62-65] was tried to be solved through the evolutionary programming after 80th century.

The second one is the genetic algorithm which belongs to the category of evolutionary computation. This method was first developed by J. H. Holland in the middle of 60th century. This algorithm is one of the popular algorithms for optimization [66-70]. It searches the candidate solutions like other evolutionary algorithm but the obvious distinguish part of genetic algorithm is that it represents the solutions in a binary form. This algorithm needs a translation function to change the actual variables into the binary vectors and vice versa. So the binary form of a solution is named as chromosome or genotype, and the actual form of it is called phenotype. Three fundamental operators exist in genetic algorithm: selection, crossover, and mutation. A rank scheme is used to select the best individuals of the population. For the procedure, the parents are picked out from the current population and then combined to generate offspring. In mutation stage, the operator will change one or more digit numbers of the chromosome of the offspring and the mutated ones will be evaluated. Finally, the population of the next generation will be selected among all parents and offspring and the algorithm will be terminated when the criteria condition is fulfilled.

The final type for evolutionary computation is called as evolution strategies. They were developed by Rechenberg, Schwefel, and Bienert in the 60th century [71-73]. Comparing with the above two evolutionary algorithm, the main difference is that this method is lack of the crossover operator and only mutation and selection operators are used. In the early research of this method, for mutating the structure shapes, the discrete probability distributions were applied. And the single search point generated one descendant for per iteration instead of the population. Then in the later period of this method research, the continuous probability distributions could be used in the implementations such as tackling numerical optimization problems. The distribution parameters play an important role in this algorithm. They introduce the severe influence to descendant qualities which are generated by the mutation operator. So the self-adaptation strategy for these parameters is always used in this algorithm. But in order to introduce the flexibility to this algorithm, the independent adaptation for each solution is much more considered in recent research [74-76]. The above characteristics make evolution strategies a proper algorithm for solving the numerical problems than other evolutionary algorithms.

Comparing with the above evolutionary computation algorithm, swarm intelligence is the other big branch of artificial intelligence. It mainly studies the properties and behavior of the complex and self-organized social problems. Swarm intelligence developed from the nature problems such as bird flocks, animal herds, and fish schools. These nature systems have the amazing capabilities for self-organization and the collective behaviors can't be simply analyzed by just aggregating each member. For the global optimization, swarm intelligence was first developed in 1989 for robotic swarms' control [77]. Then after that, there are three main algorithms developed under this branch and they are ant colony optimization, stochastic diffusion search, and particle swarm optimization.

The concept of ant colony optimization appeared in 1992 by introducing the ants' behavior of searching the food for an optimization algorithm [78] and after that this kind of optimization was named as ant colony optimization [79-80]. For natural ant colonies, when they begin to search food, they start from their nest randomly. They bring the food back to the nest until the food is found. And the pheromones lay on the ground during their paths. More pheromones lay on the shortest path between the food and the nest than other paths. So a new ant starting from the nest can follow the shortest path to the food through the higher pheromone levels and also lay its pheromones on this path. In ant colony optimization, the similar procedure is operated. The artificial ants start from the initial search point and make the new potential solution one by one. For each solution, the alternative is needed to be decided and each alternative takes a pheromone level which can be viewed as the probability of selection. Based on the quality of the solution, the alternative pheromone level is updated at the end of its tour. So the solutions with low function values can be assigned with higher pheromone levels.

Stochastic diffusion search was first developed by Bishop in 1989 [81]. Similarly to the ant colony optimization, the stochastic search is based on the populations of communicating individuals. But for individuals, the direct one-to-one communication is used for information exchange. After evaluating the solutions partially, the information is passed to other individuals by following this one-to-one scheme. The partial evaluation scheme is very helpful for the objective function which is decomposable and expensive for computation. And this algorithm owns the capabilities for synchronous and asynchronous parallelization optimization. This algorithm is widely applied on many research fields such as text and object recognition, face recognition, robotics, and wireless networks [82-85].

The last method in the branch of swarm intelligence is the youngest algorithm, particle swarm optimization. This algorithm was first developed by Kennedy and Eberhart in 1995 [86]. The particle swarm concept originated as a simulation of a simplified social system. The best position searched by each individual is named as experience, and it's diffused among the whole or part of the population. The searching direction is guided to the most promising regions of the search space. The original purpose of algorithm was to simulate the graceful but unpredictable choreography of a bird flock, fish schools, and animal herds. The concepts and rules for particle swarm optimization developed from this natural phenomenon. But unlike the ant colony optimization, the communication among individuals is exchanged directly without altering the environment in the system.

In this section, in order to fulfill a proper throwing motion which means to realize a certain throwing speed with better energy efficiency for actuating, to obtain the optimal control torque profile through the optimization algorithm is needed. So the particle swarm optimization is applied for the optimization of the manipulator control torque that is designed. At the beginning, the basic particle swarm optimization is introduced.

3.2 The Basic Particle Swarm Optimization

As which is mentioned in the former section, Particle Swarm Optimization (PSO) algorithm was generated from simulation of behavior for bird flocks or animal herds. In the early period, some researchers simulate the swarming behavior by following the main rules like nearest neighbor velocity matching and acceleration by distance. Eberhart and Kennedy analyzed the simulation model and refined it. And in 1995, they developed the first PSO [86].

If this section defines the search space $A \subset \mathbf{R}^n$, and objective function $f : A \to Y \subseteq \mathbf{R}$. In the introduction part, it is known that PSO is the algorithm based on population. So here the population is named as swarm and the individuals for it are called as the particles. So the swarm can be defined as

$$S = \{x_1, x_2, \ldots, x_N\}, \tag{3.2}$$

here the particles $x_1, x_2, \ldots, x_N$ can be defined as

$$x_i = (x_{i1}, x_{i2}, \ldots, x_{in})^{\mathrm{T}} \in A, \quad i = 1, 2, \ldots, N. \tag{3.3}$$

the indices of x are assigned arbitrarily for the particles. N is the algorithm parameter decided by the user according to detail of the problem. $f(x)$ is the objective function and can be accessed by all solutions in A search space. For each particle, there is a unique objective function value to it.

With the iteration updating, the particles can move within the search space. So a proper position shift named as velocity is introduced for the algorithm. It can be used to control and adjust the position of the algorithm. Here it can be defined as

$$v_i = (v_{i1}, v_{i2}, \ldots, v_{in})^{\mathrm{T}}, \quad i = 1, 2, \ldots, N. \tag{3.4}$$

Here with the iteration increasing, the velocity should adapt the particles in the search space to visit any region in the search space. If k is used as the counter of iteration or generation, then the current position and velocity of the ith particle can be expressed as x_i^k and v_i^k respectively.

The information gathered in the previous iteration of the algorithm is used to update the velocity. So while the particle is searching the search space, it should store the information of the best position. For the algorithm, a memory set is needed for storing the current positions of the particles and it can be expressed as

$$\mathbf{P} = (p_1, p_2, \ldots, p_N), \tag{3.5}$$

here this memory set contains the best positions as

$$p_i = (p_{i1}, p_{i2}, \ldots, p_{in})^{\mathrm{T}} \in A, \quad i = 1, 2, \ldots, N, \tag{3.6}$$

when visited by each particle, the positions can be expressed as

$$p_i^k = \arg\min_i f_i^k, \tag{3.7}$$

here k is the iteration or generation counter.

PSO algorithm is operated on the social behavior model. So the particles in the algorithm need to exchange the information or communicate their experience. The information exchange mechanism should be introduced for the algorithm. The algorithm should direct the global minimizer to the best position which is ever visited by all particles. It is assumed that g to be the index of the best position which can approximate the lowest value of objective function in **P** in an iteration k, and then this condition can be expressed as

$$p_g^k = \arg\min_i f\left(p_i^k\right). \tag{3.8}$$

So here the basic PSO can be defined as

$$v_{ij}^{k+1} = v_{ij}^k + c_1 r_1\left(p_{ij}^k - x_{ij}^k\right) + c_2 r_2\left(g_j^k - x_{ij}^k\right), \tag{3.9}$$

$$x_{ij}^{k+1} = x_{ij}^k + v_{ij}^k, \tag{3.10}$$

$$i = 1, 2, \ldots N, \quad j = 1, 2, \ldots n,$$

here k is the iteration or generation counter; c_1 and c_2 are the cognitive and the social parameters respectively; r_1 and r_2 are psedudo-random values in the [0,1] range.

For each iteration or generation, after updating and evaluating the particles, the best positions of the algorithm are also needed to be updated. So the best position for the new iteration can be defined as

$$p_i^{k+1} = \begin{cases} x_i^{k+1}, & \text{if } f\left(x_i^{k+1}\right) \le f\left(x_i^k\right) \\ p_i^k, & \text{if } f\left(x_i^{k+1}\right) > f\left(x_i^k\right) \end{cases} \tag{3.11}$$

and the iteration or generation of PSO algorithm is completed by updating the best position index g according to the above equation.

For the velocity updated by Eq. (3.9), some uncontrolled huge increase will make the swarm divergence. It is called swarm explosion effect. This is resulted from being lack of the constrict mechanism to restrict the velocity in the early PSO variants. So the strict bound conditions need to be set for the velocities in order to prevent the particles move extremely too far from their current positions just in one step of iteration updating.

The maximum velocity in the algorithm of course should be positive value. When the new velocity of each particle has been determined through Eq. (3.9), before updating the position for algorithm according to Eq. (3.10), the velocity threshold should be checked first by the following condition

$$\left|v_{ij}^{k+1}\right| \le v_{\max}, \quad i = 1, 2, \ldots N, \quad j = 1, 2, \ldots n. \tag{3.12}$$

For the violation condition, the velocity can be set to the closet velocity bound directly as

$$v_i^{k+1} = \begin{cases} v_{\max}, & if\ v_{ij}^{k+1} > v_{\max} \\ -v_{\max}, & if\ v_{ij}^{k+1} < -v_{\max} \end{cases} \tag{3.13}$$

Usually the $v_{\max}$ is viewed as a fraction of the search space size per direction. So a search space is defined as

$$A = [a_1, b_1] \times [a_2, b_2] \times \cdots [a_n, b_n], \quad a_{i,} b_i \in \mathbf{R}, \quad i = 1, 2, \ldots, n. \tag{3.14}$$

The maximum velocity for all direction can expressed as

$$v_{\max} = \frac{\min_i \{b_i - a_i\}}{d}. \tag{3.15}$$

here d is a constant value needed to be tuned by user according to the problems. Alternatively, the separate maximum velocity per component can be expressed as

$$v_{\max,i} = \frac{b_i - a_i}{d}, \quad i = 1, 2, \ldots, n. \tag{3.16}$$

The performance of the basic PSO algorithm can be improved through setting the maximum velocity threshold, but for a complex optimization problem, it may not be enough for making the algorithm efficient. Except the swarm explosion which is discussed above, on the final stage of the optimization, the swarm is commonly not easy to gather its particles around the best solutions. So although the best region in the search space can be detected roughly, the particles just hang on the wide search space around the best positions without some good strategies.

Around the best positions, to introduce some attraction for the particles towards them is necessary. And also small shift of position for escaping from the close vicinity should be considered. So here a new parameter, w, named as inertia weight, is applied for adjusting the effect of the previous velocity on the current velocity [87]. The effect for the previous velocity should fade for each particle. Then Eq. (3.9) can be modified by introducing this inertia weight as

$$v_{ij}^{k+1} = w \cdot v_{ij}^{k} + c_1 r_1 \left(p_{ij}^{k} - x_{ij}^{k} \right) + c_2 r_2 \left(g_j^{k} - x_{ij}^{k} \right), \tag{3.17}$$

$$i = 1, 2, \ldots N, \quad j = 1, 2, \ldots n,$$

Next the inertia weight factor should be selected properly. It should keep the effect of the previous velocity fading during the operation of the algorithm. In general, a linearly decreasing scheme for the inertial weight, w, can be mathematically expressed as follows:

$$w^k = \frac{w_{\mathrm{up}} - k \cdot \left(w_{\mathrm{up}} - w_{\mathrm{low}} \right)}{T_{\max}}. \tag{3.18}$$

here k is the iteration or generation counter; w_{up} and w_{low} are the designed upper and lower bounds of w; $T_{\max}$ is the total numbers of operation iterations. Equation (3.18) can produce the decreasing time dependent inertia weigh linearly. The value starts from w_{up} at the beginning and at the last iteration, the value becomes to $T_{\max}$.

For the basic PSO algorithm, the implementation procedure can be expressed roughly as Figure 3-1. Each particle keeps track of its coordinates in the search space and the program searches this problem space with the best solution which has been achieved so far. This value is called “pbest”. Another “best” value that is tracked by the global version of the particle swarm optimizer is the overall best value, and its location, obtained so far by a particle in the population. This location is called “gbest”.

The basic procedure for the global version of basic PSO can be expressed as follows:

Step one: Initialize the population of particles with random positions and velocities in the search space of the problem.

Step two: For each particle, evaluate the desired optimal cost function.

Step three: Compare each particle’s cost evaluation with its pbest. If current value is better than pbest, set the pbest value equal to the current value and the pbest location equal to the current location in the search space.

Step four: Compare cost evaluation with the population’s overall previous best. If the current value is better than gbest, reset gbest to the current particle’s array index and

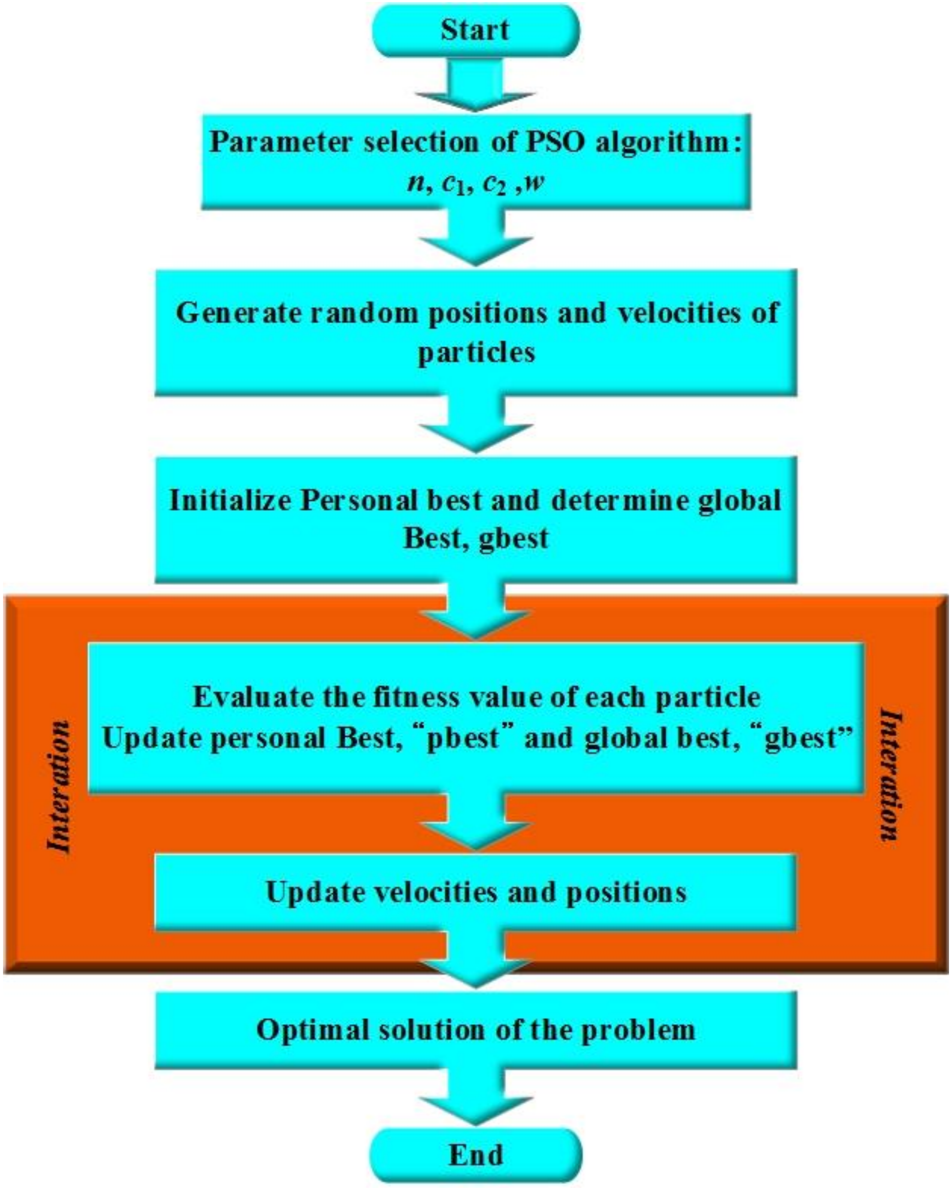


Fig. 3-1 Flowchart of the basic PSO

value.

Step five: Update the velocity and position of the particle.

Step six: Loop to Step two until a criterion is met.

The basic PSO has been used for solving some problems such as the optimization of the feedback controller for the flexible manipulator [88] and seismic control optimization of the building [89].

3.3 Control Torque Design of the Manipulator

In this section, a sinusoidal-form of input shown as in Figure 3-2 is utilized for each control torque of the link in Eq. (2.87) to obtain the optimized input.

some other kinds of input profile to the manipulator can also be applied, however in order to keep the manipulator stable or not to make the control output changing sharply on some time points, the sinusoidal-form profile was selected for this research. The amplitudes and time periods of the sinusoidal-waves for confirming the profiles of the torques can be controlled. So the amplitudes and time periods are the parameters needed to be optimized through optimization algorithm.

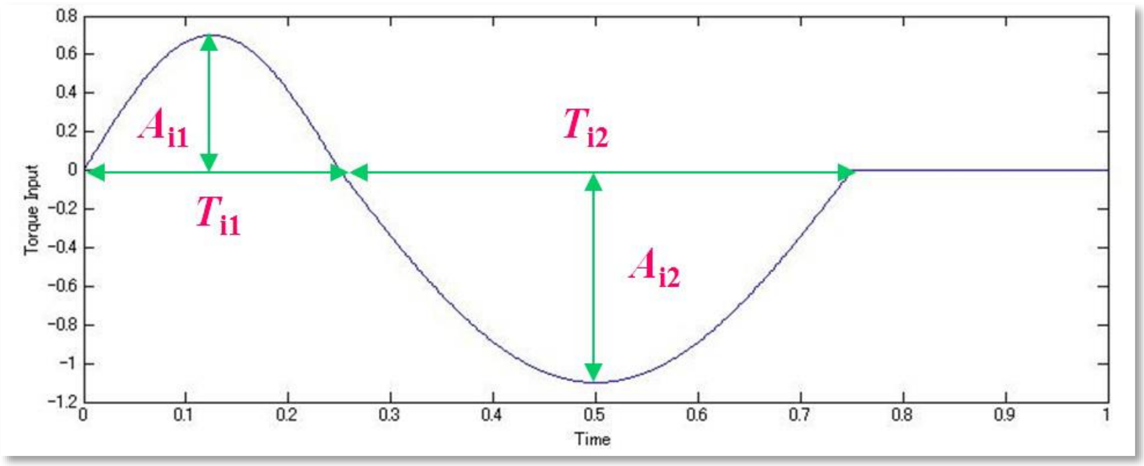


Fig. 3-2 Sinusoidal profile for the control torques

Here if the mathematical form of this sinusoidal control torque is considered, the sinusoidal-form input can be expressed as

$$\tau_i(t)=\begin{cases} A_{i1}\sin\left(\dfrac{2\pi}{T_{i1}}t\right) & 0\le t\le \dfrac{T_{i1}}{2} \\ -A_{i2}\sin\left(\dfrac{2\pi}{T_{i2}}t-\dfrac{T_{i1}}{2}\right) & \dfrac{T_{i1}}{2}\le t\le \dfrac{T_{i2}}{2} \\ 0 & t>\dfrac{T_{i2}}{2} \end{cases} \tag{3.19}$$

where i=1,2 and the parameters A_{i1}, A_{i2}, T_{i1}, T_{i2} are optimized by the basic PSO algorithm which is mentioned in the former section. Here subscript i stands for number of the actuator torque. For example, when i=1, it means the torque of the first link. In this section, the manipulator is actuated through two control torques and for each torque there are four parameters which are needed to be tuned. So the parameters that are fine evaluated here can be viewed as a particle evolution in 8-dimensional search space with respect to the fitness functions. Then the search space can be expressed as

$$\mathbf{X}_{\text{b-PSO}}=\begin{bmatrix} A_{11}^1 & T_{11}^1 & A_{12}^1 & T_{12}^1 & A_{21}^1 & T_{21}^1 & A_{22}^1 & T_{22}^1 \\ A_{11}^2 & T_{11}^2 & A_{12}^2 & T_{12}^2 & A_{21}^2 & T_{21}^2 & A_{22}^2 & T_{22}^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ A_{11}^n & T_{11}^n & A_{12}^n & T_{12}^n & A_{21}^n & T_{21}^n & A_{22}^n & T_{22}^n \end{bmatrix} \tag{3.20}$$

where n stands for the number of particle.

Our control torque and set up the search space have been determined for it, and then try to search the best control torque by utilizing the basic PSO algorithm in this search space.

3.4 Optimization result through basic PSO

In this section, to acquire an optimal global solution, a proper fitness function is needed. The choice of the objective function is very important for achieving the desired system function, because different fitness functions which generate fitness value providing a performance measure of the problem promote different PSO behaviors. For the optimization problem in this subsection, the following fitness function is designed to optimize the control torques of the manipulator

$$J_{\text{b-PSO}} = w_1 \frac{1}{Vel_{\max}} + w_2 \int_0^{\infty} \left[\tau_1(t)\dot{\theta}_1(t) + \tau_2(t)\dot{\theta}_2(t) \right] dt + w_3 t_{\text{oper}}, \qquad (3.21)$$

where $Vel_{\max}$ is the maximum speed of the ball. $\tau_1(t)$ and $\tau_2(t)$ are the torque inputs, $\dot{\theta}_1(t)$ and $\dot{\theta}_2(t)$ denote the angular velocities of the manipulator, t_{oper} is the length of operation time of the control input, and w_1, w_2, w_3 stand for the weight coefficients. These weight coefficients need to be tuned by researcher according to the conditions or the requirements of the problem. Then the problem can be formulated as the following constrained optimization problem to minimize the fitness function J under the conditions,

$$\begin{aligned} & A_{\min} \le A_{i1}, A_{i2} \le A_{\max}, \\ & T_{\min} \le T_{i1}, T_{i2} \le T_{\max}, \\ & v \le v_{\max}. \end{aligned} \qquad (3.22)$$

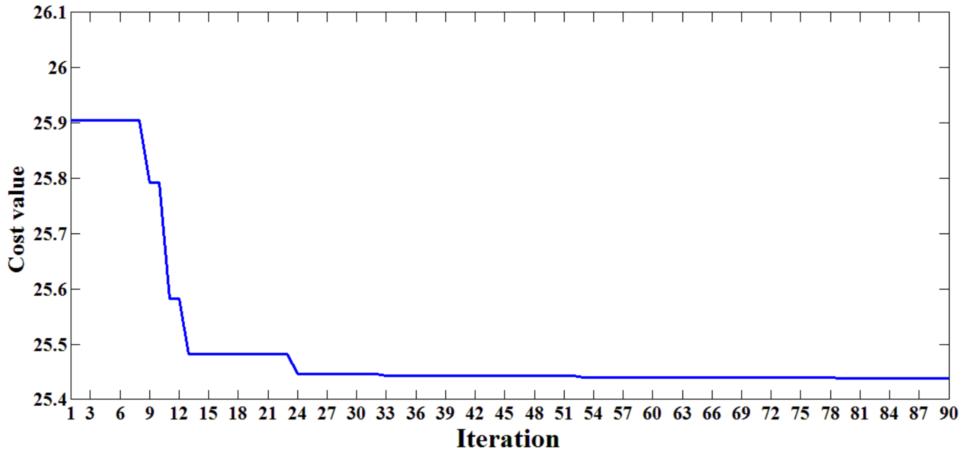


Fig. 3-3 Cost function through basic PSO

where A, T, and v are amplitude, time period of the input torque, and displacement of the flexible manipulator respectively.

Figure 3-3 shows the cost value through basic PSO algorithm. Here the weight coefficient w_1, w_2, w_3 are set to 2, 0.1, 0.1, respectively. After 90th iteration, the value of the cost function does not vary and becomes stable. So at this generation, it can be viewed that an optimal solution for our problem has been obtained. The optimal solution is [0.58, 0.72, 1.09, 0.43, 0.13, 0.54, 0.15, 1.00]. So for this manipulator, the optimal torques can be obtained through setting the aptitudes and time periods according to the above solutions of basic PSO algorithm.

3.5 Chaos Embedded Vector Evaluated Particle Swarm Optimization Algorithm

The basic PSO algorithm can operate the problems which subject to one objective function. And for the problems with one objective function, commonly the basic PSO algorithm can work well as the results which are obtained in the above section. Indeed the cost function in Eq. (3.21) contains three objective functions. In order to apply the basic PSO algorithm to our current problem, one of the most common methodologies is to aggregate the objective functions with appropriate weight factors into a single comprehensive objective function such as setting the weight coefficients w_1, w_2, w_3. But this requires a well knowledge of the problem domain to assign these weight factors appropriately. Even for the well knowledge researchers, the weight factors must be tuned for a long time to obtain the appropriate results. So the nature and appropriate way for this problem tends to handle the multiple objectives separately with multi-objective PSO.

3.5.1 Multi-objective PSO

Multi-objective Optimization named as MO is the optimization when different objective functions have to be optimized simultaneously in one problem [90]. For this kind of problem, one condition is that two or more than two competing objective functions need to be optimized together. So comparing with the basic PSO, the concept

Pareto Optimality is used to substitutes the optimality one of single objective optimization. Among Pareto solutions, two different solutions may get the same value of objective function with different inherent properties. So for a desirable MO problem, it is wished to find the suitable solutions that have different inherent properties as much as possible.

Similar to basic PSO, it can assumed that an *n*-dimensional search space $A \subset \mathbf{R}^n$, and $f_i(x), 1, 2, \ldots, k$ which are *k* objective function over the above search space. Then a vector function can be defined and expressed as

$$\mathbf{f}(x) = \left(f_1(x), f_2(x), \ldots, f_n(x)\right)^{\mathbf{T}}, \tag{3.23}$$

with *m* unequal constraint conditions

$$g_i(x) \le 0, \quad i = 1, 2, \ldots, m. \tag{3.24}$$

where $g_i(x)$ is the constraint function for expressing the boundary condition of optimal problem. The objective functions in Eq. (3.23) may conflict with each other. So it's impossible to search a single global solution for minimizing all objective functions. In this condition, the optimal solution in this kind of MO problems needs to be redefined.

It is assumed that $\mathbf{u} = (u_1, u_2, \ldots u_n)^{\mathrm{T}}$ and $\mathbf{v} = (v_1, v_2, \ldots v_n)^{\mathrm{T}}$ are two *k*-dimensional vectors. Then the vector **u** can be defined to dominate vector **v**, if and only if it fulfills the following conditions as

$$u_i \le v_i, \quad \text{all } i = 1, 2, \ldots k,$$

and at least one component *i* in above condition fulfils

$u_i < v_i$,

This property can be defined as Pareto Dominance.

For a solution, $x \in A$, if and only if there is no other solution, $y \in A$, can fulfill that $\mathbf{f}(y)$ dominates $\mathbf{f}(x)$. Or it can be defined that x is nondominated with respect to A. The set of all Pareto optimal solutions can be named as Pareto Optimal Set or denoted as P^* here.

So the set of all vector function values of all Pareto optimal solutions can be expressed as

$$F^* = \{\mathbf{f}(x) : x \in P^*\}. \tag{3.25}$$

which is named as Pareto Front.

A Pareto Front is convex if and only if, for all $\mathbf{u}, \mathbf{v} \in F^*$, and all $\lambda \in (0,1)$, a vector $\mathbf{z} \in F^*$ can be found, to fulfil the condition

$$\lambda \|\mathbf{u}\| + (1-\lambda)\|\mathbf{v}\| \geq \|\mathbf{z}\|, \tag{3.26}$$

when it's called concave, if and only if

$$\lambda \|\mathbf{u}\| + (1-\lambda)\|\mathbf{v}\| \leq \|\mathbf{z}\|. \tag{3.27}$$

However a Pareto Front can be partially convex or concave. And the multi-objective optimization problem with this kind of cases is much more difficult.

Based on the concept of Pareto Front above, each particle of the swarm could have different leaders, but only one may be selected to update the velocity. This set of leaders is stored in a repository, which contains the best non-dominated solutions found. At each generation, the velocity of the particles is also updated similar like the basic PSO

For the common MOPSO, it needs an external archive to store the non-dominated particles. And moreover the selecting rules and size of this archive are hard to be selected and decided. So in this paper, the VEPSO is used for the multi-objective optimization and crowding distance for ruling the external archive defined as *A*. If $p \in A$ with respect to *i*th cost function cf_i. Then for each cost function, the crowding distance of member *p* is as

$$cd_p(cf_i) = cf_i(q) - cf_i(r),\ i = 1, 2, \ldots, n, \tag{3.28}$$

where *q* is the member of external archive following after *p* in a sorting order which is determined through cf_i values, and *r* is the member which precedes *p* in the same order. And the total crowding distance of *p* can be calculated as

$$cd_{total}(p) = \sum_{i=1}^{n} cd_p(cf_i). \tag{3.29}$$

3.5.2 Vector Evaluated PSO (VEPSO)

Vector Evaluated Particle Swarm Optimization (VEPSO) [91-93] was introduced by Parsopoulos and Vrahatis as a multi-swarm PSO variant for multi-objective optimization (MO) problems, and it was extended to parallel implementation. The main goal in MO problems is the detection of Pareto optimal points.

In VEPSO, it assumes that n swarms, S_1, S_2, …, S_n of size L, point to optimize n functions simultaneously. And each swarm is evaluated according to one of the objective functions. Swarms exchange information among them by sharing their individual findings to direct search towards the Pareto optimal set.

The VEPSO permits the parameter configuration of each swarm independently. So, the number of particles as well as the values of the PSO parameters per swarm may differ. A notable characteristic is the insertion of the best position of another swarm in S_n. This information exchange scheme among swarms has a prominent position in VEPSO. It can be clearly viewed as a migration scheme, where particles migrate from one swarm to another according to a connecting topology.

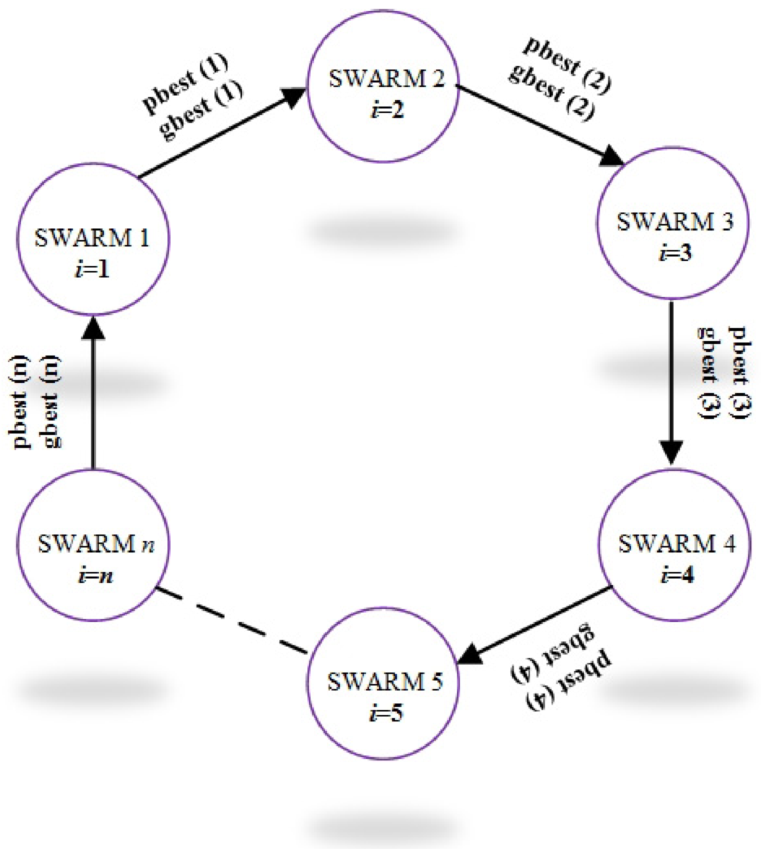


Fig. 3-4 Ring migration topology for n swarms

Figure 3-4 shows the ring migration topology corresponding to the following equation:

$$s=\begin{cases} n, & i=1, \\ i-1, & i=2,\cdots,n. \end{cases} \tag{3.30}$$

where s shows the index of the neighbor swarm. It means that current swarm obtains the information of the best solution from its former neighbor swarm. So the velocity updating formula of basic PSO for each particle

$$v_{ij}^{k+1}=w\cdot v_{ij}^{k}+c_1 r_1\left(p_{ij}^{k}-x_{ij}^{k}\right)+c_2 r_2\left(g_{j}^{k}-x_{ij}^{k}\right), \tag{3.31}$$

needs to be modified to the following equation:

$$v_{ij}^{k+1}=w\cdot v_{ij}^{k}+c_1 r_1\left(p_{sj}^{k}-x_{ij}^{k}\right)+c_2 r_2\left(g_{sj}^{k}-x_{ij}^{k}\right), \tag{3.32}$$

where x_{ij}^{k} and v_{ij}^{k} are the position and velocity of the jth dimension of the ith particle in the kth iteration, respectively; c_1 and c_2 are the cognitive and the social parameters respectively; r_1 and r_2 are psedudo-random values in the [0,1] range; p_{sj}^{k} and g_{sj}^{k} don't only stand for the individual best position and global best position of one swarm but also the transitive position information shared by multi-swarms in the algorithm. They obey the ring migration rule in Figure 3-4.

3.5.3 Chaos Embedded VEPSO (CEVEPSO)

For common PSO algorithm, it can be easily trapped in the local minima position. Moreover, it is difficult to get an acceptable solution without the use of a proper searching strategy. Chaos is a general nonlinear phenomenon in nature, and its behavior is nearly stochastic. For chaos exquisite internal structure, it owns characteristics of randomness, ergodicity, and regularity [94]. The chaos optimization algorithm owns strong ability of jumping out of the local extrema. In addition, PSO with chaos strategy can overcome the shortcoming of common PSO algorithm in which it easily arrives at the local extrema by maintaining the diversity of the swarm.

In this subsection, the chaotic system is used to generate sequences for substituting random numbers (r_1, r_2) of VEPSO given by Eq. (3.32) where it is necessary to make a random based choice. Mathematically, chaos is randomness of a simple deterministic dynamical system. For example, if two dice are thrown and counted the total point, a sum of 7 will randomly occur twice as often as 4, but the outcome of any particular roll of the dice is unpredictable. And chaotic system may be considered as source of randomness. A chaotic map is a discrete time of dynamical system which can be expressed as

$$z_{k+1} = f_c\left(z_k\right) \quad 0 < z_k < 1, \quad k = 0,1,2,\cdots. \tag{3.33}$$

here the tent mapping strategy is used in the CEVEPSO algorithm which can be expressed as

$$z_{k+1} = \mu\left(1-2\left|z_k - 0.5\right|\right), \quad 0 \le z_0 \le 1, \tag{3.34}$$

where $\mu \in [0,1]$ is the fork control parameter. When $\mu = 1$, the tent mapping is in a complete chaos state and moving through the whole range.

Then the velocity updating formula given by Eq. (3.32) is also modified correspondingly [95]. The equation is expressed as follows:

$$v_{ij}^{k+1} = CM_1^k v_{ij}^k + c_1 \cdot CM_{2j}^k \left(p_{sj}^k - x_{ij}^k\right) + c_2 \cdot CM_{3j}^k \left(g_{sj}^k - x_{ij}^k\right), \tag{3.35}$$

where

$$CM_1^{k+1} = \mu_1\left(1-2\left|CM_1^k - 0.5\right|\right), \quad 0 \le CM_1^0 \le 1,$$

$$CM_{2j}^{k+1} = \mu_2\left(1-2\left|CM_{2j}^k - 0.5\right|\right), \quad 0 \le CM_{2j}^0 \le 1,$$

$$CM_{3j}^{k+1} = \mu_3\left(1-2\left|CM_{3j}^k - 0.5\right|\right), \quad 0 \le CM_{3j}^0 \le 1.$$

CM_1^k, CM_{2j}^k, and CM_{3j}^k are functions based on the tent mapping with value between 0 and 1. Comparing with other optimal method, CEVEPSO can search the optimal solution more efficiently and simply. And it also can avoid some local trap through embedded chaos mapping strategy.

3.5.4 Optimization Result through CEVEPSO with the Conflict Objective Functions

The same sinusoidal form input is used for optimization in this section. And the fitness function in Eq. (3.21) for basic PSO can be rebuilt according to CEVEPSO as follows

$$J_1 = \frac{1}{Vel_{\max}}, \tag{3.36}$$

$$J_2 = \int_0^\infty \left[u_1(t)\dot{\theta}_1(t) + u_2(t)\dot{\theta}_2(t) \right] dt, \tag{3.37}$$

$$J_3 = t_{oper}. \tag{3.38}$$

So it is seen that there is no need to choose the weight factor for combining them to one subjective function.

After applying CEVEPSO algorithm, some solutions are detected through this optimal algorithm. These solutions are shown in Figure 3-5. The solutions with the rectangle spots were detected under the condition that the particle number and iteration were 25 and 140, respectively; the solutions with round spots were in the condition that particle number was 15 and the iteration was 160; the solutions with triangle spots were found in the setting of condition of particle 10 and iteration 120. Then by applying the interpolation method, the Pareto Fronts of this problem is tried to be estimated. The solid blue line is the interpolation line goes through the nondominated points we obtained in this problem.

Three cost functions to evaluate the particles are used in this part, but indeed these three objective functions were in conflict with each other. If the objective functions are in conflict with each other (for example, when the value of the first cost function decreases, the value of the second one will increase. And the same condition happens between the second and the third or the third and the first ones), then the task of the optimization algorithm becomes to search the Pareto Front which is the set of all non-dominated solutions of the problem [96] as which is mentioned in the former section. Even though CEVEPSO could approach the Pareto Front for solving the problem with three conflict objective functions in this section, it was not easy to search sufficient solutions for forming the Pareto Front to this problem.

So it can be found there are not enough solution values in Figure 3-5 to form an accurate Pareto Front. In the next part, the cost functions which are not in conflict with each other is tried to be converted.

Then it is considered that this problem not only on optimization of the control torques but also on the aspect of structure.

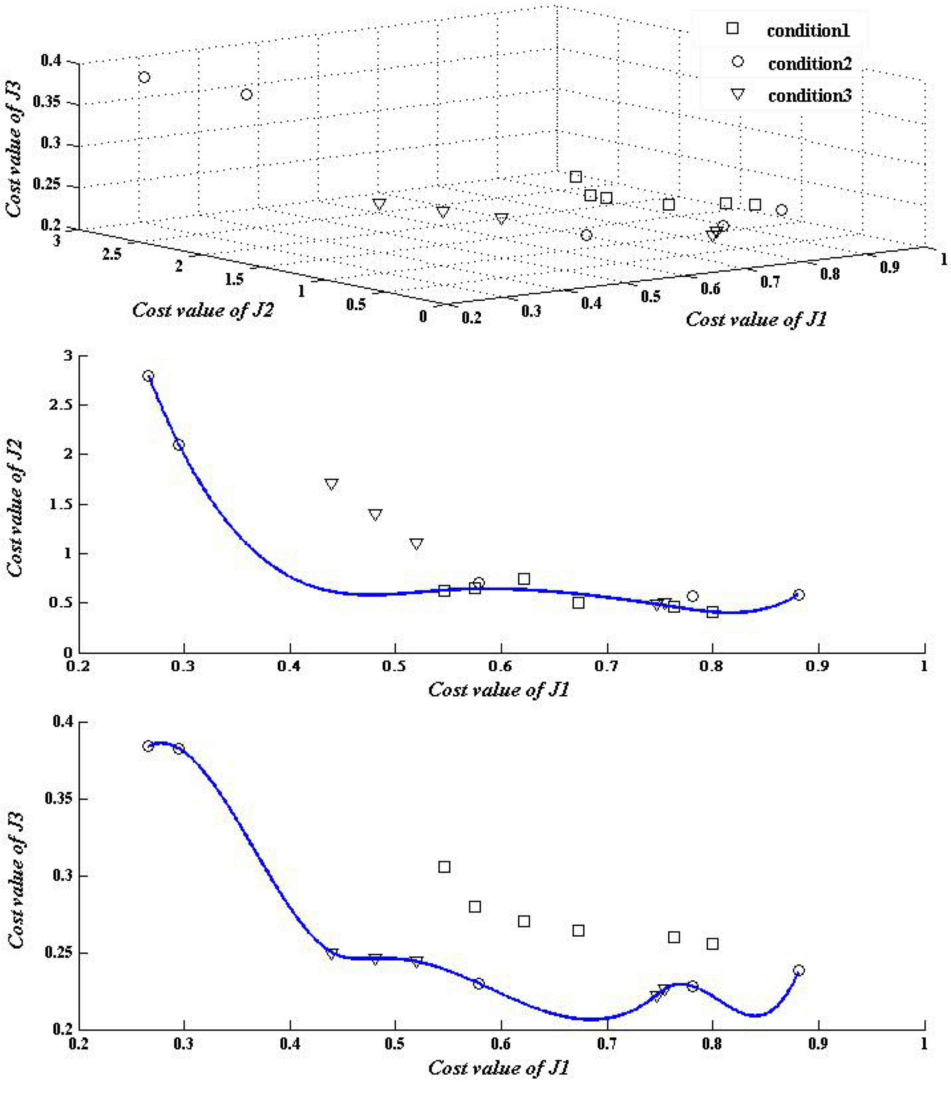


Fig. 3-5 Pareto Fronts valued by CEVEPSO with conflict objective functions

3.5.5 Beam Structure Design

The flexible link of the manipulator is designed as a kind of composite material which is the steel beam reinforced by the FRP material. In this section for beam structure design, two layers of FRP material on each side of the beam was attached as Figure 3-6. Here θ_1, θ_2, θ_3, and θ_4 are the angles between the fiber direction and *X* axis for each layer.

According to the rule of Cauchy stress change, the stress in the coordinate system O-*XY* can be derived in terms of the original stress in coordinate system O-x_iy_i and express as

$$\begin{bmatrix} \sigma_{X_i} \\ \sigma_{Y_i} \\ \tau_{X_iY_i} \end{bmatrix} = \mathbf{T}_i \begin{bmatrix} \sigma_{x_i} \\ \sigma_{y_i} \\ \tau_{x_iy_i} \end{bmatrix} \tag{3.39}$$

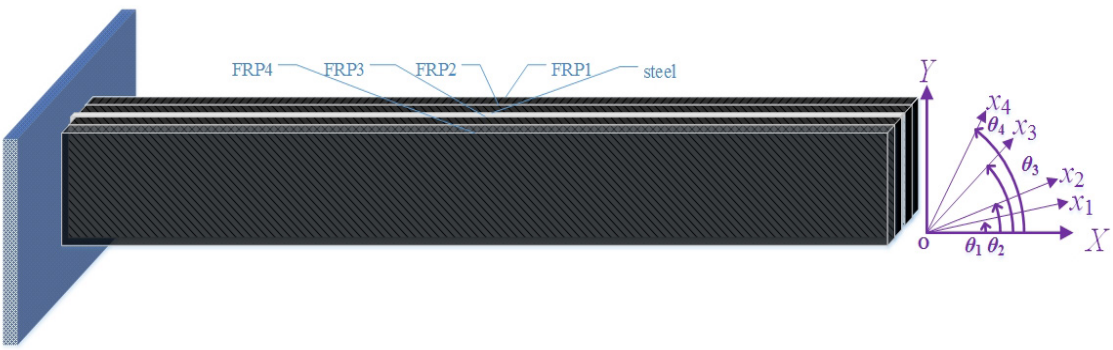


Fig. 3-6 FRP beam of the flexible link

where

$$\mathbf{T}_i = \begin{bmatrix} \cos^2\theta_i & \sin^2\theta_i & -2\sin\theta_i\cos\theta_i \\ \sin^2\theta_i & \cos^2\theta_i & 2\sin\theta_i\cos\theta_i \\ -\sin\theta_i\cos\theta_i & \sin\theta_i\cos\theta_i & \cos^2\theta_i - \sin^2\theta_i \end{bmatrix}$$

and $\mathbf{T}_i$ is called as the transformation matrix.

If the thickness of each layer is thin enough, then the laminate can be regarded as an anisotropic one. The relationship (assume that no slip occur between layers) between the stress and the strain in the *i*th layer can be expressed as

$$\begin{bmatrix} \sigma_{x_i} \\ \sigma_{y_i} \\ \tau_{x_i y_i} \end{bmatrix} = \mathbf{T}_i^{-1}\mathbf{Q}_i\mathbf{T}_i \begin{bmatrix} \varepsilon_{x_i} \\ \varepsilon_{y_i} \\ \gamma_{x_i y_i} \end{bmatrix} = \bar{\mathbf{Q}} \begin{bmatrix} \varepsilon_{x_i} \\ \varepsilon_{y_i} \\ \gamma_{x_i y_i} \end{bmatrix} \tag{3.40}$$

where $\mathbf{Q}$ is the stiffness matrix.

Consider the FRP sheet with thickness h_i of each layer as shown in Figure 3-6, the relationship between the stress and the strain of other layers can be obtained in similar manner and the resultant stress and stress of each layer can be shown as

$$\begin{aligned} \sigma_X \cdot h &= \sum_{i=1}^{M} \sigma_{X_i} \cdot h_i, \\ \sigma_Y \cdot h &= \sum_{i=1}^{M} \sigma_{Y_i} \cdot h_i, \\ \tau_{XY} \cdot h &= \sum_{i=1}^{M} \tau_{X_i Y_i} \cdot h_i. \end{aligned} \tag{3.41}$$

where σ_X, σ_Y, and τ_{XY} are the stresses of the whole laminate, σ_{X_i}, σ_{Y_i}, and $\tau_{X_i Y_i}$ are the stresses of the *i*th layer. The total thickness of the laminate is represented by h, and M is the total number of the layers.

Through the corresponding relationship between the stress of each single layer and internal force of the laminate cross section, the matrix describing the relationship between the stress and strain can be derived as

$$\begin{bmatrix} \sigma_X \\ \sigma_Y \\ \tau_{XY} \end{bmatrix} = \mathbf{D} \begin{bmatrix} \varepsilon_X \\ \varepsilon_Y \\ \gamma_{XY} \end{bmatrix}, \tag{3.42}$$

where

$$D_{kj} = \sum_{i=1}^{M} \bar{Q}_{kj}^{i} \cdot h_i / h. \quad (k, j = 1, 2, 3) \tag{3.43}$$

Here M is the total number of the layers. So the elastic modules along X axis direction can be obtained through

$$E_X = \sigma_X / \varepsilon_X . \tag{3.44}$$

In order to utilize the storage energy through the flexible beam deformation and make the releasing motion more efficiently, the fiber directions were changed to obtain different Young’s modulus which can realize different deformation on different bending direction. For example, before releasing the ball, the beam should be deformed larger to storage the energy for increasing the releasing speed of the ball. If the beam’s maximum deformation on the opposite direction of ball releasing is larger than that on the releasing direction, the beam can own more storage energy before releasing and utilize the energy more efficiently.

For two layers of FRP sheets on each side of the beam, the parameters of fiber directions can be viewed as a particle evolution in 4-dimensional search space with respected to its fitness function. Its search space can be expressed as

$$\mathbf{X}^{j}_{\text{CEVEPSO-2}} = \begin{bmatrix} \theta_1^1 & \theta_2^1 & \theta_3^1 & \theta_4^1 \\ \theta_1^2 & \theta_2^2 & \theta_3^2 & \theta_4^2 \\ \vdots & \vdots & \vdots & \vdots \\ \theta_1^j & \theta_2^j & \theta_3^j & \theta_4^j \end{bmatrix}, \quad j = 1, 2, \cdots, m. \tag{3.45}$$

where m stands for the number of particles.

The other search space is the same search space used in the former section which is used to optimize the problems with the conflict objective functions and this search space can be expressed as

$$\mathbf{X}^{j}_{\text{CEVEPSO-1}} = \begin{bmatrix} A_{11}^1 & T_{11}^1 & A_{12}^1 & T_{12}^1 & A_{21}^1 & T_{21}^1 & A_{22}^1 & T_{22}^1 \\ A_{11}^2 & T_{11}^2 & A_{12}^2 & T_{12}^2 & A_{21}^2 & T_{21}^2 & A_{22}^2 & T_{22}^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ A_{11}^j & T_{11}^j & A_{12}^j & T_{12}^j & A_{21}^j & T_{21}^j & A_{22}^j & T_{22}^j \end{bmatrix}, \tag{3.46}$$
$$j = 1, 2, \cdots, n.$$

where n stands for the number of particles.

3.5.6 Optimization Result through CEVEPSO with Non-confliction Objective Functions

For search space $\mathbf{X}^{j}_{\text{CEVEPSO-1}}$ shown as Eq. (3.46), the fitness function $J_{\text{CEVEPSO-1}}$ is designed as

$$J_{\text{CEVEPSO-1}} = \frac{1}{vel_{\max}}, \tag{3.47}$$

where $vel_{\max}$ is the maximum speed of the ball for releasing.

For search space $\mathbf{X}^{j}_{\text{CEVEPSO-2}}$ shown as Eq. (3.45), the fitness function $J_{\text{CEVEPSO-2}}$ is designed as

$$J_{\text{CEVEPSO-2}} = \frac{v_{\text{bend\_forward}}}{v_{\text{bend\_backward}}}. \tag{3.48}$$

where $v_{\text{bend\_forward}}$ represents the maximum deformation of the beam when it is bent toward the direction of throwing the ball, and $v_{\text{bend\_backward}}$ expresses the maximum deformation of the beam on the progress of bending on the opposite direction. The ratio of them can be used for evaluating the efficiency of the storage energy. Higher releasing speed requires more energy; however it is hoped to apply less control energy at the same time. In the former section, it can be viewed that the control torque is evaluated directly and form a competition relationship with $J_{\text{CEVEPSO-1}}$ in the algorithm. But if the energy utilization efficiency of the flexible beam could be increased, that would be also the optimal condition for the system without introducing the conflict cost functions for the optimal algorithm. That is the reason why the efficiency function $J_{\text{CEVEPSO-2}}$ is selected instead of the torque function in the algorithm. These two cost functions have the influence to each other. This means that for different control profile, it may lead different solution of efficiency function $J_{\text{CEVEPSO-2}}$. And vice versa, different efficiency ratio will guide the control profile $J_{\text{CEVEPSO-1}}$ to different result. So it is necessary to optimize them simultaneously.

Then the problem can be formulated as the following constrained optimization problem to minimize the fitness functions under the conditions,

$$A_{\min} \le A_i \le A_{\max},$$
$$T_{\min} \le T_i \le T_{\max}, \tag{3.49}$$
$$v \le v_{\max}.$$

where A_i and T_i are the amplitude and time period of the input torque respectively. v represents the bending deformation of the extreme end the flexible link.

After applying CEVEPSO algorithm, the cost value of each fitness function is shown as Figure 3-7.

The first plot shows optimal condition of one cost function $J_{\text{CEVEPSO-1}}$. The horizontal axis is the generation of the optimal algorithm with cost value $J_{\text{CEVEPSO-1}}$ of the vertical axis. It can be found the optimal algorithm works well and the fitness value decreased with the generation number increasing. Then after the 40$^{\text{th}}$ generation, the fitness value $J_{\text{CEVEPSO-1}}$ becomes stable. The second plot represents the variation tendency of the fitness value $J_{\text{CEVEPSO-2}}$ with the horizontal axis "generation" and vertical axis "fitness value J_2". It can be found the cost value of $J_{\text{CEVEPSO-2}}$ can be minimized with $J_{\text{CEVEPSO-1}}$ simultaneously, and this value becomes stable after 30$^{\text{th}}$ generation. For the first two plots, the cost values become stable for both fitness functions after 40$^{\text{th}}$ generation. So it can be found that a set of solutions for this problem after 40$^{\text{th}}$ generation. 50$^{\text{th}}$ generation is applied for following simulation. And the optimal result of $\mathbf{X}_{\text{CEVEPSO-1}}$ for this problem is [0.52, 0.53, 2.82, 1.09, 0.85, 1.08, 0.45, 0.84], then optimal solution for $\mathbf{X}_{\text{CEVEPSO-2}}$ is shown as this set [55.5°, 49.8°, 74.1°, 67.8°]. The third plot shows the variation condition of the fitness values $J_{\text{CEVEPSO-1}}$ and $J_{\text{CEVEPSO-2}}$ on a fitness value space. It is wished to minimize both $J_{\text{CEVEPSO-1}}$ and $J_{\text{CEVEPSO-2}}$ at the same time, so the spot on this space should approach the zero point as nearly as possible (but it can't go to the zero point). On the third plot, the value spot of the 1st generation locates on the extreme right side of the fitness value space. With the generation increasing, the value spot moves to the bottom left direction and try to approach the original point. Finally, the value spot stops on one location of the fitness value space and this location owns the most minimal distance from the zero point comparing with other locations which it passed. So it can be found that this location is the optimal position for our problem and the solution of this position is the optimal solution of this problem.

Comparing with the simple PSO algorithm the CEVEPSO algorithm with the conflict objective functions could operate the optimization without setting the weight factors which are used to form multiple cost functions into one function. However for conflict objective functions, the optimal result is not proper or sufficient for the problem in this research. So the objective functions are redesigned to the functions without conflict. It can be found that the performance of the CEVEPSO without conflict cost functions works well and finally the optimal solution for the problem in this research can be obtained properly.

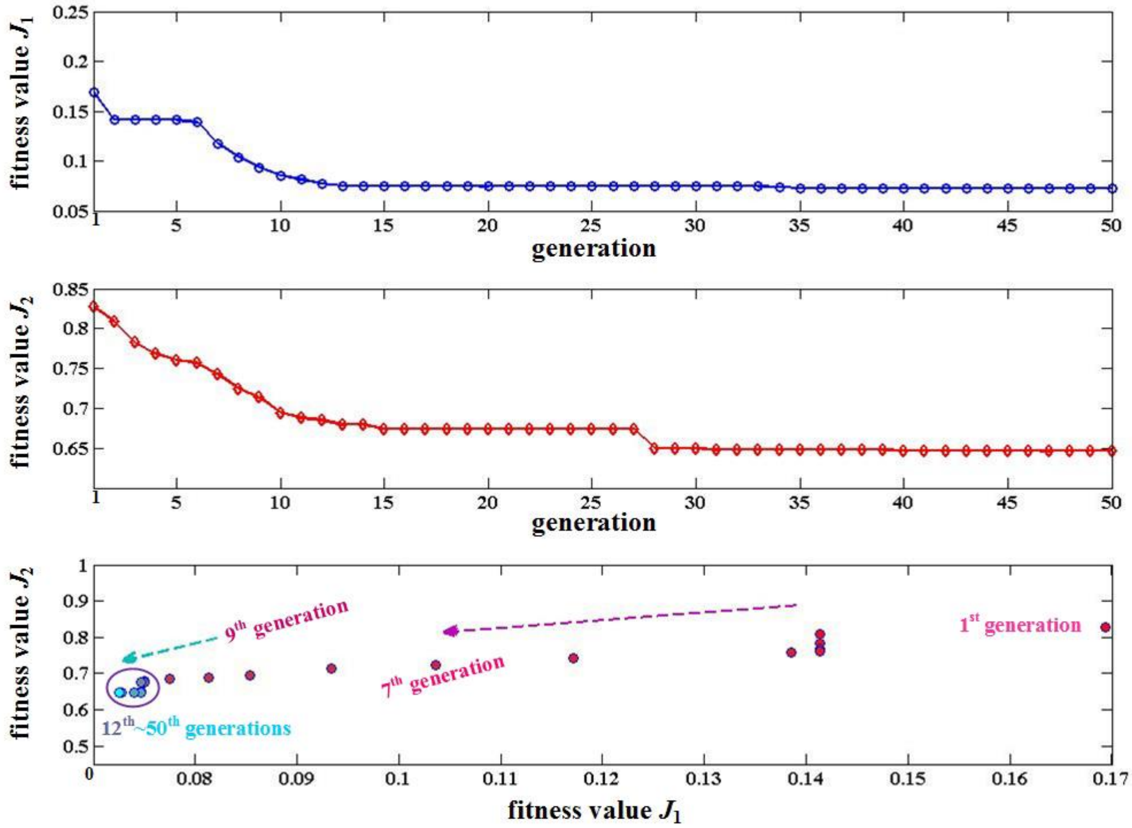


Fig. 3-7 FRP beam of the flexible link

3.6 Discussion

For realizing the swing-throw motion like human beings, the control torques with sinusoidal profile were designed. Comparing with the control torque with the sharp profile like square-wave profile, it could smooth the throwing motion and avoid serious vibration on the flexible link. However, on the current stage, different sinusoidal profiles were combined to form one profile. The sharp change which is needed to be checked may occurs on the combination point. So some other profile such as Gaussian shaped profile will be tested in the further research.

The Particle Swarm Optimization was used for designing the optimal control inputs and structure of the manipulator. Comparing with the Genetic Algorithm which is widely used in many researches, PSO algorithm is a much newer optimization algorithm. Some theories for PSO still need to be improved, however it is much more efficiency than GA. In order to reduce the number of initial parameter which is needed to be tuned by researcher, the advance PSO algorithm with multi-object functions was used. The price was to increase the complexity of the codes and operation time sharply. However for the objective functions designed in Chapter 3.5.4, the Pareto Front could not be formed precisely. Much proper objective functions can be found to form better Pareto Fronts. With conflict objective functions, more operation time is required for calculation and the researchers need to select an optimal solution from the Pareto Front by themselves. Sometimes it’s not easy to decide which a proper solution is. So for this problem, the design of objective functions without confliction is more important. For the result in Chapter 3.5.6, the performance was better than the former design.

3.7 Summary

In this chapter, for designing the control torques of the manipulator, the optimization algorithm for this throwing manipulator is needed to be introduced. Among many kinds of optimization strategies such as evolutionary programming, genetic algorithm, ant colony optimization and so on, the Particle Swarm Optimization (PSO) is selected for designing the optimal control torques of the throwing manipulator. It's the newest method comparing with other optimization methods. Indeed, the operation efficiency for PSO algorithm is much higher and also owns some other advantages.

The control torques with the sinusoidal form was designed. With this kind of control torques, the manipulator can realize a throwing motion and at the same time avoid introducing the serious impact influence for the flexible link of the manipulator. Then the basic PSO algorithm is applied for getting the optimal solutions for the designed torques and one optimal control input set is obtained.

But for basic PSO algorithm, many factors or coefficients need to be tuned by users. In order to reduce the number of this kind of factors, the multi-objective optimization algorithm is introduced. And also for avoiding the local minimum trap, the chaos strategy is embedded. So the Chaos Embedded Vector Evaluated Particle Swarm Optimization (CEVEPSO) method is designed in this chapter.

However, for MO problem, if the objective functions are in conflict with each other, the Pareto Front formed by the optimal solution set need to be searched. For this problem, at current stage, not enough optimal solutions can be found to form an accurate Pareto Front. So objective functions were reformed to avoid the confliction problem. Finally, the optimal solutions are derived for this throwing manipulator.

As which is mentioned in the introduction part of this paper, for the manipulator with the flexible link, the vibration control strategy is much necessary for keeping the performance and accuracy of the flexible manipulator. The above optimization work is used to design the control torques used in the process shown as Figure 2-3 (a) in the second chapter of this thesis, then after the ball throwing, the residual vibration will occur due to the sudden mass loss from the payload and flexibility of the manipulator. In the following process shown as Figure 2-3 (b) in the second chapter of the paper, it also needs to design the control torques for residual vibration suppression. So in the next chapter, the control strategy used for vibration suppression in “after throwing motion” process is presented.

Chapter 4 Input Shaping Technique for Residual Vibration Suppression

4.1 Introduction

There are many researches on the vibration control of the flexible manipulator. It's becoming much more important with the increasing ratio of the structure's length to its cross section area. For example, a manipulator in an underground waste storage tank, it must go through a 30cm diameter access hole, and then clean this tank which is 24 meter in diameter and 11 meter in deep. For such high ratio of length over cross section area, the vibration grows up a serious and fatal problem. The vibration suppression becomes an important procedure to be tackled with.

And among the control research for vibration suppression, most of the control strategies are feed-back controls [97-99]. But for feed-back control, it requires an additional sensor for gathering the vibration signals which can also introduce much disturbance signals at the same time. In order to get the proper control output, the hardware for control should calculate the results in time, then this requires the equipment owns higher performance especially for the system with complex model (even in the simulator, feed-forward control have quicker response performance than feed-back one). Also for real experiment, sometimes feed-back controller generates the unrealistic command due to the uncertainty in practical situation; even a proper control in simulation can be obtained. So owing to above reasons, this section introduced the input shaping technique which is a kind of typical feed-forward controls as the torque design strategy in after throwing motion shown as Figure 2-3 (b) in the second chapter.

Input shaping was first presented by Singhose et al [100-102]. This technique is a kind of feed-forward control method and is an effective way to optimize the performance of robots, flexible structures, spacecraft, telescopes, and other systems that have vibration, control authority, tracking, and pointing constraints. These constraints along with the dynamics and kinematics of the system under consideration can be included in a trajectory optimization or path planning procedure to ensure that the system satisfies the desired performance.

4.2 Input Shaping

4.2.1 Zero Vibration Shaper

As which is mentioned in the former part, the vibration problem of the flexible manipulator has to be tacked with. Especially for the flexible manipulator under the rapid motion, the residual vibration is tried to be suppressed to a low level. But commonly it needs to make an assumption that the exact parameters of the manipulator model can be obtained. In the real application, this assumption is always not available, however the robustness to this input shaping technique can be introduced in the later part of this chapter.

If the impulse or the step response to a linear system has been obtained, through making the command as a superposition of these elementary inputs, the response to this kind of shaped command signal can be gathered. For a pulse input, it can be viewed as a superposition of two step inputs, and one of them is defined as negative value with a time delay. And the response to these two impulses can be viewed as the superposition of the responses to these two individual step inputs.

Also the pulse shown in Figure 4-1 can be viewed as a step input convolved with a positive and a negative impulse delayed in time.

In Figure 4-2, it can assume that an input command is a 2 second pulse function. The corresponding impulse sequence can be expressed as

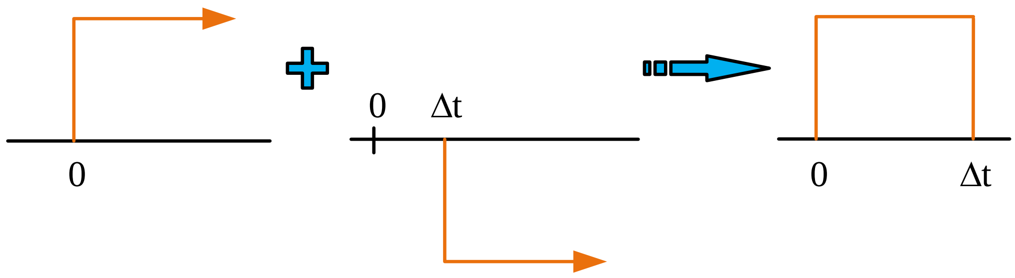


Fig. 4-1 A pulse formed by superimposing two step functions

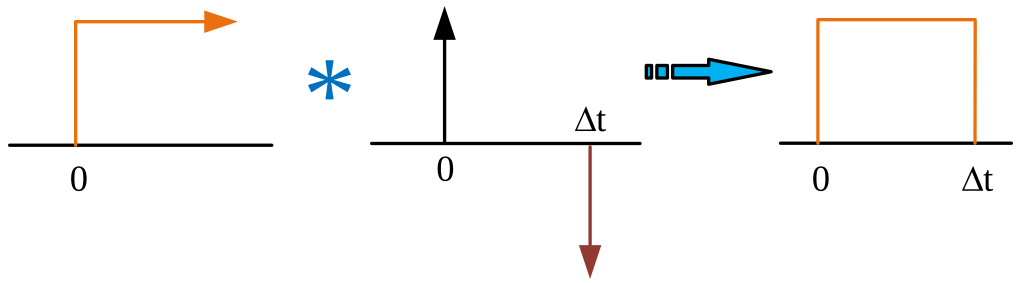


Fig. 4-2 A pulse formed by convolving a step with two impulses

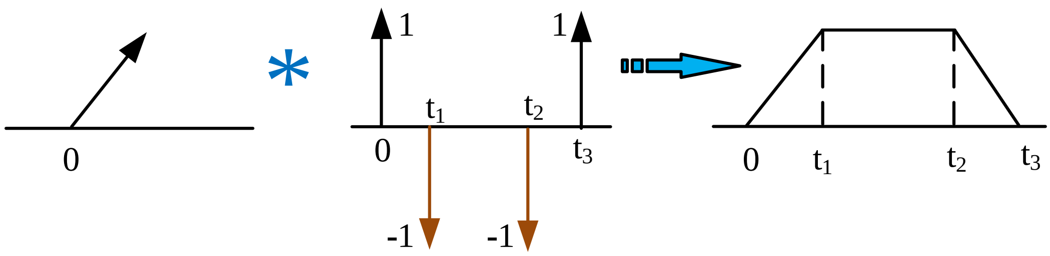


Fig. 4-3 Trapezoid profile function formed by convolving a ramp with impulses

$$\begin{bmatrix} AP_i \\ tp_i \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ 0 & 2 \end{bmatrix}. \tag{4.1}$$

The decomposition demonstrated in Figure 4-2 is not restricted to step functions convolved with impulse sequences. For a trapezoidal velocity profile, it can be viewed as a ramp input convolved with a sequence of impulses which is shown as Figure 4-3. It should be known that if a finite length command is to be decomposed to a semi-infinite length function like the ramp or step function convolved with the impulse sequence, the sum of total impulse amplitudes should be zero. On the other way, if a command is generating through convolving impulse sequence with a finite length command, the amplitudes of impulse should sum to one for the shaped command reaches the same final set point as the unshaped one.

In order to understand that how to generate input shaper which can suppress the vibration of the flexible system, it is better to start with the simple impulses. As Figure 4-1 (a) shown, if an impulse A1 is applied to this underdamped flexible system, such impulse will make vibration response to the system like the black solid line shown. Then in a later time, another impulse A2 is introduced to the same system in Figure 4-1 (a), and such impulse will also induces the vibration like impulse A1. The system response is shown as the red dotted line in figure 4-1 (a). If the amplitudes and time locations of each impulse could be adjusted or tuned properly, after combining the two responses excited by the impulses A1 and A2, the response of A1 can be cancelled by the response of A2.

If an underdamped, second-order system has an underdamped natural frequency of ω_n and a damping ration of ξ, then the residual vibration resulted from a sequence of impulses can be described as

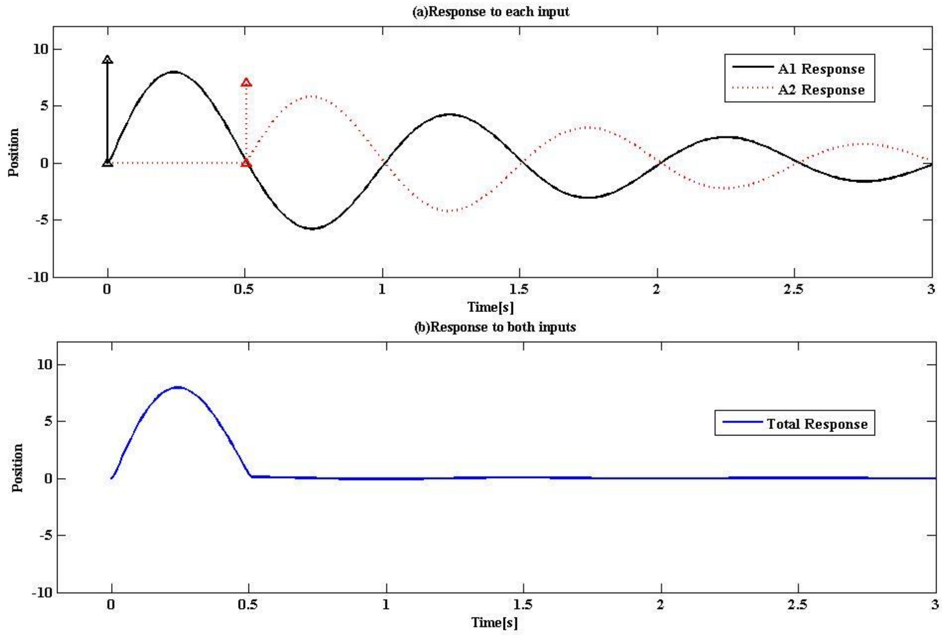


Fig. 4-4 Vibration caused by two impulses

$$VR(\omega_n,\xi)=e^{-\xi\omega_n t_n}\sqrt{\left[C(\omega_n,\xi)\right]^2+\left[S(\omega_n,\xi)\right]^2}, \tag{4.2}$$

where

$$C(\omega_n,\xi)=\sum_{i=1}^{I}A_i e^{\xi\omega_n t_i}\cos(\omega_d t_i),$$

$$S(\omega_n,\xi)=\sum_{i=1}^{I}A_i e^{\xi\omega_n t_i}\sin(\omega_d t_i).$$

here A_i and t_i are the amplitudes and time locations of the impulses. I is the number of impulses in the impulse sequence, t_I is the time location of the final impulse. The damping natural frequency is defined as

$$\omega_d = \omega_n \sqrt{1-\xi^2} \tag{4.3}$$

To generate an impulse sequence that yields no residual vibration, it is needed to set *VR* in Eq. (4.2) equal to zero and solve for the impulse amplitudes and time locations. And it also needs placing a few more restrictions on the impulses because there are infinite numbers of solutions that satisfy this condition. Here the impulse can be constrained as

$$\sum_{i=1}^{I} A_i = 1 \tag{4.4}$$

Sometimes, the impulses which satisfy the above equation may take on very large positive and negative values. The actuators will have the problems with these large impulses. So to obtain a proper solution, the amplitudes of the impulses are limited to the positive values as

$$A_i > 0 \quad i = 1, 2, \cdots, I \tag{4.5}$$

The impulses with minimum time sequence are tried to be found. They can set the vibration shown as Eq. (4.2) equal to zero, and also satisfy the constraints of impulse amplitudes shown in Eq. (4.4) and Eq. (4.5). Then the solution which is needed to be found can be restrict to the two impulse sequence shown as figure 4-4 with constrains in Eq. (4.4) and Eq. (4.5). For this problem, it has four unknown values which are two amplitudes (A_1, A_2) of impulses and two time locations (t_1, t_2) for the two impulses. So the time locations of the first impulse can be set to equal to zero as

$$t_1 = 0 \tag{4.6}$$

Then this problems can be restrict to find three unknown A_1, A_2, and t_2. If the Eq. (4.2) is set to zero, the $C(\omega_n, \xi)$ and $S(\omega_n, \xi)$ in this equation need to equal to zero at the same time. So the impulses must satisfy the below conditions as

$$A_1 e^{\xi\omega_n t_1} \cos(\omega_d t_1) + A_2 e^{\xi\omega_n t_2} \cos(\omega_d t_2) = 0 \tag{4.7}$$

$$A_1 e^{\xi\omega_n t_1} \sin(\omega_d t_1) + A_2 e^{\xi\omega_n t_2} \sin(\omega_d t_2) = 0 \tag{4.8}$$

then substitute Eq. (4.6) into above conditions and the conditions can be simplified as

$$A_1 + A_2 e^{\xi\omega_n t_2} \cos(\omega_d t_2) = 0 \tag{4.9}$$

$$A_2 e^{\xi\omega_n t_2} \sin(\omega_d t_2) = 0 \tag{4.10}$$

For Eq. (4.10), in order to get a solution, the sine term should equal to zero. So the following solution for time location t_2 can be got as

$$\begin{aligned} t_2 &= \frac{pT_d}{2}, \quad p = 1, 2, \ldots, \\ T_d &= \frac{2\pi}{\omega_d}. \end{aligned} \tag{4.11}$$

here T_d is the damped period of vibration. The infinite numbers of possible values exist for the time location of the second impulse. So in order to obtain the shortest command impulse, the smallest time location value for t_2 should be

$$t_2 = \frac{T_d}{2} \tag{4.12}$$

Now the time locations of two impulses have been confirmed. Then for this two impulses case, the amplitude constraint given in Eq. (4.4) can be simplified as

$$A_1 + A_2 = 1. \tag{4.13}$$

Then considering the damped natural frequency given in Eq. (4.3) and substituting the Eq. (4.12) and Eq. (4.13) into Eq. (4.9), which can get

$$A_1 - (1 - A_1)e^{\left(\frac{\xi\pi}{\sqrt{1-\xi^2}}\right)} = 0 \tag{4.14}$$

by solving this equation, which can get

$$A_1 = \frac{1}{1+P} \tag{4.15}$$

where

$$P = e^{\left(\frac{-\xi\pi}{\sqrt{1-\xi^2}}\right)}.$$

Then substitute the solution for A_1 to Eq. (4.13), the solution for A_2 can be obtained as

$$A_2 = \frac{P}{1+P}. \tag{4.16}$$

If the solution is rewritten, the sequence of two impulses that leads to zero vibration can be stated in matrix form as

$$\begin{bmatrix} A_i \\ t_i \end{bmatrix} = \begin{bmatrix} \frac{1}{1+P} & \frac{P}{1+P} \\ 0 & 0.5T_d \end{bmatrix} \quad i = 1, 2. \tag{4.17}$$

The two impulses for this solution can be expressed as following Figure 4-5

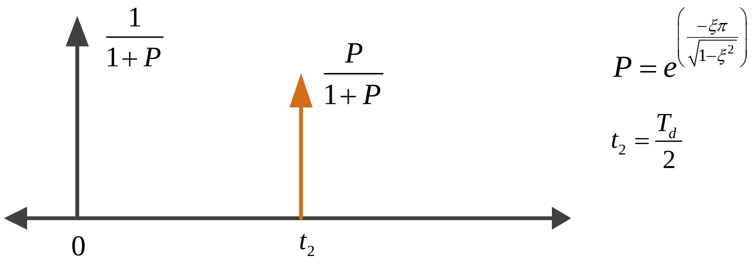


Fig. 4-5 Two impulse input shaper

The impulse input shaper shown as Figure 4-5 is referred to as a Zero Vibration (ZV) input shaper which can result in no vibration.

4.2.2 Zero Vibration Derivative Shaper and Multi-mode Shaper

In order to increase the robustness of the zero vibration shaper, the shaper can be designed by requiring the partial derivative of the residual vibration with respect to its frequency to equal to zero at the modeling frequency. So in a mathematical form, it can be expressed as

$$\frac{\partial VR(\omega,\xi)}{\partial \omega} = 0 \tag{4.18}$$

by adding this constraint, the vibration near zero can be kept as the actual frequency starts to deviate from the modeling frequency. So this derivation of the zero vibration shaper can be derived in matrix form as

$$\begin{bmatrix} A_i \\ t_i \end{bmatrix} = \begin{bmatrix} \frac{1}{(1+P)^2} & \frac{2P}{(1+P)^2} & \frac{P^2}{(1+P)^2} \\ 0 & 0.5T_d & T_d \end{bmatrix} \quad i = 1, 2, 3. \tag{4.19}$$

Note that this ZVD shaper has duration of one vibration period. However, for a small increase in rise time, a substantial amount of robustness is obtained.

The three impulses of the solution for this ZVD shaper can be expressed as following Figure 4-6.

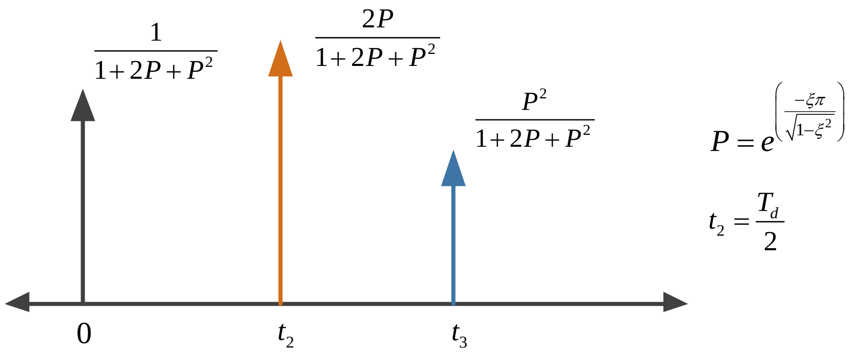


Fig. 4-6 Three impulse ZVD shaper

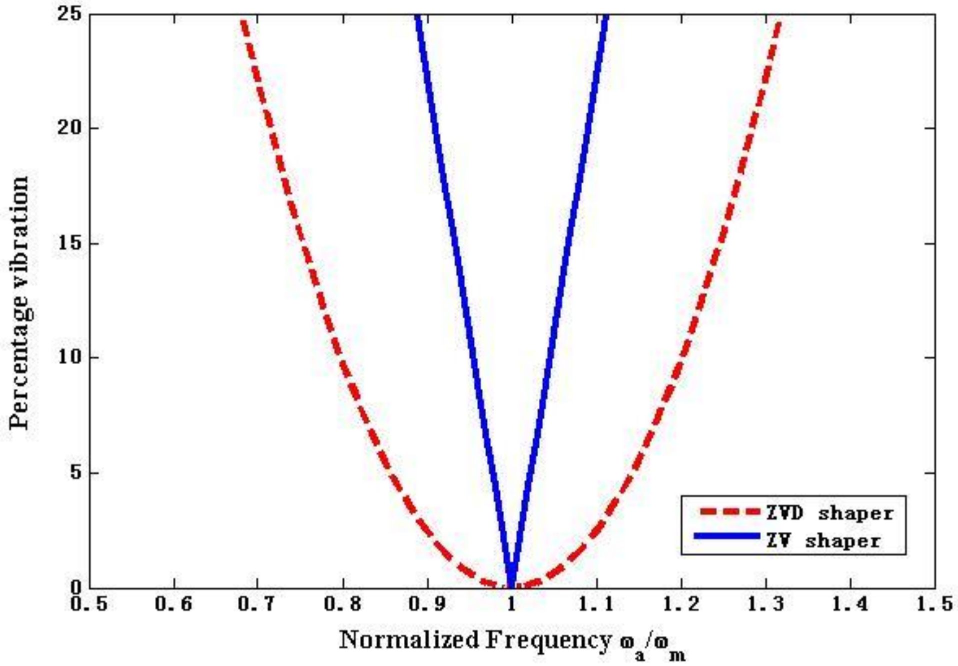


Fig. 4-7 Sensitivity curve for ZV and ZVD shaper

Through the sensitivity curve of the input shaper, the robustness for it can be viewed as Figure 4-7. This curve can show the amplitude of residual vibration as a function of system parameter. Figure 4-7 shows the sensitivity curve for ZV shaper and ZVD shaper. ω_a shows the actual frequency, while ω_m is derived from the modeling frequency. So it can be found that comparing with ZV shaper, ZVD shaper owns wider tolerance of model error and can add a certain level of robustness to the frequency error.

If the vibration suppression of multi-mode is considered, the ZVD shaper for each mode can be designed separately and then combine them together following the procedures as Figure 4-8.

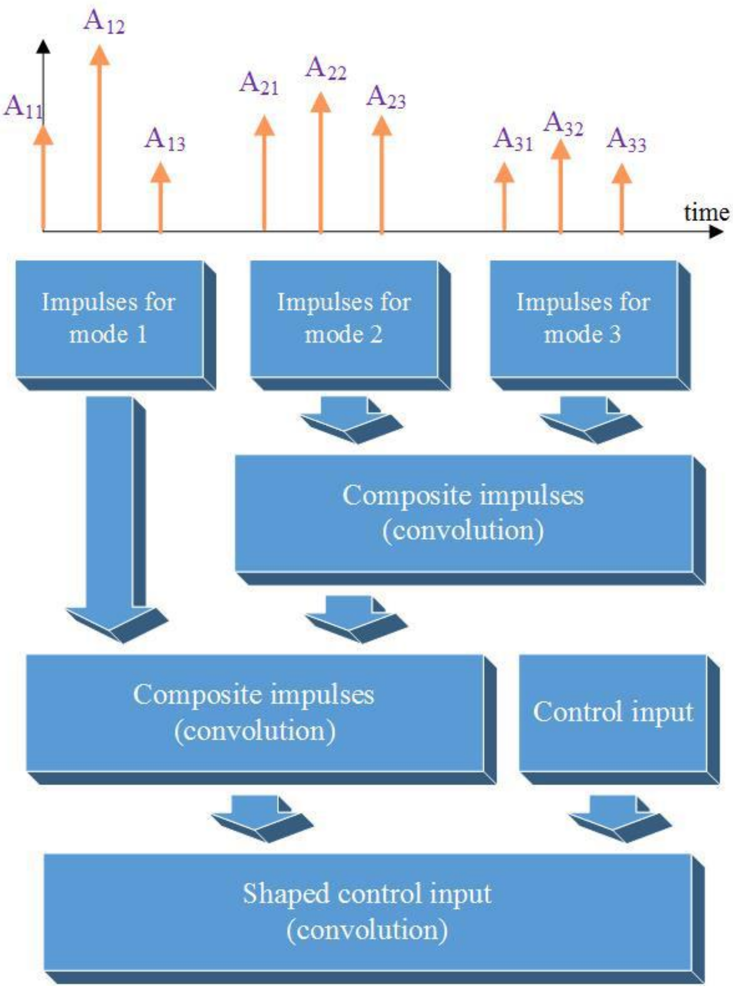


Fig. 4-8 Multi-mode ZVD shaper

4.3 Simulation Implementation of the ZVD Shaper

For the implementation of the input shaper, the reference command can be convolved with a sequence of impulses or the input shaper we know. Then the control input modified through the input shaping filter is introduced to the system controller. Finally, the controller can control the robot system as it designed. The whole procedure is shown in Figure 4-9.

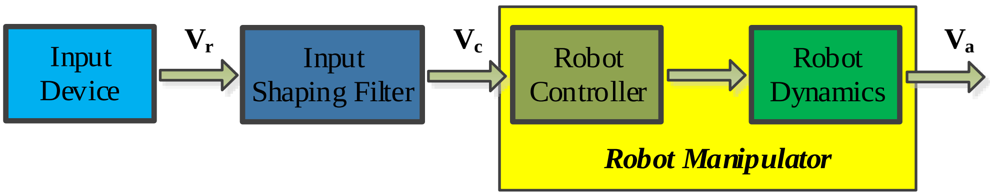


Fig. 4-9 Input shaper for the robot manipulator

Figure 4-10 shows the vibration suppression of the flexible beam with the ZVD shaper. Here the input torque modified by ZVD shaper is the sinusoidal form torque optimized in the third section (which torque parameters are [0.52, 0.53, 2.82, 1.09, 0.85, 1.08, 0.45, 0.84]) and the fiber angle of each FRP layer is also the optimized one shown in section three (which laminator angles for each layer of FRP are [55.5°, 49.8°, 74.1°, 67.8°]) . The pink dash line is condition without any vibration suppression strategy. The blue solid line is the result in which input torque is modified through the ZVD shaper. Figure 4-10 (a), (b), (c), and (d) plots show the angles and angular velocities of the second link, strains of the flexible link at 0.15 m which is measured from the second hub, and torques of the second link. The residual vibration of the flexible link has been reduced to some extent (from plot (e), it can be found that the residual vibration is reduced), but some amount of time lag has also been introduced to the system which is the price for vibration suppression. Indeed this time delay is not so long. It is so obvious in the strain data plot, but the lag is about 0.15s averagely. If the lag gap between the original and control systems hope to be narrowed, just ZV shaper can be used. But the price is the low robustness of the system which means the system parameters are needed to be obtained so exactly. In practical condition, this is too difficult to realize. Another method is to design the control torque through the infinite impulse filter which is also a kind of input shaping technique. But this filter is in progress.

Comparing with the feed-back control, this ZVD feed-forward control requires no additional sensors for feedback. And with knowing some coefficients of the manipulator, the ZVD shaper can be designed in advance. So the computing cost of the system can be reduced. The overload for the actuators and some dangerous condition can be avoid on design period. For the performance, it shows that the residual vibration can be reduced properly. There are time delay introduced to the system, however comparing with the whole operation period for throwing the ball, this time delay can be neglected indeed. If the time delay is required to be reduced further, the infinite impulse filter should be introduce in the further research.

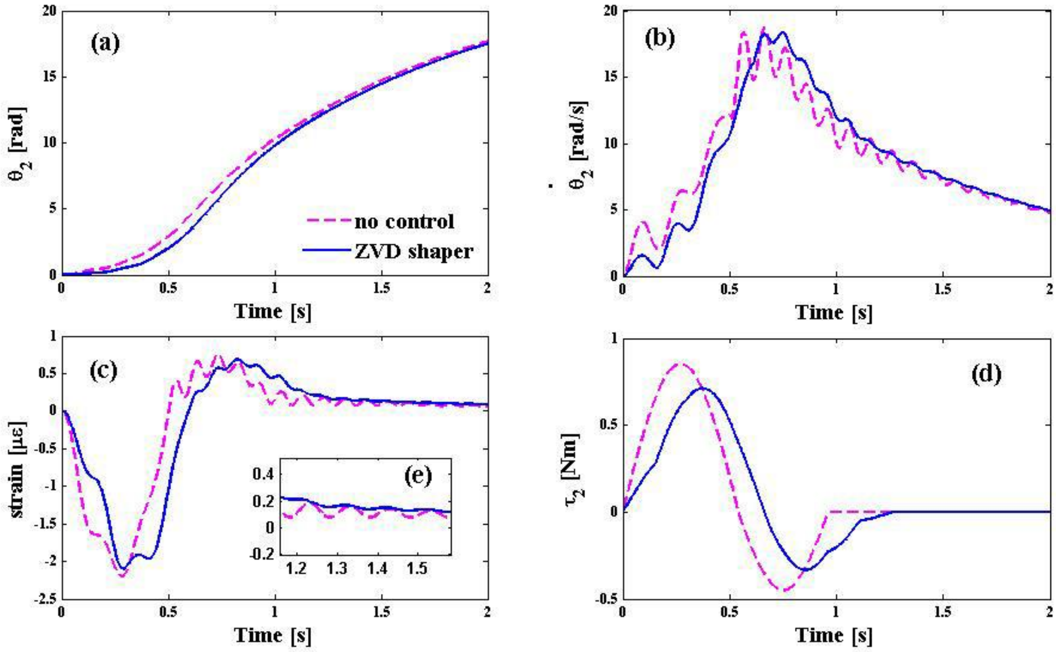


Fig. 4-10 Vibration suppression through ZVD shaper

4.4 Discussion

For residual vibration control of the flexible link, a feedforward control strategy-input shaping was applied to the system in this research. Indeed the feedback control strategy are commonly used in many research for vibration suppression. It can control the vibration at the desired point. However for feedback control, it requires additional sensors and these sensors may introduce the undesirable noise signals. Also for feedback control with complex calculation, it requires high performance for the hardware. In this research, the control torque could be designed and optimized in advance and required no additional sensor. So the feedforward control was much more suitable.

The basic Zero Vibration shaper could reduce the residual vibration but own no robustness for errors of the system parameters. So we introduced the Zero Vibration Derivative shaper. The price was that more time delay was introduced to the system and the residual vibration could not be cancelled completely but just be limited within a range like 5% of the maximum vibration without control. If much better robustness performance is required, some other input shaper can be applied with the price of larger time delay and wider range of tolerance to residual vibration. By using some advanced shapers, good robustness with low system time delay can be realized, however can't reduce vibration to zero without any time delay. In the further research, some other advance shapers will be tested for this system.

4.5 Summary

For this section, in order to suppress the vibration after the ball releasing, the vibration suppression strategy is needed for the manipulator. There are many feedback vibration control strategy on this research field. But comparing with the feedback control, the feedforward control owns some advantages such as less equipment requirement for calculation as the control profile can be pre-calculated.

In this paper, the input shaping technique is introduced. A two impulses Zero Vibration (ZV) shaper is derived at the beginning. Then in order to add some robustness to this shaper, the ZV shaper is modified as the Zero Vibration Derivative (ZVD) shaper. Also the sensitivity for these two shapers is presented through the sensitivity curve. And it can be found that the ZVD shaper owns much error tolerance than the common ZV shaper. For the multi-mode system, the procedure for forming the multi-mode input shaper is presented through a diagram.

Finally, the simulation result for the ZVD shaper applied to this throwing manipulator is presented. The vibration of the flexible link can be reduced obviously. However a little bit time delay and amplitude reduction is the price for the method of ZVD shaper.

In the next chapter, the system model which is derived, the performance of the optimization algorithm, and the performance of the control strategy are checked and evaluated through the real experimental manipulator which is set up.

Chapter 5 Experiments and Results

5.1 Introduction

In this chapter, in order to validate the proposed theory and simulation in the former chapter, some experiments are designed to confirm them.

The throwing manipulator owns a flexible link. So the system model becomes much more complex than the common rigid link one. For making the optimization and vibration control correctly, the accuracy of the system model which is set up need to be checked through the experiment. So at the beginning, the experiment is designed for confirming the dynamic model of this throwing manipulator.

Then for the design and optimization of the control torque, it also need to design the real experiments to check if the optimal solutions are acceptable or not. And the optimal ones should have better performance than the other arbitrary control inputs in the real implementation.

For the residual vibration of the manipulator after throwing, the experiment is implemented to the view if the control strategy can be applied or not and whether the vibration can be suppressed to an acceptable level for the manipulator.

5.2 Experimental Devices

The devices for the experiments are illustrated in this section. The real experimental manipulator is shown in Figure 5-1.

As the above figure shown, the manipulator consists of a rigid link, a flexible link with a ball holder on the extreme end of the beam, and two rotational joints. The first rigid link is actuated by a direct driven servo motor SGMCS05B3C11 (YASKAWA Electric Corp.) shown as in Figure 5-2(a) with its servo pack in Figure 5-2(b). And the second flexible link is rotated through a small AC servo motor SGMJVA5ADAHC01 (YASKAWA Electric Corp.) shown as in Figure 5-3(a) with its servo pack in Figure 5-3(b). And the specifications of these two motors are listed on Table 1 and Table 2. A strain gauge pasted on the flexible link is used to measure the strain of the location where the distance is 0.15 m from the second hub. The signal of the strain is amplified through dynamic strain amplifier DC204a (Tokyo Sokki Kenkyujo Company) which is shown in Figure 5-4.

(a) DC motor (SGMCS05B3C11)

(b) Servo pack

Fig. 5-2 DC motor for first link

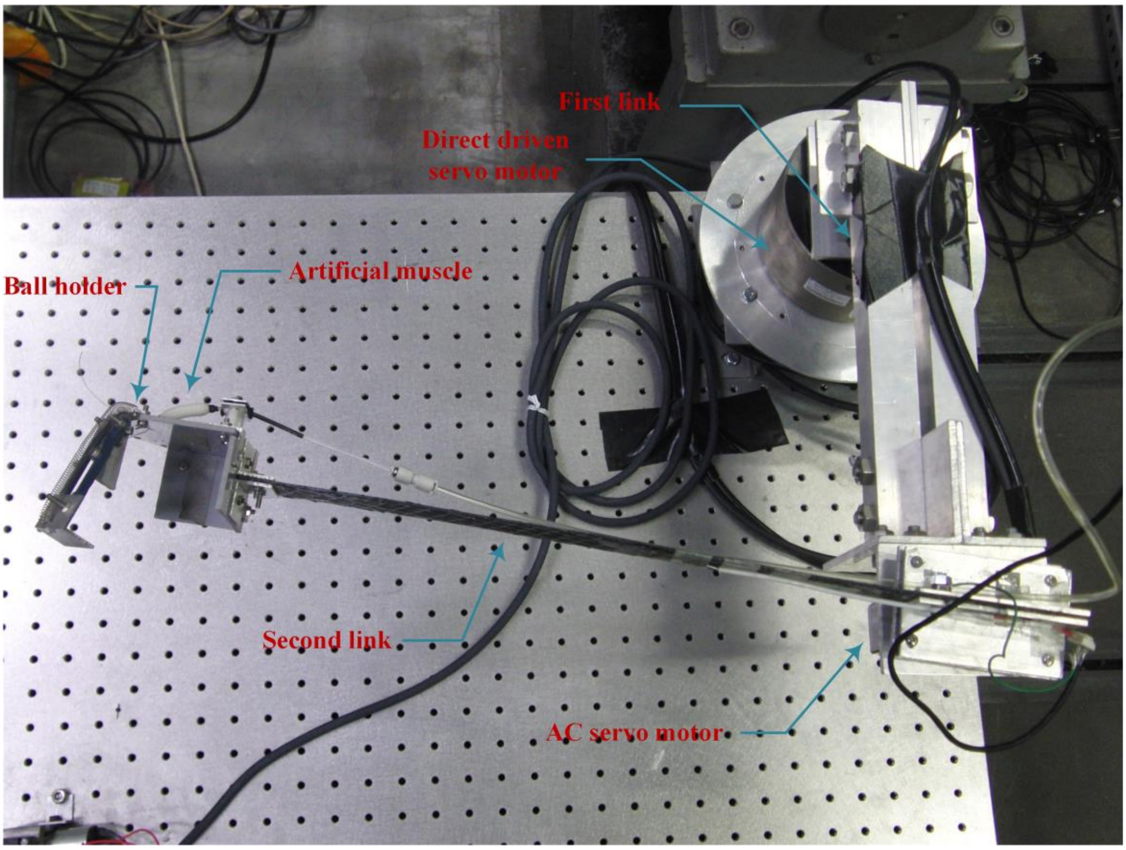


Fig. 5-1 Experimental manipulator

(a) AC servo motor SGMJVA5ADAHC01 **(b) Servo pack**

Fig. 5-3 AC servo motor for second link

Table 1 Specification of DD motor

SGMCS-05B3C11	
rated output	105W
rated torque	5.0N•m
instantaneous maximum torque	15.0N•m
rated rotation speed	200rpm
maximum ratation speed	500rpm

Table 2 Specification of AC motor

SGMJV-A5ADAHC01	
rated output	50W
rated torque	2.84N•m
instantaneous maximum torque	10.6N•m
rated rotation speed	143rpm
maximum ratation speed	286rpm

Fig. 5-4 Dynamic strain amplifier (DC204a)

Table 3 Specification of artificial muscle

SiktBW12PE30S	
rated pressure	0.2MPa
maximum extention part length	28.9mm
minimum extention part length	19,1mm
towing distance	9,8mm
shrink ratio	33.90%
maximum towing torque under rated torque	20.3N

An artificial muscle SiktBw12PE30S shown as Figure 5-5 is used to actuate the finger of the ball holder for holding and releasing a ball. Table 3 shows the specification of this artificial muscle. This artificial muscle is controlled through pneumatic system, and this system is controlled by a controlling circuit shown in Figure 5-6.

Fig. 5-5 Artificial muscle (SiktBw12PE30S)

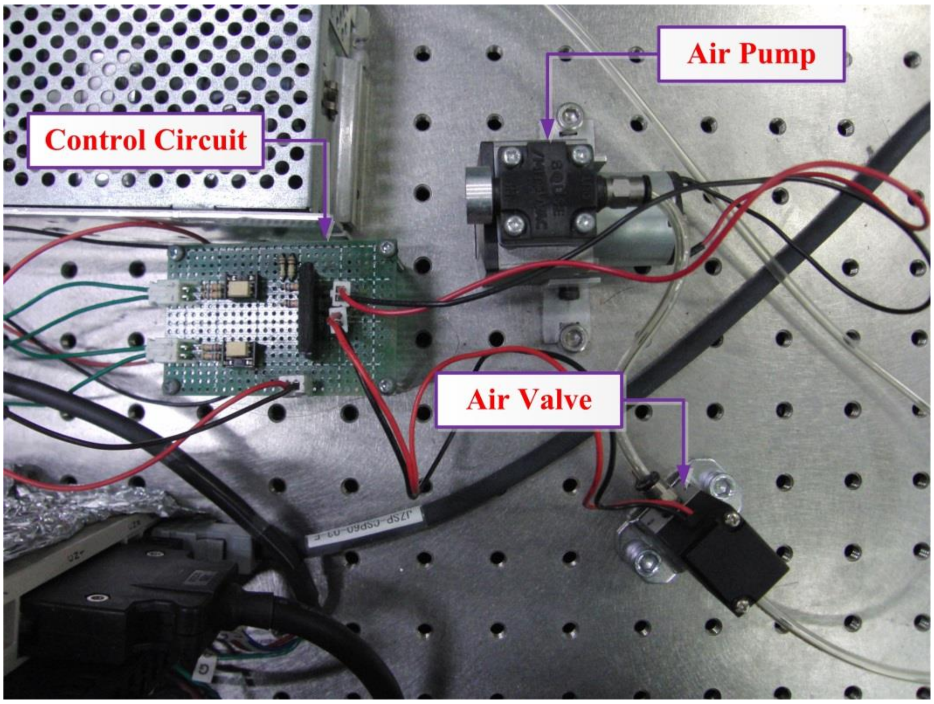


Fig. 5-6 Pneumatic system for the artificial muscle

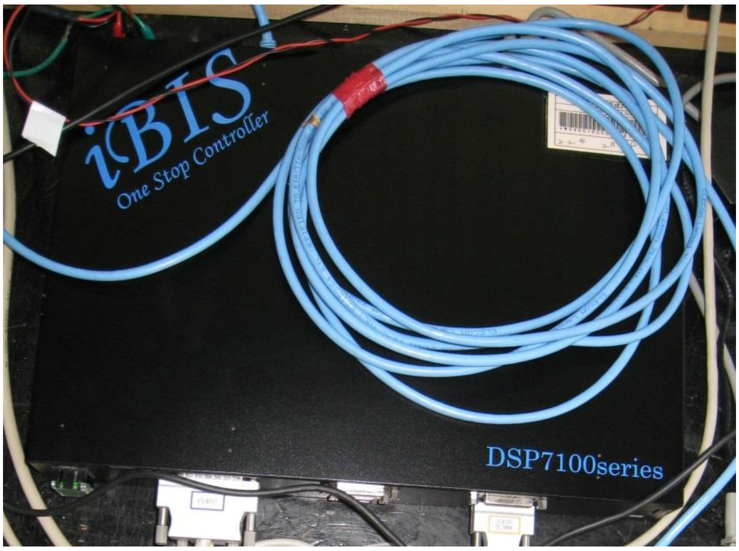


Fig. 5-7 Real time DSP (DSP7100)

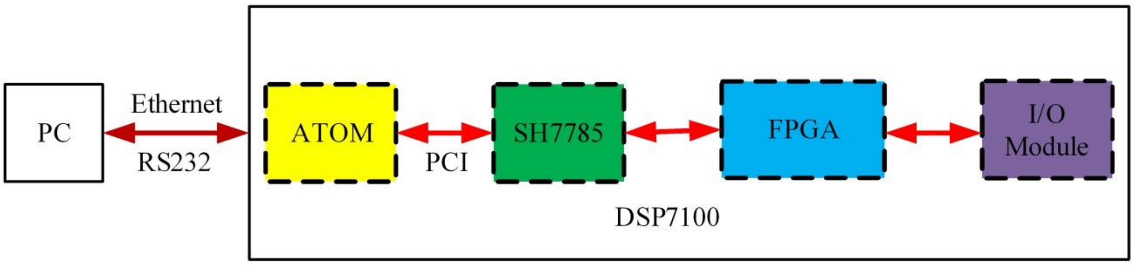


Fig. 5-8 Inter structure of the controller

A real time DSP control system DSP7100 (MTT Corp.) shown in Figure 5-7 is used to deal with the algorithm.

The system structure is shown as Figure 5-8 [103]. The PC can communicate with this DSP system with the RS232 cable which is just used for initializing the system and Ethernet cable which is applied for real time communication as figure shown. This DSP system includes three cores. One of them is Intel ATOM which is used to control the communication with PC terminal. Another one is SH7788. It is applied for real time control. The final core is the FPGA programmer. It can allocate and manage the I/O module for input and output of the DSP system.

If the system of DSP7100 is started, this DSP firmware needs about one and half minutes to initialize the system. Before the initialization of the system, the windows program and application program are needed to be programed in the host PC side. The application program enables the control assignment. A GNU C compiler is applied to make the executable files for application program. The windows code is compiled by the visual C editor and the interface window is also created. The executable files of the application program are downloaded into the DSP side through the Ethernet cable. After finishing the initialization, the host PC will delivery and gather data through the LAN cable communication with DSP side. The block diagram of the application for this DSP system can be shown as Figure 5-9 [103]. The system block is shown as Figure 5-10.

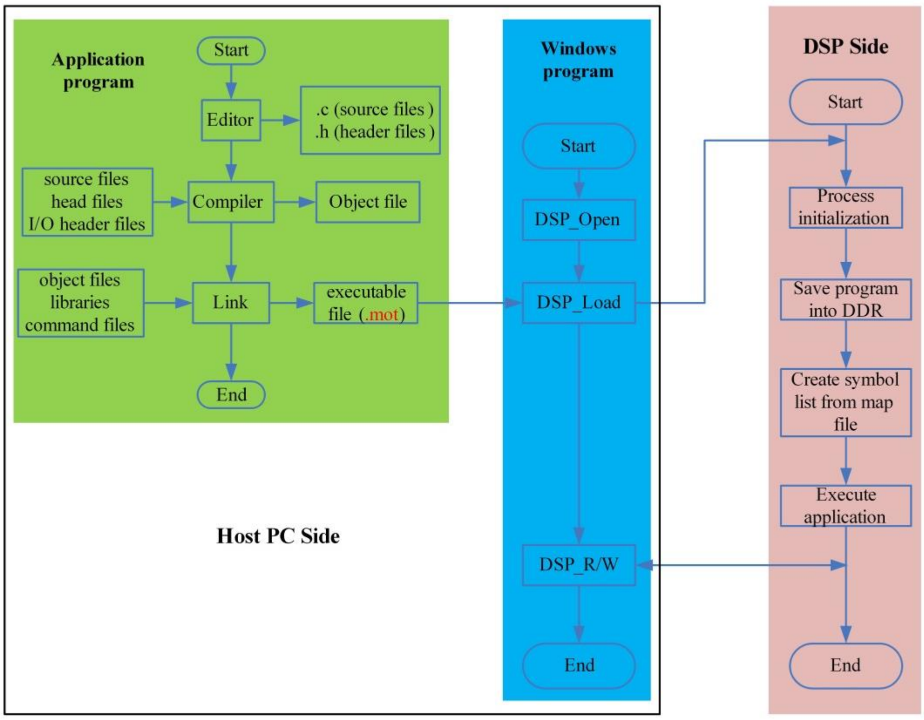


Fig. 5-9 Block diagram of the application for DSP7100

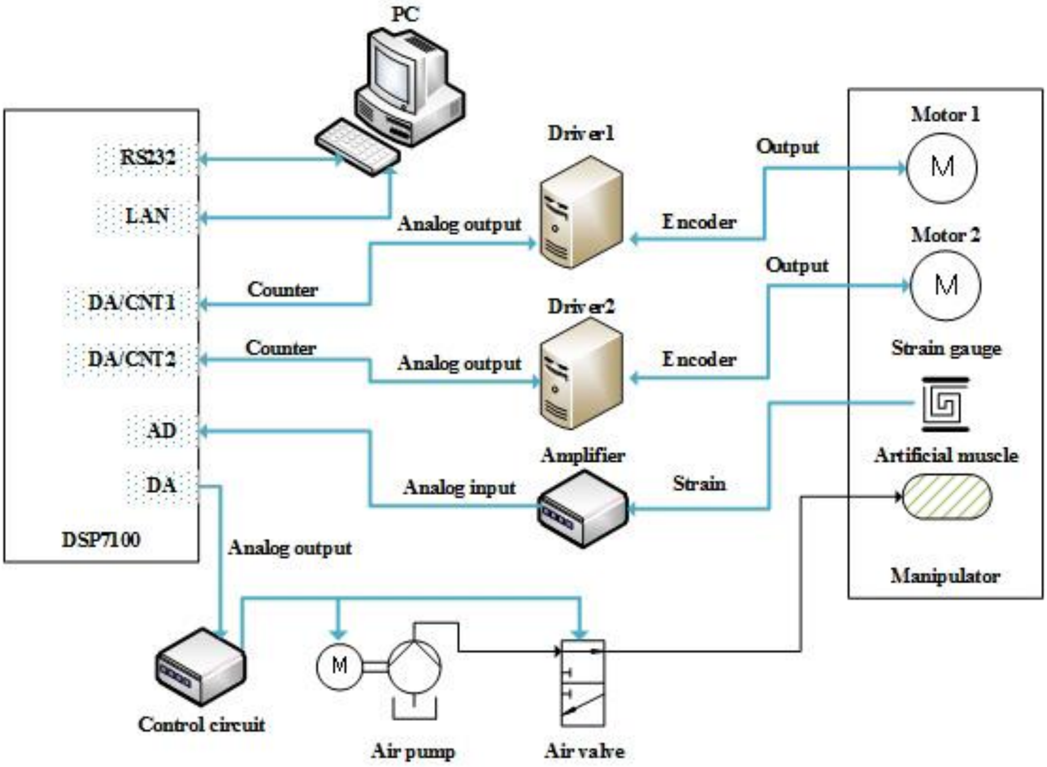


Fig. 5-10 Experimental system block

5.3 Experimental Results

5.3.1 Experimental Verifications for System Models

In this section, the experiments are operated to verify the validity of the models built up in the second section. The control torques are the optimal ones which are determined through CEVEPSO (which torque parameters are [0.52, 0.53, 2.82, 1.09, 0.85, 1.08, 0.45, 0.84]) and the structure is also the optimized one (which structure angles for each layer of FRP are [55.5°, 49.8°, 74.1°, 67.8°]).

Table 4 Parameters of experimental manipulator

Symbol	Meaning	Quantity
m_1	Mass of first hub	1.45 kg
J_1	Inertia of first hub	3.45×10^{3} kg·m^2
r_1	Radius of first hub	0.47×10^{-1} m
m_{L1}	Mass of rigid link	0.19 kg
L_1	Length of rigid link	0.26 m
m_2	Mass of second hub	1.04 kg
J_2	Inertia of second hub	1.30×10^{-1} kg·m^2
r_2	Radius of second hub	4.75×10^{-2} m
L_2	Length of flexible link	0.37 m
ρ_e	Density of FRP beam	7.70×10^{3} kg/m^3
E_s	Young's modulus for steel	2.00×10^{2} GPa
E_f	Young's modulus for FRP	9.34×10 GPa
I_e	Inertia of cross section of the beam	5.25×10^{-11} m^4
A_e	Cross sectional area of the beam	8.64×10^{-5} m^2
m_c	Mass of the ball holder	0.10 kg
J_c	Inertia of the ball holder	6.00×10^{-5} kg·m^2
r_c	Radius of the ball holder	0.03 m
m_b	Ball mass	5.50×10^{-2} kg
J_b	Ball inertia	3.75×10^{-5} kg·m^2

The comparison of results between experiment and simulation are shown in Figure 5-11 and Figure 5-12. Figure 5-11 is the result when the manipulator does not take the payload (ball). And Figure 5-12 is the condition when the manipulator has the payload. Table 4 lists the parameters of the experimental manipulator. The solid blue lines depict the experimental data and the broken pink line show the simulation results. The cyan and green solid lines in (f) plots of Figure 5-11 and Figure 5-12 are the actuating torques of first and second links of the manipulator. Figures (a) and (c) plot the angles and angular velocities for the first link; (b) and (d) plots are the angles and angular velocities of the second link; (e) plot is the strain for the flexible link.

The errors between experiment and simulation exist, but it can be accepted in a certain extent. The friction factors of the real device own nonlinear relationship to rotation speed; however in the simulation, the constant friction factors are used. Also some influence factors like temperature and binding material between the FRP sheets can make the parameters of the flexible beam hard to be measured accurately. All of these factors introduce the errors between experiment data and simulation.

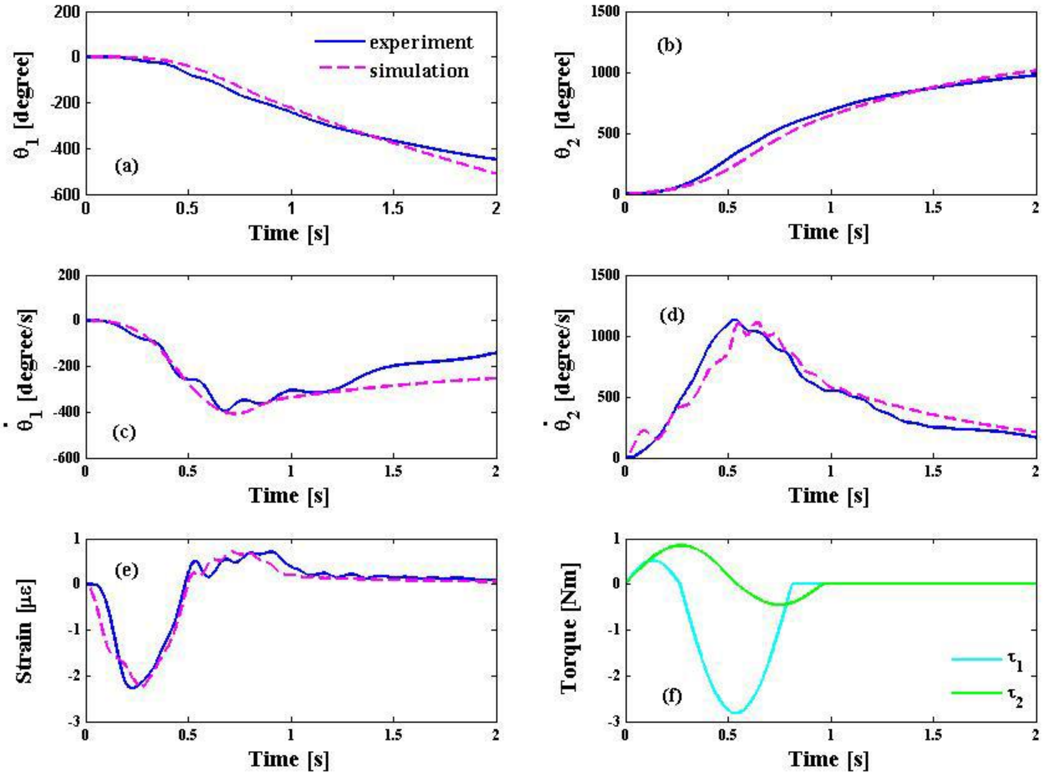


Fig. 5-11 Comparison between simulation and experiment without ball

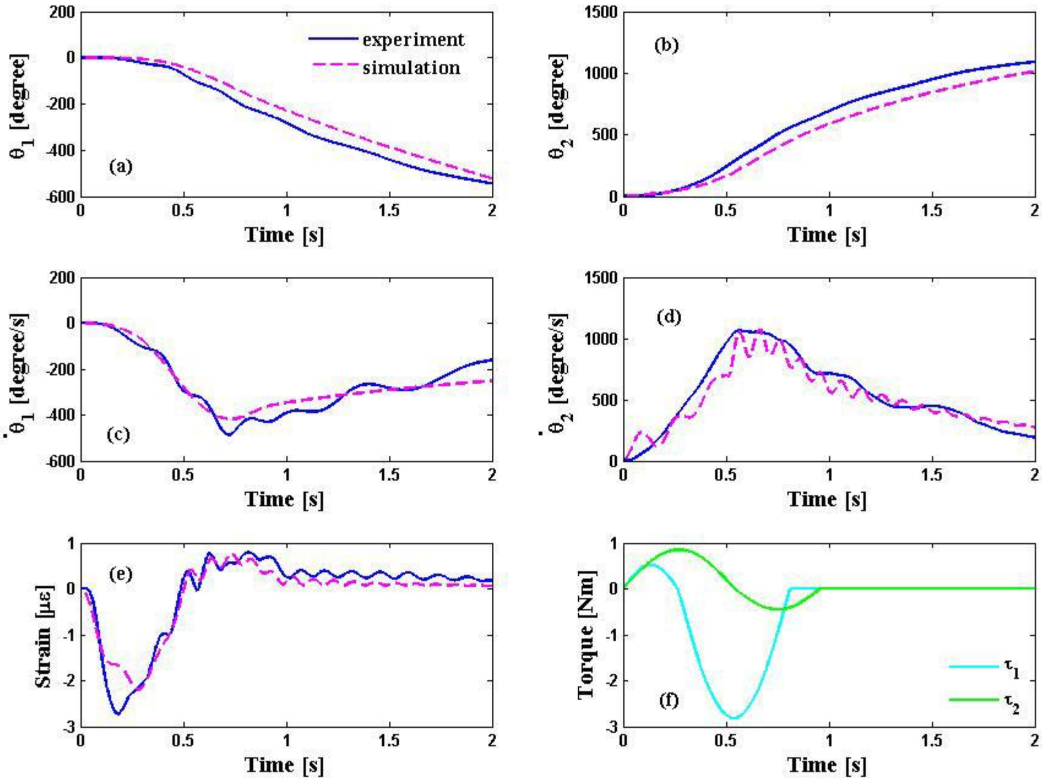


Fig. 5-12 Comparison between simulation and experiment with ball

5.3.2 Experimental Verifications for CEVEPSO

In order to verify the optimal results of CEVEPSO, some parameters are selected randomly within the solution range of the algorithm and compare the results with the optimal one. Owing to the limitation of the time and materials, four random groups of structure are chosen and check their results comparing with the optimal one for beam structure. Figure 5-13 shows the FRP beams used in part of experiment and the specification for steel base and FRP sheet is listed in Table 5. The strain gauge used for obtaining the deformation data of the flexible link is set on the position 0.15m which is measured from the right side of each beam shown in the picture. The FRP fiber directions for test are listed on Table 6. Group A are [24.9°, 4.15°, 8.75°, 74.11°]; Group B are [62.5°, 28.5°, 85.5°, 9.1°]; Group C are [39.5°, 34.3°, 68.8°, 71.5°]; Group D are [16.8°, 44.1°, 40.0°, 58.17°]; Optimal Group are [55.5°, 49.8°, 74.1°, 67.8°]. And figure 5-14 shows the experimental testing result of these five groups under the optimal torque inputs ([0.52, 0.53, 2.82, 1.09, 0.85, 1.08, 0.45, 0.84] substitute into Eq. (3-19)). The x axis is the releasing speed of the ball and y axis indicates the bending ratio which is the ratio of the flexible deformation between bend backward and forward given by Eq. (3-48). The higher ratio makes better energy utility efficiency of the deformation energy. The optimal group owns the best releasing speed comparing with other group and better bending ratio within all groups except for group D. Group D owns a higher ratio, but the releasing speed is the lowest among all of these groups. So the optimal group makes the best performance in this testing experiment.

Table 5 Specification of steel and FRP sheet

	Steel	Carbon fiber
Young's modulus	206GPa	93.4GPa
Density	7.86X10^{3}kg/m^3	1.50X10^{3}kg/m^3
Poisson's ratio	0.3	0.31

Table 6 FRP fiber directions for test

Groups	Fiber Directions			
Optimal Group	55.5°	49.8°	74.1°	67.8°
Group A	24.9°	4.2°	8.8°	74.1°
Group B	62.5°	28.5°	85.5°	9.1°
Group C	39.5°	34.3°	68.8°	71.5°
Group D	16.8°	44.1°	40.0°	58.2°

Table 7 Control torques for test

	Control Torque of First Link				Control Torque of Second Link			
Groups	A_{11}	T_{11}	A_{12}	T_{12}	A_{21}	T_{21}	A_{22}	T_{22}
Optimal Group	0.52	0.53	2.82	1.09	0.85	1.08	0.45	0.84
Group 1	2.12	1.39	0.75	1.40	0.66	0.50	0.35	1.00
Group 2	2.41	1.46	0.81	1.46	0.96	0.93	0.82	0.55
Group 3	1.34	1.40	2.08	1.45	0.69	0.43	0.82	1.42
Group 4	1.85	1.23	1.98	0.83	0.68	0.58	0.73	0.43
Group 5	1.05	0.45	0.69	1.30	0.72	0.74	0.95	0.43
Group 6	1.37	0.81	2.03	1.27	0.26	0.93	0.50	1.11

Next the optimal torques is tested. Like the structure test, six groups of torque inputs are selected randomly. Table 7 lists the testing groups of control torques. From Group 1 to 6, they are [2.12, 1.39, 0.75, 1.40, 0.66, 0.50, 0.35, 1.00], [2.41, 1.46, 0.81, 1.46, 0.96, 0.93, 0.82, 0.55], [1.34, 1.40, 2.08, 1.45, 0.69, 0.43, 0.86, 1.42], [1.85, 1.23, 1.98, 0.83, 0.68, 0.58, 0.73, 0.43], [1.05, 0.45, 0.69, 1.30, 0.72, 0.74, 0.95, 0.43], and [1.37, 0.81, 2.03, 1.27, 0.26, 0.93, 0.50, 1.11]. The testing result under optimal structure is shown as figure 5-15. The optimal group shows the best performance both on releasing speed and bending ratio among all of the testing groups. So the Optimal solution obtained by the CEVEPSO algorithm is acceptable through this part of experiment.

Fig. 5-13 Testing FRP beam

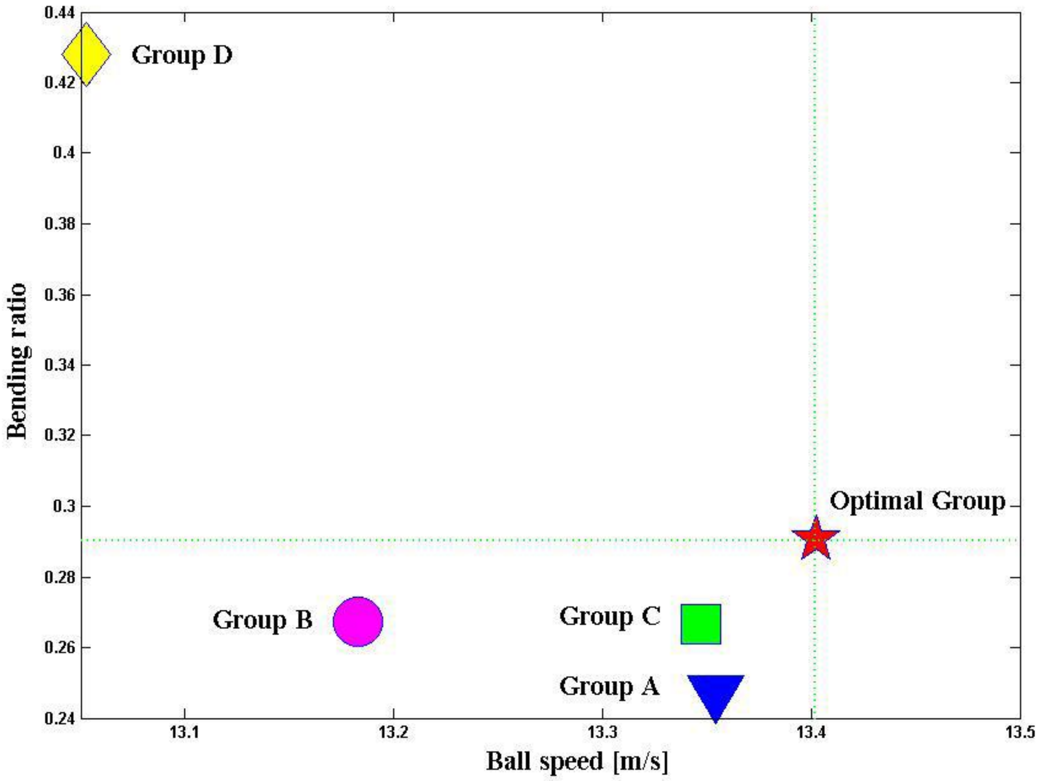


Fig. 5-14 Testing result for beam structures

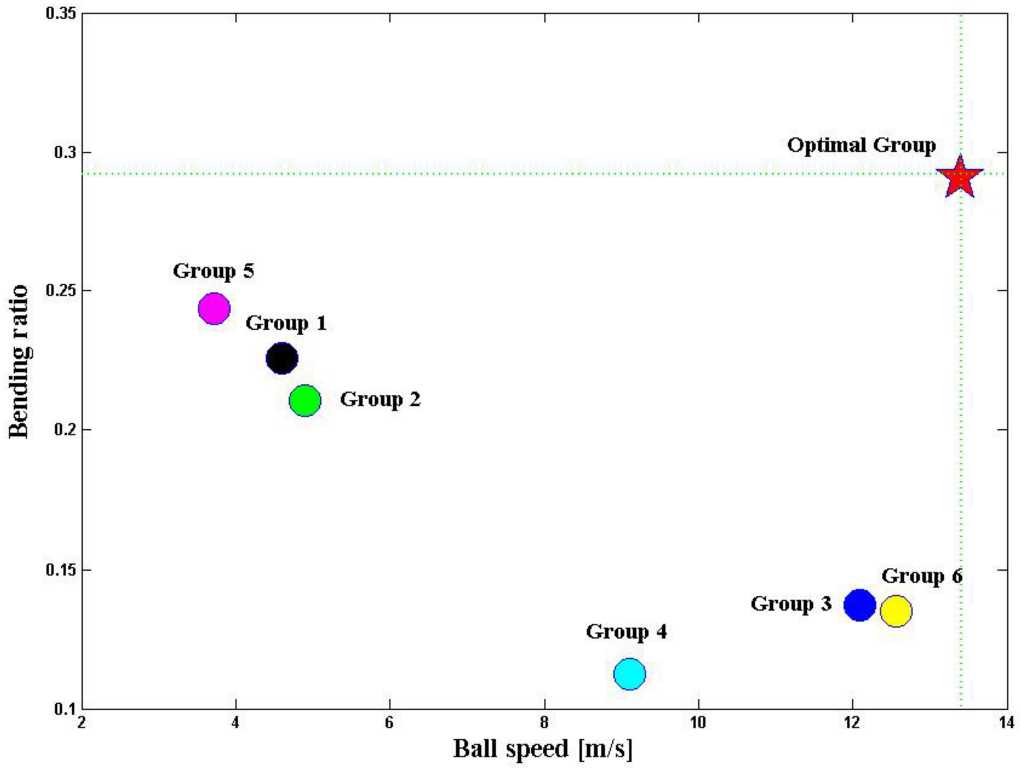


Fig. 5-15 Testing result for beam torque

5.3.3 Experiment for Residual Vibration Suppression

The performance of the input shaper introduced in the fourth section is evaluated in this part. Figure 5-16 shows the result which the manipulator is actuated by the optimal control torque and modified one through ZVD input shaper. The left plot shows the strain of the flexible beam and the right plot represents the control profiles for the flexible link. The blue solid line is experiment data without any control strategy and the pink dash line shows the experiment result when the control profile is modified through the ZVD shaper. The result shows that the input shaper can smooth the deformation and reduce the residual vibration after ball releasing to a certain extent. As the simulation, the designed control profile modified by the ZVD shaper can work properly in the realistic manipulator. But the price is to reduce some peak values and introduce a certain time delay to the system which also can be found in the simulation result.

In this experiment section, the accuracy of the mathematical model is confirmed and this model can be accepted even some errors exist. If more accuracy model is required, in further research the model is needed to be modified by considering more influence factors. The experiment for the CEVEPSO shows the optimal results obtained through simulation can be applied in the real condition. The results are not absolutely accurate since the real condition contains many influence factors which can't be calculated totally in the simulation and the system model is not absolute accurate without errors. The same condition also exits on the experiment for the suppression of residual vibration.

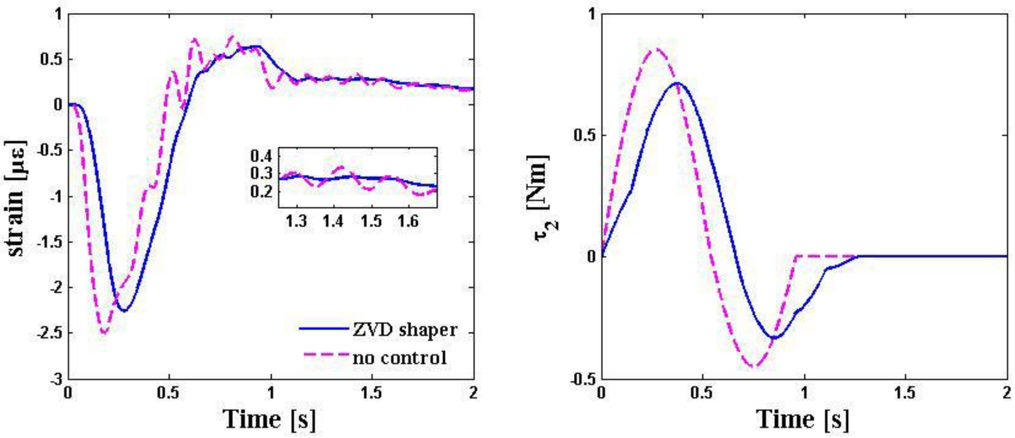


Fig. 5-16 Experiment for input shaper

5.4 Discussion

In the experiment to verify the system model, the errors mainly caused through the friction of each motor, unconsidered nonlinear terms of the model, and imprecise parameters of the system. The friction played the most important role among them. For the nonlinearity of the friction, it was not easy to obtain the accurate value of friction factor. Indeed the friction of the motor not only depends on the rotation speed. Some other factors may have the influence to it like temperature of the environment. So before experiments, the warm-up program would be operated for reducing the effect of environment temperature. In the further research, through introducing more factors to the model, the accuracy of the model can be improved.

The performance of the CEVEPSO algorithm was verified in the second experiment. For the result of the optimal control torque. It showed that optimal group had the best performance comparing with the other group without optimization. For the structure, optimal group showed better performance than others except one group which owned higher bending ration but with low releasing speed. This may be caused from the imprecise parameters of the flexible structure measured from experiment. Steel beam with multiple layers of FRP sheets was used for the flexible link. It was made by hand and binding material was used to attach the FRP sheets. All of these made the parameters of the flexible link was hard to be measured accurately. Some methods may be found to measure the parameters of the flexible link more exactly in the further research.

The input shaper could be worked well in the real experimental manipulator as the third experiment shown. And the residual vibration was reduced obviously. Indeed the time delay for system response is not so serious in the experiment. But if less time was required, some other advance input shaper would be tested in the further research.

5.5 Summary

In the former chapters of this paper, the manipulator for realizing a ball throwing motion was built up. Then the system was analyzed and derived the equations of motion and state equations for this system. In order to control the manipulator properly and imitate a throwing motion optimally, the input profiles of the control torques are designed and introduced the PSO algorithm for obtaining the optimal solutions for the control inputs. Also the throwing motion after ball releasing excites the vibration. For vibration suppression of the residual vibration, the input shaping technique is applied to this throwing manipulator.

All of these theories and simulations need to be firmed or identified through the real experiment. So in this section, the throwing manipulator is designed and the experimental system firstly. And then the experiment was divided into three main parts according to the former contents which are mentioned.

The experiments in the first part are designed to verify the model accuracy derived for the manipulator. And it can be found the results are acceptable even with some errors. Through the group experiments of the beam structures and control torques, the performances of the CEVEPSO algorithm which is introduced in the third chapter was confirmed. The results show that the optimal control torques own better performance than other non-optimal ones. Finally the results of vibration suppression by input shaper are tested through the real experiment. The result shows the control inputs modified by the input shaper which is applied in this paper can suppress or reduce the vibration to a certain extent.

Chapter 6 Conclusions and Further Works

6.1 Conclusions

In this paper, a two link ball throwing manipulator with one flexible link as the second link was designed with a ball holder at the extreme end of this flexible link, and the equations of motion and state equations were derived for it. This flexible manipulator could realize a swing-throwing motion as human beings as it was designed.

In order to obtain the proper control torques and structure for this manipulator, the sinusoidal input torque-profile and the FRP structure of the flexible link were designed. The sinusoidal profile torque helped the flexible manipulator avoid the serious deformation and vibration. Then the CEVEPSO algorithm was introduced to optimize the governing parameters for the control torques and the FRP structure. The optimal control torques and beam structure were obtained at the end of this section. Both of the basic PSO algorithm and CEVEPSO could obtain the optimal solutions for the problem in this paper. The advanced PSO algorithm needed less initial setting parameters to be tuned and shown better performance than the basic one. However, for the same CEVEPSO algorithm, according to different objective functions, the performance and optimal results were obviously different. In this research, the cost function in confliction with each other could obtain the optimal solutions, but for forming a clear Pareto Front, the calculation cost was high and the performance was not so reliable. Comparing with the problem with the designed conflict cost functions, the algorithm without conflict functions got better optimal solution with less cost for calculation.

For the residual vibration of the flexible structure after throwing, the input shaping technique was applied for it. The common ZV shaper was derived at beginning and the robustness shaper called as ZVD shaper was introduced. The ZVD shaper was used for suppressing residual vibration occurred after the ball throwing motion. Through the simulation, it could suppress the system vibration to some extent; however a certain time lag and amplitude attenuation were the price. ZV shaper owned less time lag than ZVD shaper with the price of low robustness. However, comparing with the whole operating period of the throwing motion, this time lag time could be accepted. And the robustness is more important than the time lag here.

In the fifth chapter, an experimental manipulator was built up to verify the accuracy of the system models which were derived. The errors caused by some unconsidered nonlinear terms and frictions could be found in the experiment. For considering the friction of the actuator, the accuracy of the simulation can be increased. Then the solutions of CEVEPSO algorithm are checked. It could be found that the optimal result obtained through the object functions without conflict showed better performance than the arbitrary ones. The CEVEPSO algorithm can direct the particles to the optimal solution. Finally the performance of the input shaper was evaluated in the real experimental device. The input shaper designed in simulation could also worked well in the real experimental setup. The residual vibration could be reduced but not completed canceled for the errors introduced through system model.

6.2 Further Works

In this paper, some problems for this ball throwing robot was studied. Some results for these problems were obtained, however there are still many research problems left for this complex flexible system.

For the dynamic model of the throwing manipulator, too large or severe deformation of the flexible link were not considered. In the future research, the severe deflection of the flexible structure will be taken into consideration. The profile of the control input which is used in this paper is a sinusoidal profile. It can actuate each link smoothly; however the peak value of the releasing speed for the ball is also limited. So for the further research, other types of control profiles will be detected and tested. For the structure of the flexible link, some parameters like the cross section area of the composite beam are the constant value. By introducing the variable values for the structure parameters, the flexible structure can obtain much better mechanical properties. For the PSO algorithm applied in this paper, there are still many aspects that can be and need to be improved. In the fourth chapter, the ZVD shaper is introduced to suppress the residual vibration of the flexible link. For the further research, some other advanced feedforward control will be tested and the performance of them will be used to compare with each other.

References

[1] Book, W. J., Modeling, Design, and Control of Flexible Manipulator Arms: A Tutorial Review, *Proceedings of 29th Conference on Decision and Control*, Honolulu, pp. 500-506, 1990.

[2] Book, W. J., Controlled Motion in an Elastic World, *Journal of Dynamic Systems, Measurement, and Control*, Vol. 115, pp. 252-261, 1993.

[3] Dwivedy, S. K. and Eberhard, P., Dynamic Analysis of Flexible Manipulators, A Literature Review, *Mechanism and Machine Theory*, Vol. 41, pp. 749-777, 2006.

[4] Plunkell, R. and Lee, C. T., Length Optimization for Constrained Viscoelastic Layer Damping, *Journal of the Acoustical Society of America*, Vol. 48, pp. 150-161, 1970.

[5] Aubrun, J. –N., Theory of the Control Structures by low-authority controllers, *Journal of Guidance and Control*, Vol. 3, pp. 444-451, 1980.

[6] Book, W. J., Alberts, T. E. and Hastings, G. G., Design Strategies for High-speed Lightweight Robots, *Computers in Mechanical Engineering*, Vol. 5, pp.26-33, 1986.

[7] Cannon, R. H. and Schmitz, E., Initial Experiments on the End-point Control of a Flexible One-link Robot, International Journal of Robotics Research, Vol. 3, pp. 62-75, 1984.

[8] Harashima, F. and Ueshiba, T., Adaptive Control of Flexible Arm Using the End-point Position Sensing. *Proceedings of Japan-USA Symposium on Flexible Automation*, Osaka, pp. 225-229, 1986.

[9] Hasings, G. G. and Book, W. J., A Linear Dynamic Model for Flexible Robotic Manipulator. *IEEE Control Systems Magazine*, Vol. 7, pp. 61-64, 1987.

[10] Nagathan, G. and Soni, A. H., Non-linear Flexibility Studies for Spatial Manipulators. *Proceedings of the IEEE International Conference on Robotics and Automation*, San Francisco, pp. 373-378, 1986.

[11] Hu, A., A Survey of Experiments for Modeling Verification and Control of Flexible Robotic Manipulators, *Proceedings of The 1st IEEE Regional Conference on Aerospace Control Systems*, Westlake, pp. 344-353, 1993.

[12] Nagarkatti, S. P., Rahn, C, D., Dawson, D. M. and Zergeroglu, E., Observer-based Modal Control of Flexible Systems Using Distributed Sensing, *Proceedings of the 40th IEEE Conference on Decision and Control*, Florida, pp. 4268-4273, 2001.

[13] Li, Y. F. and Chen, X. B., End-point Sensing and State Observation of a Flexible-link Robot, *IEEE/ASME Transactions on Mechatronics*, Vol. 6, pp. 351-356, 2001.

[14] Su, Z. and Khorasani, K., A Neural-network-based Controller for a Single-link Flexible Manipulator Using the Inverse Dynamics Approach, *IEEE Transactions on Industrial Electronics*, Vol. 48, pp. 1074-1086, 2001.

[15] Landau, I., Langer, J., Rey, D. and Barnier, J., Robust Control of a 360^0 Flexible Arm Using the Combined Pole Placement/Sensitivity Function Shaping Method. IEEE Transactions on Control systems Technology, Vol. 4, pp. 369-383, 1996.

[16] Goh, S. P., Plummer, A. R. and Brown, M. D., Digital Control of a Flexible Manipulator. *In Proceedings of the American Control Conference*, Chicago, pp. 2205-2209, 2000.

[17] Oakley, C. M. and Cannon, R., Anatomy of an Experimental Two-link Flexible Manipulator under End-point Control. *Proceedings of the IEEE 29th Conference on Decision and Control*, Honolulu, pp.507-513, 1990.

[18] Bossert, D., Lyu, -L. and Vagners, J., Experiment Evaluation of Robust Reduced-order Hybrid Position/Force Control on a Two-link Manipulator, *Proceedings of the IEEE International Conference on Robotics and Automation*, Minneapolis, pp. 2573-2578, 1996.

[19] Romano, M., Agrawal, B. and Bernelli-zazzera, F., Experimants on Command Shaping Control of a Manipulator with Flexible Links, *Journal of Guidance, Control and Dynamics*, Vol. 25, pp.232-239, 2002.

[20] Kino, M., Goden, T., Murakami, T. and Ohnishi, K., Reaction Torque Feedback Based Vibration Control in Multi-degrees of Freedom Motion System, Proceedings of the 24th Annual Conference of the Industrial Electronics Society of the IEEE, Germany, Vol. 3, pp.1807-1811, 1998.

[21] Lin, I. –C. and Fu, L. –C. Adaptive Hybrid Force/Position Control of a Flexible Manipulator for Automated Deburring with On-line Cutting Trajectory Modification, *Proceedings of the IEEE International Conference on Robotics and Automation*, Belgium, pp. 818-825, 1998.

[22] Cheong, J., Chung, W. and Youm, Y., Fast Suppression of Vibration for Multi-link Flexible Robots Using Parameter Adaptive Control, *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Hawaii, pp. 913-918, 2001.

[23] Xu, W., Zhang, X. and Nair, S., Dynamic Modelling of a Two-link Flexible Manipulator Incorporating Experimental Data, *Proceedings of the American Control*

Conference, Texas, pp. 4508-4513, 2001.

[24] Benosman, M. and LE VEY, G., Joint Trajectory Tracking for Planar Multi-link Flexible Manipulator: Simulation and Experiment for a Two-link Flexible Manipulator. *Proceedings of the IEEE International Conference on Robotics and Automation*, Washington DC, pp. 2461-2466, 2002.

[25] Miyabe, T., Yamano, M., Konno, A. and Uchiyama, M., An Approach toward a Robust Object Recovery with Flexible Manipulator, *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Hawaii, pp. 907-912, 2001.

[26] Wang, Z., Zeng, H., Ho, D. and Unbenauen, H., Multiobjective Control of a Four-link Flexible Manipulator: A Robust H-infinity Approach, *IEEE Transactions on Control Systems Technology*, Vol. 10, pp.866-875, 2002.

[27] Mori, W., Ueda, J. and Ogasawara, T., A 1-Dof Dynamics Pitching Robot that Independently Controls Velocity, Angular Velocity and Direction of a Ball, *Advanced Robotics*, pp.921-942, 2010.

[28] Senoo, T., Namiki, A. and Ishikawa, M., High-Speed Batting Using a Multi-Jointed Manipulator, *International Conference on Robotics and Automation*, New Orleans, pp.1191-1196, 2004.

[29] Trunckenbrodt, A., Truncation Problems in the Dynamics and Control of Flexible Mechanical Systems, *Proceedings of 8th Triennial World Congress of IFAC*, Vol. 7, pp. 1909-1914, 1982.

[30] Judd, R. P. and Falkenburg, D. R., Dynamics of Non-rigid Articulated Robot Linkages, *IEEE Transactions*, Vol. AC-30, pp. 499-502, 1985.

[31] Kanoh, H. and Lee, H. G., Vibration Control of One-link Flexible Arm, *Proceedings of 24th IEEE Conference on Decision and Control*, Vol. 3, pp. 1172-1177,

1985.

[32] Geradin, M., Robot, G. and Bernardin, C., Dynamic Modeling of Manipulators with Flexible Members, *Advanced Software in Robotics*, pp. 27, 1984.

[33] Usoro, P. B., Mahil, S. S. and Nadir, R., A Finite Element/Lagrange Approach to Modeling Lightweight Flexible Manipulator, *Winter Annual Meeting of ASME*, 1984.

[34] Sunada, W. and Dubowsky, S., On the Dynamic Analysis and Behavior of Industrial Robotic Manipulator with Elastic Members, *Transactions of ASME*, Vol. 105, 1985.

[35] Sakawa, Y., Masuno, F. and Fukushima, S., Modeling and Feedback Control of a Flexible Arm, *Journal of Robotic System*, Vol. 2, No. 4, pp. 453-472, 1985.

[36] Fukuda, T., Flexibility Control of Elastic Robotic Arm, *Journal of Robotic System*, Vol. 2, No. 1, pp. 73-88, 1985.

[37] Vukobratovic, M. and Potokonjak, V., Dynamics of Manipulation Robots: Theory and Application, *Springer-Verlag*, 1982.

[38] Naganathan, G. and Soni, A. H., Nonlinear Flexibility Studies for Spatial Manipulators, *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 373-378, 1986.

[39] Dado, M. and Soni, A. H., A Generalized Approach for Forward and Inverse Dynamics of Elastic Manipulators, *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 359-364, 1986.

[40] Khorrami, F. and Ozguner, U., Perturbation Methods in Control of Flexible Link Manipulators, Proceedings of the *IEEE International Conference on Robotics and Automation*, pp. 310-315, 1988.

[41] Book, W. J. and Majette, M., Controller Design for Flexible Distributed

Parameter Mechanical Arms Via Combined State-space and Frequency Domain Techniques. *Transactions of ASME: Journal of Dynamic Systems, Measurement and Control*, Vol. 105, pp. 245-254, 1983.

[42] Young, K. –S., Book, W. J. and Siciliano, B., Direct Adaptive Control of a One-link Flexible Arm with Tracking, *Journal of Robotic Systems*, Vol. 6, pp. 663-680, 1989.

[43] Irie, T. and Kobayashi, Y., General Theory of Mechanical Vibration, *Asakura Press, Third Edition*, 2006.

[44] Spall, J. C., Introduction to Stochastic Search and Optimization, Wiley & Sons Ltd, 2003.

[45] Luenberger, D. G., Linear and Nonlinear Programming, Addison-Wesley, 1989.

[46] Torn, A. and Zilinskas, A., Global Optimization, Springer, 1989.

[47] Horst, R. and Pardalos, P. M., Handbook of Global Optimization, Kluwer Academic Publishers, 1995.

[48] Polak, E., Optimization: Algorithms and Consistent Approximations, Springer, 1997.

[49] Horst, R. and Tuy, J., Global Optimization: Deterministic Approaches, Spriger-Verlag, 2003.

[50] Nocedal, J. and Wright, S. J., Numerical Optimization, Springer, 2006.

[51] Zhigljavsky, A. and Zilinskas, A., Stochastic Global Optimization, Springer, 2008.

[52] Chiang, A. C., Elements of Dynamic Optimization, Waveland Press, 1991.

[53] Deb, K., Multi-objective Optimization Using Evolutionary Algorithms, Wiley

& Sons Ltd, 2001.

[54] Branke, J., Evolutionary Optimization in Dynamic Environments, Kluwer Academic Publishers, 2002.

[55] Ehrgott, M. and Gandibleux, X., Multiple Criteria Optimization: State of the Art Annotated Bibliographic Surveys, Kluwer Academic Publishers, 2002.

[56] Floudas, C. A., Nonlinear and Mixed-integer optimization: Fundamentals and Applications, Oxford University Press, 1995.

[57] Boros, E. and Hammer, P. L., Discrete Optimization: the State of the Art, Elsevier Science, 2003.

[58] Dixon, L. C. W. and Szego. G. P., The Global Optimization Problem: An Introduction, *Towards Global Optimization 2*, pp. 1-15, 1987.

[59] Archetti, F. and Schoen, F., A Survey on the Global Optimization Problem: General Theory and Computational Approaches, *Annals of Operations Research*, pp. 87-110, 1884.

[60] Friedberg, R. M., A Learning Machine, *IBM Journal part I*, Vol. 2, pp. 2-13, 1958.

[61] Fogel, L. J., Autonomous Automata, Industrial Research, pp. 14-19, 1962.

[62] Fogel, D. B. and Fogel, L. J., Route Optimization through Evolutionary Programming, *In Proceedings of the 22nd Asilomar Conference on Signals, Systems and Computers*, Vol. 2, pp. 679-680, 1988.

[63] Fogel, D. B., An Evolutionary Approach to the Travelling Salesman Problem, *Biological Cybernetics, 60*(2), pp. 139-144, 1988.

[64] Fogel, D. B., Fogel, L. J. and Porto, V. W., Evolving Neural Networks, *Biological Cybernetics, 63*(6), pp. 487-493, 1990.

[65] McDonnell, J. R., Andersen, B. D., Page, W. C. and Pin, F, Mobile Manipulator Configuration Optimization Using Evolutionary Programming, *In Proceedings of the 1st Annual Conference on Evolutionary Programming*, pp. 52-62, 1992.

[66] Mitchell, M., An introduction to Genetic Algorithm, Cambridge, MIT Press, 1996.

[67] Falkenauer, E., Genetic Algorithms and Grouping Problems, Wiley & Sons Ltd, 1997.

[68] Michalewicz, Z., Genetic Algorithms Adds Data Structure Equals Evolution Programs, Springer, 1999.

[69] Vose, M. D., The Simple Genetic Algorithm: Foundations and Theory, MIT Press, 1999.

[70] Gorberg. D. E., The Design of Innovation: Lessons From and For Competent Genetic Algorithms Reading, Addison-Wesley, 2002.

[71] Rechenberg, I., Cybernetic Solution Path of an Experimental Problem, Royal Aircraft Establishment Library, Translation 1122, 1965.

[72] Schwefel, H. -P., Kybernetische Evolution als Strategie der experimentellen Forschung in der Stromungstechnik, Dipl.-Ing. thesis, Technical University of Berlin, Hermann Fottinger Institute for Hydrodynamics, Germany, 1965.

[73] Bienert, P., Aufbau einer optimierungsautomatik fur drei Parameter, Dipl,-Ing, thesis, Technical University of Berlin, Institute of Measurement and Control Technology, Germany.

[74] Beyer, H. –G., The Theory of Evolution Strategies, Springer, 2001.

[75] Hansen, N. and Ostermeier, A., Completely Derandomized Self-adaptation in

Evolution Strategies, *Evolutionary Computation*, *9* (1), pp. 159-195, 2001.

[76] Beyer, H, -G. and Schwefel, H. -P., Evolution Strategies: A Comprehensive Introduction, *Natural Computing, 1* (1), pp. 3-52, 2002.

[77] Beni, G. and Wang, J., Swarm Intelligence in Cellular Robotic Systems, *Robotics and Biological Systems: Towards a New Bionics, NATO ASI Series, Series F: Computer and System Science*, Vol. 102, pp. 703-712, 1989.

[78] Dorigo, M., Optimization, Learning and Natural Algorithms, Ph.D. thesis, Politecnico di Milano, Italy, 1992.

[79] Bonabeau, E., Dorigo, M. and Theraulaz, G., Swarm Intelligence: From Natural to Artificial Systems, Oxford University Press, 1999.

[80] Dorigo, M. and Stutzle, T., Ant Colony Optimization, MIT Press, 2004.

[81] Bishop, J. M., Stochastic Searching Network, In proceedings of the 1st IEEE Conference on Artificial Neural Networks, pp. 329-331, 1989.

[82] Bishop, J, M. and Torr, P., The stochastic Search Network, *Neural Networks for Images, speech and Natural Language*, pp. 370-387, 1992.

[83] Grech-Cini, H. J. and McKee, G. T., Locating the Mouth Region in Images of Human Faces, *In Proceedings of SPIE- The International Society for Optical Engineering, Sensor Fusion*, Boston, USA, pp. 458-465, 1993.

[84] Beattie, P. D. and Bishop, J. M., Self-localization in the “Scenario” autonomous Wheelchair, *Journal of Intelligence and Robotic Systems*, Vol. 22, pp. 255-267, 1998.

[85] Whitaker, R. M. and Hurley, S., An Agent Based Approach to Site Selection for Wireless Networks, *In Proceedings of the 2002 ACM Symposium on Applied Computing*, pp. 574-577, 2002.

[86] Kennedy, J. and Eberhart, R. C., Particle Swarm Optimization, *In Proceedings of the IEEE International Conference on Neural Networks*, Australia, pp.1942-1948, 1995.

[87] Shi, Y. and Eberhart, R. C., A Modified Particle Swarm Optimizer, *In Proceedings of the 1998 IEEE International Conference on Evolutionary Computation*, pp.69-73, 1998.

[88] Solihin, M. I., Akmeliawati, W. and Akmeliawati, R., PSO-based optimization of state feedback tracking controller for a flexible link manipulator, *International Conference of Soft Computing and Pattern Recognition*, pp. 72-76, 2009.

[89] Shayeghi, A., Shayeghi, H. and Kalasar, H. E., Application of PSO technique for seismic control of tall building, *World Academy of Science, Engineering and Technology*, Vol.28, pp. 851-858, 2009.

[90] Nadia Nedjah, Leandro dos Santos Coelho and Luiza de Macedo de Mourelle, Multi-Objective Swarm Intelligent Systems (2009), Springer.

[91] Omkar SN, Mudigere D, Naik GN and Gopalakrishnan S., Vector evaluated particle swarm optimization (VEPSO) for multi-objective design optimization of composite structures. *Computers and Structures* 86: 1-14, 2008.

[92] Parsopoulos KE and Vrahatis MN, *Particle Swarm Optimization and Intelligence: Advances and Applications*. Information Science Reference, 2009.

[93] Vlachogiannis JG and Lee KY, Multi-objective based on parallel vector evaluated particle swarm optimization of optimal steady-state performance of power systems. *Expert Systems with Applications* 36:10802-10808, 2009.

[94] Jiang H, Kwong CK, Chen Z and Ysim YC, Chaos particle swarm optimization and T-S fuzzy modeling approaches to constrained predictive control.

Expert Systems with Applications 39: 194-201, 2012.

[95] Alatas B, Akin E. and Ozer AB, Chaos embedded particle warm optimization algorithms. *Chaos, Solutions and Fractals* 40: 1715-1734, 2009.

[96] Reyes-Sierra M and Coello CAC, Multi-objective particle swarm optimizers: A survey of the state-of-the-art. *International Journal of Computational Intelligence Research* 2: 287-308, 2006.

[97] Chen, X., Guo, S. and Fukuda, T., Vibration Suppression Control of Flexible Arms Using Sliding Mode Method, *In Proceedings of the 1999 IEEE International Conference on Control Applications*, Vol. 1, pp.309-314, 1999.

[98] Li, P. and Du, X., A Fuzzy Logic Learning Control for Vibration Suppression of Manipulator Robot Systems, *In Proceedings of 2004 International Conference on Intelligent Mechatronics and Automation*, pp. 327-331, 2004.

[99] Ahmad, M. A., Vibration and Input Tracking Control of Flexible Manipulator Using LQR with Non-collocated PID Controller, *Second UKSIM European Symposium on Computer Modeling and Simulation*, pp. 40-45, 2008.

[100] Singer, N. C. and Seering, W. P., Preshaping Command Inputs of Reduce System Vibration, *Journal of Dynamic Systems, Measurement and Control*, Vol. 112, pp. 76-82, 1990.

[101] Singhose, W., Seering, W. and Singer, N., Shaping Inputs to Reduce Vibration: A Vector Diagram Approach, *IEEE Conference on Robotics and Automation*, pp. 922-927, 1990.

[102] Singhose, W., Seering, W. and Singer, N., Residual Vibration Reduction Using Vector Diagrams to Generate Shaped Inputs, *Journal of Mechanical Design*, Vol. 116, pp.654-659, 1994.

[103] Yin, H. B., Controller Design Considering the Nonlinearity of Flexible Robotic Manipulator Under Fast Motion, Ph.D. thesis, Sapporo, Japan, 2011.

[104] Lopez-Linares, S., Konno, A. and Uchiyama, M., Vibration Suppression Control of 3D Flexible Robots Using Velocity Inputs, *Journal of Robotics Systems*, 14-12, pp.823-837, 1997.

[105] Alam, M. S. and Tokhi, M. O., Hybrid Fuzzy Logic Control with Genetic Optimization for a single-link Flexible Manipulator, *Engineering Applications of Artificial Intelligence 21*, pp.858-873, 2008.

[106] Alam, M. S. and Tokhi, M. O., Designing Feedforward Command Shapers with Multi-objective Genetic Optimization for Vibration Control of a Single-link Flexible Manipulator, *Engineering Applications of Artificial Intelligence 21*, pp.229-246, 2008.

[107] Hoshino, Y. and Kobayashi, Y., Shock and Vibration Control of a Golf-Swing Robot at Impacting the Ball, *Journal of System Design and Dynamics*, Vol.2, No.5, pp.1069-1080, 2008.

[108] Yin, H. B., Yukinori Kobayashi, Y., Hoshino, Y. and Emaru, T., Modeling and Vibration Analysis of Flexible Robotic Arm under Fast Motion in Consideration of Nonlinearity, *Journal of System Design and Dynamics*, Vol. 5, No. 5, pp.909-924, 2011.

Acknowledgements

At the beginning, I want to show my grateful thankfulness to my supervisor Professor Yukinori Kobayashi. For Kobayashi Sensei, I could own the chance to study and do the research in this laboratory and it's my great honor to be his student. For the period of my master and doctor course, I received the good guidance, research experience, and methodology of research thinking from Kobayashi Sensei. They are the priceless treasure for my future research activities. I had obtained many help and concern from him not only on the field of research activities, but also on the aspect of my daily life. During my doctor course, I had a traffic accident and was taken to the hospital with serious injured skull and neck. Thanks to my Sensei, I could get a contact with my family, tackle with many documents and problems for this traffic accident finish the procedures of entering Japan for my mum in a short time and many other troublesome. So again, I want to show my grateful appreciate to Professor Yukinori Kobayashi and thanks a million for his valuable time, suggestion and kind attention to me.

Of course, I hope to show my feelings of appreciation to my beloved parents. They give me birth to this world, help me growing up, teach me how to be responsible man, and support my studying abroad as much as they can without any requirement for return. Especially when I had an accident and was sent to the hospital in Japan, my mother left all the things she was working at and came to my side as soon as she could. When I had some problems on my Ph.D. course and needed to delay my graduation date, my parents didn't complain to me. They encouraged me and gave me suggestions to help me go out of the blue mood.

Also show my thanks to the Associate Professor Takanori Emaru. Thanks for his attention and good care for me. Then I hope to my special thankfulness to former Assistant Professor Yohei Hoshino. Thanks so much for checking my paper draft and giving the valuable suggestions for it during the busy period for him. I learned many important points on research works from him.

Thank you to the members of the lab of Robotics and Dynamics and my friends who gave me a lot of help and enriched my life in Sapporo, Hokkaido. It is my pleasure to study and have the school life with the fifth floor member, Koichi Imaoka, Ravankar Ankit, Jixin Lv, Shuhei Yoshida, Ravankar Abhijeet, Chattunyakit Sarun, master course students, and undergraduate members. Well, they are many people I hope to show my thankfulness here and Thank you so much. Also special thanks to Imaoka san, thank you so much for encouraging and helping me during my tough time.

At last of this paper, I gratefully acknowledge the funding sources and support. The China Scholarship Council supports me for four years and my study is also provided the support by Hokkaido University.