2013 Second International Conference on Robot, Vision and Signal Processing

The Implementation of Smoothing Robust Control for a Delta Robot

Hui-Hung Lin
Energy and Agile System Department
Metal Industries Research & Development Centre
Taiwan, R.O.C.
hlin@mail.mirdc.org.tw

Yuan-Hung Ta
Institute of Mechanical and Precision Engineering
National Kaohsiung University of Applied Sciences
Kaohsiung City, Taiwan, R.O.C.
1100303107@kuas.edu.tw

Chao-Shu Liu
Department of Mechanical Engineering
National Kaohsiung University of Applied Sciences
Kaohsiung City, Taiwan, R.O.C.
joshualiu@kuas.edu.tw

***Abstract*—The implementation of smoothing robust control for a delta robot has been explored in this paper. Delta robots are usually used for the high speed pick-and-place or highly precise applications in the manufacturing factories. For the inherent mechanical structural constraints of delta robots, they would make the motion control more complex than serial manipulators. A delta robot control system is a highly coupling system and easy to induce position errors or chatter due to the effects of other attached limbs. In order to solve the serious problem, this paper implemented a smoothing robust control method, MDDS, to guarantee the smoothly and robustly dynamic behavior as we designed. The method has the capabilities of dynamics prediction and disturbance estimation, and then outputs the control efforts to deform the dynamic manifold of the controlled plant into the desired manifold. In the hardware structure, the paper uses a self-designed kernel development board with PC-based HMI to execute the experiments. The control methods are designed by the combined kernel of DSP and FPGA. We also develop the PID control method in the development board to make comparisons. Moreover, the validation of this control architecture for a delta robot is verified successfully through the results of experiments.**

Keywords-Delta Robot, Smoothing Robust, Manifold Deformation

I. INTRODUCTION

Robots have been applied in automation industries for many years. For their specific mechanical structures, mechanical design and how to control stably and precisely are still challenging problems. In 1980, a kind of delta mechanism addressed by Clavel at Swiss Lausanne Engineering Institute was developed as industrial robots and applied in manufacturing or packaging lines. Especially in medicine area, for the requirements of highly accurate positioning and chattering free for human operations, the kind of delta robots is applied in surgical operation. [1]

Delta robots are a kind of parallel robots with many advantages comparing to the serial robots, such as high speed, high stiffness, and high accuracy. However, to achieve a higher accuracy, the static and dynamic behavior of the system must be well understood. In addition, parallel robots are highly coupled systems and very easy to be disturbed by other attached limbs for asynchronous convergence and then to induce chatters. In order to solve the serious problems, various robot control methods have been developed in lately years, such as CTC (Calculation Torque Control), Fuzzy Control, Adaptive Control, and Learning Control, etc. [2]-[5] Those control methods are often used to attenuate the effects of disturbances which are induced by loading variations or system dynamics, and always need a large number of mathematical equations to describe the disturbances and uncertainties. However, it is very hard to obtain the exact model of disturbances and uncertainties for such complex dynamic systems.

After considering various approaches for the implementation of robust controls, Manifold Deformation Design Scheme (MDDS) provided by Liu in 1998 with the characteristic of smooth robustness under the preset responding bandwidth of motion based on the topological analysis about system dynamics is very suitable for this kind of robot systems. The scheme is a kind of mapping technology to perform the system dynamics in the state space, and then to analyze the dynamic behavior and obtain the control efforts directly. Therefore, by the way of setting the desired manifold to be the reference of the system dynamic behavior in the state space, and calculating and compensating the difference between the current state and the desired state for control efforts, then the control efforts will force the behavior of the controlled system to approach the desired manifold instantly during each sampling interval without the worry of parameters selection. [6] The related literatures for the simulations and experiments of robot controls were also explored by Liu. [7]-[8]

For validation, some experiments of motion controls for a delta robot are performed successfully in the paper. In the hardware structure, the main controller is implemented by the combined kernel of FPGA and DSP, which is used to realize the motion controls of a delta robot. The FPGA is used to calculate and translate the feedback signals and output the control efforts synchronously for each joints, and the calculations of MDDS and PID are executed by the DSP. After various experiments and tests, MDDS has excellent control performances obviously compared with PID control scheme.

We implement MDDS on the self-designed kernel development board and perform motion controls for a delta robot to achieve the following objectives:

(a) To construct a servo control kernel for the various applications of delta robots.

978-1-4799-3184-2/13 $31.00 © 2013 IEEE
DOI 10.1109/RVSP.2013.11

(b) To solve the asynchronous problems induced by time-sharing or digital delay of hardware through the design and programming of hardware architecture.

(c) To solve the practical problems of coupling and disturbance without the worry of parameters selection by the way of the implementation of smoothing robust control algorithm, MDDS.

II. System Description

A delta robot has been presented by Clave as three-degree-of-freedom parallel robot, is dedicated to high speed applications. The structure of a delta robot is shown in Fig 1 and consists of moving platform and a fixed base, connected by several linkages.

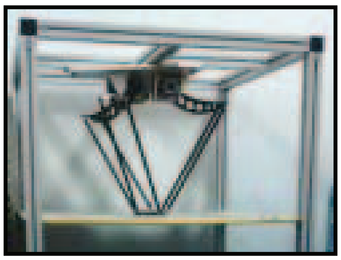

Fig 1. A delta robot

2.1 Inverse Kinematics

We consider the coordinate systems of a delta robot and the axis angles of the i'th limb are shown in Fig 2. A reference coordinate system O-xyz connects to the central point O on the fixed plane, in which x axis and y axis locate on the fixed plane, and z axis points up orthogonally. Another coordinate system A_i-$x_iy_iz_i$ connects to the point Ai on the fixed plane, in which the direction of x_i axis is the extension of $\overrightarrow{OA_i}$, the direction of y_i axis is from A_i to the rotational axis, the direction of z_i axis is parallel to z axis and angle ψ_i is measured from x axis to x_i axis.

In which P is the vector of the central point on the moving plane, θ_{1i} is from xi axis to $\overrightarrow{A_iB_i}$, θ_{2i} is the intersection of the extension of $\overrightarrow{A_iB_i}$ and the plane of x_i-z_i, θ_{3i} is the angle from y_i to $\overrightarrow{B_iC_i}$.

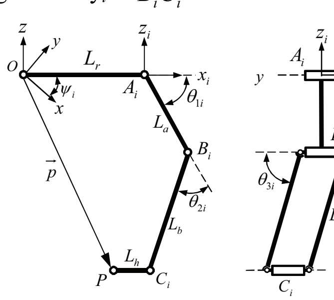

Fig 2. The dimensional parameters of the i'th limb

The closed loop for each linkage is written as follow:

$$\overrightarrow{A_iB_i} + \overrightarrow{B_iC_i} = \overrightarrow{OP} + \overrightarrow{PC_i} - \overrightarrow{OA_i} \tag{1}$$

Modify (1) on coordinate system, A_i-$x_iy_iz_i$, we can get:

$$L_a\begin{bmatrix}\cos\theta_{1i}\\ 0\\ \sin\theta_{1i}\end{bmatrix} + L_b\begin{bmatrix}\sin\theta_{3i}\cos(\theta_{1i}+\theta_{2i})\\ \cos\theta_{3i}\\ \sin\theta_{3i}\sin(\theta_{1i}+\theta_{2i})\end{bmatrix} = \begin{bmatrix}c_{xi}\\ c_{yi}\\ c_{zi}\end{bmatrix} \tag{2}$$

In which

$$\begin{bmatrix}c_{xi}\\ c_{yi}\\ c_{zi}\end{bmatrix} = \begin{bmatrix}\cos\psi_i & \sin\psi_i & 0\\ -\sin\psi_i & \cos\psi_i & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}p_x\\ p_y\\ p_z\end{bmatrix} + \begin{bmatrix}L_h - L_r\\ 0\\ 0\end{bmatrix} \tag{3}$$

C_i is the position relative to the coordinate system A_i-$x_iy_iz_i$, L_a and L_b are the lengths of $\overrightarrow{A_iB_i}$ and $\overrightarrow{B_iC_i}$, respectively. $\vec{p} = \begin{bmatrix}p_x & p_y & p_z\end{bmatrix}^T$ is the position of P relative to the coordinate system, O-xyz.

Angle θ_{3i} can be obtained from (2),

$$\theta_{3i} = \cos^{-1}\frac{c_{yi}}{L_b} \tag{4}$$

Then, θ_{2i} can be determined by the sum of square c_{xi}, c_{yi}, and c_{zi}.

$$2L_aL_b\sin\theta_{3i}\cos\theta_{2i} + L_a^{\ 2} + L_b^{\ 2} = c_{xi}^{\ 2} + c_{yi}^{\ 2} + c_{zi}^{\ 2} \tag{5}$$

Therefore,

$$\theta_{2i} = \cos^{-1}\left(\frac{c_{xi}^{\ 2} + c_{yi}^{\ 2} + c_{zi}^{\ 2} - L_a^{\ 2} - L_b^{\ 2}}{2L_aL_b\sin\theta_{3i}}\right) \tag{6}$$

Finally, θ_{1i} can be also obtained form (2).

2.2 Robot Dynamics

From Fig 1, we have known that the Jacobian matrix J of robot expresses the relation between the input joint rates and the output velocity of the end-effector as follows (4):

$$\dot{\mathrm{P}} = J\dot{\Theta}, \tag{7}$$

where $\dot{\mathrm{P}} = \begin{bmatrix}v_{px} & v_{py} & v_{pz}\end{bmatrix}^T$ is the velocity of the end-effector, $\dot{\Theta} = \begin{bmatrix}\dot{\theta}_{11} & \dot{\theta}_{12} & \dot{\theta}_{13}\end{bmatrix}^T$ is the input joint velocity vector.

In the case of the delta robot, the constraint equations in each limb can be chosen as

$$\Phi_i = \begin{bmatrix}p_x\\ p_y\\ p_z\end{bmatrix} - \begin{bmatrix}\cos\psi_i & -\sin\psi_i & 0\\ \sin\psi_i & \cos\psi_i & 0\\ 0 & 0 & 1\end{bmatrix} \cdot\left(\begin{bmatrix}L_r - L_h\\ 0\\ 0\end{bmatrix} + L_a\begin{bmatrix}\cos\theta_{1i}\\ 0\\ -\sin\theta_{1i}\end{bmatrix}\right) \quad i = 1,2,3. \tag{8}$$

The Jacobian matrix of the delta robot can be obtained

$$J=-\begin{bmatrix}\Phi_1{}^T\\ \Phi_2{}^T\\ \Phi_3{}^T\end{bmatrix}^{-1}\begin{bmatrix}\Phi_1{}^T R_1 & 0 & 0\\ 0 & \Phi_2{}^T R_2 & 0\\ 0 & 0 & \Phi_3{}^T R_3\end{bmatrix}, \tag{9}$$

where

$$R_i=\begin{bmatrix}\cos\psi_i & -\sin\psi_i & 0\\ \sin\psi_i & \cos\psi_i & 0\\ 0 & 0 & 1\end{bmatrix}\cdot L_a\begin{bmatrix}\sin\theta_{1i}\\ 0\\ -\cos\theta_{1i}\end{bmatrix}\quad i=1,2,3.$$

Let $\ddot{\mathrm{P}}=\begin{bmatrix}a_{px} & a_{py} & a_{pz}\end{bmatrix}^T$ is the acceleration of the end-effector, $\dot{\Theta}=\begin{bmatrix}\dot\theta_{11} & \dot\theta_{12} & \dot\theta_{13}\end{bmatrix}^T$ is the input joint velocity vector, $\ddot{\Theta}=\begin{bmatrix}\ddot\theta_{11} & \ddot\theta_{12} & \ddot\theta_{13}\end{bmatrix}^T$ is the input joint acceleration vector, therefore, we can get the relation as bellow:

$$\ddot{\mathrm{P}}=-\begin{bmatrix}\Phi_1{}^T\\ \Phi_2{}^T\\ \Phi_3{}^T\end{bmatrix}^{-1}\left(\begin{bmatrix}\dot\Phi_1{}^T\\ \dot\Phi_2{}^T\\ \dot\Phi_3{}^T\end{bmatrix}J+\right.$$

$$\left.\begin{bmatrix}\dot\Phi_1{}^T R_1+\Phi_1{}^T\dot R_1 & 0 & 0\\ 0 & \dot\Phi_2{}^T R_2+\Phi_2{}^T\dot R_2 & 0\\ 0 & 0 & \dot\Phi_3{}^T R_3+\Phi_3{}^T\dot R_3\end{bmatrix}\right)\dot\Theta+J\ddot\Theta' \tag{10}$$

where

$$J=-\begin{bmatrix}\Phi_1{}^T\\ \Phi_2{}^T\\ \Phi_3{}^T\end{bmatrix}^{-1}\begin{bmatrix}\Phi_1{}^T R_1 & 0 & 0\\ 0 & \Phi_2{}^T R_2 & 0\\ 0 & 0 & \Phi_3{}^T R_3\end{bmatrix},$$

$$\Phi_i=\begin{bmatrix}p_x\\ p_y\\ p_z\end{bmatrix}-\begin{bmatrix}\cos\psi_i & -\sin\psi_i & 0\\ \sin\psi_i & \cos\psi_i & 0\\ 0 & 0 & 1\end{bmatrix}\cdot\left(\begin{bmatrix}L_r-L_h\\ 0\\ 0\end{bmatrix}+L_a\begin{bmatrix}\cos\theta_{1i}\\ 0\\ -\sin\theta_{1i}\end{bmatrix}\right),\quad i=1,\ 2,\ 3.$$

$$\dot\Phi_i=\begin{bmatrix}\dot p_x\\ \dot p_y\\ \dot p_z\end{bmatrix}+\begin{bmatrix}\cos\psi_i & -\sin\psi_i & 0\\ \sin\psi_i & \cos\psi_i & 0\\ 0 & 0 & 1\end{bmatrix}\cdot L_a\begin{bmatrix}\cos\theta_{1i}\\ 0\\ -\sin\theta_{1i}\end{bmatrix}\dot\theta_{1i},\quad i=1,\ 2,\ 3.$$

$$R_i=\begin{bmatrix}\cos\psi_i & -\sin\psi_i & 0\\ \sin\psi_i & \cos\psi_i & 0\\ 0 & 0 & 1\end{bmatrix}\cdot L_a\begin{bmatrix}\sin\theta_{1i}\\ 0\\ -\cos\theta_{1i}\end{bmatrix},\quad i=1,\ 2,\ 3.$$

$$\dot R_i=\begin{bmatrix}\cos\psi_i & -\sin\psi_i & 0\\ \sin\psi_i & \cos\psi_i & 0\\ 0 & 0 & 1\end{bmatrix}\cdot L_a\begin{bmatrix}\sin\theta_{1i}\\ 0\\ -\cos\theta_{1i}\end{bmatrix}\dot\theta_{1i},\quad i=1,\ 2,\ 3.$$

The dynamics of a robot, usually, can be described as the following nonlinear differential equations:

$$\mathrm{T}=\mathrm{I}(\Theta)\ddot\Theta+\mathrm{N}(\Theta,\dot\Theta) \tag{11}$$

and

$$\mathrm{N}(\Theta,\dot\Theta)=\mathrm{V}(\Theta,\dot\Theta)+\mathrm{G}(\Theta)+\mathrm{F}(\Theta,\dot\Theta)+\mathrm{T_d}, \tag{12}$$

in which Θ is a p_{x1} vector of joint variables, T is a p_{x1} vector of input torque, $\mathrm{I}(\Theta)$ is a pxp symmetric and positive definite matrix of inertia, $\mathrm{V}(\Theta,\dot\Theta)$ is a pxp matrix of centrifugal and Coriolis terms, $\mathrm{G}(\Theta)$ is a px1 vector of gravitational terms, $\mathrm{F}(\Theta,\dot\Theta)$ is a p_{x1} vector of frictional terms, and T_d is a p_{x1} vector of any unknown but bounded disturbances.

Because the robot manipulator is actuated by some actuators equipped with gear trains, the dynamics of actuators are considered as below:

$$\mathrm{T_a}=\mathrm{G_{ear}}^{-1}\cdot\mathrm{T}+\mathrm{I_a}\cdot\mathrm{G_{ear}}\cdot\ddot\Theta, \tag{13}$$

and

$$\mathrm{T_a}=\mathrm{D}\cdot\mathrm{U}, \tag{14}$$

in which T_a is a p_{x1} vector of actuators' input torque, Gear = diag [g_1 ,g_2 ,…,g_p] is a pxp diagonal matrix of gear ratio, I_a = diag [I_{a1} ,I_{a2} ,…,I_{ap}] is a pxp diagonal matrix of actuators' inertia, D = diag [d_1 ,d_2 ,…,d_p] is a pxp diagonal matrix of input gains for voltage to torque, and U is a p_{x1} vector of input voltage commands for actuators.

Combining (11), (12), (13), and (14), the dynamic equations can be written as follows:

$$\mathrm{G_{ear}}\cdot\mathrm{D}\cdot\mathrm{U}=[\mathrm{I}(\Theta)+\mathrm{G_{ear}}\cdot\mathrm{I}]\ddot\Theta+\mathrm{N}(\Theta,\dot\Theta) \tag{15}$$

Let Θ_d and $\dot\Theta_d$ characterize the desired trajectory vector that is to be tracked, and x_1, x_2 denote the position and velocity tracking error vectors defined by $\mathrm{x_1}=\Theta_\mathrm{d}-\Theta$ and $\mathrm{x_2}=\dot\Theta_\mathrm{d}-\dot\Theta$, respectively. Then we write the state equations as follows.

$$\begin{aligned}\dot{\mathrm{x}}_1&=\mathrm{x}_2\\ \dot{\mathrm{x}}_2&=\ddot\Theta_\mathrm{d}-[\mathrm{I}(\Theta)+\mathrm{G_{ear}}\cdot\mathrm{I_a}\cdot\mathrm{G_{ear}}]^{-1}\cdot\mathrm{G_{ear}}\cdot\mathrm{D}\cdot\mathrm{U}\\ &\quad+[\mathrm{I}(\Theta)+\mathrm{G_{ear}}\cdot\mathrm{I_a}\cdot\mathrm{G_{ear}}]^{-1}\cdot\mathrm{N}(\Theta,\dot\Theta),\end{aligned} \tag{16}$$

in which

$$\mathrm{x}_1=\begin{bmatrix}x_{11}, x_{12},\cdots,x_{1p}\end{bmatrix}^T$$

Let $\mathrm{B}=-[\mathrm{I_n}+\mathrm{G_{ear}}\cdot\mathrm{I_a}\cdot\mathrm{G_{ear}}]^{-1}\cdot\mathrm{G_{ear}}\cdot\mathrm{D}$ and $\Delta\mathrm{B}=[\mathrm{I_n}+\mathrm{G_{ear}}\cdot\mathrm{I_a}\cdot\mathrm{G_{ear}}]^{-1}\cdot\mathrm{G_{ear}}\cdot\mathrm{D}-[\mathrm{I}(\Theta)+\mathrm{G_{ear}}\cdot\mathrm{I_a}\cdot\mathrm{G_{ear}}]^{-1}\cdot\mathrm{G_{ear}}\cdot\mathrm{D}$, we can simplify (16) and rewrite it as follows.

$$\begin{aligned}\dot{\mathrm{x}}_1&=\mathrm{x}_2\\ \dot{\mathrm{x}}_2&=\ddot\Theta_\mathrm{d}+(\mathrm{B}+\Delta\mathrm{B})\cdot\mathrm{U}+[\mathrm{I}(\Theta)\\ &\qquad+\mathrm{G_{ear}}\cdot\mathrm{I_a}\cdot\mathrm{G_{ear}}]^{-1}\cdot\mathrm{N}(\Theta,\dot\Theta)\end{aligned} \tag{17}$$

III. CONTROLLER INFRASTRUCTURE

3.1 System Identification

For practical applications, we need to use the open-loop identification to identify the servo motor which is applied in the delta robot firstly. Usually the servo motor system can be simplified as a first-order system as below:

$$G(s) = \frac{Y(s)}{R(s)} = \frac{b}{s+a} = \frac{b}{a} * \frac{1}{\frac{s}{a}+1}, \tag{18}$$

where R(s) is the input signal and the unit is voltage; Y(s) is the output velocity of motor and the unit is rad/sec.

We can observe the step response of the first-order system and get the parameters a and b from the two indexe

$$x_n^*(t+h) = -\sum_{i=1}^{n-1} c_i[x_i(t) + h \cdot x_{i+1}(t)] \tag{30}$$

Combining (29) and (30),

$$\begin{aligned}&\bar{b} \cdot u_{pr}(t) \cdot h + [\bar{b} \cdot u_{es}(t) + \ b(X(t),t) \cdot u(t) + g(X(t),t)] \cdot h \\ &= -\sum_{i=1}^{n-1} c_i[x_i(t) + h \cdot x_{i+1}(t)] - x_n(t)\end{aligned} \tag{31}$$

Then we can separate (31) as

$$u_{pr}(t) = -\{\sum_{i=1}^{n-1} c_i[x_i(t) + h \cdot x_{i+1}(t)] + x_n(t)\}/(\bar{b} \cdot h) \text{ and} \tag{32}$$

$$[\bar{b} \cdot u_{es}(t) + \ \Delta b(X(t),t) \cdot u(t) + g(X(t),t)] \cdot h = 0 \tag{33}$$

From manifold deformation analysis, we can find that

$$\begin{aligned}&[\bar{b} \cdot u_{es}(t-h) + \Delta b(X(t-h),t-h) \cdot u(t-h) + g(X(t-h),t-h)] \cdot h \\ &= x_n(t) - x_n^*(t)\end{aligned} \tag{34}$$

Combining (33) and (34), we can obtain

$$u_{es}(t) = u_{es}(t-h) - [x_n(t) - x_n^*(t)]/(\bar{b} \cdot h) \text{ and } u_{es}(0) = 0 \tag{35}$$

Therefore, the overall control effort has the form:

$$u(t) = -\frac{f(X(t))}{\bar{b}} - \{\sum_{i=1}^{n-1} c_i[x_i(t) + h \cdot x_{i+1}(t)] + x_n(t)\}/(\bar{b} \cdot h) + u_{es}(t)$$

$$u_{es}(t) = u_{es}(t-h) - [x_n(t) - x_n^*(t)]/(\bar{b} \cdot h) \text{ and } u_{es}(0) = 0 \tag{36}$$

Due to practical constraint, the control efforts must be bounded (worst case) and the sampling interval is not zero, so ill condition will never happen.

IV. EXPERIMENTAL VALIDATION

For the needs of experiments, we use a self-designed kernel development board to implement the MDDS control method and the trajectory path is produced by a PC.

Therefore, we design the path firstly and send the data to the controller instantly during one sampling interval through the communication interfaces and then the delta robot can be controlled by the kernel board simultaneously. Fig 4. is the experimental bench and Fig 5. is the self-designed kernel development board.

Fig 4. The experimental bench for a delta robot

Fig 5. The self-designed kernel development board

4.1 Experiment 1 - Point-to-Point Motion Control

In this case, we let each actuators run to -1 (rad) in 100ms. The parameters are shown in Table 1 and the control results are shown in Fig 6.

Table 1. Parameters for Point-to-Point control

Parameters	Setting Values
Input Command	Point-to-Point Step command
Command Function	$\theta_{d1} = \theta_{d2} = \theta_{d3} = \begin{cases} 0 & , \ t < 0.1\,\text{sec} \\ -1 & , \ t \geq 0.1\,\text{sec} \end{cases}$
Sampling Rate	1 KHz
Max control effort	± 10 volt
PID control parameter	$K_{P1} = K_{P2} = K_{P3} = 0.5$ $K_{I1} = K_{I2} = K_{I3} = 0$ $K_{D1} = K_{D2} = K_{D3} = 0.01$

parameters are shown in Table 2. The results are shown in Fig 8 ~ Fig 10 and very easy to be found that PID has bigger overshoot and position errors than MDDS.

Table 2. Parameters for Trajectory Tracking Control

Parameters	Setting Values
Initial point [x, y, z]	[X, Y, Z]= [0, 0, -335] (mm)
Final point [x, y, z]	[X, Y, Z] =[10, 10, -335] (mm)
Command Function	$P_d(t)=\begin{bmatrix} P_{dx} \\ P_{dy} \\ P_{dz} \end{bmatrix}=\begin{bmatrix} 5\times[1-\cos(2\pi\times t)] \\ 5\times[1-\cos(2\pi\times t)] \\ -335 \end{bmatrix}$ (mm)
Sampling Rate	1 KHz
Max control effort	$\pm$10 volt
PID parameter	$K_{P1}= K_{P2}= K_{P3}=0.8$ $K_{I1}= K_{I2}= K_{I3}=0.01$ $K_{D1}= K_{D2}= K_{D3}=0.002$
MDDS parameter	$c_1 = c_2 = c_3 = 20$

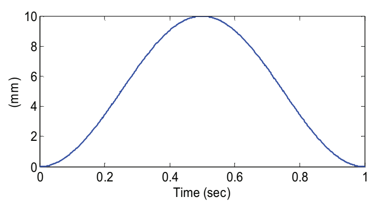

Fig 7. The acceleration and deceleration curve for straight line trajectory tracking control

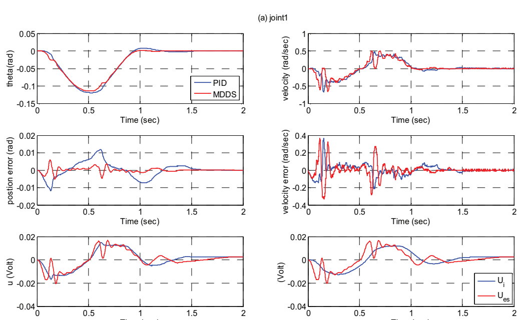

Fig 8. Trajectory Tracking – joint 1 Response

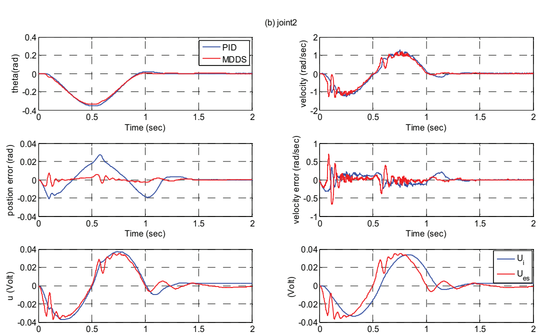

Fig 9. Trajectory Tracking – joint 2 Response

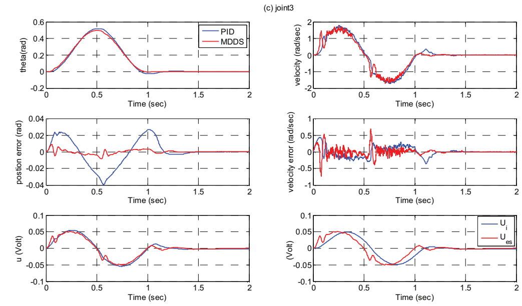

Fig 10. Trajectory Tracking – joint 3 Response

V. CONCLUSIONS

This paper presents a smoothing robust control infrastructure for a delta robot. We apply a self-designed kernel development board as the controller and implement MDDS control method to make the delta robot move smoothly and precisely. The validation of the control architecture is verified through experiments. The results show that MDDS has better control performances than PID. In addition, we can complete the design quickly without the worry of parameters selection by the way of system identification and MDDS control method. We just give a time constant