Assembly and Control of a Humanoid Torso with Variable Stiffness Actuators

eingereichte
Forschungspraktikum Protokoll
von

cand. ing. Jiabin Wu

geb. am 24.08.1988 wohnhaft in:
Felsennelkennanger 19
80937 München
Tel.: 0176 45640883

Lehrstuhl für
Dynamische Mensch-Roboter-Interaktion für
Automatisierungstechnik
Technische Universität München

Prof. Ph. D. Dongheui Lee

Betreuer:	M.Sc. Matteo Saveriano
Beginn:	Aug.3rd 2015
Zwischenbericht:	Oct.15th 2015
Abgabe:	Nov.17th 2015

In your final hardback copy, replace this page with the signed exercise sheet.

Abstract

The first task is to assemble the robot, then set the Denavit-Hartenberg parameters and calculate the forward and inverse kinematics for the arms; By the simulation and the realization on the robot, zero deflection control and position control through Matlab/Simulink are applied.The last step is to divide and modularize all the Simulink blocks into standard blocks, so as to support the following research on this robot.
Challenges on this topic lie on simulation/realization part and the division/modularization part. How to drive the arms to move to the predefined position is the major concern.

Zusammenfassung

Die erste Aufgabe dieser Forschung ist, den Roboter einzubauen,danach werden Denavit-Hartenberg Parameter eingesetzt und werden Vorwärts-Kinematik auch Rückwärts-Kinematik der Arm berechnet.Bei der Simulation -und Ausführung werden Zero Deflection Control und Position Control durch Matlab/Simulink verwendet.Letztendlich sollen alle der Simulink Blocks standardisiert werden, um die nachfolgende Aufgabe zu unterstützen.
Die Herausforderungen liegen meist in die Simulation -und Ausführung und die Standardisierung.Die Hauptaufgabe is, den Roboter an vorgegebene Position zu treiben.

Contents

Chapter 1

Introduction

The goal of this research practise is : drive the robot-arm to predefined position, in consideration of speed, stability and accuracy.
To reach this target, building a mathematic model for the arms in the first place is indispensable, because the mathematic model is the basics of applying all the algorithms. The second step is to develop the model algorithm,which will generate the values of the joint variables and theoretically drive the arms to the target point.The third step is the realization of the algorithm on the robot.Repeat the second and the third steps so as to satisfy the requirements: speed,stability and accuracy.
After the requirements are satisfied, all the Simulink block need to be standardized,without degrading the performace of the movements.

1.1 Problem Statement

The main problems come from different aspects:

- how to build a accurate model for the robot.
- what kind of algorithm should be used to control the robot.
- how to prevent the joints from over rotating in some special cases.
- by the block-standardizing, how to reduce the delay caused by blocks' communication.
- how to compensate the gravity

1.2 Related Work

[PCC11] proposes a method to transfer to robots several tasks demonstrated by the user through kinesthetic teaching and subsequently learned using a weighted combination of dynamical systems (DS).

[JTC11]achieves the stiffness regulation not through the control of the spring pretension (as in most of the existing variable stiffness joints) but by controlling the location of the spring elements.
[CGB+10] presents design, implementation and performance of a new Variable Stiffness Actuator (VSA) based on Harmonic Drives (VSA-HD).
[PMDL08]presents a basic control study for a general class of multi-dof manipulators with variable joint stiffness, taking into account different possible modalities for changing the joint stiffness on the fly by an additional set of commands.
[VVHL+09]presents a design formulation to link the different mechanical designs together, and a study on the power consumption of these actuators.
[CHL+11]presents a variable stiffness joint (VSJ) designed for a robot manipulator, as well as a control scheme to control the stiffness and position of the VSJ.
[LTC10]introduces the development of a semi-active friction based variable physical damping actuator (VPDA) unit.
[ASOH07]describes a general passivity-based framework for the control of flexible joint robots.
[TLVC09]presents the development of a new compact soft actuation unit intended to be used in multi degree of freedom and small scale robotic systems such as the child humanoid robot.
[JTVC10]describes the design and development of a new actuator with adjustable stiffness (AwAS) which can be used in robots which are necessary to work close to or physically interact with humans.
[VHVVD+07] describes a rotational actuator with a novel adaptable compliance (inverse of stiffness);
[HSV+09]describes the state of the art in the design of actuators with adaptable passive compliance.
[EHW+10]introduces a classification of intrinsically compliant joint mechanisms.
[GASB+11]is aimed to reach its human archetype regarding size, weight and performance in the consideration of robustness, dynamic performance and dexterity.
[YGH+11] presents a novel human-like learning controller to interact with unknown environments.
[VASB+13] presents an overview of the different VIAs developed and proposes a classification based on the principles through which the variable stiffness and damping are achieved.
[GPB+11]follows a similar program of understanding the use of VSA in performance enhancement, looking at very dynamic tasks where impacts are maximized.
[TSB05] is concerned with the design and control of actuators for machines and robots physically interacting with humans.
[CGG+11]proposes a prototype of a Variable Stiffness Actuator (VSA) conceived with low cost as its first goal. This approach was scarcely covered in past literature.
[SSVO09] describes the basic theories of the robotics, includes how to model a robot and how to control it.

Chapter 2

Main Part

2.1 Design of mathematic model

In this section, how to build the mathematic model will be discussed.The input of the model is the coordinate of a targeting point, and the output is the joint variables θ which will be used to control the robot.

2.1.1 Setting DH-Parametrers

As shown in figure 2.1,with reference to the rules demonstrated in ([SSVO09] chapter 2.8.2), DH parameters of the robots can be calculated.

Link	a_i	α_i	d_i	θ_i(initial)
1	0	$\pi/2$	0.16	$\pi/2$
2	0	-$\pi/2$	0	$\pi/2$
3	0	$\pi/2$	0.17	$\pi/2$
4	0.09	0	0	$\pi/2$

Table 2.1: DH-parameters for the right arm

Link	a_i	α_i	d_i	θ_i(initial)
1	0	-$\pi/2$	0.16	0
2	0	-$\pi/2$	0	$\pi/2$
3	0	$\pi/2$	0.17	$\pi/2$
4	0.09	0	0	$\pi/2$

Table 2.2: DH-parameters for the left arm

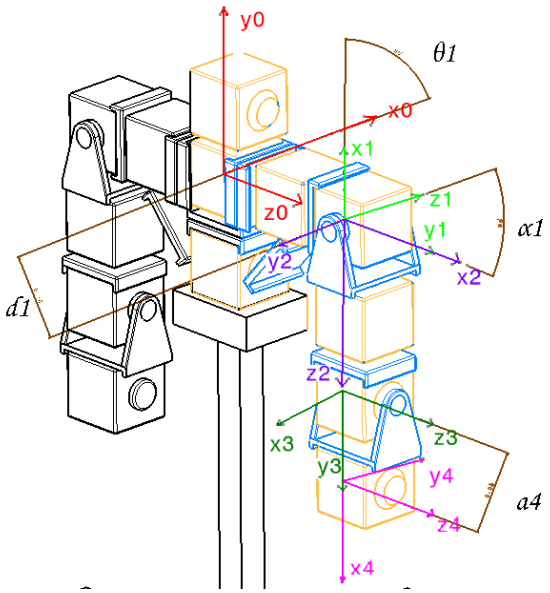

Figure 2.1: DH-parameters for the right arm

2.1.2 Kinematics

Basic idea of Forward kinematics: The position of the end-effector is a function of joint variables $\theta([\theta_1,\theta_2,\theta_3,\theta_4])$

$$position(x, y, z) = func(\theta_1, \theta_2, \theta_3, \theta_4) \tag{2.1}$$

The coordinate transformation between two frames is expressed by the homogeneous matrix A_i^{i-1}:([SSVO09] chapter 2.8.2)

$$A_i^{i-1}(q_i) = \begin{bmatrix} cos(\theta_i) & -sin(\theta_i)cos(\alpha_i) & sin(\theta_i)sin(\alpha_i) & a_i cos(\theta_i) \\ sin(\theta_i) & cos(\theta_i)cos(\alpha_i) & -cos(\theta_i)sin(\alpha_i) & a_i sin(\theta_i) \\ 0 & sin(\alpha_i) & cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} R_{i-1}^i(\theta) & o_{i-1}^i(\theta) \\ 0 & 1 \end{bmatrix} \tag{2.2}$$

Where $R_{i-1}^i(\theta)$ refers to the rotation and $o_{i-1}^i(\theta)$ the transformation.
Then the transformation matrix from frame-4 to frame-0 can be expressed as

$$A_4^0 = A_1^0 A_2^1 A_3^2 A_4^3 = \begin{bmatrix} R_4^0(\theta) & o_4^0(\theta) \\ 0 & 1 \end{bmatrix} \tag{2.3}$$

Reference to equation 2.1,the forward kinematic equation shown in figure2.2 can be expressed as

$$p_0 = o_4^0(\theta) + R_4^0(\theta)p_4 \tag{2.4}$$

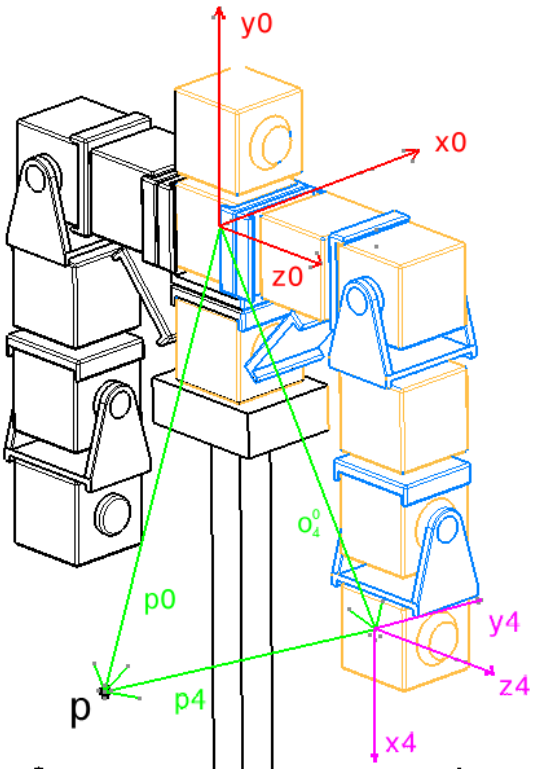

Figure 2.2: Coordinates transformation from end-effector to base frame

Theoretically the joint variables θ can be deduced from equation 2.1,

$$\theta = func^{-1}(position(x, y, z)) \tag{2.5}$$

However it is unrealistic to calculate $func^{-1}$ directly, because $func$ is nonlinear (see equation 2.2).
One method to fix this problem is using **Differential Kinematics** instead:

$$\theta_{next} = \theta + \dot{\theta}dt \tag{2.6}$$

Differential Kinematics describe the Relationship between the joint velocities and end-effector linear and angular velocity ($\dot{p}$,ω), as illustrated in figure 2.3

$$\dot{p} = J_P(\theta)\dot{\theta} \tag{2.7}$$
$$\omega = J_O(\theta)\dot{\theta} \tag{2.8}$$
$$J = [J_P; J_O] \tag{2.9}$$

J is Jacobian Matrix, which can be obtained from equation 2.2

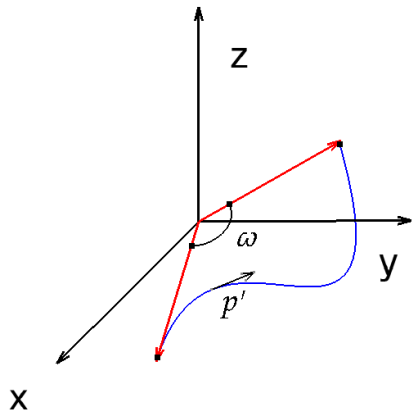

Figure 2.3: linear and angular velocity

$$J = \begin{bmatrix} J_{Pi} \\ J_{Oi} \end{bmatrix} = \begin{bmatrix} z_{i-1} \times (p_e - p_{i-1}) \\ z_{i-1} \end{bmatrix} \tag{2.10}$$

In order to get $\dot{p}$, **Minimum Jerk Trajectory** is applied:

$$function[x_{next}, xd_{next}, xdd_{next}, tmove_{next}] = minjerkstep(x, xd, xdd, goal, tmove, dt)$$

There are different ways moving to the predefined point, **Minimum Jerk Trajectory** can choose the smoothest one,which has the Minimum Jerk.
Apply this algorithm to z and y dimension as well. Then

$$\dot{p} = [xd, yd, zd] \tag{2.11}$$

Combine equation 2.6,2.7 and 2.11

$$\theta_{next} = \theta + J_P^{-1}(\theta)\dot{p}dt \tag{2.12}$$

Note:This research focuses on position control, so only J is used and J_O is ignored.

2.1.3 Structure of the model

The input of the model is the coordinate of a targeting point, and the output is the joint variables θ ,see figure 2.4

2.2 Design of the controller

Controller transfers the signal(output of the model) to the real robot,and adjust the robot arms to reach the target point.
A PID controller is applied to satisfy the accuracy requirement,refer to figure2.5 .

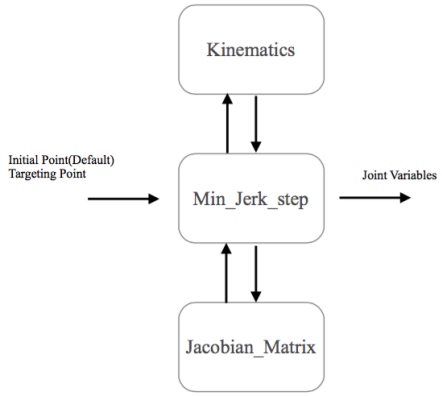

Figure 2.4: Model structure

Figure 2.5: PID controller

2.2.1 Base frame transformation

As shown in figure2.1, frame-0 is the base frame of right arm, and the z-axis(z0) is the rotate axis of the first joint. This is the requirement of setting DH parameter.Similar to the left arm. So two arms have different base frame. It is better to use only one base frame in one robot: global base frame.
Definition of global base frame:$[x_g, y_g, z_g]$,origin is the same as frame0 of right arm; z-axis is perpendicular to the ground and points upwards; x-axis is perpendicular to z-axis and points forwards; y-axis is chosen to complete a right-hand frame.
So the transformation between $[x_g, y_g, z_g]$ and $[x_0, y_0, z_0]$ is

$$[x_g; y_g; z_g] = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} [x_0; y_0; z_0] \tag{2.13}$$

Apply the similar operation to the left arm.

Chapter 3

Experimental results

In this section, the above mentioned model and controller will be realized.

3.1 Implementation of the Model

3.1.1 Flowchart of the model algorithm

As shown in figure3.1, the algorithm takes the targeting point as the input, then passes it to function blocks for further processing. Firstly the values of joint variables for forward moving is calculated step by step, then after the Inverse moving switch signal is activated, all the parameters are resetted,and values of joint variables for inverse moving is calculated.

3.1.2 Option 1

The first trial of Implementation is: divide the whole program into individual function blocks. Dividing and packaging each function block into standard block can improve the algorithm flexiblity of the following tasks which base on this work.

With reference to figure3.2, $Kinamatics function$, $Minjerkstep function$ and $Jacobi-matrix function$ are individual to each other and communicate through I/O interface. The $flag$ signals are used to synchronize the blocks,so as to insure that each step to be calculated with the right arguments.

In the parameters setting, the sampling time is 8 ms. Figure3.3 shows that the time gap between two adjacent values is 16 ms which means that the program needs two sampling time periods to calculate the value for one step.

The possible reason: the synchronization or the data-transfer between blocks causes the delay.

One way to reduce the delay is to merge the blocks to reduce the delay.

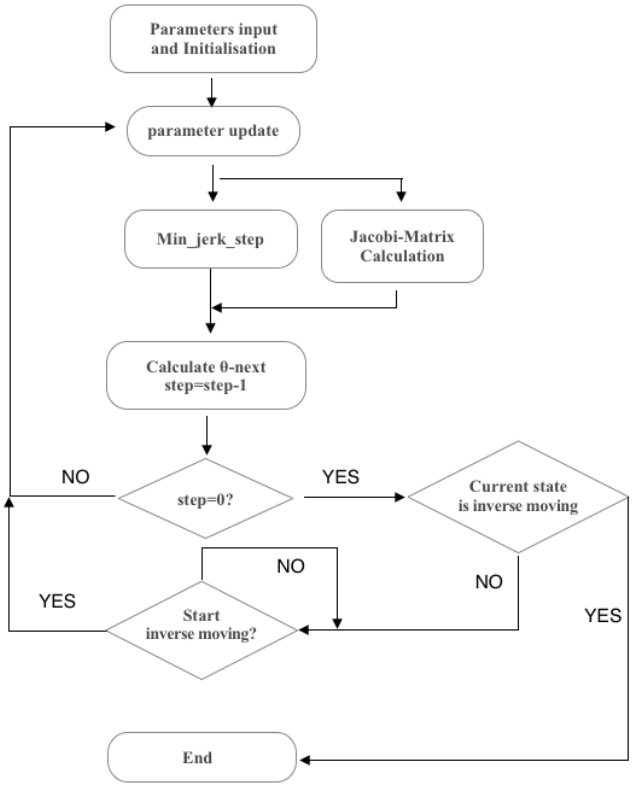

Figure 3.1: Flowchart of the model algorithm

3.1.3 Option 2

In option 2 ,$Kinamatics function$ and $Jacobi - matrix function$ are merged into one block(Figure3.4) The result is the same as in option 1(Figure3.5).

3.1.4 Option 3

In option 3 , all the function blocks are merged into one block. See Figure3.6 and Figure 3.7;
It is can be observed that the time gap between two adjacent values is 8 ms,one sampling period.

3.2 implementation of PID controller

Since the dynamic behavior of the robot is unknown,so it difficult to design a PID controller directly. However with help from *pidtool* in Matlab/Simulink it is possible to design a PID controller without knowing the system's dynamics.

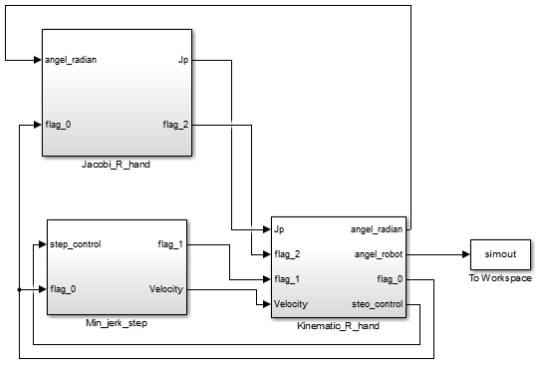

Figure 3.2: Individual function blocks-1

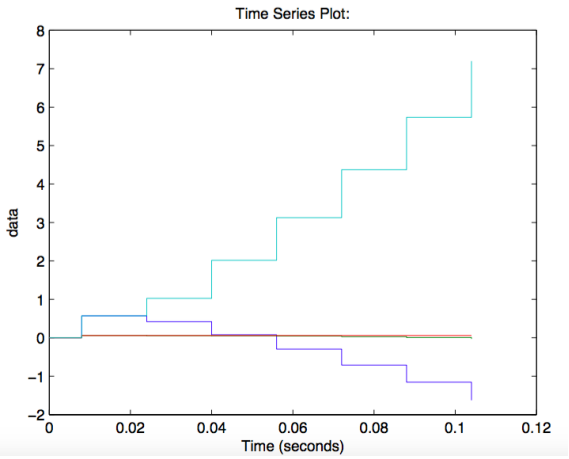

Figure 3.3: Simulation result of Option 1

Design the PID controller

Step 1:sampling the step response of the system(figure3.8).
Step 2:use *pidtool* in Matlab/Simulink to generate the PID parameters(figure3.9).
Step 3:manually adjust the PID parameters to get a better performance.

Extension of the controller

Add Saturation at the PID input and feedback rout to ensure a better PID performance; Add Saturation at the PID output to protect the robot from over-rotating.See figure3.10

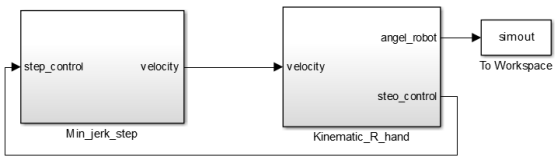

Figure 3.4: Individual function blocks-2

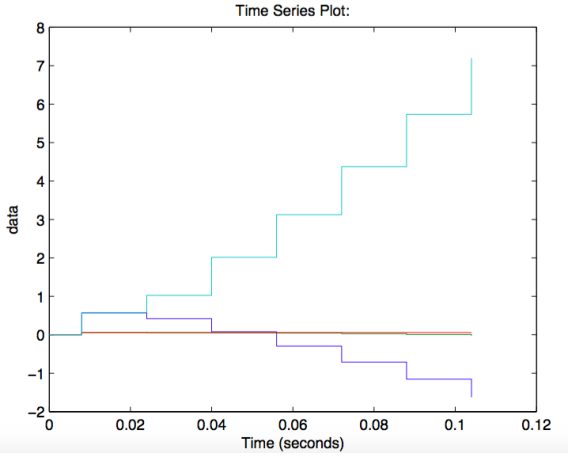

Figure 3.5: Simulation result of Option 2

3.3 Test and Results

Since the right arm and left arm are symmetrical to each other,so this report only record the test/result of right arm.

3.3.1 test setup and test procedure

Test setup

The test system mainly consists of 4 parts(figure 3.11):
1, A switch to change moving direction
2,Pace generator generates the joint variables' value step by step
3,Controller contains the PID controller and the interface to the robot
4, Two display to show the real-time values(input and feedback of the robot) and one block "simout" to record the input values.

Test procedure

1,Connect and power-on the robot;

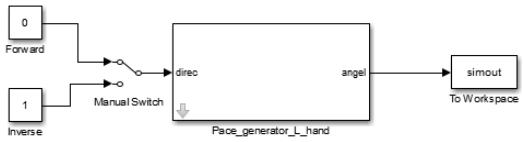

Figure 3.6: Individual function blocks-3

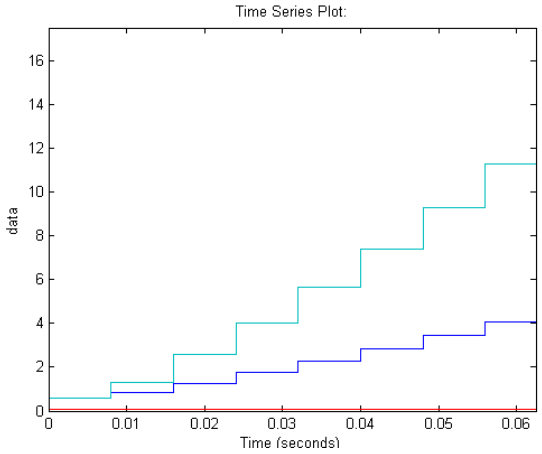

Figure 3.7: Simulation result of Option 3

2,set the switch to "Forward",double click the pace-generator block to set the targeting point;

3,run the simulation, when the values in both display stop changing, means the robot has reached the targeting point,then set the switch to "Inverse"

4, after the simulation stop automatically, analyse the recording data.

5,adjust the PID parameters according to the robot performance.

3.3.2 Results analyse

Result analyse-1

Figure3.12 shows the four Variables traces of four joints.It can be observed that all the joints move smoothly, however not all the traces converge to origin position$[0, 0, 0, 0]$

possible reason: assembly error and measurement error in the DH parameters

solution:add extra function to dampen the errors in the traces.

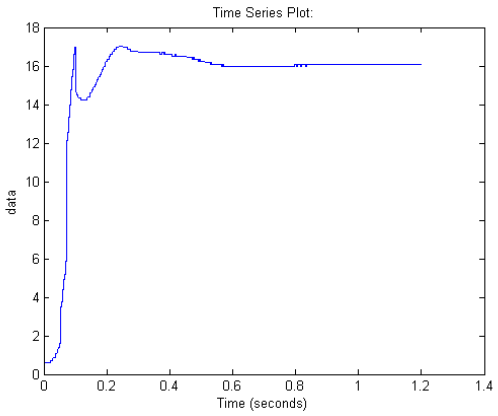

Figure 3.8: step response of the joint

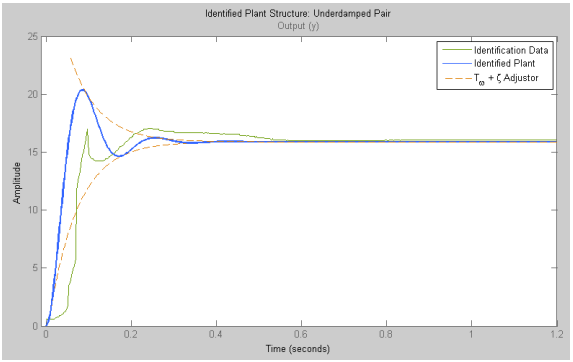

Figure 3.9: adjust the parameters

Result analyse-2

Run the simulation again and the result is in figure3.13 The traces finally all converge to origin.

Result analyse in robot movements

From the video,also from the real joints trajectories(figure3.14) it can be observed that

- The arm can finally reach the right position;
- The arm moves fast and smoothly in the first place and stop in the middle way for about 2 seconds then continue moves relative slowly and even jerkily to the pre-defined position.

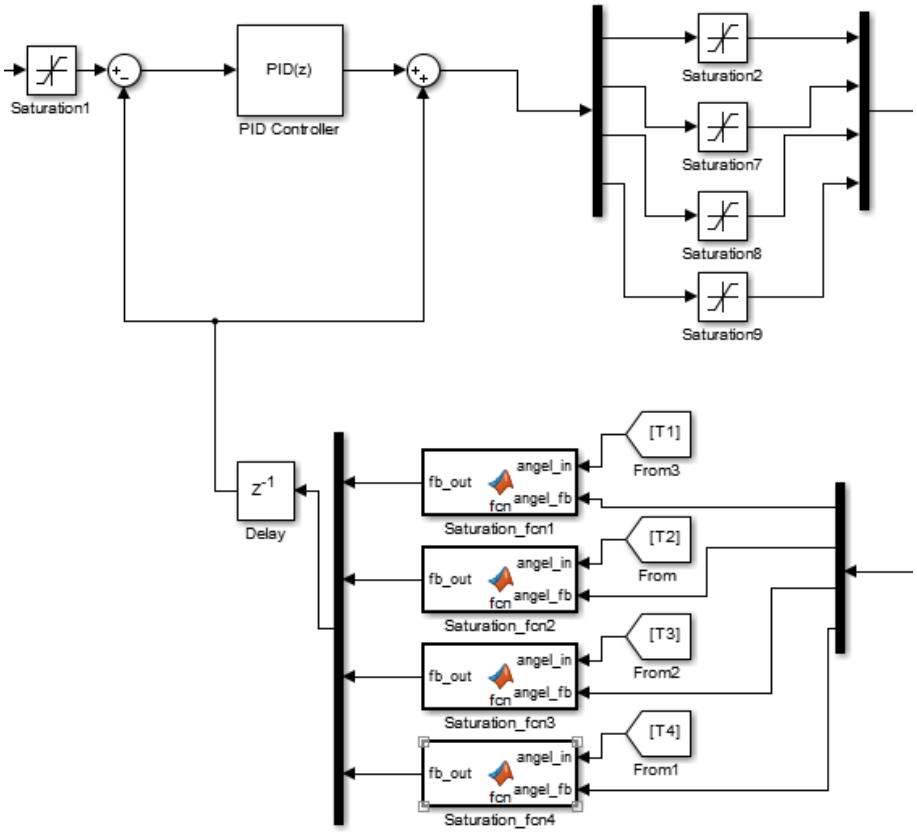

Figure 3.10: PID controller with extension

3.4 Discussion

The PID parameters obtained from the *pidtool* do not fit the robot well, need to be manually adjust by observing the robot moves;
Potential reason for the moving performance is influence of the gravity. In this research compensation method to eliminate the gravity influence isn't included.
Three options for blocks packaging are discussed in section $Implementation of the Model$, and the test only records the result of the first option. Because the first and the second options do not meet the timing requirement.

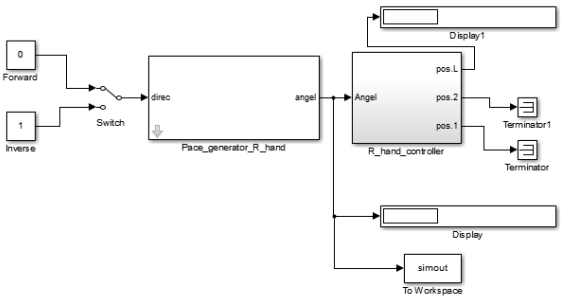

Figure 3.11: Test set-up

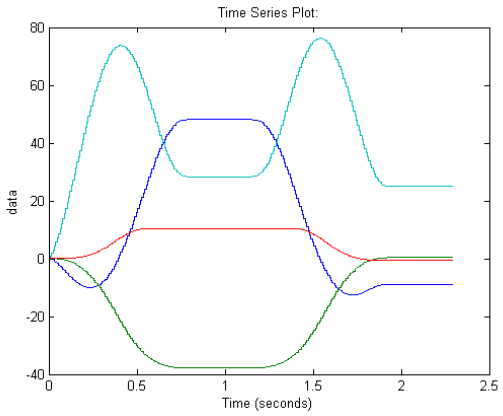

Figure 3.12: joints trajectories do not converge to origin

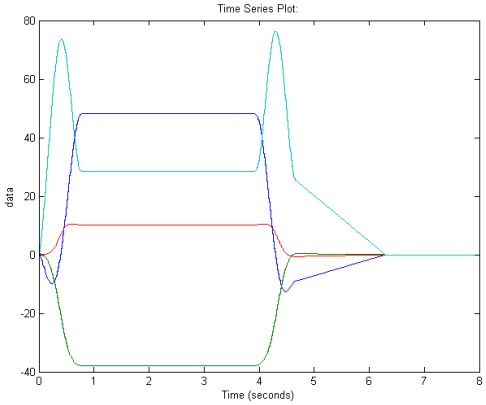

Figure 3.13: joints trajectories converge to origin

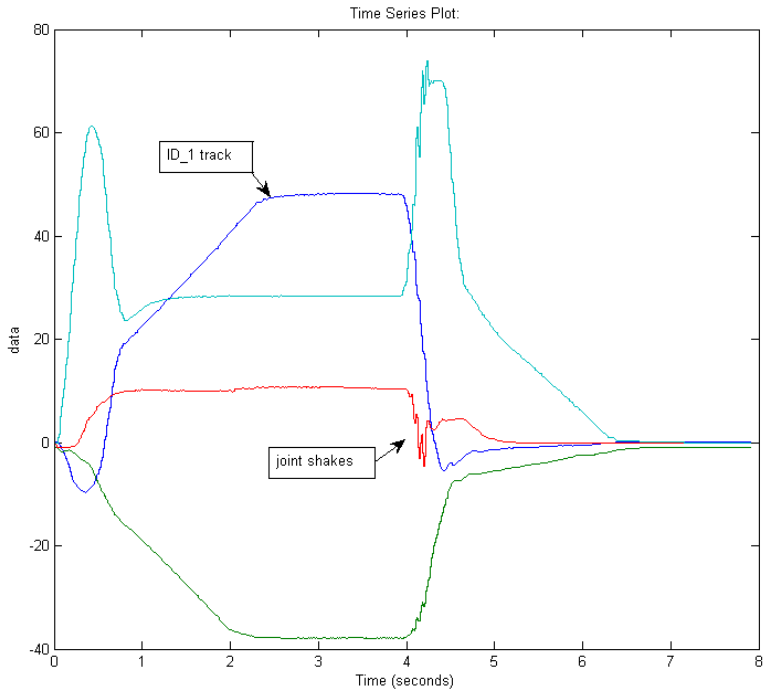

Figure 3.14: real Joints trajectories

Chapter 4

Conclusion

Different methods have been discussed to improve the robot performance.
This work has basically reached the goal:drive the robot arm moving to the predefined position.However the speed and smoothness are not as expected.Reason can be that the Dynamics is not part of the concern in this work.

List of Figures

Bibliography

[ASOH07] Alin Albu-Schäffer, Christian Ott, and Gerd Hirzinger. A unified passivity-based control framework for position, torque and impedance control of flexible joint robots. *The International Journal of Robotics Research*, 26(1):23–39, 2007.

[CGB+10] Manuel G Catalano, Giorgio Grioli, Fabio Bonomo, Riccardo Schiavi, and Antonio Bicchi. Vsa-hd: From the enumeration analysis to the prototypical implementation. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 3676–3681. IEEE, 2010.

[CGG+11] Manuel G Catalano, Giorgio Grioli, Manolo Garabini, Fabio Bonomo, Michele Mancini, Nikolaos Tsagarakis, and Antonio Bicchi. Vsa-cubebot: A modular variable stiffness platform for multiple degrees of freedom robots. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 5090–5095. IEEE, 2011.

[CHL+11] Junho Choi, Seonghun Hong, Woosub Lee, Sungchul Kang, and Munsang Kim. A robot joint with variable stiffness using leaf springs. *Robotics, IEEE Transactions on*, 27(2):229–238, 2011.

[EHW+10] Oliver Eiberger, Sami Haddadin, Michael Weis, Alin Albu-Schäffer, and Gerd Hirzinger. On joint design with intrinsic variable compliance: derivation of the dlr qa-joint. In *ICRA*, pages 1687–1694, 2010.

[GASB+11] Markus Grebenstein, Alin Albu-Schäffer, Thomas Bahls, Maxime Chalon, Oliver Eiberger, Werner Friedl, Robin Gruber, Sami Haddadin, Ulrich Hagn, Robert Haslinger, et al. The dlr hand arm system. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 3175–3182. IEEE, 2011.

[GPB+11] Manolo Garabini, Andrea Passaglia, Felipe Belo, Paolo Salaris, and Antonio Bicchi. Optimality principles in variable stiffness control: The vsa hammer. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 3770–3775. IEEE, 2011.

[HSV+09] R v Ham, Thomas G Sugar, Bram Vanderborght, Kevin W Hollander, and Dirk Lefeber. Compliant actuator designs. *Robotics & Automation Magazine, IEEE*, 16(3):81–94, 2009.

[JTC11] Amir Jafari, Nikos G Tsagarakis, and Darwin G Caldwell. Exploiting natural dynamics for energy minimization using an actuator with adjustable stiffness (awas). In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 4632–4637. IEEE, 2011.

[JTVC10] Amir Jafari, Nikos G Tsagarakis, Bram Vanderborght, and Darwin G Caldwell. A novel actuator with adjustable stiffness (awas). In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 4201–4206. IEEE, 2010.

[LTC10] Matteo Laffranchi, Nikos G Tsagarakis, and Darwin G Caldwell. A variable physical damping actuator (vpda) for compliant robotic joints. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 1668–1674. IEEE, 2010.

[PCC11] Antonio Pistillo, Sylvain Calinon, and Darwin G Caldwell. Bilateral physical interaction with a robot manipulator through a weighted combination of flow fields. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 3047–3052. IEEE, 2011.

[PMDL08] Gianluca Palli, Claudio Melchiorri, and Alessandro De Luca. On the feedback linearization of robots with variable joint stiffness. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 1753–1759. IEEE, 2008.

[SSVO09] Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, and Giuseppe Oriolo. *Robotics: modelling, planning and control*. Springer Science & Business Media, 2009.

[TLVC09] Nikolaos G Tsagarakis, Matteo Laffranchi, Bram Vanderborght, and Darwin G Caldwell. A compact soft actuator unit for small scale human friendly robots. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 4356–4362. IEEE, 2009.

[TSB05] Giovanni Tonietti, Riccardo Schiavi, and Antonio Bicchi. Design and control of a variable stiffness actuator for safe and fast physical human/robot interaction. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 526–531. IEEE, 2005.

[VASB+13] Bram Vanderborght, Alin Albu-Schäffer, Antonio Bicchi, Etienne Burdet, Darwin G Caldwell, Raffaella Carloni, M Catalano, Oliver Eiberger, Werner Friedl, Ganesh Ganesh, et al. Variable impedance actuators: A review. *Robotics and Autonomous Systems*, 61(12):1601–1614, 2013.

[VHVVD+07] Ronald Van Ham, Bram Vanderborght, Michaël Van Damme, Björn Verrelst, and Dirk Lefeber. Maccepa, the mechanically adjustable compliance and controllable equilibrium position actuator: Design and implementation in a biped robot. *Robotics and Autonomous Systems*, 55(10):761–768, 2007.

[VVHL+09] Bram Vanderborght, Ronald Van Ham, Dirk Lefeber, Thomas G Sugar, and Kevin W Hollander. Comparison of mechanical design and energy consumption of adaptable, passive-compliant actuators. *The International Journal of Robotics Research*, 28(1):90–103, 2009.

[YGH+11] Chenguang Yang, Gowrishankar Ganesh, Sami Haddadin, Sven Parusel, Alin Albu-Schäeffer, and Etienne Burdet. Human-like adaptation of force and impedance in stable and unstable interactions. *Robotics, IEEE Transactions on*, 27(5):918–930, 2011.

License