Proc. 2002 Australasian Conference on Robotics and Automation Auckland, 27-29 November 2002

Computer Simulation of Robot Closed-loop Dynamics for Force Control Study

Yueshi Shen
Department of Systems Engineering
Research School of Information
Sciences and Engineering
Australian National University
Canberra ACT 0200
yueshi@syseng.anu.edu.au

Roy Featherstone
Department of Systems Engineering
Research School of Information
Sciences and Engineering
Australian National University
Canberra ACT 0200
roy@syseng.anu.edu.au

Abstract

This paper presents a practical computer simulation method for a manipulator robot's closed-loop dynamics. A two-order acceleration stabiliser has been applied as the motion constraint of the closed-loop dynamics, and 6-D vector has been used in its corresponding mathematical analysis instead of the conventional 3-D vector. Then we present some simulation result for a simple motion/force control law, which is applied on an actual 4-DOF manipulator robot model, the function of acceleration stabiliser has also been discussed in later section. Lastly, we describe some expected future work on both advanced contact dynamics simulation involving contact detection, and friction compensation in force control of robot-environment interaction.

1 Introduction

1.1 Background

Control of manipulator robot can be conceptually divided into motion control and force control. Motion control techniques have been fully developed and widely used in many industry areas, but so far there are still few successful applications for force control, especially when the friction between robot and its working environment can not be ignored [Yoshikawa, 2000].

Before applying the new-established control law on real robot, normally we first run some computer simulation to test its theoretical correctness or even practical performance. Under this circumstance, people have poured much interest on this issue and already developed several successful applications for simulating the manipulator robot's kinematics and dynamics [Corke, 1995]. However, most of such simulators are designed to cope with motion control simulation, in other words, open-loop dynamics, which assumes that robot is allowed to move freely in the positional space.

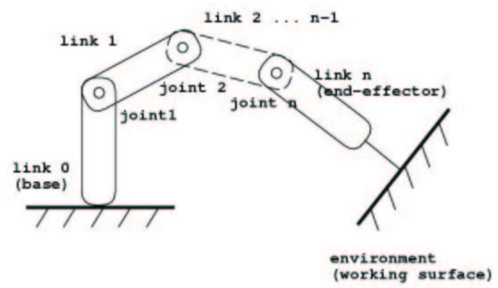

Figure 1: n-joint robot in contact with working surface

But if we wish to do some researches on the force or motion/force hybrid control, then the closed-loop dynamics must be considered, because in this case robot's end-effector will be in contact with some environment, thus the motion behaviour is constrained to some space with limited degree of freedoms.

To study the contact dynamics, usually some motion constraint functions will be introduced and combined with the normal motion equation of forward dynamics to form the complete motion/force equation (Eqs. 1).

$$\begin{cases} \mathbf{H}\ddot{\mathbf{q}} + \mathbf{C} = \mathbf{Q} + \mathbf{T}_{equ} \\ c(\mathbf{q}) = 0 \end{cases} \qquad (1)$$

where $\mathbf{Q}$ is the driving torque, $\mathbf{T}_{equ}$ is the joint torque equivalent to the contact force, it has also been called constraint force. $\mathbf{H}$, $\mathbf{C}$ are inertial matrix and Coriolis-gravitation vector. The function c represents the motion constraint.

1.2 Problem Description

To simulate the robot closed-loop dynamics, let's first start with a simple case as showed in Fig. 1. Suppose the robot has n joints and the tip of its only end-effector is in contact with some frictionless working surface, also we assume that the robot is not in some singularity pose. Then the problem is to calculate the joint acceleration and contact force that comply with the motion and force constraints.

Copyright © ACRA 2002

So, compare with the general case of kinematic loop mechanism [Featherstone, 1987], our particular problem depends on the following several assumptions:

1. The whole mechanism has only one closed-loop, it is closed through the contact between the robot end-effector and the fixed surface
2. The contact between robot and environment is limited to a single point
3. There will be no loss of contact during the simulation
4. The interaction between robot and working surface is frictionless
5. The robot is not in some strange pose, which causes such problems as singularity etc.

2 Mathematical Analysis

2.1 General Analysis of Closed-loop Dynamics for Chain-structured Robot

For an n-joint chain-structured robot with its end-effector in contact with the environment as described above, we introduce a loop-closure joint j_{n+1}, which connects link 0 (predecessor) and link n (successor).

The difficulties of kinematic loop on robot dynamics lie on two complications. Firstly, it introduces the motion constrain

$$\mathbf{R}^T(\mathbf{v}_n - \mathbf{v}_0) = 0 \tag{2}$$

where $\mathbf{v}_0$, $\mathbf{v}_n$ are respectively the velocities of link 0 and link n, $\mathbf{R}$ is the contact force space of the loop-closure joint, the number of constraints imposed on the velocity is equal to the dimension of $\mathbf{R}$. By differentiating Eqs. 2, we get the acceleration constraint of closed-loop dynamics

$$\mathbf{R}^T(\mathbf{a}_n - \mathbf{a}_0) + \dot{\mathbf{R}}^T(\mathbf{v}_n - \mathbf{v}_0) = 0 \tag{3}$$

Also, we have

$$\mathbf{v}_n = \mathbf{J}_n\,\dot{\mathbf{q}} \tag{4}$$

$$\mathbf{a}_n = \mathbf{J}_n\,\ddot{\mathbf{q}} + \dot{\mathbf{J}}_n\,\dot{\mathbf{q}} \tag{5}$$

where $\mathbf{J}_n$ is Jacobian matrix mapping joint velocity to velocity of link n.

Now assume the base link is stationary, in other words, $\mathbf{v}_0 = 0$ and $\mathbf{a}_0 = 0$. Insert Eqs. 4 and Eqs. 5 into Eqs. 3, we get the motion constraint equation expressed in joint acceleration as

$$\mathbf{R}^T\,\mathbf{J}_n\,\ddot{\mathbf{q}} = -\mathbf{R}^T\,\dot{\mathbf{J}}_n\,\dot{\mathbf{q}} - \dot{\mathbf{R}}^T\,\mathbf{J}_n\,\dot{\mathbf{q}} \tag{6}$$

Secondly, the closed-loop dynamics involves the contact force f (positive scalar) applied on the robot's end-effector from environment, the equivalent joint torque due to f can be expressed as

$$\mathbf{Q}_{equ} = \mathbf{J}_n^T\,\mathbf{R}f \tag{7}$$

The common motion equation [Featherstone, 1987] for robot dynamics is

$$\mathbf{H}\ddot{\mathbf{q}} = \mathbf{Q} - \mathbf{C} \tag{8}$$

Combine Eqs. 6, 7 and 8, we can complete the Eqs. 1, which represents the motion/force mechanism for robot closed-loop dynamics

$$\begin{bmatrix} \mathbf{H} & \mathbf{J}_n^T\mathbf{R} \\ \mathbf{R}^T\mathbf{J}_n & 0 \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ -f \end{bmatrix} = \begin{bmatrix} \mathbf{Q}-\mathbf{C} \\ b \end{bmatrix} \tag{9}$$

where

$$b = -\mathbf{R}^T\,\dot{\mathbf{J}}_n\,\dot{\mathbf{q}} - \dot{\mathbf{R}}^T\,\mathbf{J}_n\,\dot{\mathbf{q}} \tag{10}$$

The above $\mathbf{v}_0$, $\mathbf{v}_n$, $\mathbf{a}_0$, and $\mathbf{a}_n$ are all 6-D spatial velocity and acceleration [Featherstone, 1987; 2001] and $\mathbf{R}$, $\mathbf{J}_n$, $\mathbf{H}$, $\mathbf{C}$ are expressed in spatial algebra definition.

2.2 Mathematical Revision for Practical Calculating Consideration

For practical simulation, first establish the model of working surface, which reflects the contact force space or motion constraint

$$\mathbf{n}\cdot\mathbf{p} = d \tag{11}$$

where $\mathbf{n}$ is the normal of the working surface, $\mathbf{p}$ is for any point on this plane, d is a scalar representing the distance between plane and coordinate origin.

Since $\mathbf{n}$ here can be considered as $\mathbf{R}$ in Eqs. 2, the equation of motion now becomes

$$\mathbf{H}\ddot{\mathbf{q}} + \mathbf{C} = \mathbf{Q} + \mathbf{J}_r\,\mathbf{n}\,f \tag{12}$$

where

$$\mathbf{J}_r = \begin{bmatrix} \mathbf{r}\times^T & \mathbf{I} \end{bmatrix}\mathbf{J} \tag{13}$$

$\mathbf{J}_r$ is 3 by n matrix mapping the joint velocities to the linear component of spatial velocities at the point of robot's end-effector. $\mathbf{r}$ is the 3-D vector of x,y,z position of robot end-effector expressed in the base coordinate. The cross operator is defined in *Robot Dynamics Algorithm* [Featherstone, 1987]

$$\mathbf{r}\times = \begin{bmatrix} 0 & -r_z & r_y \\ r_z & 0 & -r_x \\ -r_y & r_x & 0 \end{bmatrix} \tag{14}$$

$\mathbf{I}$ is 3 by 3 unit matrix, f is the magnitude of contact force applied on robot from environment.

Now construct the acceleration constraint (motion constraint), first start with forward kinematics

$$\mathbf{r} = k_r(\mathbf{q}) \tag{15}$$

where k_r is the forward kinematics function yielding end-effector's position.

Eqs. 15 yields

$$\dot{\mathbf{r}} = \mathbf{J}_r\,\dot{\mathbf{q}} \tag{16}$$

and

$$\ddot{\mathbf{r}} = \mathbf{J}_r\,\ddot{\mathbf{q}} + \dot{\mathbf{J}}_r\,\dot{\mathbf{q}} \tag{17}$$

where

$$\dot{\mathbf{J}}_r \dot{\mathbf{q}} = [\ \mathbf{r}\times^T \quad 1\ ]\, \dot{\mathbf{J}}\, \dot{\mathbf{q}} + [\ \dot{\mathbf{r}}\times^T \quad 0\ ]\, \mathbf{J}\, \dot{\mathbf{q}} \tag{18}$$
$$= [\ \mathbf{r}\times^T \quad 1\ ]\, \dot{\mathbf{J}}\, \dot{\mathbf{q}} + [\ (\mathbf{J}_r \dot{\mathbf{q}})\times^T \quad 0\ ]\, \mathbf{J}\, \dot{\mathbf{q}}$$

Then, based on Eqs. 11 and 15, we introduce the closed-loop error function e, which represents the discrepancy between robot end-effector and working surface

$$e(\mathbf{q}\ ) = \mathbf{n}\cdot\mathbf{r} - d \tag{19}$$

which yields

$$\dot{e}\ (\mathbf{q}\ ) = \mathbf{n}^T \mathbf{J}_r \dot{\mathbf{q}} \tag{20}$$

$$\ddot{e}\ (\mathbf{q}\ ) = \mathbf{n}^T \mathbf{J}_r \ddot{\mathbf{q}} + \mathbf{n}^T \dot{\mathbf{J}}_r \dot{\mathbf{q}} \tag{21}$$

In actual simulation, e will probably go off zero due to various practical calculation errors, so it is necessary to introduce an acceleration stabiliser to compensate this kind of numeric drifts

$$\ddot{e} + s_v \dot{e} + s_p e = 0 \tag{22}$$

which yields

$$\mathbf{n}^T \mathbf{J}_r \ddot{\mathbf{q}} = -\mathbf{n}^T [\ (\mathbf{J}_r \dot{\mathbf{q}})\times^T \quad 0\ ]\, \mathbf{J}\, \dot{\mathbf{q}} - \mathbf{n}^T \dot{\mathbf{J}}_r \dot{\mathbf{q}} - s_v\, \mathbf{n}^T \mathbf{J}_r \dot{\mathbf{q}} - s_p\, (\mathbf{n}^T \mathbf{k} - d)$$
$$= b_s \tag{23}$$

So, the complete equation of motion/force is like

$$\begin{bmatrix} \mathbf{H} & \mathbf{J}_r^T \mathbf{n} \\ \mathbf{n}^T \mathbf{J}_r & 0 \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ -f \end{bmatrix} = \begin{bmatrix} \mathbf{Q} - \mathbf{C} \\ b_s \end{bmatrix} \tag{24}$$

In summary, the whole calculation process can be divided into four steps as follows:

Step 1, calculate standard forward dynamics for $\mathbf{H}(\mathbf{q})$, $\mathbf{C}(\mathbf{q})$.

There are several ways to calculate the inertia matrix $\mathbf{H}$, the double recursive Newton-Euler method is easy to implement but computationally expensive [Walker and Orin, 1982], it has been applied in *Matlab Robotics Toolbox* [Corke, 1995]. Here we use the 6-D vector composite-rigid-body algorithm [Featherstone, 1987]. In authors' opinions, it is comparatively more efficient and easier to program.

Step 2, calculate standard forward kinematics function k($\mathbf{q}$) for contact point on working surface.

The standard homogenous transformation should be the best way, 6-D vector analysis is not necessary to be used on this issue.

Step 3, calculate the Jacobian $\mathbf{J}_r$ and its time derivative as explained below

Step 4, calculate the complete motion/force equation by using Eqs. 24

To calculate the Jacobian $\mathbf{J}_r$ and its time derivative, first calculate the conventional Jacobain which maps the joint spatial velocity to the rigid-body velocity of link n:

$$\mathbf{J} = [\ {}_0\mathbf{X}_1 \mathbf{s}_1 \quad {}_0\mathbf{X}_2 \mathbf{s}_2 \quad \cdots \quad {}_0\mathbf{X}_n \mathbf{s}_n\ ] \tag{25}$$

where $\mathbf{s}_i$ is the joint axis vector of joint i and ${}_0\mathbf{X}_i$ is the coordinate transform matrix from frame i to frame 0.

Matlab Robotics Toolbox [Corke, 1995] applies the coordinate differential method [Paul *et al.*, 1981]. The Jacobian matrix $\mathbf{J}_0$, $\mathbf{J}_n$ calculated by command *jacob0* and *jacobn*, are respectively expressed in frames whose orientation are same as frame 0 and frame n, but both origins locate at the origin of frame n. The relationship among $\mathbf{J}$, $\mathbf{J}_0$ and $\mathbf{J}_n$ are

$$\mathbf{J}_0 = \begin{bmatrix} \mathbf{I} & 0 \\ \mathbf{r}\times^T & \mathbf{I} \end{bmatrix} \mathbf{J} \tag{26}$$

$$\mathbf{J}_n = \begin{bmatrix} \mathbf{E} & 0 \\ 0 & \mathbf{E} \end{bmatrix} \mathbf{J}_0 \tag{27}$$

where $\mathbf{E}$ is the 3 by 3 orthogonal rotation matrix.

Since it is not easy to calculate $\dot{\mathbf{J}}_r$, we can calculate $\dot{\mathbf{J}}_r\dot{\mathbf{q}}$ instead, which equals to the velocity-product term of spatial acceleration of link n.

Proof

$$\mathbf{a}_i = \mathbf{a}_{i-1} + \mathbf{s}_i \ddot{\mathbf{q}}_i + \dot{\mathbf{s}}_i \dot{\mathbf{q}}_i \tag{28}$$

$$\mathbf{a}_n = \frac{d}{dt}(\mathbf{J}\dot{\mathbf{q}}) = \mathbf{J}\ddot{\mathbf{q}} + \dot{\mathbf{J}}\dot{\mathbf{q}} \tag{29}$$

$$\mathbf{a}_i^{vp} = \dot{\mathbf{J}}\dot{\mathbf{q}} \tag{30}$$

Calculation

$$\mathbf{a}_i^{vp} = \mathbf{a}_{i-1}^{vp} + \mathbf{v}_i \times \mathbf{s}_i \dot{\mathbf{q}}_i \tag{31}$$

$$\mathbf{a}_0^{vp} = 0 \tag{32}$$

$\mathbf{J}_r$ is then obtained from Eqs. 13 and $\dot{\mathbf{J}}_r\dot{\mathbf{q}}$ from Eqs. 18.

3 Computer Simulation

3.1 Simulate 4-DOF Manipulator Robot WAM

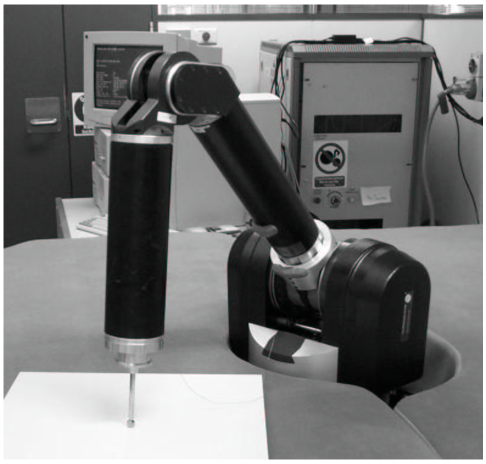

Figure 2: WAM robot platform for proposed force control experiments

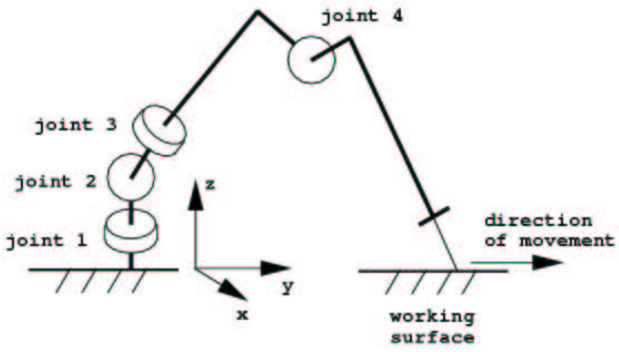

Figure 3: illustration of simulating WAM's closed-loop dynamics

The BA-300 WAM (Whole Arm Manipulation) robot (Barrett Technology Inc.) is a lightweight, high-speed, 4-degree-of-freedom manipulator (see Fig. 2). Due to its advanced cable-drive systems, WAM has no backlash, extremely low friction and high-bandwidth performance [Barrett, 1998], so it can be considered as an ideal platform for researches on high precision trajectory control and advanced force control.

Our project is to study the effect of friction in robot-environment interaction, and then design a practical real-time motion/force hybrid control law, which is capable of compensating the robot-environment fiction's effect. To achieve this objective, the simulation of non-frictional on-going contact dynamics will be our first step.

The simulation task here is as shown in Fig. 3. The robot's end-effector is supposed to move along the y-axis on a horizontal plane and a 10-Newton contact force is applied from robot to environment along negative z-axis. Intuitively, the control law consists of three parts: PD controller looking after the motion trajectory tracking, Coriolis-gravitation compensation, and the equivalent torque of contact force.

So, the control law can be expressed as Eqs. 33:

$$\begin{aligned}\mathbf{Q} &= p(\mathbf{q}-\mathbf{q}_{des})+d(\dot{\mathbf{q}}-\dot{\mathbf{q}}_{des})+\\ &\quad \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})+\mathbf{J}_r(\mathbf{q})^T\,\mathbf{n}(-10) \qquad (33)\end{aligned}$$

The simulation runs under the frequency of 1000 Hz, which is high enough to guarantee a good quality of simulation. Some results generated by SIMULINK are shown as follows,

The PD controller activates after 0.5 second (the other two parts activates immediately after the simulation starts) and makes the robot to stretch out slowly (the velocity of end-tip is less than 2cm/sec), which guarantees the dynamics of robot will not change severely during the movement. From Fig. 4 and 5, we can find that the robot tracks the desired trajectory quite well and the contact force stabilises at 10 Newton eventually. The overshoot of f comes from the transition from stationary to moving of the robot at the moment of PD control signal initially activates.

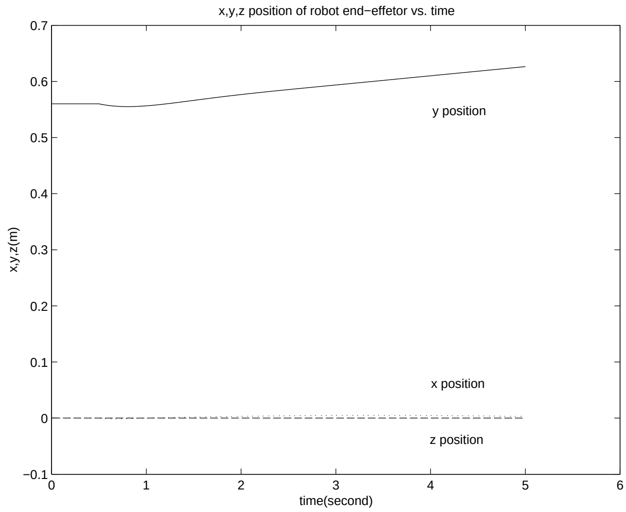

Figure 4: x,y,z position of robot's end-effector vs. time

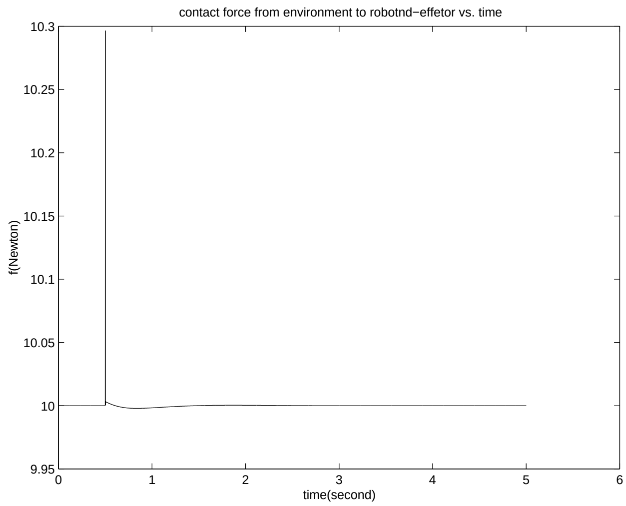

Figure 5: contact force from environment to robot vs. time

3.2 Necessity of Acceleration Stabiliser

As we discussed above, on the basis of conventional motion constraint we apply the acceleration stabiliser to compensate the numerical drift, which is due to various practical computational errors.

In the closed-loop simulation of WAM, the configuration of acceleration stabiliser is deliberately chosen to match the sampling frequency of whole robot system. We pick the damping ratio as 0.707, and time constant as 0.001, in other words, Sv=1414, Sp=1000000. The simulation results of closed-loop error e for the cases of stabiliser switched on or off are shown as follows.

From Fig. 6 and 7, we can find that once acceleration stabiliser has been cut off, the closed-loop error will accumulate gradually, in other words, there will

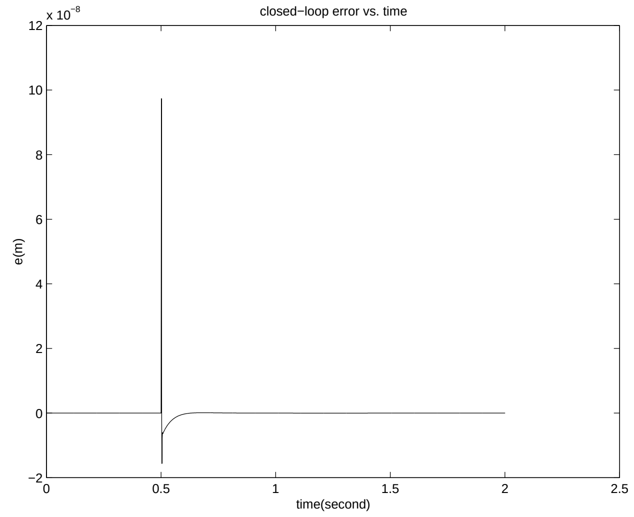

Figure 6: closed-loop error vs. time (stabiliser on)

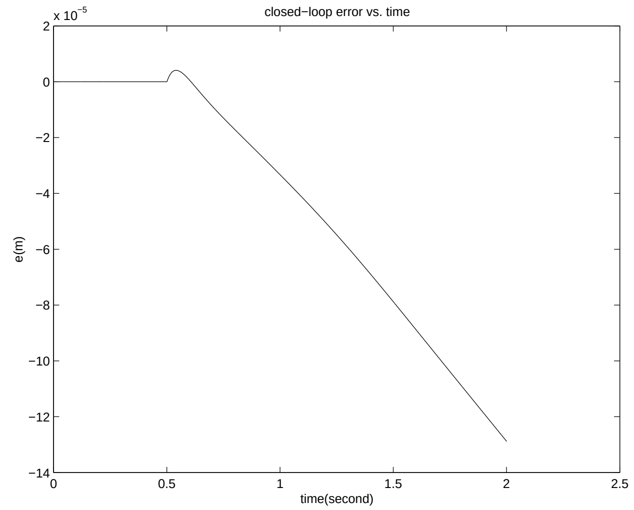

Figure 7: closed-loop error vs. time (stabiliser off)

be a steady numeric drift when the simulation keeps running for longer time. The problem is much worse when the sampling frequency has been set to a lower value or the robot is moving in a higher velocity.

4 Future Work

The work described in this paper can be regarded as a part of the research of friction compensation in robot-environment interaction. In future, we will mostly focus our attention on two major aspects: simulation of advanced closed-loop dynamics and hybrid motion/force control in frictional environment.

The on-going contact dynamics discussed here is just a special case of complete closed-loop dynamics, the missing bit is the contact detection mechanism, which gives the most possible state in terms of contact/contact-free at the next time instant depending on the current motion state and driving torque input. So, altogether there are four possibilities in this kind of mode switching:

- Free motion → Free motion
- Free motion → Constrained motion (collision)
- Constrained motion → Constrained motion
- Constrained motion → Free motion (contact breaking)

The first and third cases are respectively conventional forward dynamics and on-going contact dynamics discussed in this paper. For contact breaking, once the simulator notices that robot is about to leave the working surface, it then abolishes the contact constraint and applies the forward dynamics equation instead. The collision case is comparatively more complicated, because it involves the analysis of impact effect, which consists of two phases: compression and decompression. In compression phase, impulse is applied to make the relative velocity of robot end-effector and environment along the contact normal to zero. In the decompression phase, another impulse takes place to cause the two colliding rigid-bodies to bounce apart, and this effect depends on the elasticity extent of collision.

Another difficulty in complete closed-loop dynamics is the effect of friction. Friction only exists in the motion space, for instance, x and y direction in the problem described in this paper. Also, we need to pick a suitable method for friction modeling, for example, seven-parameter-model [Armstrong *et al.*, 1994].

The hybrid motion/force control is to push the end-effector of the manipulator robot with a specified force while moving the end-effector along the working surface in a specified way [Yoshikawa, 2000]. The basic idea is to introduce two control loops, one for compensating position error, the other for compensating force error by using PID control strategy etc. However, so far most of hybrid control techniques assume that the surface is perfectly smooth, which is not true in practical situation. So, to find some hybrid control law that works in frictional environment is greatly valuable in both industry usage and scientific research.

References

[Armstrong *et al.*, 1994] Armstrong-Helouvry, B. *et al.* A Survey of Models, Analysis Tools and Compensation Methods for the Control of Machines with Friction. *Automatica*, 30(7): 1083-1138, 1994.

[Barrett, 1998] Barrett Technology Inc. *User Manual, Version 1.* 1998.

[Corke, 1995] Corke, P.I. A computer tool for simulation and analysis: the Robotics Toolbox for MATLAB. In *Proceedings of the 1995 National Conference of the Australian Robot Association*, pages 319-330, Melbourne, Australia, 1985. National Conference of the Australian Robot Association.

[Featherstone, 1987] Featherstone, R. *Robot Dynamics Algorithms.* Kluwer Academic Publishers, 1987.

[Featherstone, 2001] Featherstone, R. The Acceleration Vector of a Rigid Body. *The International Journal of Robotics Research*, 20(11): 841-846, 2001.

[Paul *et al.*, 1981] Paul, R.P., Shimano B. and Mayer, G.E. Kinematic Control Equations for Simple Manipulators. *IEEE Systems, Man and Cybernetics*, 11(6): 449-455, 1981.

[Walker and Orin, 1982] Walker, M.W., and Orin, D.E. Efficient Dynamic Computer Simulation of Robotic Mechanisms. *Journal of Dynamic Systems, Measurement, and Control*, 104: 205-211, 1982.

[Yoshikawa, 2000] Yoshikawa, T. Force Control of Robot Manipulators. In *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, pages 220-226, San Francisco, USA, 2000. IEEE International Conference on Robotics and Automation.