



PERGAMON

Mechatronics 12 (2002) 791–811

MECHATRONICS

Mechatronic design and control of a robot system interacting with an external environment

Prabhakar R. Pagilla ^{*}, Biao Yu

*School of Mechanical and Aerospace Engineering, Oklahoma State University,
Stillwater, OK 74078-5016, USA*

Received 26 July 2000; accepted 27 February 2001

Abstract

In this paper, an efficient mechatronic approach is used to integrate various components of a robot system interacting with an external environment. Both hardware and software integration are considered in the development of the open-architecture experimental platform. Real-time control software is developed to provide optimal use of the hardware components and available processing capability. Extensive experiments for a surface following robot task are conducted using the experimental platform. Emphasis is given to two important aspects, rate of sampling and data filtering. To minimize feedback and control algorithm switching delays, various modes of operation that involve efficient data communication and data filtering are considered and compared via experiments. Applications of the developed experimental platform include robotic surface following and surface finishing operations. © 2002 Elsevier Science Ltd. All rights reserved.

1. Introduction

Mechatronic systems are becoming common in many engineering applications [1]. The goal of a mechatronic approach is to take all the advantages that can result from an integrated design considering the best properties of the components within the system. In the 1980s, significant progress of microcomputer and power electronic technology made more complicated control algorithms realizable in mechatronic systems. In recent years, the emphasis of mechatronic systems is shifting

^{*} Corresponding author. Tel.: +1-405-744-6579; fax: +1-405-744-7873.
E-mail address: pagilla@ceat.okstate.edu (P.R. Pagilla).

to the soft part, which is handling information processing and decision making. Higher level system integration can be achieved by using mechatronic design approaches [2].

Mechatronic approaches are quite evident in the robotics field. A typical robot system is an integration of mechanical and electronic components in synergistic combination with computer control. Considering information flow in the system, a robot system for most applications sequentially performs the following: collects information from sensors, processes the information, communicates information to the controller, executes the control algorithm to generate the control signal, and amplifies the control signal to drive actuators. This is an information integration procedure that crosses the boundaries of mechanical, electrical, and computer science fields.

There are numerous applications in manufacturing industry that involve a robot interacting with an external environment. Examples include surface finishing operations such as deburring, grinding, chamfering, scraping, polishing and painting, etc. Conventional machine tools such as CNC machines are used in general to remove large amount of material to shape a part to its desired geometry. Finishing of the machined part is required to remove material in small amounts to bring the part to the required tolerance. Automation of such processes is still in its rudimentary stages. Robot manipulators are employed in manufacturing industry to automate surface finishing operations. A typical robot task performing a surface finishing operation involves: robot moving freely in its workspace, making contact with the workpiece, finishing the workpiece, and leaving the workpiece to return to the home position. This involves free motion control of the robot, transition control when the robot makes contact with the surface, and simultaneous motion and force control during surface finishing.

A robot system performing surface finishing operations can be divided into two parts, namely, hardware and software. The hardware consists of a manipulator, a computer for data acquisition and real-time control, force sensor, surface finishing tool, position and velocity sensors, digital signal processors, pneumatic system for controlling tool speed, etc. The software part consists of the off-line software for trajectory generation and simulations, real-time control software for data acquisition, data processing, data communication, and implementation of the control algorithm. To achieve high performance from such a system, a mechatronic approach to integration of all components of the system is required.

A large body of research in free motion control and constrained motion and force control has been reported in [3,4], and the references therein. A survey of different robot interaction control schemes can be found in [9]. Extensive research has been done in the area of the contact transition control problem; see [5] and their extensive bibliography. In this paper, the primary focus will be on mechatronic aspects of the experimental platform design and implementation of a switching controller for a complete robot task. Several integration modes with synchronous and asynchronous sampling that minimize the sampling and control algorithm switching delay are considered and discussed. Extensive experiments were conducted on the experimental platform for a complete robot task. In all the control experiments

emphasis is given to mechatronic aspects such as data filtering and communication between various processors in the system.

The rest of the paper is organized as follows: Section 2 gives a summary of the dynamic model and the switching control design. In Section 3, mechatronic design of the experimental platform is given. Several system integration modes that involve synchronous and asynchronous sampling are given in Section 4. Section 5 gives the experimental procedure and experimental results of surface following of a robot. Conclusions and future work are given in Section 6.

2. Dynamic model and control

In this section, the dynamic model of a robot interacting with an external environment together with the control strategy for a complete robot task will be presented. Details of the dynamic model and controller design can be found in [6–8].

Let the kinetic and potential energy functions of the n -link robot be given by $\mathcal{H}(q, \dot{q}) = \frac{1}{2} \dot{q}^T M(q) \dot{q}$ and $\mathcal{P}(q)$, where $q, \dot{q} \in \mathbb{R}^n$ are the generalized position and velocity, respectively, and $M(q) \in \mathbb{R}^{n \times n}$ is the symmetric positive definite mass matrix. Let the geometric constraint on the robot be modeled by the following unilateral constraint [8],

$$\phi(x(q)) \leq 0, \tag{1}$$

where $x(q)$ is the Cartesian position of the robot end-effector. Define the following orthogonal projection matrix whose image represents the normal direction of the constraint,

$$P_\phi(q) = [\nabla \phi(x(q))] [\nabla \phi(x(q))]^T / \|\nabla \phi(x(q))\|^2,$$

where $\nabla \phi(x(q))$ is the gradient of the function $\phi(x(q))$. The kernel of $P_\phi(q)$ gives the tangential direction of the constraint surface, which is given by

$$Q_\phi(q) = I_n - P_\phi(q),$$

where I_n denotes the $n \times n$ identity matrix. The gradient of the constraint surface equation is assumed to be known and hence, both the normal and tangential projection matrices are known. The dynamics of the geometrically constrained robot is

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau + J^T(q)f, \tag{2}$$

where $C(q, \dot{q}) \in \mathbb{R}^{n \times n}$ is the matrix composed of Coriolis and centripetal terms, $g(q) \in \mathbb{R}^n$ is the gravity vector, $\tau \in \mathbb{R}^n$ is the generalized force applied by the motors at each joint of the robot, $f \in \mathbb{R}^n$ is the contact force due to the constraint, which can be written as

$$f = n(x(q))f_n + t(x(q))f_t, \tag{3}$$

where $n(x(q))$ and $t(x(q))$ represent the unit normal and tangential directions of the constraint surface, respectively, in the Cartesian space, and f_n and f_t represent the normal force and tangential force magnitudes, respectively. The value of the contact force, $\|f\|$, depends on the activation/deactivation of the constraint, that is, if

$\phi(x(q)) < 0$ then $\|f\| = 0$, and if $\phi(x(q)) = 0$ then $\|f\| \geq 0$. In the following section the dynamic model of a robot for a complete task is given.

2.1. Dynamic model for a complete task

A complete task of the robot in the presence of the unilateral constraint can be divided into three phases: (a) when $\phi(x(q)) < 0$, then the robot is said to be in the unconstrained motion phase, (b) when $\phi(x(q)) = 0$ and the velocity normal to the surface is zero, then the robot is said to be in the constrained motion phase, and (c) transition from free motion phase to constrained motion phase is termed as the transition phase. Assuming friction on the surface, the tangential force f_t and the normal force f_n are related by the coefficient of friction, ξ , as

$$f_t = \xi f_n. \quad (4)$$

The above expression (4) is generic in the sense that it can be used for both surface following with friction and also surface finishing operations. In the case of surface finishing operations, f_n is also called the coefficient of grinding friction [10]. Thus, the dynamic model for the complete task is given by:

- Unconstrained motion phase:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau. \quad (5)$$

- Transition phase:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau \quad \text{and} \quad \dot{q}_+ = \mathcal{D}(q, \dot{q}_-). \quad (6)$$

- Constrained motion phase:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau + v(q)f_n + v'(q)\xi f_n, \quad (7)$$

where \dot{q}_+ and \dot{q}_- are the post- and pre-impact velocities, respectively, $\mathcal{D}(q, \dot{q}_-)$ maps the pre-impact velocity to post-impact velocity which is obtained from an impact model, $v(q) = J^T(q)n(x(q))$ maps the normal surface force magnitude into corresponding joint forces and $v'(q) = J^T(q)t(x(q))$ maps the tangential surface force magnitude into corresponding joint forces. In the transition phase the robot end-effector may bounce on the constraint surface. Notice that the robot dynamics in the transition phase is the same as that of the unconstrained motion phase except at impact points. At impact points, the post-impact velocity is obtained from pre-impact velocity and the configuration of the robot via an impact model. The constrained motion phase starts when the robot end-effector makes a stable contact with the surface.

2.2. Control design

The control goal in the unconstrained motion phase is to track the desired motion trajectory considering manipulator model uncertainties. During the constrained motion phase, the control goal is to simultaneously track the desired motion in tangential direction and regulate the desired force normal to the constraint surface.

Switching directly from motion control to simultaneous motion and force control could lead to severe repeated impacts of the robot end-effector on the surface. A stable transition controller assures that repeated impacts do not occur. The controller for a complete task is a switching controller that is composed of a control algorithm for each phase. The control algorithm in each phase is given below.

2.2.1. *Unconstrained motion phase*

For this phase, a model-based control law is chosen as follows:

$$\tau = Y(q, \dot{q}, \ddot{q}_r)\beta - F_v e_v, \tag{8}$$

where F_v, Γ are positive definite gain matrices, $q_d, \dot{q}_d, \ddot{q}_d$ are the desired joint position, velocity, and acceleration, respectively, $\dot{q}_r = \dot{q}_d - A_p e$, A_p is a positive definite gain matrix, $e_v = \dot{q} - \dot{q}_r$, $\beta \in \mathbb{R}^p$ is a vector of unknown parameters of the robot, and the matrix $Y(q, \dot{q}, \ddot{q}_r) \in \mathbb{R}^{n \times p}$ is obtained from

$$Y(q, \dot{q}, \ddot{q}_r)\beta = M(q)\ddot{q}_r + C(q, \dot{q})\dot{q}_r + g(q). \tag{9}$$

Eq. (9) is generally known as the linear parameterization property, which gives the dynamics of the robot as a known matrix times a vector of unknown coupled inertial parameters of the robot such as link masses and lengths.

2.2.2. *Transition phase*

In the transition phase, the control goal is to bring the robot end-effector back onto the surface. We choose the following control law:

$$\tau = Y(q, \dot{q}, \ddot{q}_r)\beta - F_v e_v - \lambda_{tn} P_\phi(q) \text{sgn}(e_{vn}), \tag{10}$$

where $\text{sgn}(e_{vn})$ represents the component-wise sign of the vector e_{vn} , $\dot{q}_r = \dot{q}_d - \lambda_p e = Q_\phi(q)\dot{q}_d - \lambda_p e$, $e_v = \dot{q} - \dot{q}_r$, $e_{vn} = P_\phi(q)e_v$, and λ_p is a positive gain. Notice that there is an additional discontinuous term in the normal direction in the transition phase control law, (10). This additional discontinuous term in the control law assists in bringing back the robot end-effector onto the constraint surface.

2.2.3. *Constrained motion phase*

For this phase, we choose the following control law:

$$\tau = Y(q, \dot{q}, \ddot{q}_r)\beta - F_v e_v - v(q)f_{nd} - v'(q)\hat{\xi}f_n \tag{11}$$

and the adaptation law for the grinding coefficient as

$$\dot{\hat{\xi}} = -\gamma_f v^T(q)e_v f_n, \tag{12}$$

where f_{nd} is the desired normal force, $e_{fn} = f_n - f_{nd}$, γ_f is a positive gain, $e_v = \dot{q} - \dot{q}_r$, and

$$\begin{aligned} \dot{q}_r &= Q_\phi(q)[\dot{q}_d - A_p e] + \alpha_f \bar{v}(q)e_{vfn} \\ e_{vfn} &= \int_0^t e_{fn}(\omega) d\omega, \end{aligned}$$

where α_f is a positive gain, and $\bar{v}(q) = v(q)/\|v(q)\|^2$.

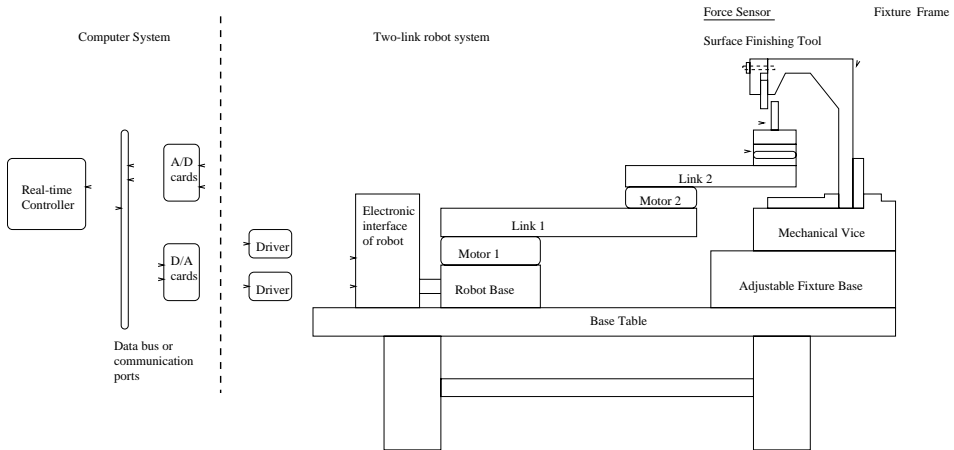


Fig. 1. Robot system.

The overall controller for a complete task is a switching controller composed of the controllers from each individual phase. An event-based control switching strategy is developed for the complete task. In the unconstrained motion phase, a model-based adaptive control law is applied for trajectory tracking considering robot model uncertainties. Upon contact with the surface, control is switched to the one given by Eq. (10). In the transition phase the primary goal is to stabilize the robot end-effector onto the surface. When the end-effector is on the surface, we switch to the constrained motion control law given by Eq. (11). An event-based online trajectory planning is utilized along with the control switching strategy. The desired trajectory is pre-computed based on the initial knowledge of the location of the constraint surface. Immediately after the first impact, pre-computed trajectories are modified using the projection matrices, such that the desired velocity normal to the constraint surface is zero. After the completion of the transition phase, the trajectory planner prescribes a desired force in the direction normal to the constraint surface and a desired joint space trajectory projected into the tangential direction of the constraint surface.

3. Mechatronic design of the experimental platform

A mechatronic design of a robot interacting with an external environment involves a number of hardware components coupled with the software. An efficient integration of the hardware components together with the optimal use of all the components is essential to obtain maximum performance from the system. The hardware and software aspects of the robot system are described in sections below.

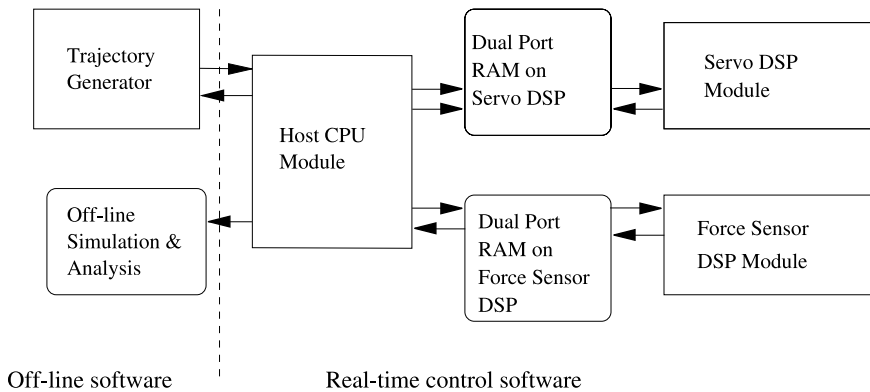


Fig. 2. Sketch of software of the robot system.

3.1. Hardware

The robotic surface finishing system consists of a two-link manipulator, computer for real-time control, force sensor, surface finishing tool, a vice, and an adjustable fixture frame. Fig. 1 illustrates the complete hardware platform of the robotic surface finishing system.

The main part of the robot system is a two-axis direct drive manipulator, which is shown in Fig. 1. Each axis is driven by an NSK Megatorque direct drive servo-motor which is capable of up to 3 rps maximum velocity and position feedback resolution of up to 156,400 counts per revolution. The NSK Megatorque motor consists of a high torque direct drive brush-less actuator, a high-resolution brush less resolver, and a heavy duty precision bearing. The Megatorque motor is capable of producing extremely high torque at low speeds suitable for direct drive applications. The base motor can deliver up to 245 N m of torque output, and the elbow motor can deliver up to 40 N m torque output.

The computer system consists of the direct drive manipulator controller (host computer), a servo DSP, a force sensor DSP, and I/O cards associated with the sensors. The overall control system is a three-processor system consisting of a host Pentium processor, a servo DSP, and a force sensor DSP. The host processor is used for reference trajectory generation, user interface, and coordination of other processors. Real-time control is performed using the servo DSP board. The force sensor DSP is used for processing of raw force data. The three processor architecture provides flexibility in terms of collecting force data from the force sensor and position/velocity data from the motor resolvers at different sampling rates.

A six-axis force/torque sensor is mounted on the end of the second link of the robot manipulator. The force sensor has an on-board controller, which can provide force sensor data up to a rate of 4 kHz. The primary function of the force sensor DSP is to convert strain gauge data to Cartesian force/torque components. Provision is also available for tool weight offset, data filtering, temperature compensation, and coordinate frame rotation.

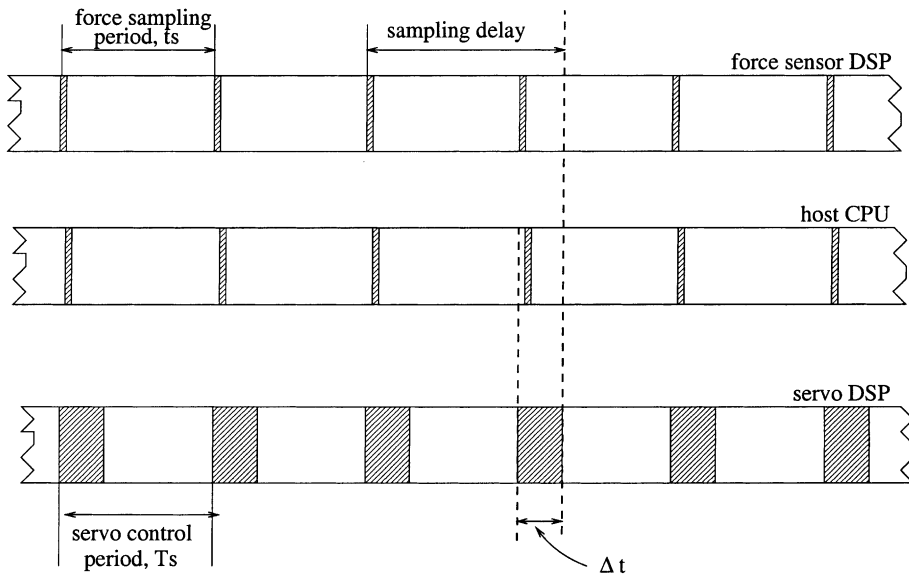


Fig. 3. Mode 1: Synchronous operation.

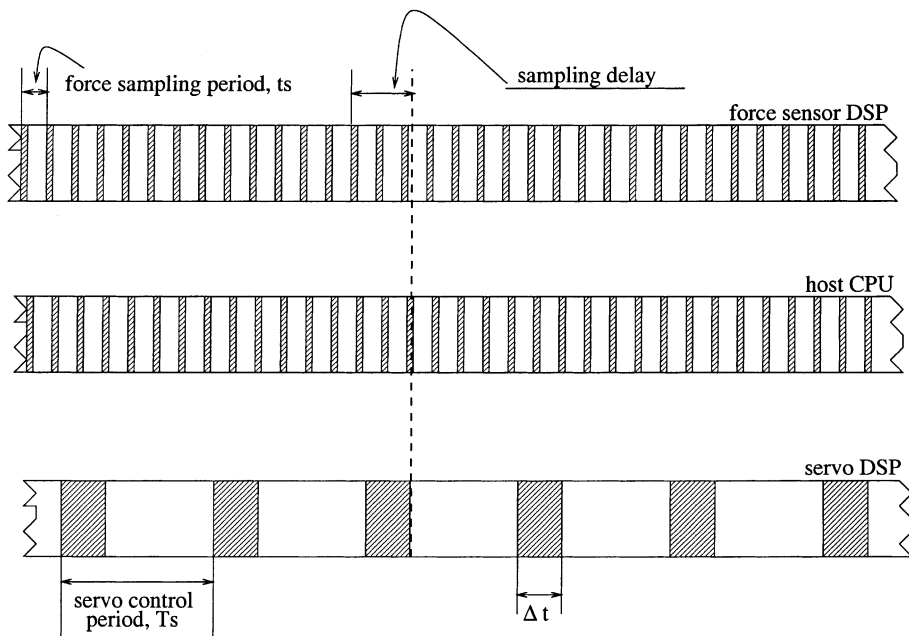


Fig. 4. Mode 2: Asynchronous operation.

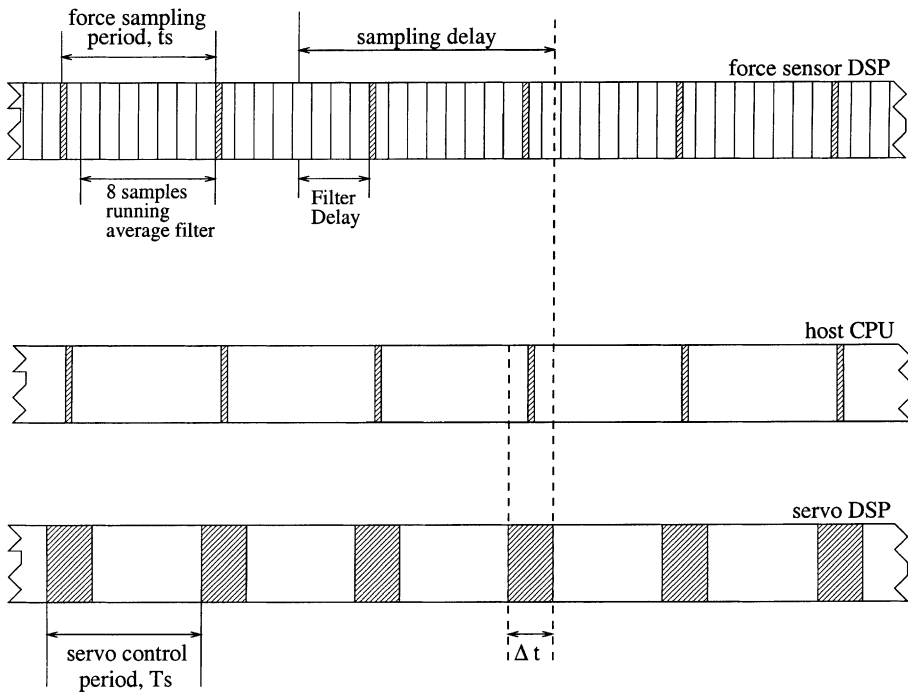


Fig. 5. Mode 3: Asynchronous operation, running average filter on force sensor DSP.

The robotic surface finishing tool used in the system is a high-speed multi-position robotic deburring tool with integrated axial compliance. One of the important feature of this tool is its ability to provide axial motion, which although limited in travel, 8 mm maximum travel, can provide the much needed axial movement of the tool in the plane perpendicular to the robot motion. In robotic applications, deviation between the burr edge and the desirable path exists. The axial motion of the tool can compensate for this type of deviation. The axial motion and the tool speed are controlled via a pneumatic system.

3.2. Software

Corresponding to the multi-processor structure, there are three software modules running on the three processors. The software modules running on the host computer, servo DSP and force sensor DSP exchange data via the dual-port RAM as shown in Fig. 2. The data saved in the dual-port RAM on the servo DSP can be read by the real-time display module and uploaded to host computer for further analysis. The dual-port RAM operation module sends commands to force sensor DSP to read force data. At the same time, the module delivers force data to the dual-port RAM on the servo DSP board. The run-time executive block in the host

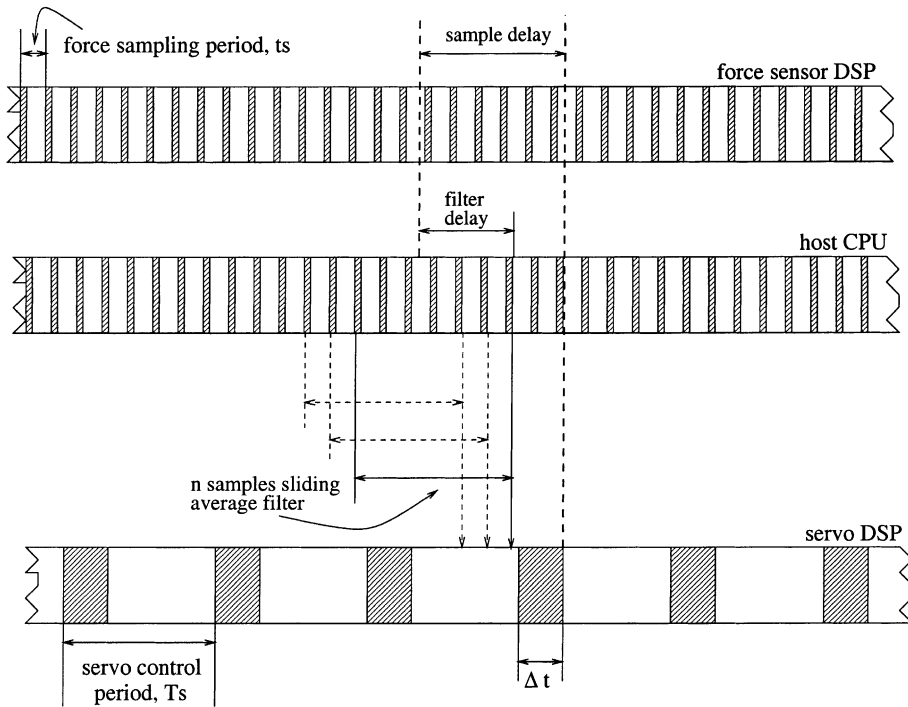


Fig. 6. Mode 4: Asynchronous operation, fixed length sliding average filter on host processor.

CPU acts as a user interface; coordinates the servo control DSP, the force sensor DSP, and the host CPU; handles real-time communication of data among the three processors; and performs safety checks and “housekeeping functions”.

For a multi-processor system, time synchronization is another important aspect that affects the performance of the entire software system. Managing data exchange operations is critical to proper functioning of the real-time system. This problem will be addressed in the following section.

4. Implementation of the switching controller

An important aspect during implementation of the switching controller is to detect precisely when the robot end-effector first makes contact with the constraint surface. This helps in reducing the time delay in switching to the transition phase controller. Contact of the end-effector with the surface is detected when the force signal exceeds a threshold value. Since non-zero impact velocity may be involved it is crucial to detect the first contact with minimal time delay so that the controller can be switched to the transition phase controller. Time delay due to switching and sampling plays a key role in the stability and performance of the closed-loop control system. The

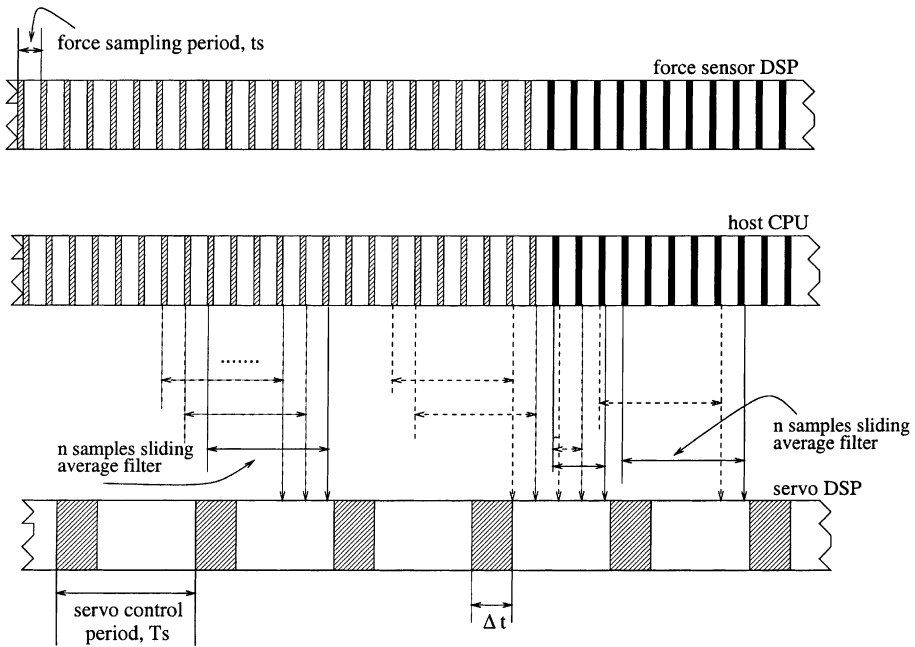


Fig. 7. Mode 5: Asynchronous operation, flexible length sliding average filter on host processor.

Table 1
Sample and switching delay

Mode	SMD	SWD
1	$T_s + \Delta t$	$T_s + \Delta t$
2	$(\Delta t, t_s + \Delta t)$	$t_s + \Delta t$
3	$(\Delta t + \Delta T_f, T_s + \alpha_{if})$	$(\Delta T_f + \Delta t, 2T_s + \Delta T_f + \Delta t)$
4	$(\Delta t + \Delta T_f, t_s + \alpha_{if})$	$(\Delta T_f + \Delta t, \Delta T_f + T_s + \Delta t)$
5	$(\Delta t + \Delta T_f, t_s + \alpha_{if})$	$(\Delta t, T_s + \Delta t)$

method in which force data is acquired and processed is critical to the overall real-time system performance. Also, it is generally known that the raw force signal is very noisy. Filtering of the raw force signal is necessary before it can be used for feedback. For the multi-processor robot system, we consider five modes of operation. These five modes of operation are distinguished by considering various sampling frequencies and filtering strategies. The performance of the overall closed-loop real-time control system in terms of the sampling delay and control switching delay is discussed and compared for these modes. The five modes of operation are:

1. Mode 1: Synchronous operation mode (Fig. 3).
2. Mode 2: Asynchronous operation mode with multi-frequency sampling (Fig. 4).
3. Mode 3: Asynchronous operation mode with multi-frequency sampling using fixed length filter on the force sensor DSP (Fig. 5).

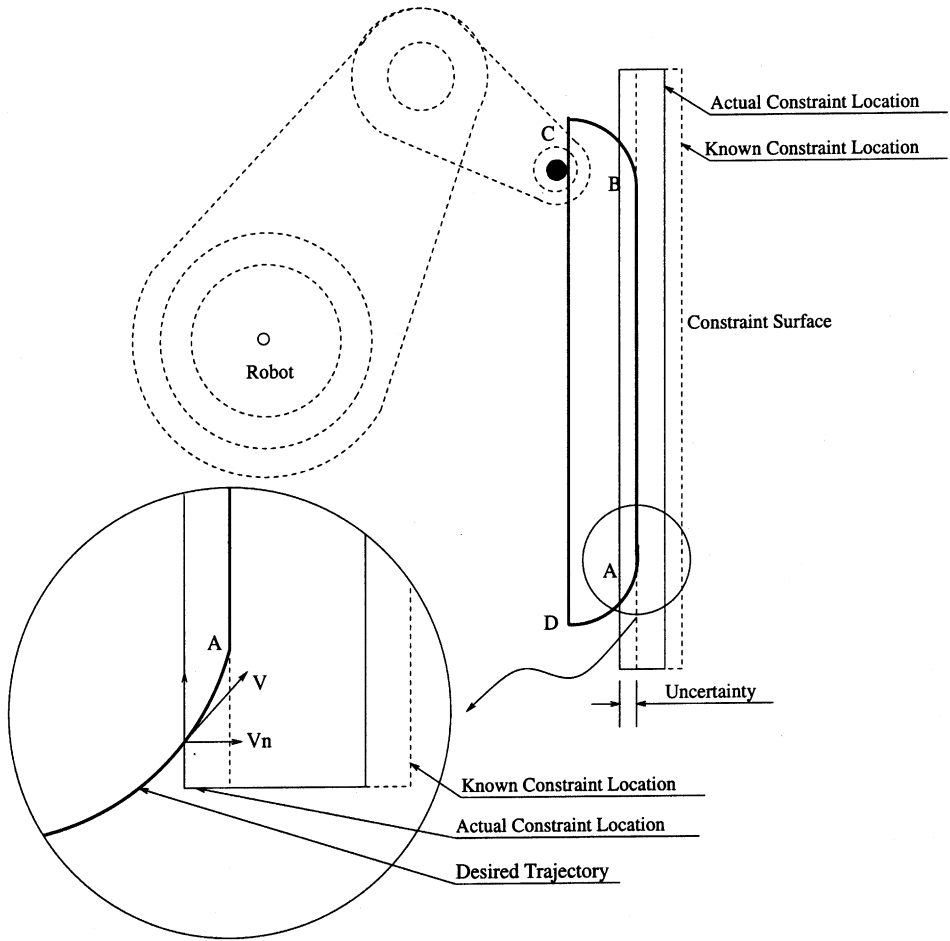


Fig. 8. Robot and the constraint.

- 4. Mode 4: Asynchronous operation mode with multi-frequency sampling using sliding fixed length filter of the force signal on the host processor (Fig. 6).
- 5. Mode 5: Asynchronous operation mode with multi-frequency sampling using sliding variable length filter of the force signal on the host processor (Fig. 7).

Figs. 3–7 illustrate the five operation modes, where T_s and t_s represent the control sampling period and the force sampling period, respectively, Δt represents the time required by the servo DSP to run the control algorithm once and to read the force signal at the beginning of its program code, and ΔT_f represents the time delay due to filtering of the force data.

In mode 1, Fig. 3, the force sensor DSP and the servo DSP use the same sampling period, i.e., $t_s = T_s$, and the operations are synchronized. The signal processing

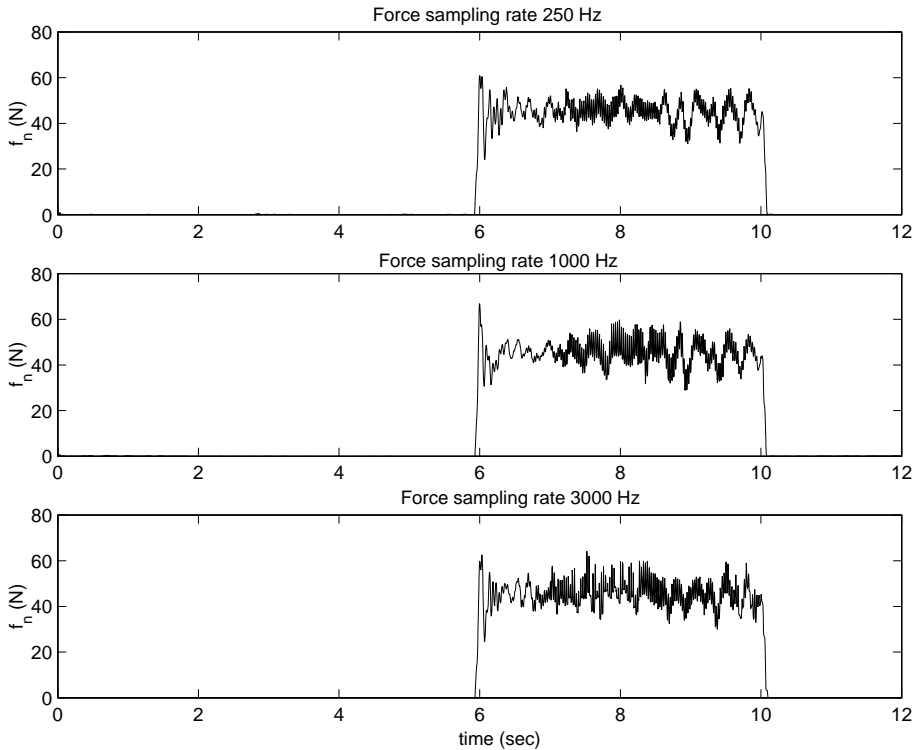


Fig. 9. Modes 1 and 2: Normal force.

module runs on an interrupt mode and the interrupt is triggered once per sample period. The data exchange module on the host computer works in the polling mode. When the force signal is ready, the host computer fetches the data and delivers it to the servo DSP. As shown in Fig. 3, the force sampling delay, which is defined as the time delay between sampling of the force signal and the control output being sent to the robot drivers, is $T_s + \Delta t$.

To minimize the force sampling delay asynchronous operation is considered in mode 2 by utilizing the availability of faster force signal sampling rate. Since the computational load of computing the control input at each sampling period is on the servo DSP, increasing the force sensor sampling period does not affect the time required for implementation of the control algorithm. The force sensor that is used in this experimental platform can provide force data up to a rate of 4 kHz. Fig. 4 shows asynchronous operation, where the force sensor sampling frequency is considerably higher than the servo sampling frequency. In this case, the maximum sampling delay is $t_s + \Delta t$. Thus the sampling delay is minimized without changing the servo sampling frequency.

To alleviate the problem caused by force signal noise, low-pass filtering of the force signal is considered. In mode 3, an eight sample running average filter of the

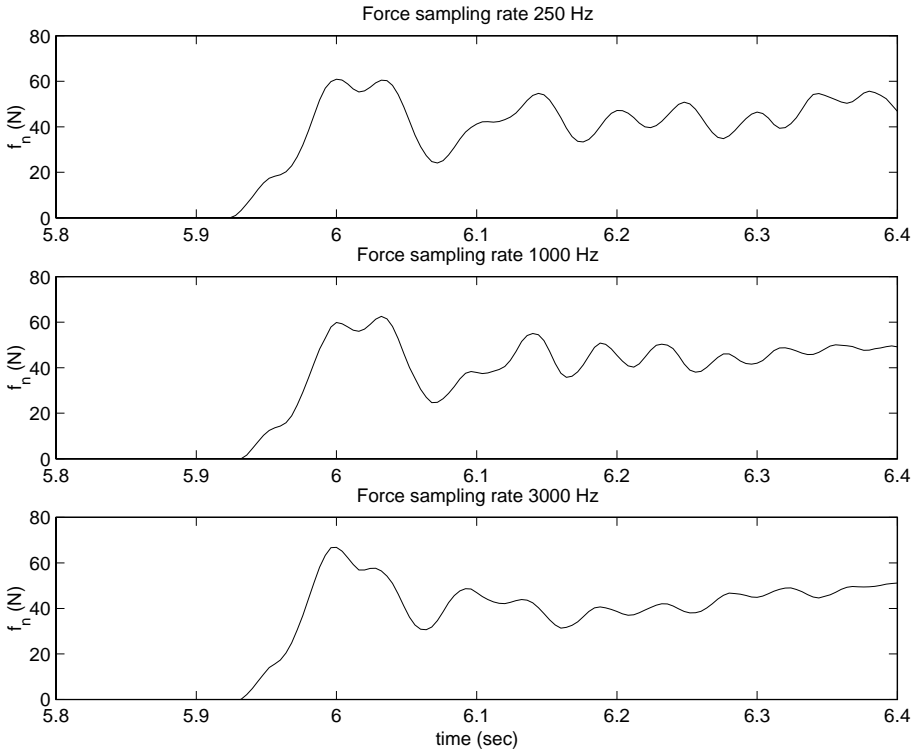


Fig. 10. Modes 1 and 2: Normal force transient.

force signal is implemented on the force sensor DSP. Instead of sending the force data signal to the host processor, the force sensor DSP collects eight samples, calculates the average, then transmits the average value to the host processor, as shown in Fig. 5. In mode 3, a force sampling frequency of 2 kHz is used. The maximum and minimum time delay due to sampling are $T_s + \Delta t + \Delta T_f$ and $\Delta t + \Delta T_f$, respectively.

In mode 4, a fixed length sliding average filter is implemented on the host processor as shown in Fig. 6. Host processor working in the polling mode fetches the force signal at a rate of force sampling frequency, $1/t_s$, which is much larger than $1/T_s$. Using the previous $n - 1$ points of the force signal, host processor implements an n sample length sliding filter at every data sampling period. Fig. 6 indicates that the maximum force sample delay is $t_s + \Delta t + \Delta T_f$, and the minimum sample delay is $\Delta t + \Delta T_f$.

Recall that the controller for a complete task is a switching type of control algorithm. Switching of the control algorithm from the unconstrained motion phase to the transition phase is triggered based on a threshold force signal level. If the end-effector makes contact with the surface just before the servo DSP reads the force data for the next control period, it is possible that the force signal after filtering is less than the threshold value. This leads to a time delay of one control period in

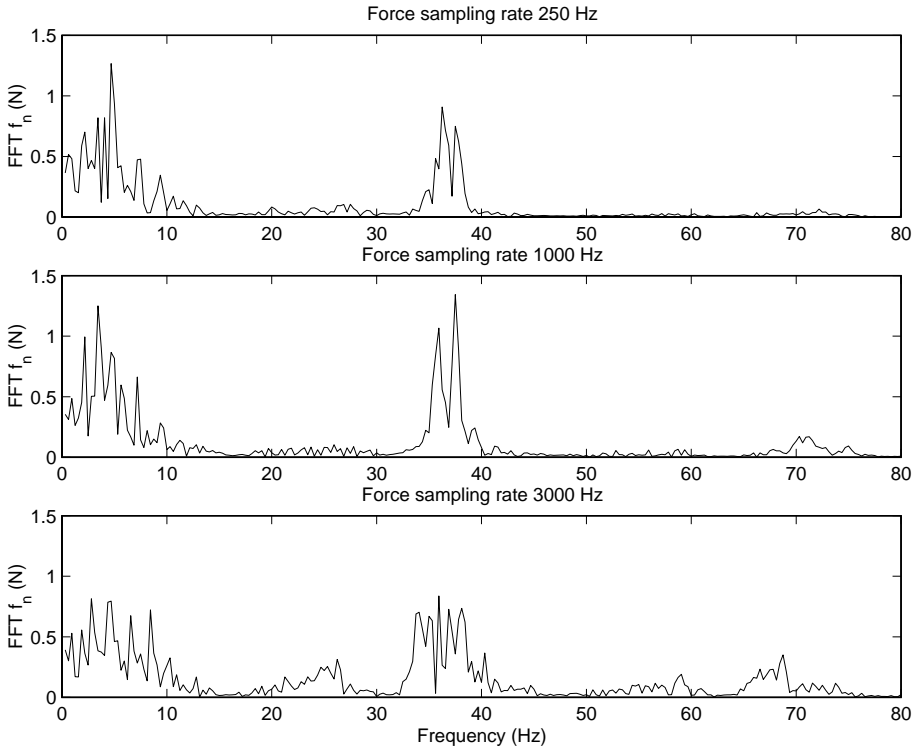


Fig. 11. Modes 1 and 2: FFT of normal force.

switching to the transition controller. To circumvent this problem, flexible length sliding filter is proposed in mode 5, Fig. 7. The solid black blocks in Fig. 7 indicate that the force signal is larger than the threshold value.

Mode 5 operation is similar in operation to mode 4 except when the host processor reads a force signal larger than the threshold value, the filter length is modified to one. Further, the length of the filter is increased as the new force data is available until it equals the length of the fixed filter length. Using mode 5, controller can switch to transition control algorithm at the earliest control period after the end-effector makes contact with the surface. In mode 5, the maximum and minimum force sample delay are $t_s + \Delta t + \Delta T_f$ and $\Delta t + \Delta T_f$, respectively. Although the force sampling delay is the same for mode 4 and mode 5, the control algorithm switching delay is different. For mode 4, the range of algorithm switching delay at impact is $(\Delta T_f + \Delta t, \Delta T_f + T_s + \Delta t)$, and for mode 5 it is $(\Delta t, T_s + \Delta t)$. This time delay plays a crucial role in improving the overall system performance in the presence of constraint uncertainty.

Table 1 gives a summary of the range of force sampling delay and switching delay for the five modes. The following key is used in the table. SMD: sample delay, and SWD: switching delay.

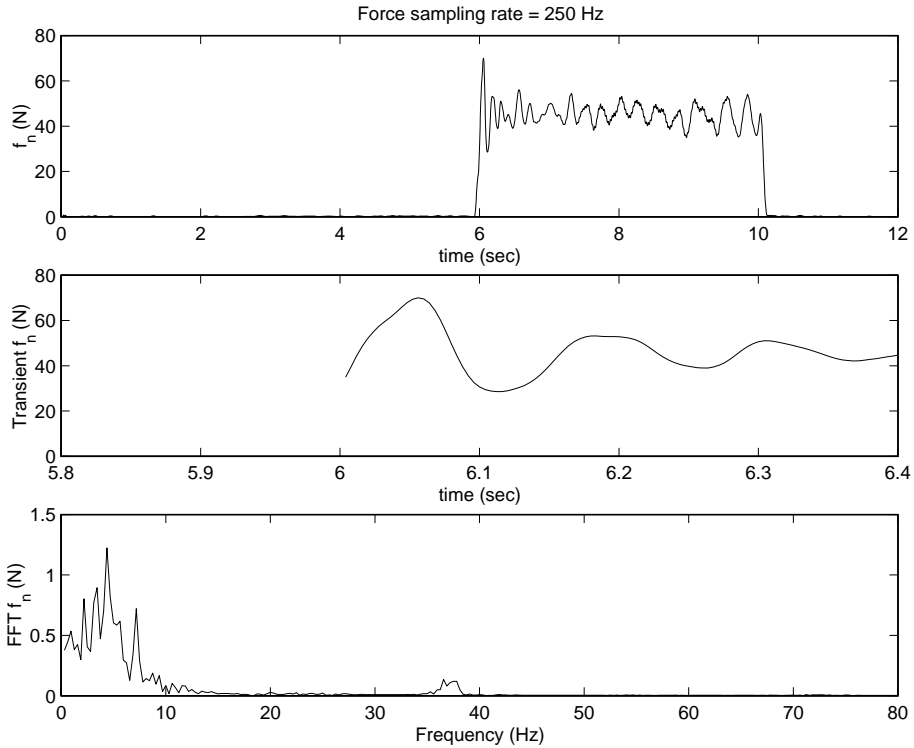


Fig. 12. Mode 3: Normal force, transient and FFT.

5. Experimental results

The constraint surface $\phi(x(q))$ is chosen as a rigid straight wall, which is a thick aluminum sheet firmly held by a vice as shown in Fig. 1. Fig. 8 illustrates the two-link robot (top view), the constraint surface, and the desired trajectory of the robot (DABC). The desired trajectory is constructed based on the known constraint location. As shown in Fig. 8, uncertainty in the location of the constraint surface leads to the robot end-effector impacting the surface at contact with non-zero normal velocity. Different levels of constraint uncertainty were considered. Extensive experiments were conducted using the five modes described in the previous section. A representative sample of the experimental results are presented and discussed. All the experiments use a control sampling frequency of 250 Hz, i.e., $T_s = 4$ ms. The cycle time to traverse the desired trajectory DABC as shown in Fig. 8 is 12 s. The desired normal force during the constrained motion phase is 45 N. Force signal, its transient at contact, and its FFT are plotted and analyzed.

Experimental results for the complete task in mode 1 and mode 2 are shown in Figs. 9–11. The top plot in all the three figures correspond to synchronous operation, i.e., mode 1. Two other force sampling frequencies, 1 and 3 kHz, are considered to

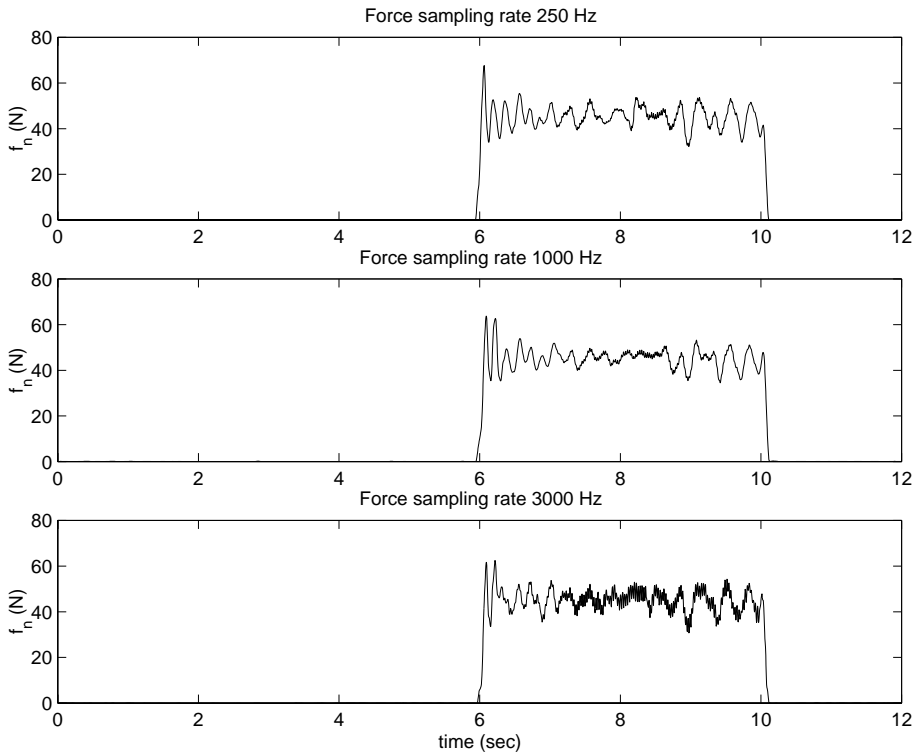


Fig. 13. Mode 4: Normal force.

compare the performance of synchronous operation with asynchronous operation. Choosing higher force sampling rate reduces force signal delay. Fig. 10 is the transient force signal of three experiments after contact of the robot end-effector with the constraint surface. The experimental data shows that choosing higher force sensor sampling frequency results in better transient force response. Fig. 11 shows the existence of high frequency noise components in the force signal. The amplitude and frequency of the noise is influenced by the force sensor sampling frequency. Higher force sampling frequency introduces high frequency components in the force signal.

Notice the appearance of frequency components in the range of 30 to 40 Hz in the force signal, Fig. 11. These components are undesirable as the force/position control algorithm in the constrained motion phase uses force feedback signal, thus leading to noisy control torque for each motor. To alleviate this noise problem, low pass filtering of the force signal is considered in mode 3. In mode 3, an eight sample running average filter is implemented on the force sensor DSP. The servo DSP control sampling frequency is 250 Hz, the force sampling frequency is 2 kHz. Fig. 12 shows the normal force signal, its transient and its FFT. Comparison of the force signal in

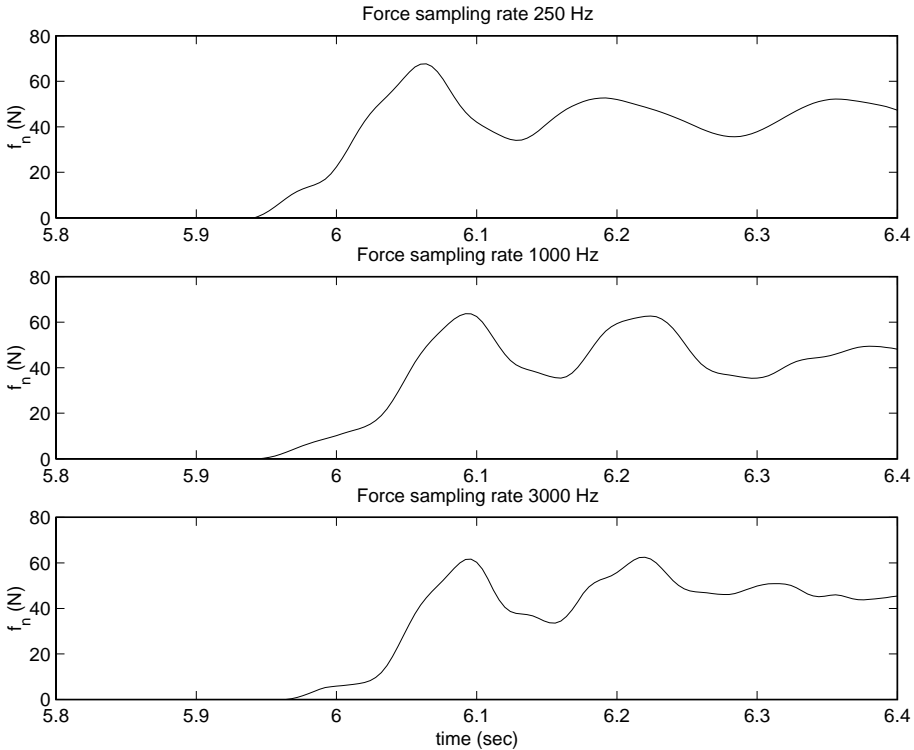


Fig. 14. Mode 4: Normal force transient.

mode 1 and mode 2 with mode 3 shows that the high frequency components no longer appear in the force signal in mode 3. Due to the time delay introduced by the low pass filter, the rise time of the normal force transient response becomes longer for mode 3 when compared with mode 1 and mode 2.

To improve the performance of the switching controller as explained in Section 3, mode 4, experiments were conducted with a fixed length sliding average filter on the host processor. Three force sensor sampling frequencies, 250 Hz, 1 and 3 kHz, were considered in the surface following experiments using mode 4. Figs. 13–15 show the normal force, its transient, and its FFT, respectively. Comparing the experimental results of mode 2 given by Fig. 10, with results of mode 4, we can observe that the rise time increases because of the time delay, ΔT_f , introduced by filter. The experimental data shows that the choice of higher force sensor sampling frequency results in better transient response.

When there is uncertainty in the constraint location as shown in Fig. 8, there is a tendency of the robot end-effector to bounce on the surface. In this case, it is critical that the controller switches to the transition phase control algorithm as quickly as possible after the first impact. The force signal delay due to filtering may delay the switching of the controller. To solve this problem, mode 5 is implemented. In ex-

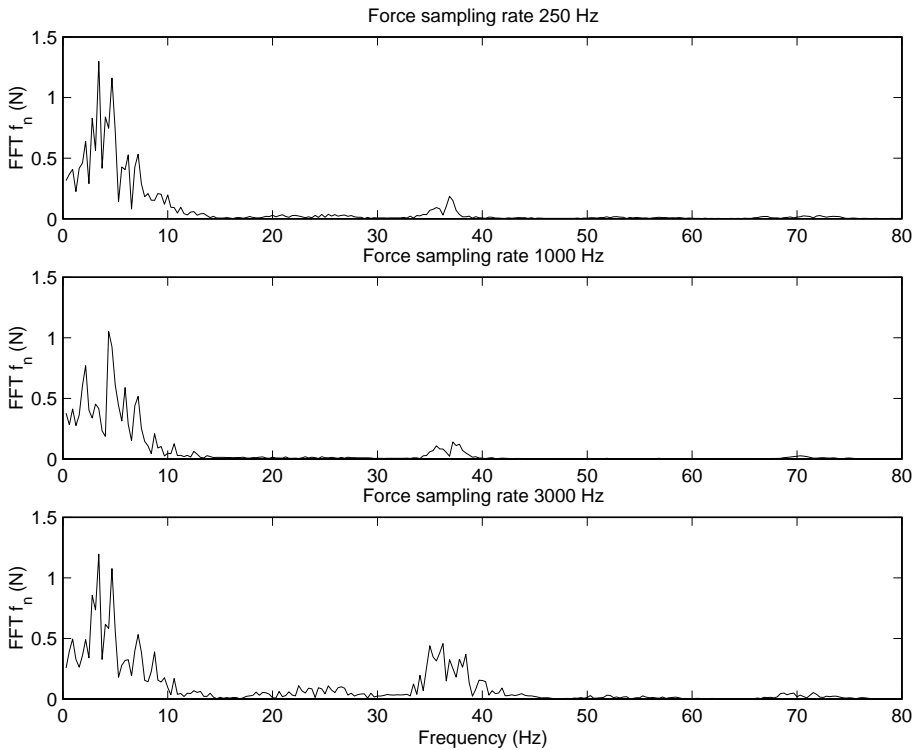


Fig. 15. Mode 4: FFT of normal force.

periments, it is assumed that the uncertainty in the constraint surface location is 5 mm, the control sampling frequency is unchanged at 250 Hz, three force sensor sampling frequencies 250 Hz, 1 and 3 kHz, were considered. Fig. 16 shows the transient response of the force signal in mode 4 and mode 5. Notice that mode 5 has faster rise time and a smaller force peak at impact when compared with mode 4. This is mainly due to the fact that the switching delay is reduced in the case of mode 5. In mode 5, a flexible length sliding filter is used as explained in Section 3. The experimental data also shows that the choice of higher force sensor sampling frequency results in smaller force signal overshoot over the desired force level.

In summary, the following observations can be made based on the experimental results. In mode 2, the force sampling delay and algorithm switching delay are reduced by using higher force sampling frequency. Raw force signal obtained directly from the force sensor is noisy and thus cannot be used directly in the control algorithm. In mode 3, a running average filter is implemented on the force sensor DSP. Filtering introduces time delay but the control input torque is smooth when compared to the one given by mode 2. In mode 4, force signal delay due to filtering can be reduced by increasing the force sensor sampling frequency and running the filter

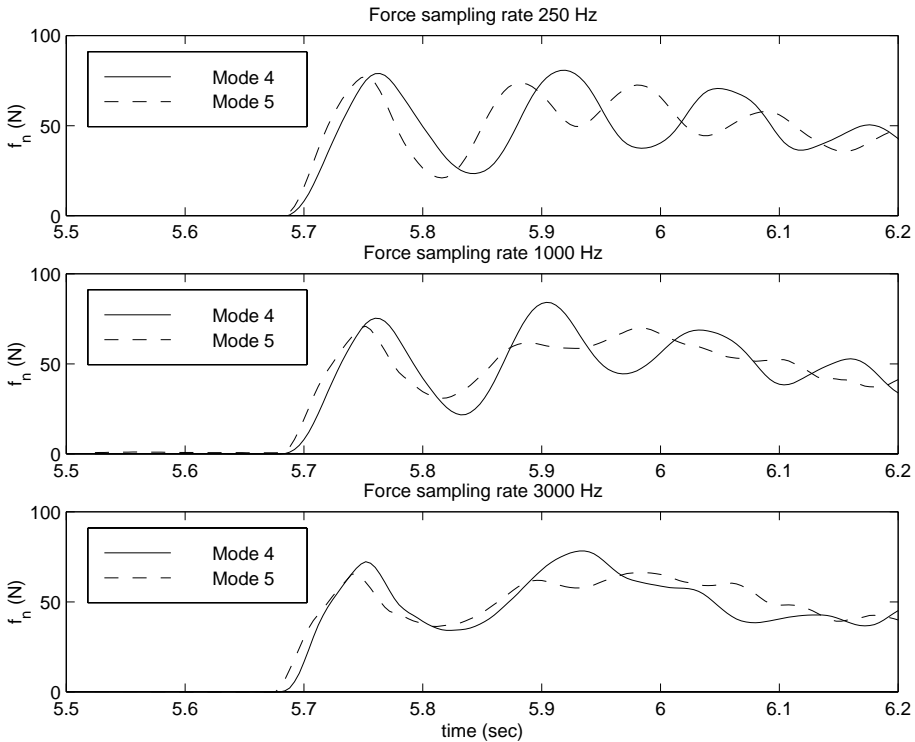


Fig. 16. Modes 4 and 5: Normal force transient.

on the host processor. Implementation of the controller using mode 5 can reduce the algorithm switching delay.

6. Conclusion

The main focus in most of the literature on robots interacting with an external environment has been on dynamic modeling and control design for a complete task. There is a lack of adequate effort towards efficient implementation of the controller to get better closed-loop performance. In this paper, an efficient mechatronic design has been considered in developing a robotic experimental platform interacting with an external environment. In applications such as surface following and surface finishing, the normal force control during the constrained motion phase is important to obtain the required finish. In this paper, analysis of the performance of the closed-loop system with different implementation modes is considered. The implementation modes differ with respect to the communication between different processors and force signal filtering strategies. The focus in this paper was on surface following experiments with a desired normal force. Future work will focus on implementation

of the strategies described in this paper to surface finishing applications such as polishing, deburring and chamfering.

References

- [1] Auslander DA. What is mechatronics?. *IEEE/ASME Trans Mechatronics* 1996;1(1):5–9.
- [2] Hashimoto H. Intelligent mechatronics. In: *Proceedings of the 19th Annual International Conference on Industrial Electronics, Control and Instrumentation*, vol. 1; 1993. p. 445–50.
- [3] Arimoto S. *Control theory of non-linear mechanical systems – a passivity-based and circuit-theoretic approach*. NY: Oxford University Press; 1996.
- [4] Slotine JJE, Li W. *Applied nonlinear control*. Englewood Cliffs, NJ: Prentice-Hall; 1991.
- [5] Pagilla PR, Tomizuka M. Contact transition control of nonlinear mechanical systems subject to a unilateral constraint. *ASME J Dyn Syst Measurement Control* 1997;119:749–59.
- [6] Pagilla PR, Yu B. Adaptive motion and force control of robots performing a complete task. In: *Proceedings of the 1999 ASME Design Engineering Technical Conferences*. Las Vegas, NV; September 1999.
- [7] Pagilla PR, Yu B. Design and experimental evaluation of a stable transition controller for geometrically constrained robots. In: *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*. San Francisco, CA; 2000.
- [8] Pagilla PR, Yu B. A stable transition controller for constrained robots. *IEEE/ASME Trans Mechatronics* 2001;6:65–74.
- [9] Chiaverini S, Siciliano B, Villani L. A survey of robot interaction control schemes with experimental comparison. *IEEE/ASME Trans Mechatronics* 1999;4(3).
- [10] King R, Hahn R. *Handbook of modern grinding technology*. London: Chapman and Hall; 1986.