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

Application of Shared Telerobotic Control for Contour Following Processes

Patrick Lim, Jeffery Yang, Nicholas Hildreth and Werner Friedrich
Automation Systems Team
Industrial Research
24 Balfour Road, Auckland,
New Zealand
p.lim@irl.cri.nz

Abstract

The paper outlines the development of a telerobotic control system with a focus on maximising the versatility of both production equipment and human operators. This was achieved by integrating sensory feedback from end effectors into the human-machine control loop. For most robotic applications, the robot executes a series of point-to-point movements along a predefined path. When an application requires the tool to *follow* an undefined object geometry, the process of contour following demands continuous adaptation of its path and complex articulation of the robotic arm. To overcome the resultant limitation in processing speed, a 3 DOF compliant sensing tool was developed. It features tactile sensors to determine surface geometry and active compliance to allow the tool to maintain orthogonality with surface during rapid contour following. The adaptive controller derives a vector-based projection of the robot path based on sensory and user data. This paper presents the concept, tool development and laboratory testing involving a 6-DOF industrial robot.

1 Introduction

Robotic contour following is a process that demands continuous adaptation of its path and therefore a suitable application for this system. It requires real time kinematic calculations to be performed with very fast sensing update rate for optimal adaptive control. As a result the processing speed for such applications is greatly reduced and tight boundaries are imposed on deviations from the path. The task becomes more demanding when the robotic tool must be normal to the surface requiring the robot to make complex articulated motions to position and orientate the tool while traversing a non-linear surface. This further reduces the processing speed.

One particular application of this nature is steam vacuuming of bovine carcasses. Once the pelt has been cut during the dressing stage, the cut path must be sterilised. Steam vacuuming is preformed with a rectangular nozzle that simultaneously injects steam and vacuums over the cut path. For an effective vacuum, the nozzle face should have a good seal with the carcass surface. Manually, the operator has to move the nozzle along the cut path while adapting to the surface profile while simultaneously orientating the nozzle towards the surface. Clearly teleoperation of the process will be difficult for the operator. Telerobotics would be a better proposition where task control can be shared between the operator and robot.

One important issue concerning telerobotic systems is the reliability of human input. Studies have shown that whilst humans are capable of very complex operations, they are limited in their abilities for space orientation and interpretation of geometrical data [Lumelsky, 1991]. Apart from the psychological limitations, other factors such as fatigue, loss of concentration and repetitive strain do affect the quality of control.

A number of schemes have been proposed to address these limitations. Lee [1992] suggested reducing the operator's control burden by allowing the robot to perform control tasks such as compliance and force control in the form of task sharing. Schryver [1996] proposed a supervisory control where the user is responsible only for strategic inputs leaving low-level control responsibilities to the robot. Brunner [1995] presented a task–directed programming approach where methods are provided to instruct the robot at an implicit level, to generate the reference values for the local feedback loops, and to design hybrid controllers for shared-control applications. Other shared control examples are found in Masson[1998] and Wai[1999].

To address the technological demands of a robotic solution, the authors introduced a 3-DOF compliant tool head. The mechanical compliance allows the tool to maintain orthogonality with the surface. Built-in sensors measure the degree of compliance and the robot corrects its position and orientation with respect to the surface. Therefore the robot could traverse the surface at a greater speed but at a lower update rate for adaptive correction. In this paper, the authors present the robotic system, development of the compliant tool head, the robotic motion control and results of laboratory tests.

2 Conceptual design

The robotic system shown in figure 1 consists of a 6 DOF robot, a compliant tool head, human machine interface (HMI) and a supervisory controller. The supervisory controller is central to the system receiving instructions from the HMI unit and monitoring the robot. It also monitors the tool sensors, processes that data and sends instructions to the robot for adaptive corrections.

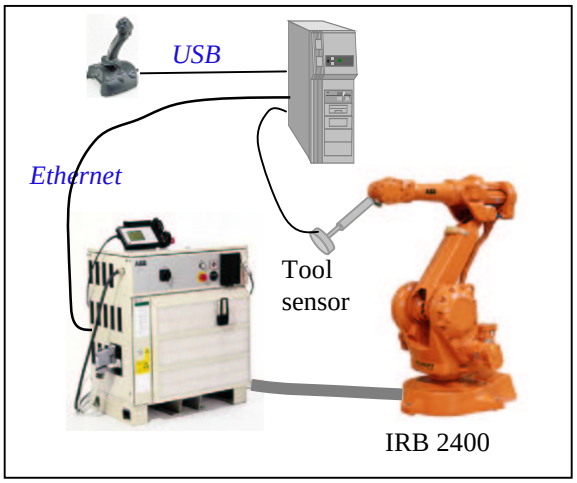


Figure 1: Components of the robotic system.

Motions are defined with reference to the tool centre point (TCP) described by a direction vector. This vector is an integration of direction vectors resulting from adjustment of the TCP to surface and tangential direction defined either manually or preprogrammed.

2.1 Task description and definition

The task is to develop a robotic system that is capable of adapting the tool to the surface contour while an operator simultaneously controls the direction and velocity. To achieve this, the tool must have at least 3 DOF with two rotational joints pivoted at the TCP and a linear axis along the shaft of the tool. The rotational axes enables the tool to orientate its face normal to the surface while the linear axis allow the tool to extend or retract depending on the curvature.

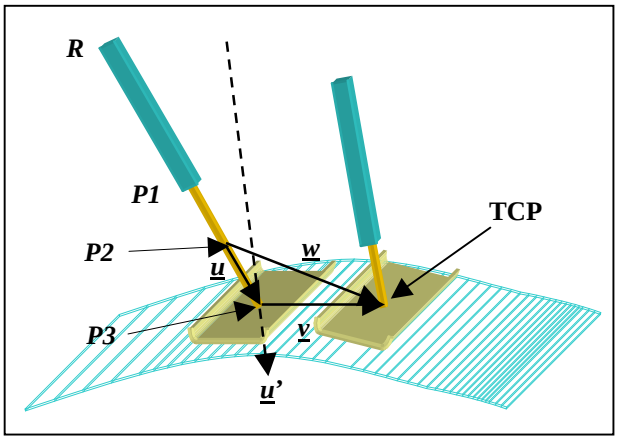


Figure 2: Description of tool and vectors

A description of the tool and resulting vectors are shown in figure 2. The position and orientation of the tool head are characterised by ***P3*** and ***u*** respectively. Motion of the tool head along the contour is specified b direction vector ***v*** with reference to position ***P3***. The robot position, ***R*** moves along the same vector when no angular compensation is required. If θ_x and θ_y are non-zero, then ***R*** will need to re-adjust its position to satisfy a new ***u***'.

Table 1: List of variables

Variables	Description
R	Robot mounting plate position
P1	Position at zero extension.
P2	Position at mid extension
P3	Position of plate mid point.
u	Unit direction vector of sensing tool.
v	Unit direction vector for tangential motion
w	Resultant direction vector
dz	Linear displacement
θ_x, θ_y	Angular displacements

Notes:

R, P1, P2 and ***P3*** are in Cartesian coordinates (x, y, z) while ***u, v, w*** are unit vectors represented by cosine direction angles (i, j, k).

2.2 Mechanical design

The tool head has 3 DOF comprising of two revolute joints and one prismatic joint (linear) as shown in figure 3. This arrangement allows flat plate of the tool to comply with the surface geometry. The two revolute joints merged in a hexagonal head and socket arrangement allowing the joint to move like a ball and socket joint while preventing rotation about the shaft.

The prismatic joint is a linear guide consisting of a rotary pneumatic actuator to provide a constant contact force with the surface. The plate orientation with respect to the shaft is measured by two potentiometers. Similarly, the linear movement is measured by a potentiometer.

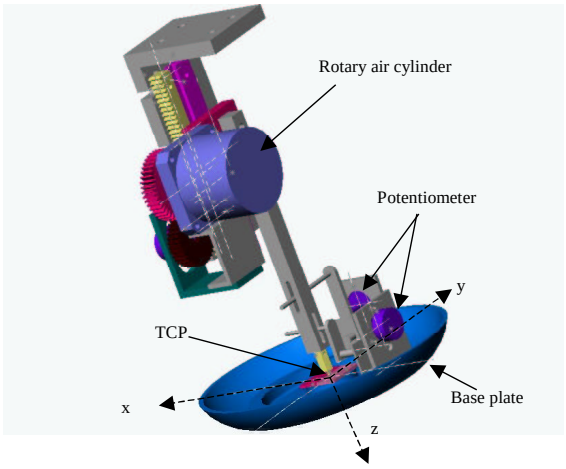


Figure 3: A conceptual design for the tool

3 System controller

3.1 Control modes

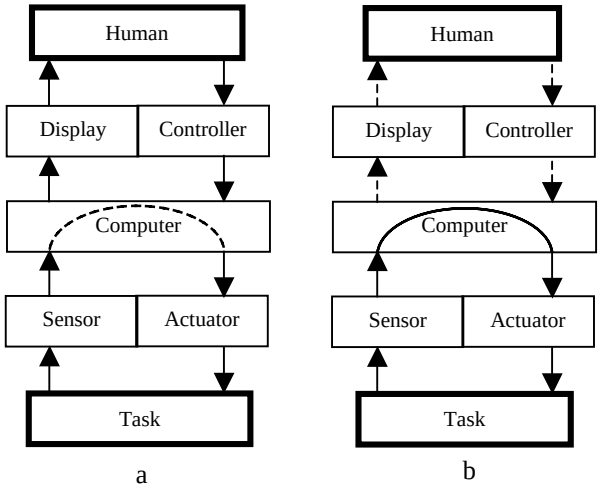


Figure 4: Supervisory control scenarios.

Two common supervisory control scenarios are depicted in figures 4a and 4b [Sheridan 1992]. In figure 4a, the human directly controls the manipulator while the computer monitors the system. The computer can assume control when a situation demands it. In figure 4b, the computer controls the manipulator while the human performs a supervisory role and may provide occasional input. Using a combination of these scenarios, the controller has the flexibility to operate either traded control or shared control.

Traded control switches system control between the operator and computer. Therefore the controller is trading between two scenarios. Shared control involves both human and computer control of the system with each entity controlling different aspects of the system. Control of this type tends to be inter-dependent. In situations when the system is unable to cope sufficiently with the dynamic changes, the computer will override instructions given by the operator.

A block diagram of the shared supervisory controller is shown in figure 5. It consists of a main controller and three sub controllers. The main controller performs the following functions:

- Tool motion control based on user commands
- Robot position and orientation correction

The system dynamics was designed to satisfy the following criteria:

- Tool orientation remain orthogonal ($\theta x = 0$, $\theta y = 0$)
- Tool extension is at mid position, ***P2*** ($dz = 0$)

3.2 Tool linear motion control

The operator defines the tangential direction of the tool along the surface via a 2 DOF joystick. Two vectors $\underline{\boldsymbol{v}}_x$ and $\underline{\boldsymbol{v}}_y$ are derived from the absolute positioning of the joystick. The resultant vector $\underline{\boldsymbol{v}}$ becomes the motion direction. Vectors $\underline{\boldsymbol{v}}$, $\underline{\boldsymbol{v}}_x$ and $\underline{\boldsymbol{v}}_y$ are direction vectors aligned tangential to the surface. The angles to their respective axes are derived from sensor readings and current tool orientation $\underline{\boldsymbol{u}}$.

$$\underline{v}_{sum} = \underline{v}_x + \underline{v}_y$$

$$\underline{v}_n = \frac{\underline{v}_{sum}}{\sqrt{v_i^2 + v_j^2 + v_k^2}}$$

Vector $\boldsymbol{v_n}$ is the normalised unit direction vector for the joystick command. This vector is scaled by a factor, $\boldsymbol{v_f}$ to produce the actual surface velocity. The larger of the two is selected.

$$v_f = \begin{cases} |J_x| & if\ |J_x| \geq |J_y| \\ |J_y| & if\ |J_y| > |J_x| \end{cases}$$

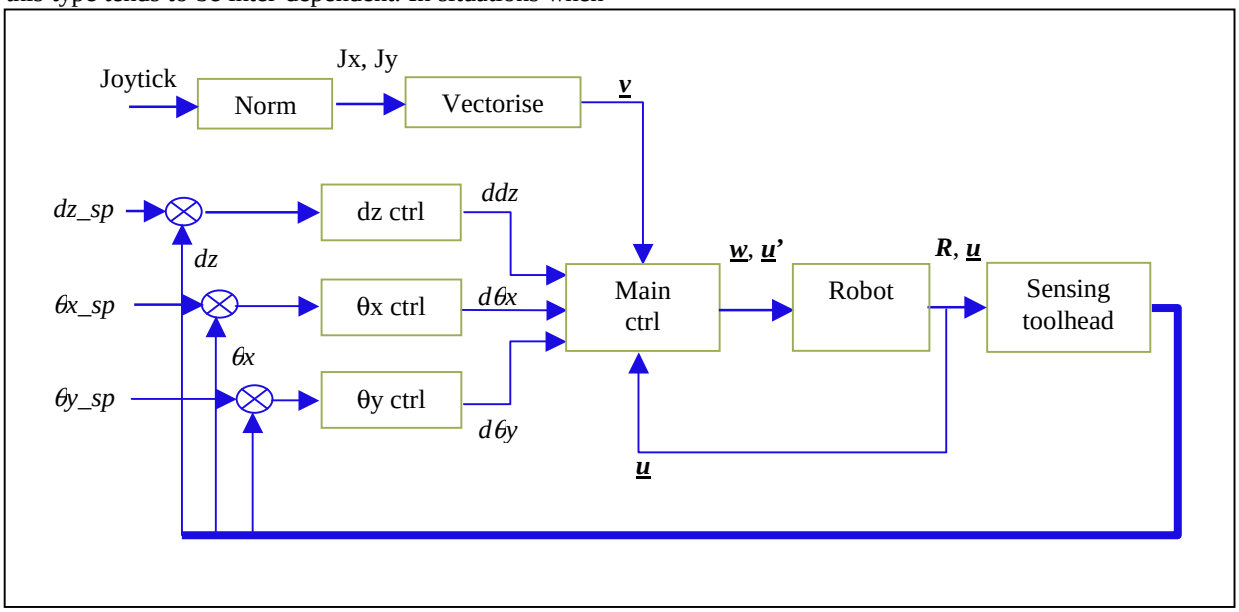


Figure 5: Block diagram of the shared supervisory controller

$\underline{v} = \underline{v}_n * v_f * v_{\max}$

where v_{max} is the maximum surface velocity.

3.3 Tool compensation control

The sub controllers for *dz, θx and θy* are independent controllers determine the required change for the respective variables in order to satisfy the above criteria. Each output is an incremental adjustment of the variables until the set points are reached. It uses a PID loop in the form

$$m = P\left(e + I.\int_0^t e.dt + D.\frac{de}{dt}\right)$$

where

P, I and *D* are the control coefficients.

e is the error variable *(dz, θx* or *θy)*

m is the change required (*ddz, dθx* or *dθy*)

3.4 Motion priority

The motion priority is a factor that determines the degree of control sharing between the user and robot. It resolves the conflict that exists between tool compensation and tangential motion commands. Tool compensation has been given priority over motion control. In the event of a large error detected, the robot will reduce the tangential motion speed to allow for adaptive correction. This minimises the possibility of instability when the tool is unable to reach the setpoint because the tool is moving too quickly forward. However, when the tool is in vicinity of the setpoint, then tool motion will assume the required speed. The degree of sharing is defined as follows:

$$s = \left(1 - \frac{dz}{dz_{\max}}\right) * \left(1 - \frac{\theta x}{\theta x_{\max}}\right) * \left(1 - \frac{\theta y}{\theta y_{\max}}\right)$$

3.5 Integration of vectors

The integrated linear motion is represented by the direction vector ***w***, is the sum of the linear correction vector, ***u*** and user commanded vector, ***v***. Hence,

$\underline{w} = ddz.\underline{u} + t.s.\underline{v}$ where *t* is the sampling period.

Notice that when *s* is small, motion in ***v*** is reduced so that time is given for *dz* correction to take place. Orientation adjustment is done by rotating ***u*** about the x and y axes to produce ***u'***. This adjustment is incremental and set to limit the speed of the robot.

$\underline{u}' = Rx(d\theta x).Ry(d\theta y).\underline{u}$

where $R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c & s \\ 0 & -s & c \end{bmatrix}$ and $R_y = \begin{bmatrix} c & 0 & -s \\ 0 & 1 & 0 \\ s & 0 & c \end{bmatrix}$

4 Implementation

4.1 System integration

As shown in figure 6, the supervisory controller is central to the system comprising of a computer program written in C++ language operating under Linux OS on a Pentium III computer. It executes a control loop where data from joystick and sensors is acquired, processed and the next target pose downloaded to the robot controller.

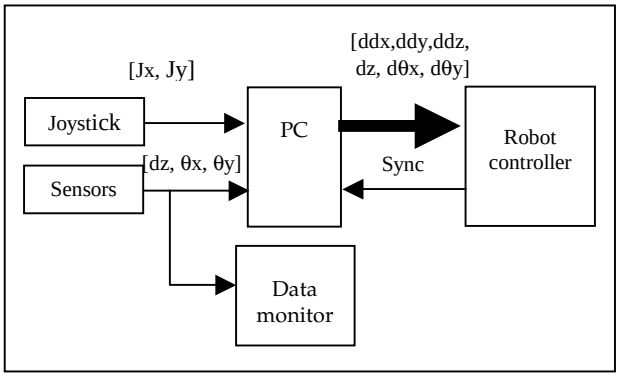


Figure 6: Data transfer

To ensure a consistent control loop period, a 'time specified' motion command was used. This function is supplied by the robot programming language for robotic motions to be completed by a declared period. However, this function has a variability of up to 40ms. Due to this restriction, a cycle time of 150ms was selected.

Another important aspect is the process time lag that exists between reading sensor data and executing motion instruction. While the PC and the robot controller can run very consistent cycles independently, the time lag could be as long as one sampling period and will drift over time. Ideally, the robot must execute a motion command immediately upon receiving a new set of data or at least maintain a constant time lag. To achieve this, the robot sends a digital signal at the start of each motion instruction to the PC. Using a suitable delay after each sync signal and taking into account the time for data acquisition (0.5ms) and for Ethernet data transfer (1.5ms), the time lag was minimised but more importantly it was consistent between cycles.

A dataset from the dual axes joystick represents absolute positioning while from the tool sensors; the dataset represents displacements of the three compliant axes. Based on these data, the supervisory controller evaluates the relative linear translation in Cartesian coordinates and rotational translation in Euler angles and downloads a dataset of 6 values to the robot.

4.2 Robot program

The robotic program was written in *RAPID*, a programming language supplied for IRB2400 robot. The program essentially operates a series structure beginning with the initialisation of variables; determination of the new target pose followed by making an adjustment to the TCP. Then a sync signal is generated before executing the motion command.

The dataset transferred from the PC has six data values that are grouped in a pose type variable that was declared as a persistent variable so that it can be rewritten remotely.

Pose transformation

The linear and angular translations of the tool are described with reference to the TCP in the tool coordinate

frame. Pose translation is presented in Cartesian coordinates (***ddx***, ***ddy***, ***ddz***). These variables define the relative linear movements for the next position. Therefore they represent direction and travel distance. Pose orientation is represented by quarternions (*q1, q2, q3, q4*) but the control program computes the orientation in Euler angle and are transferred as such. The new pose is obtained by pose multiplying the current pose with the translation pose.

Adjustment of TCP

The tool TCP is described as the pivot point on the base plate. Since the point is aligned to the tool coordinate frame and offsets only along the z-axis, it is defined as (0,0,***dl***) for position and (1,0,0,0) for orientation. Variable ***dl*** is the linear displacement between two coordinate frames in the z-axis.

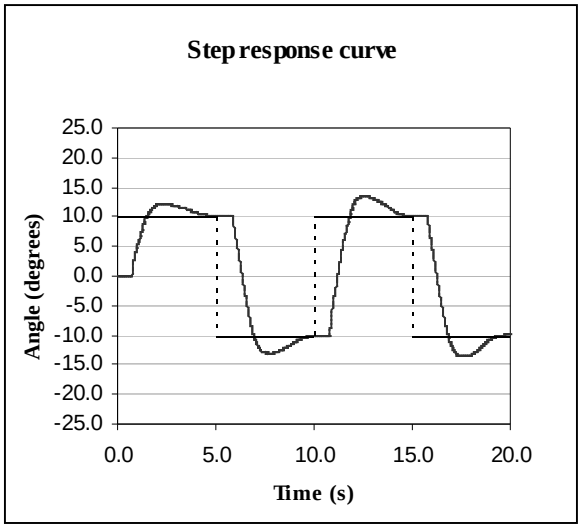
When the tool moves along an object, ***dl*** varies with the surface therefore the TCP must be redefined accordingly. Changes in the orientation will then be executed with respect to ***dl*** as the tool rotates about this pivot point. Otherwise the tool will rotate about an incorrect tool radius and will result in a positional offset.

Motion command and sync signal

When the new pose has been determined, a sync signal is generated before the motion command. The sync signal consists of a digital pulse of period 50ms. The motion command instructs the TCP along a linear path towards target position ***p3*** with the required rotational adjustment with respect to the tool length. The motion is 'time specified' rather than 'distance specified' so that the robot will complete by the allocated time. A 'fly-by' zone setting rather than 'stopover' was implemented for smoother continuous motion. The robot will not come to a complete stop instead proceeds to the next target point when the TCP is within the specified vicinity of ***p3***.

5 System tests and analysis

Two tests were performed. Firstly a static test was conducted for tuning of the PID coefficients and secondly a dynamic test to observe the behaviour of the system during motion.



Graph 1: Step response curve

5.1 Static test

For the static test, a step response test is performed to each compliant axis. The TCP remains stationery but the setpoint is varied by a 0.1Hz square wave at 50% amplitude test signal. The PID coefficients were then tuned to achieve low overshoot (<10%), rapid rise time but no secondary overshoot. Graph 1 shows the typical optimum response.

5.2 Dynamic test

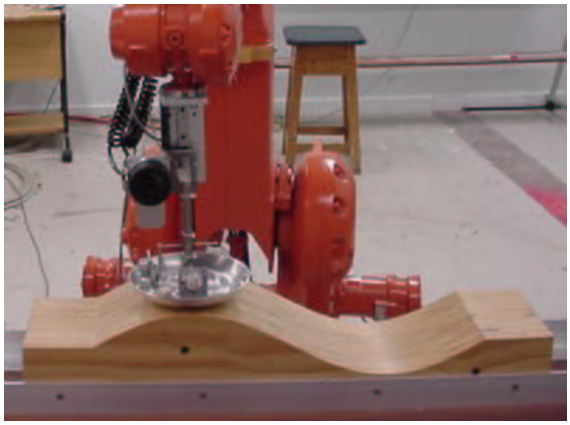


Figure 7: Test object

This test demonstrates the effectiveness of the system adaptive control during motion. A test platform was built made of laminated wooden layers glued firmly together as shown in figure 7. It is 70mm wide and a sinusoidal contour surface of 500mm long and peak-to-peak amplitude of 20mm.

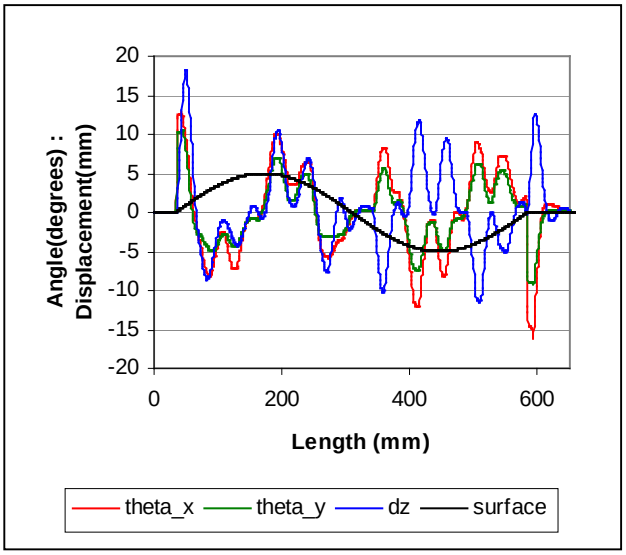
A linear path is projected between two end-points for the robot to travel at the set velocity. When the tool moves along this path, the dynamic responses of the sensors with respect to the geometry is shown in graph 2. The change in the TCP velocity caused by sensors readings is shown in graph 3.

5.3 Results and analysis

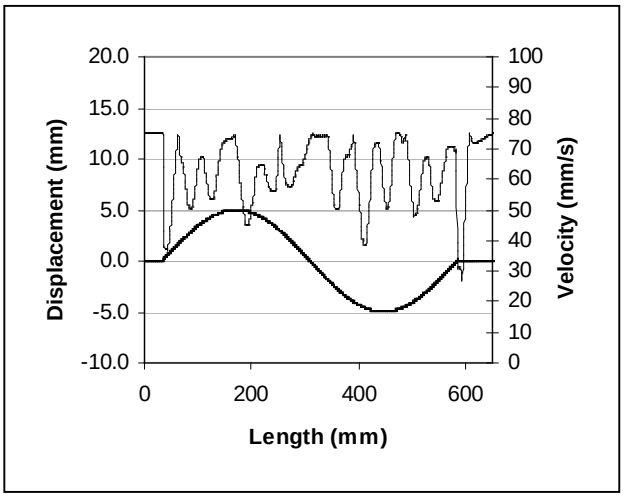
The greatest variation is found at the peaks of the sine curve where the tool must make the biggest linear and angular adjustments. In graph 2, the response shows that the robot and tool velocities vary according to the contour. The tool velocity is kept constant except for situations where large adjustments must be made. This was largely influenced by the motion priority factor. On the graph, it can be seen that the robot velocity is greatest when the tool moves over the peak because the robot has to move at a greater velocity in order to orientate the tool. However at the trough, the robot needs to wait for the tool to complete the arc.

The optimised system works well even at 100mm/s with reduced angular displacements and fast recovery rates. Surprisingly the system behaved differently when all axes are active as compared with only one active axis. The former had greater overshoot. Although it worked relatively well, the system may not be decoupled as initially assumed. To reduce the overshoot, gain settings for each axis was slightly reduced.

The response of the adaptive control system depends on



Graph 2: Dynamic sensor response at 75mm/s



Graph 3: Variation of tool velocity caused by sensor readings

the process cycle time despite having the mechanical compliance in place. However, the cycle time was not limited by the data transfer rate but by an inherent time lag of the robot controller moving from point to point. This was observed during the static test procedure recording a time lag of 800ms during step changes. This was unexpected as a delay of only 300ms equivalent to 2 cycle time lengths was anticipated. A simple test confirmed that 'time specified' motion introduced greater lag than 'distance specified' motion and similarly for fly-by points over stop over points.

6 Conclusion

The technological demands of robotic contour following were addressed by introducing mechanical compliance to a specialised tool. With compliance, the traverse speed can be increased while the sensing data rate is reduced. This provides greater flexibility in terms of processing speed without being limited by process dynamic constraints. The results demonstrated that the tool adhered to the surface continuously with the faceplate normal to the surface during motion. The adaptive control was accomplished efficiently by a vector-based projection and prioritising method.

References

[Brunner, et. al, 1993] Brunner B., Hirzinger G., Landzettel K., Heindl J. (1993) *Multisensory shared autonomy and tele-sensor-programming – key issues in the space robot technology experiment ROTEX*. IEEE/RSJ Int. Conf. on Intelligent Robots & Systems (IROS), Yokohama, July 26-30, 1993.

[Brunner, et. al, 1995] Brunner B., Landzettel K., Steinmetz B.-M., Hirzinger G. (1995) *Tele-sensor-programming – A task-directed programming approach for sensor-based space robots.* Int. Conf. on Advanced Robotics (ICAR), Saint Feliu de Guixols, Spain, Sept 20-25, 1995.

[Lee & Lee, 1992] S. Lee & H.S. Lee, *Advanced teleoperator control system design with human in control loop*, Winter Annual Meeting of the American Society of Mechanical Engineering, 1992.

[Lee & Lee, 1992] S. Lee & H.S. Lee, *An advanced teleoperator control system: Design and evaluation*, Proceedings 1992 IEEE International Conference on Robotics and Automation, 1992

[Lumelsky, 1992] V. Lumelsky, *On human performance in telerobotics*, IEEE Transactions on Systems, Man and Cybernetics, 21(5), pp 971-982, 1991.

[Schryver & Draper, 1996] J.C. Schryver & J.V. Draper, *Network simulation analysis of level of control for the single-shell tank waste retrieval manipulator system*, (ORNL/TM-12752). Oak Ridge, TN: Oak Ridge National Laboratory, 1996.

[Sundstrom et. al., 1995] E. Sundstrom, J.V. Draper, A. Fausz & H. Woods, *Test bed control centre design concept for tank waste retrieval manipulator systems*, Proceedings of the ANS Sixth Topical Meeting on Robotics and Remote Systems, pp. 187-193. Monterey, CA: The American Nuclear Society, 1995.

[Sheridan, T.B. 1992] *Telerobotics automation and human supervisor control*, ISBN:0262193167, Boston, MA: MIT Press, 1992

[Masson, et. al.,1998] Masson Y., Gravez P. and Fournier R. (1998) *A graphical supervision concept for telerobotics*. Int. Symp. on Robotics, Birmingham, 1998.

[Wai, et. al, 1999] Wai Yu, Pretlove, J., Parker G. (1999). *From supervisory control to cooperative control*. Proc. SPIE Conference on Telemanipulator and Telepresince Technologies VI, Boston, Massachusetts, September 1999.