FORCE CONTROL OF AN INDUSTRIAL ROBOT WITH ADAPTIVE COMPENSATION OF THE ENVIRONMENT STIFFNESS

Bojan Nemec and Leon Žlajpah
Jožef Stefan Institute, University of Ljubljana
Ljubljana, Jamova 39, Slovenia
bojan.nemec@ijs.si

Keywords: Force control, robot control, adaptive control

Edited by: Jadran Lenarčič

Received: September 15, 1993 **Revised:** March 20, 1994 **Accepted:** March 30, 1994

The paper describes the implementation of the adaptive force control of an industrial robot. The implemented algorithm is a position based hybrid control scheme with adaptation to the environment stiffness. Control scheme is sensitive to the changes in environment stiffness. We solved this problem by the adaptive controller. Implementation problems on the robot controller are also discussed. The proposed control method is easy to implement and can be applied to existing industrial robots fitted with a conventional position controller. The performance of the force controlled manipulator with the proposed control law was tested with the computer simulation and by using the real robot.

1 Introduction

Many of robot industrial applications, such as automated assembly, deburring, teleoperation, etc., require exact control of interaction forces with the environment. The problem of controlling interaction forces has been investigated by many authors. According to Kazerooni [7], active force control strategy can be classified into two major approaches. The first approach force or torque is commanded along those directions constrained by the environment, while position or orientation is commanded in the direction unconstrained by the environment. The above approach was formalized by Mason [9]. Craig and Raibert [12] introduced a hybrid force/position controller by controlling the actuator torque. Whitney [14] proposed damping control where sensed force error is transformed into the commanded velocity of the actuator. A similar approach was used by Paul and Shimano [11]. Some advantages can be obtained if the decoupling of the manipulator is done in the task space, like in the operational space approach introduced by Kathib [6]. The second approach is based on establishing a relationship between the position of the manipulator and interaction forces. Error in position, velocity and force generates joint torque commands. Salisbury [13] introduced the stiffness control approach which acts like a six-dimensional active spring in the tool coordinates. Impedance control which combines stiffness and damping control was introduced by Hogan [5]. Our approach is modified hybrid/position force controller where force error is converted to the position offset. This method is easy to implement and requires no modification at the servo level of the robot controller. The stability and response of the proposed force controller depend on the sensor and environment stiffness. For applications on unknown or changing environment stiffness we propose a simple adaptive controller which adapts the gain of the force control loop to the environment and sensor stiffness.

2 Force control

The problem of compliant control can be broken down into pure position and pure force control. In a direction where the robot task is unconstrained by the environment, pure position control can be used, while in the direction constrained by the environment pure force control is used. The constrains imposed by the environment are called natural constrains. In order to specify the desired task, artificial constraints are introduced. Natural and artificial constraints together form N-dimensional constrained space C, where N is the number of the Cartesian degrees of freedom. The task of the controller is to map the C space into the manipulator joint movement. The hybrid force control method controls motor torque directly. Here, another approach was used due to the hardware limitation of the controller of our robot. The force control is implemented in the outer loop of the existing position/velocity control [1] and generates new N dimensional input vector $\mathbf{y}_d$ in tool coordinate system

$$\mathbf{y}_d = \mathbf{S}\mathbf{x}_f + (\mathbf{I} - \mathbf{S})\mathbf{x}_p \tag{1}$$

where $\mathbf{x}_p$ is the desired displacement vector of the robot (translations and orientations), $\mathbf{I}$ is the identity matrix and $\mathbf{S}$ is the compliance selection matrix [12], and $\mathbf{x}_f$ is

$$\mathbf{x}_f = \mathbf{K}_f(\mathbf{F}_d - \mathbf{F}) \tag{2}$$

where $\mathbf{K}_f$ is the force controller transfer function and $\mathbf{F}$ and $\mathbf{F}_d$ are the measured force and the desired force, respectively. The compliance selection matrix is defined as a binary NxN-tuple which specifies which degrees of freedom in C are under force control and which are under position control. The first term in the Eq.1 corresponds to the force control loop where the last term is the position (orientation) command vector. The position (orientation) command vector is transformed from the tool coordinate system to the robot base coordinate system and then to the joint coordinates. Joint coordinates $\mathbf{q}_d$ are passed to the position/velocity controller. This transformation can be described by the equation

$$\mathbf{q}_d = \Psi^{-1}(\mathbf{A}(\mathbf{y}_d)) \tag{3}$$

where Ψ^{-1} describes the transformation from the Cartesian space to the joint angles and $\mathbf{A}$ denotes the transformation from the tool coordinates to the robot base coordinates.

A simple PI controller with the discrete transfer function $K_f\frac{1-\xi z^{-1}}{1-z^{-1}}$ was used for the force controller transfer function. In order to improve the stability, first order anti alias filter was used in the force feedback loop.

2.1 Design of the force controller : Single–joint case

We will first design the closed loop system for the single joint case. Stability analyses will be done in the S domain by a root locus design. A model of the one-joint robot system with DC (AC) motors is presented on Fig. 1. The parameters of the transfer functions were estimated by using the test signals and LS estimation procedure and compared with the known parameters of the system to validate results. For the third joint of our robot the transfer function parameters are as follows:

K_f = 0.06 rad/N	gain of the force control
ξ = 0.997	damping of the force control
K_p = 1400 1/s	gain of the position control
K_w = 15900	gain of the velocity comp.
K_v = 1000 Vs/rad	gain of the P velocity control
K_{vi} = 5000 V/rad	gain of the I velocity control
K_t = 0.031831 V/rad^2	tachometer gain
K_τ = 0.23 Nm/A	torque constant
K_b = 0.101 Vs/rad	back EMF constant
R = 0.91 Ω	motor resistance
B_{eff} = 0.0003 Nms/rad	effective damping
H_{eff} = 0.00046 Nms^2/rad	effective inertia
K_{se} = 350 N/rad	sensor/environment stiffness
n = 1/70	gear ratio
a = 50	anti alias filter pole
Ts = 0.01 s	sampling time

The position controller consists of simple K_p gain with feed-forward velocity compensation realized by a digital computer. Since the sampling time of the position control loop is much smaller than the sampling time of the force feedback loop, it is assumed that the position controller is realized with an analog feedback. The force control loop is realized by a digital computer, therefore we will assume zero-order sample/hold element at the input of the position controller. We assume simple model environment, described by the Eq. 4, where q_c and q is the environment contact position and measured position in joint coordinates respectevely.

$$F = K_{se}(q - q_c) \tag{4}$$

From Fig. 1 we can compute the open loop [2] [3]

[1] this method is referred to as position based force control

[2] open loop with respect to the force loop

[3] we will omit the subsystem index i in the equations for the single joint case

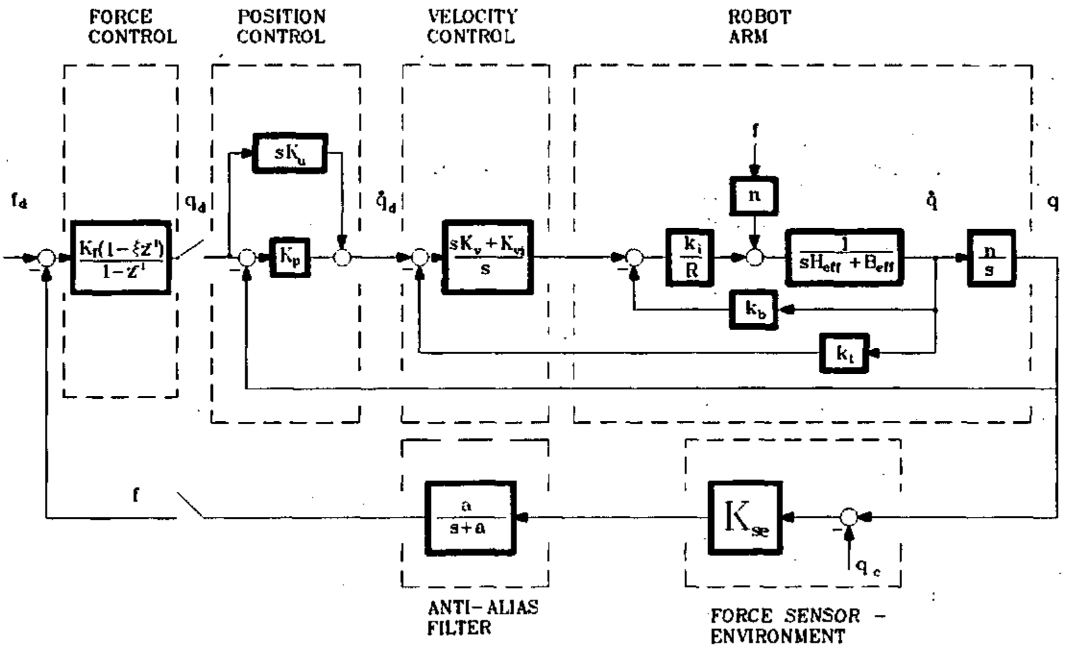

Figure 1: Model of one robot joint with velocity and position control

transfer function for the single joint case in the form

$$q = G_1(s)q_d + G_2(s)f \tag{5}$$

where

$$G_1(s) = \frac{nK_i(K_p + K_u s)(K_{vi} + K_v s)}{W(s)} \tag{6}$$

$$G_2(s) = \frac{n^2 Rs}{W(s)} \tag{7}$$

and

$$\begin{aligned} W(s) = {} & H_{eff}s^3R + \\ & (B_{eff}R + K_iK_vK_t + K_iK_b)s^2 + \\ & K_i(K_{vi}K_t + K_pK_vn)s + K_iK_pK_{vi}n \end{aligned} \tag{8}$$

Discrete PI force control law for the single joint case is

$$q_d(k) = q_d(k-1) + K_f(e(k) - \xi e(k-1)) \tag{9}$$

where $e = (f_d(k) - f(k))$, f and f_d are the measured and the desired joint forces respectively. Factor ξ was chosen to meet the desired dynamic performance of the closed loop system. The overall discrete transfer function of the reduced [4] open loop system with one sample delay in the control loop and first order anti alias filter for the sampling time $T_s = 0.01$ s is thus

$$G(z) = \frac{0.0221z^{-1} - 0.0180z^{-2} - 0.0029z^{-3} - 0.00007z^{-4}}{1 - 1.4488z^{-1} + 0.4738z^{-2} - 0.0005z^{-3} - 0.00004z^{-4}}$$

The above model was used to determine suitable gain for the force control loop via discrete root locus analyses. The root locus for the 3rd joint discrete model of our robot is presented in Fig.2. The gain K_f where system becomes unstable is 0.132 rad/N and suitable gain at dominant damping factor $\zeta = 0.5$ is 0.06 rad/N.

2.2 Design of the force controller : Multi-joint case

Robot dynamics is described by using the Lagrangean formulation, with the Eq. [5]

$$\tau = \mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{d}(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{J}^T\mathbf{F} \tag{10}$$

where τ is the N-dimensional actuator force (torque) vector, $\mathbf{H}(\mathbf{q})$ is the NxN dimensional manipulator and actuator inertia matrix, $\mathbf{d}(\mathbf{q},\dot{\mathbf{q}})$ is the N-dimensional vector of Coriolis, centrifugal, gravity and friction forces, $\mathbf{J}$ is the manipulator Jacobian and $\mathbf{F}$ is the compliant force in Cartesian coordinates.

[4] The non-reduced system is of 5th order. The system was reduced by canceling non-dominate poles and non-dominate zeroes

[5] for the sake of simplicity we will omit time dependence in the equations that follow

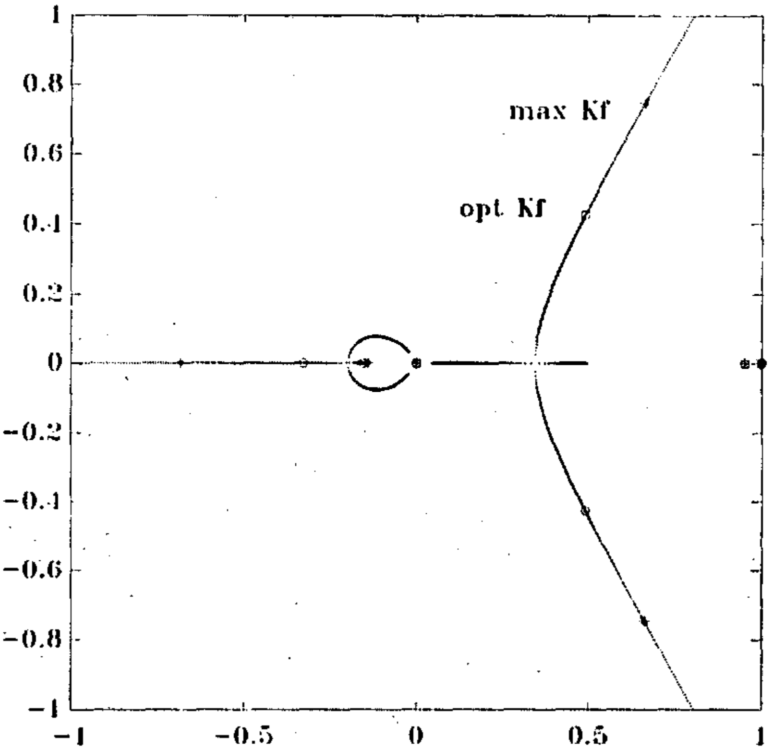

Figure 2: Discrete root locus of the system

In a general case Eq.10 describes a highly nonlinear and strongly coupled multivariable system. A stability analysis with the proposed control law is very difficult, if not impossible for such a system. The following is assumed for the stability analyses

- The gravity is compensated either by mechanical construction of the robot or by online calculation and compensation with a controller.
- the manipulator is operating at low speed, centrifugal and Coriolis forces are therefore negligible.
- the deflections in the position around the desired force are low, because force sensors have high stiffness. This assumption will allow linearisation of the system around the set point.

Furthermore the majority of existing industrial robots have a high gear ratio between the drive motors and the joint. With the above assumptions the matrix of inertia $\mathbf{H}(\mathbf{q})$ can be approximated by a diagonal matrix with constant terms $\mathbf{H}_d$ and $\mathbf{d}(\mathbf{q},\dot{\mathbf{q}})$ can be approximated with constant damping $\mathbf{B}\dot{\mathbf{q}}$.

$$\tau = \mathbf{H}_d\ddot{\mathbf{q}} + \mathbf{B}\dot{\mathbf{q}} + \mathbf{J}^T\mathbf{F} \qquad (11)$$

For a noncompliant motion Eq.11 describes a decoupled system, which is generally not true in case of compliant motion. First we will analyse the open loop transfer function (Eq.5) for multi joint case. Matrices $\hat{\mathbf{G}}_1$ and $\hat{\mathbf{G}}_2$ are diagonal matrices consisting of the subsystem transfer functions described by Eq.6 and Eq.8 respectively.

$$\mathbf{q} = \hat{\mathbf{G}}_1\mathbf{q}_d + \hat{\mathbf{G}}_2\mathbf{J}^T\mathbf{F} = \hat{\mathbf{G}}_1\mathbf{q}_d + \hat{\mathbf{G}}_2\mathbf{J}^T\mathbf{K}_{se}(\mathbf{x} - \mathbf{x}_c) \qquad (12)$$

where $\mathbf{x}$ is the position and $\mathbf{x}_c$ is the contact position in the Cartesian coordinates. We will define position as position for the desired force plus the deflection from that set point $\mathbf{x} = \mathbf{x}_d + \Delta\mathbf{x}$. Then Eq.12 can be rewritten in the form

$$\begin{aligned} \mathbf{q} &= \hat{\mathbf{G}}_1\mathbf{q}_d + \hat{\mathbf{G}}_2\mathbf{J}^T\mathbf{K}_{se}((\mathbf{x}_d - \mathbf{x}_c) + \Delta\mathbf{x}) \\ &= \hat{\mathbf{G}}_1\mathbf{q}_d + \hat{\mathbf{G}}_2\mathbf{J}^T(\mathbf{F}_d + \mathbf{K}_{se}\Delta\mathbf{x}) \qquad (13) \\ &= \hat{\mathbf{G}}_1\mathbf{q}_d + \hat{\mathbf{G}}_2(\mathbf{f}_d + \mathbf{J}^T\mathbf{K}_{se}\mathbf{J}\Delta\mathbf{q}) \end{aligned}$$

Matrix $\mathbf{J}^T\mathbf{K}_{se}\mathbf{J}$ is the joint stiffness matrix $\mathbf{K}_{sq}$. The control law for cartesian coordinates is in the form

$$\Delta\mathbf{x}_f = \mathbf{K}_f\mathbf{K}_{se}(\mathbf{x}_d - \mathbf{x}_c - \mathbf{x} + \mathbf{x}_c) = \mathbf{K}_f\mathbf{K}_{se}\Delta\mathbf{x} \qquad (14)$$

Next we multiply both sides of the Eq. 14 by Jacobian inverse and assume, that all subsystems are tuned using inner position and velocity controller to have equal close loop dynamic properties. Then, the matrix $\mathbf{K}_f\mathbf{K}_{se}$ is diagonal matrix with equal terms and control law (Eq. 14) can be rewritten into the form

$$\Delta\mathbf{q}_d = \mathbf{K}_f\mathbf{K}_{se}\Delta\mathbf{q} \qquad (15)$$

Control law for the multi-joint case is thus identical to the control law in single-joint case.

From Eq. 5 and 13, we can see that the dynamics of the multi joint case is thus similar to the dynamics of the single-joint case except that the joint compliance $\mathbf{K}_{sq}$ matrix introduces nonlinearity and cross-coupling between joints. Joint stiffness matrix can be calculated and compensated on-line. This will assure stability of the overall system regarding the assumptions presented at the beginning of this paper section. In our robot with high gear ratio the influence of the last term in the Eq.13 is almost negligible and the results of the single-joint case are also valid for the multi-joint case.

2.3 Adaptation to the variable sensor and environment stiffness

The stability of the proposed force control loop is mainly affected by the environment and sensor stiffness. If the stiffness is not known in advance or is changing during the task, the response of the force control may be to slowness when the expected stiffness is lower than real stiffness. When the real stiffness is greater than the expected stiffness, the response of the robot can be very oscillatory, bouncing and even unstable. This problem can be slightly reduced by diminishing time delays in the force control loop (see the results of [1]), but this may be impossible with some robot controller architectures. The above problem can be efficiently solved by the adaptive control loop. Sensor stiffness can be computed from Eq. 4. Unfortunately, the contact position $\mathbf{x}_c$ vector is usually not known in advance. Differentiating the Eq. 4 poses implementation problems. Robot position signals are usually read from encoders and are not so affected by noise as force signals, which are read as analog values from an A/D converter. Differentiating noisy signals gives less useful results. In [2] averaging was proposed to avoid this problem. Namely, Eq. 4 can be expressed also as $\mathbf{F} = \mathbf{K}_{se}\mathbf{x} - \mathbf{F}_0$, where $\mathbf{F}_0$ is a constant offset if contact position remains unchanged. However, averaging slows the adaptation speed. We propose a state variable filter to solve differentiation problems. In this case, force readings and position vectors are lead to the simple, stable, first order filter with transfer function

$$G_f = \frac{s}{bs+1} \qquad (16)$$

which can be realized by a computer program or by a simple analog circuit. The realization of the filter is presented on Fig. 3. Filtered derivatives $\dot{\mathbf{F}}_f$ and $\dot{\mathbf{x}}_f$ are then used for the estimation of the sensor and environment stiffness. A direct adaptive controller was used in our control scheme. In order to increase the adaptation speed and avoid the computational burden we chose simple reference model in the form

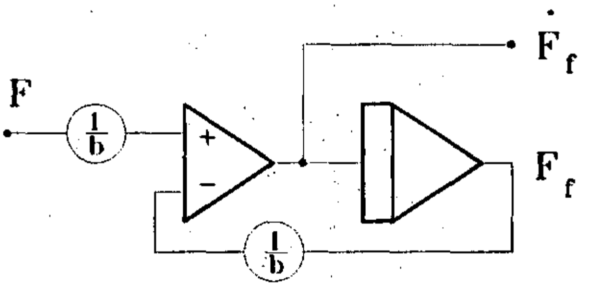

Figure 3: Derivative of an signal obtained by filtering

$$\mathbf{F}_m = \hat{\mathbf{G}}\mathbf{K}_0\mathbf{x} \qquad (17)$$

with the desired response. Root locus design was used to determine the required gain $\mathbf{K}_0$ for the desired behaviour of the reference model. Note that sensor-environment stiffness is included in $\mathbf{K}_0$. The aim of the adaptive controller is to minimize the output error between the reference model and the system with variable gain $\mathbf{K}_f$

$$\mathbf{F} = \hat{\mathbf{G}}\mathbf{K}_f\mathbf{K}_{se}\mathbf{x} \qquad (18)$$

It can easily be verified that the proposed adaptive control satisfies the criteria for the perfect linear model following control [8]. The gain K_f [6] for the each subsystem is calculated using RLS

[6] we will omit the subsystem index i in the equations that follow

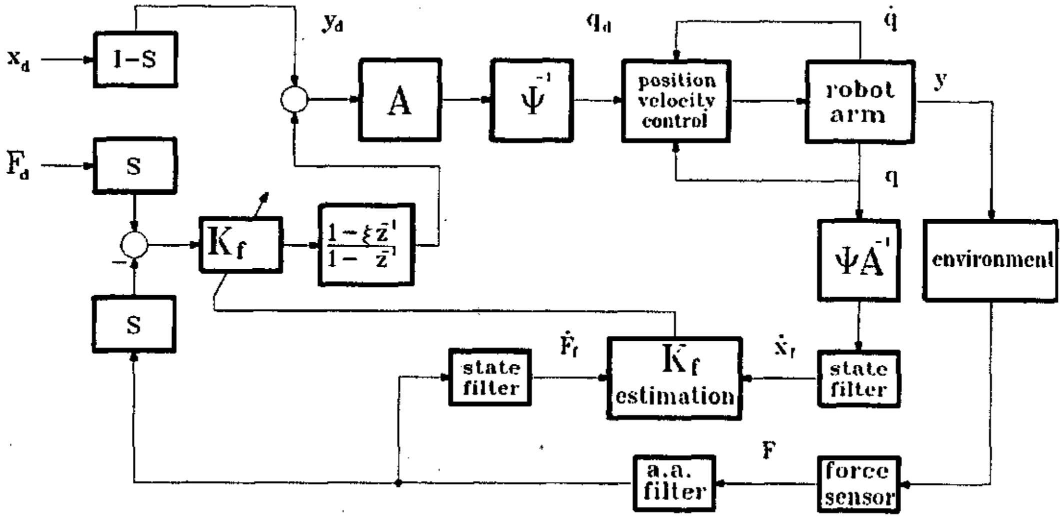

Figure 4: Adaptive control scheme

estimation procedure in the form

$$K_f(k) = K_f(k-1) + \frac{P(k-2)\psi(k-1)}{1+\psi(k-1)P(k-2)\psi(k-1)} e_m(k-1) \tag{19}$$

$$\psi(k-1) = \frac{\dot F_f(k-1)}{K_0} \tag{20}$$

$$e_m(k-1) = \dot x_f(k-1) - \psi(k-1) \tag{21}$$

$$P(k-1) = (P(k-2) - \frac{P(k-2)\psi(k-1)\psi(k-1)P(k-2)}{1+\psi(k-1)P(k-2)\psi k-1})\lambda \tag{22}$$

λ is the appropriate chosen forgetting factor. Normally, the forgetting factor is set between $\lambda = 0.95$ and $\lambda = 0.99$. However, a large forgetting factor slows down the convergence of the estimation, but the algorithm is more resistant to a sudden change in the estimated parameters due to noise . In our case we get the best results with a forgetting factor of about $\lambda = 0.7$. In this case the behaviour of the RLS estimation will approach the projection estimation algorithm. With a low forgetting factor λ, there is a danger of covariance blowup during the period when the system is insufficiently excited [3]. In our case we solved the problem by identifying only during the period when filtered derivative of the force was above the noise limit.

If the contact position remains unchanged during the adaptation, the adaptive system is linear in the unknown parameter and the stability of such a system can be easily verified under the assumption of a persistently excited system [4]. However, if the contact position changes during the adaptation, it is impossible to estimate both contact position and stiffness of environment. Namely, changes in contact position have the same effect as changes in environment stiffness. To solve this problem we propose to estimate K_f only a few samples after the sensor reaches the obstacle, i.e. to identify only few samples after the filtered force derivative changes from 0 to the ϵ, where ϵ is a suitably chosen constant according to the sensor reading resolution and noise in the measurement.

The adaptive control scheme is presented in Fig.4. Note that at low speed calculation of the robot position in the task space can be replaced by the desired position in the task space. The closed loop behaviour was simulated by using continuous time simulation of the DC motor, gears, velocity controller and sensor, discrete simulation for the position control and trajectory interpolation at a sampling interval of 0.0016 s and discrete simulation of the force control loop at a sampling interval of 0.01 s.

The simulation results for the step response of the proposed control scheme are presented in Fig.5 for the non-adaptive and adaptive controller with filtered signals respectively. Both, the adaptive and non-adaptive controller were tuned for the environment stiffness 1 N/mm, while actual environment stiffness was 4 N/mm. From the simulation results we can see that the non-adaptive controller starts to bounce when environment stiffness increases and goes to a limit cycle. In contrary, the adaptive controller quickly adjusts to the new environment stiffness.

The simulation results were compared to the measurement obtained on the real robot. The step response of the adaptive and non adaptive controller for the stiff environment are presented in Fig.6. The adaptive controller estimates correct gain and is stable, but some oscillation can be noticed during the impact, which are not obtained in the simulation. This is mainly due to the nonlinear friction and backlash in the gears, which were not included in the simulation.

3 Implementation on the robot controller

The proposed compliance control scheme was implemented on a 6. d.o.f. industrial robot RIKO 106. The architecture of the control system is presented in Fig.7. The main CPU of the robot controller is dedicated to trajectory generation, kinematic transformation and man-machine interface. The axis computer is used for the digital position controller with feed-forward speed compensation and for interfacing with the controller periphery. Because of hardware limitations, force feedback was realized via the main CPU. The sampling interval of the force controller, as well as the sampling interval for the trajectory generation module was set to 0.01 s. The desired trajectory is passed to the axis CPU by a shared VME RAM. The axis computer generates trajectory with sampling time 0.0016 s by polynomial interpolation. Due to the interpolation algorithm and data exchange between the main and the axis CPU a delay of 0.02 s appears in the force feedback loop. RRL robot programming language is implemented on the robot controller [10]. Three additional commands were added to RRL for the compliant motion definition. Natural constrains are defined with command ForceSELect, `FSEL s1 s2 .. s6`. Nonzero parameters s1 .. s6 corresponds to the pure force control in the direction x y z roll pitch yaw, while the zero parameter specifies the pure position control in the tool coordinates. The value of the parameters s1 ..s6 selects the A/D channel where the corresponding force signal appears. The negative parameter reverses the signal input sign. Artificial constraints are defined with command ForceTRACK, `FTRACK c1 .. c6`, where c1 .. c6 is the desired velocity (angular velocity) or force (torque) vector according to the artificial constraints definition. The offset of the sensor and A/D converter, as well as the effect of gravity on the sensor and tool is removed using command `CALIBRATE FS`. Of course, during the calibration, the force sensor should not be in contact with the environment. Calibration activates also adaptation procedure.

4 Example

Force control was tested on the deburring process of an irregulary shaped workpiece. The task of the robot was to apply constant force 70 N in the orthogonal direction of the free movement of the robot and to maintain zero torque at the tool during movement at constant speed 10 mm/s along the X axis of the workpiece. We used a three-dimensional wrist mounted force sensor, developed at our institute. The RRL program for the required task is listed in Fig.8. The response of the robot is presented in Fig.9 for orthogonal force and wrist torques respectively. In the Fig.9 plot between (t=5sec) and (t=9sec) shows the tracking of the sensor when change in the shape of workpiece occur. We can see that the signals are rather chattering. It was found that this is caused mainly by poor resolution, cross-coupling and noise of the sensor. A higher sampling frequency improves the transient response, but does not eliminate chattering from the response.

5 Conclusion

A force control algorithm based on a hybrid control scheme was presented. The main difference between the original method and our approach is that force is controlled by changing the desired position. This approach allows implementation on existing robot controllers with a position and velocity control loop. The limitations of our approach are the following:

- position resolution of the robot controller affects the force resolution of the system. In

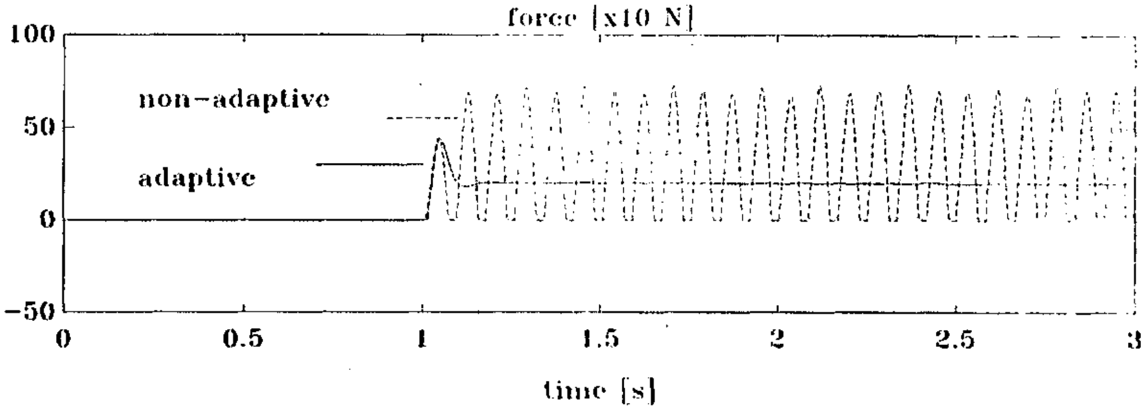

Figure 5: Step response of the simulated adaptive and non-adaptive system

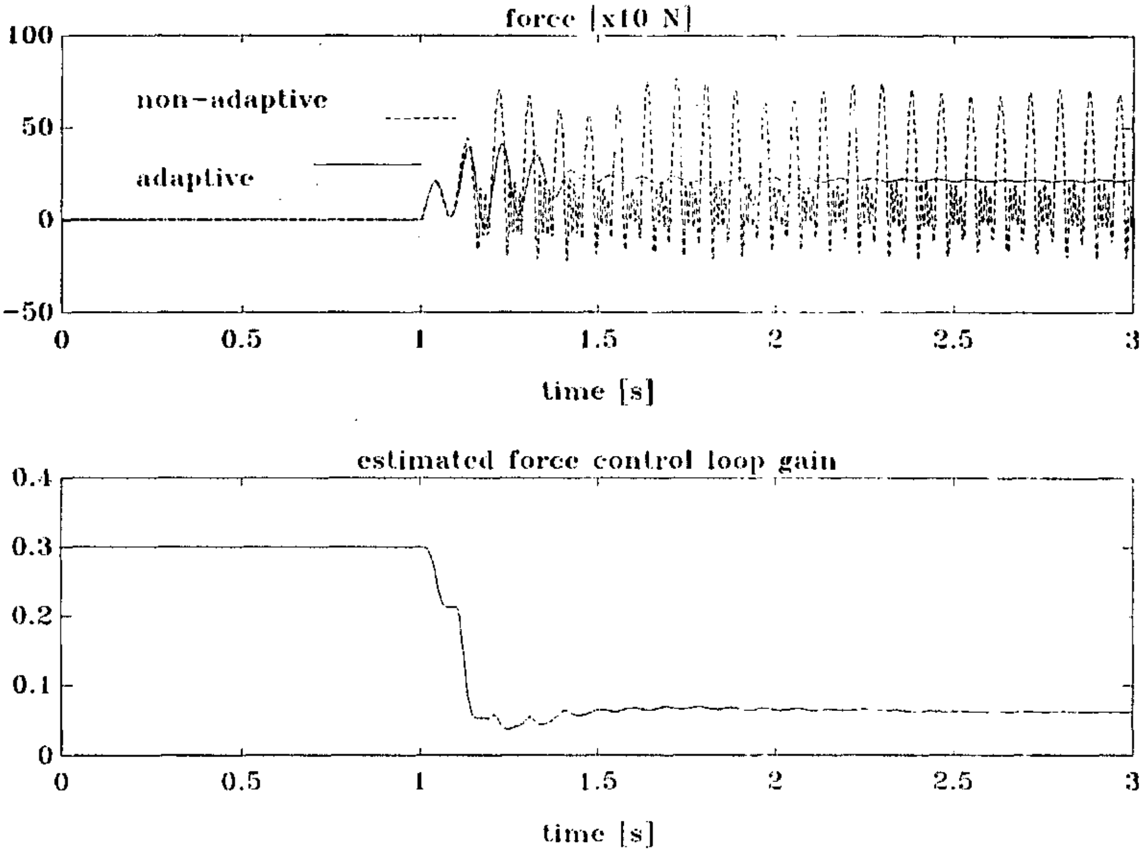

Figure 6: Step response of the adaptive and non-adaptive system and estimated gain of the adaptive controller

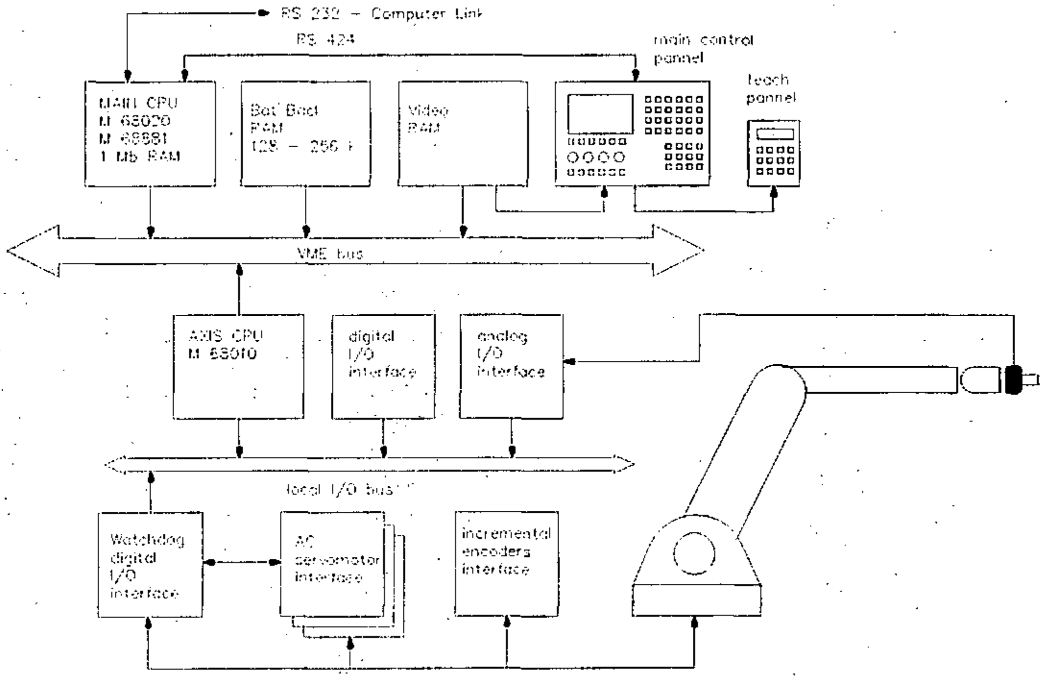

Figure 7: Architecture of the RIKO 106 robot controller

```
* RRL sample program for the deburring process
*
* define maximal, actual speed and toll center point
MAXSP = 1000 40
SPEED = 50.0
TCP 1 = 0 350 0 0 0 0
* approach start point of the deburring and calibrate sensor
APPRO TO 1 FOR 0 -10 0
CALIBRATE FS
* natural constraints
FSEL 0 1 0 0 2 3
* artificial constraints , start deburring, stop on external signal
FTRACK 10.0 70.0 0 0 0 0 UNTIL SIG 0
DEPART FOR 0 -10 0
HOME
```

Figure 8: RRL program for the deburring process

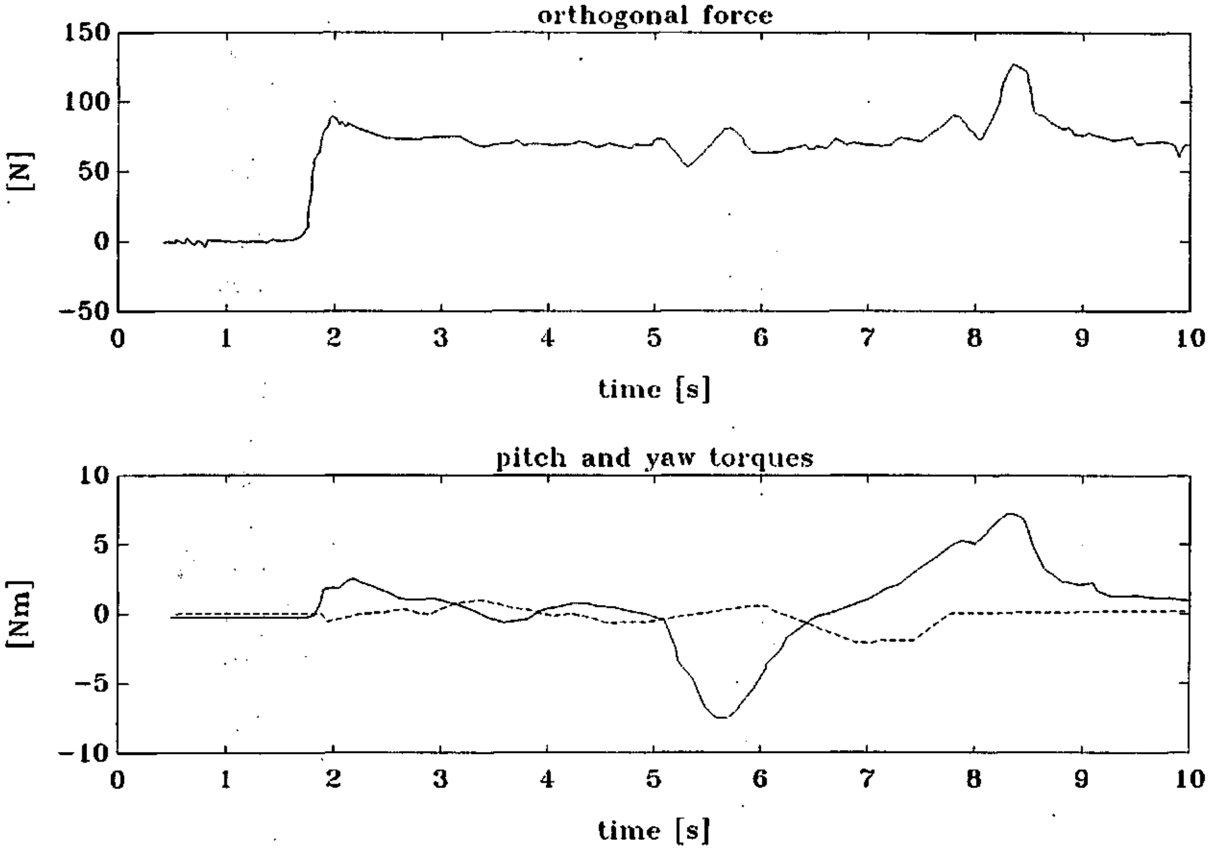

Figure 9: Maintained force of the tool (in the direction normal to the surface) and torques during deburring (in the direction of the movement and orthogonal to the direction of the movement). Note that the shape of the object changes (Fig. 8)

other words, the proposed control scheme will not work with sensors with high stiffness and robots with poor position resolution.

- the sampling interval of the force control loop is the same as the sampling frequency of the trajectory generation module.

Therefore, the proposed method is suitable for compliant tasks at low speed. In the proposed control law the environment stiffness is directly multiplied by the force control gain. Additionally, time delays introduced by the interpolation algorithms and communications between main and axis processor affect the stability of the control algorithm. To avoid this problem we proposed simple direct model reference adaptive controller. A discrete root locus was used for the force controller design. The results were verified with a simulation and compared with the response of the actual robot. The paper shows that the root locus design is also suitable for a multi-joint case in the case of the high gear robot and low speed robot movements.

References

[1] S.D. Eppinger and W.P Seering. On dynamic models of robot force control. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 29–34, 1986.

[2] V. Hayward L. Daneshemend and M. Pelletier. Adaptation to environment stiffness in the control of manipulators. *Lecture Notes in Control and Information Science, Experimental Robotics I.*, Vol. 1, No. 1, pages 150–164, 1989.

[3] L.S. Kersenbaum T.R Fortescue and B.E. Ydstie. Implementation of self–tuning regulators with variable forgetting factors. *Automatica*, Vol. 17, No. 6, 1981.

[4] C.G. Goodwin and K.S. Sin. *Adaptive Prediction, Filtering and Control.* Prentice–Hall, 1984.

[5] N. Hogan. Impedance control of industrial robots. *Journal of Robotics and Computer-Integrated Manufacturing*, Vol. 1, No. 1, pages 97–113, 1984.

[6] O. Kathib. A unified approach for the motion and force control of robot manipulators: The operational space formulation. *IEEE Journal of Robotics and Automation*, Vol. 2, No. 2, pages 83–106, 1987.

[7] H. Kazerooni, T.B. Sheridan, and P.K. Houpt. Robust compliant motion for manipulators. *IEEE Journal of Robotics and Automation*, Vol. 2, No. 2, pages 83–106, 1986.

[8] Y. Landau. *Adaptive Control.* Marcel Dekker, 1979.

[9] M.T. Mason. Compliance and force control for computer controlled manipulatotrs. *IEEE trans. on System Man and Cybernetics*, Vol. 11, No. 6, 1981.

[10] B. Nemec, A. Ružić, V. Ilc. RRL — an integrated envirnonment for robot programming. *Informatica*, Vol. 16, No. 1, pages 27-33, 1992

[11] R.P. Paul and B. Shimano. Compliance and control. In *Proc. of the Joint Automatic Control Conference*, pages 694–699, July 1976.

[12] M.H. Raibert and J.J. Craig. Hybrid position/force control of manipulators. *Journal of Energy Resources Technology*, Vol. 102, pages 126–133, 1981.

[13] J.K. Salisbury. Active stiffness control of a manipulator in cartesian coordinates. In *Proceedings of 19th Conference on Decision and Control*, pages 95–100, 1980.

[14] D.E. Whitney. Force feedback control of manipulator fine motions. *Trans. on ASME and Journals of Dynamics Systems Measurement and Control*, pages 91–97, 1977.