Proceedings of DETC98
1998 ASME Design Engineering Technical Conference
September 13-16, 1998, Atlanta, GA

DETC98/MECH-5870

NATURALLY-TRANSITIONING RATE-TO-FORCE CONTROL

Robert L. Williams II
Mark A. Murphy
Ohio University
Athens, OH 45701

ABSTRACT

The Naturally-Transitioning Rate-to-Force Controller (*NTRFC*) for manipulators is presented. In free motion rate control is provided, while in contact the same rate commands are proportional to the force exerted on the environment by the manipulator. The transition between free motion and stable contact with the environment requires no changes in control mode or gains and hence is termed natural. The *NTRFC* has been experimentally implemented and shows great promise. This paper demonstrates the *NTRFC* concept and provides a basis for its modeling and design.

1. INTRODUCTION

Two fundamental problems in manipulator control are free motions in Cartesian space and contacting the environment during task performance. Resolved-rate control has been around for a long time (Whitney, 1969). However, rate control has not been widely implemented in practical industrial and remote operations, perhaps due to the difficulty of rate control in contact. If a constant rate is commanded while the manipulator contacts the environment, joint angles integrate until unacceptably large forces are exerted.

Compared to an inverse pose algorithm, the resolved-rate algorithm is attractive because it is a linearized, unique solution (assuming full rank for the Jacobian matrix). Also, control inputs from various sources can be summed linearly to form the total input command. Both inverse pose and inverse rate schemes are subject to the same singularities.

Craig and Raibert (1981) presented a hybrid control method wherein some Cartesian axes are controlled in position while the remaining axes are force controlled. While this method is effective in practical tasks, it does not use rate control, and one must choose either position or force on each Cartesian axis. Hogan (1985) presented an impedance controller where the behavior of a manipulator is controlled to mimic a 6-dof Cartesian *m-c-k* system. Whitney (1985) reviews various force control architectures.

Goldenberg et.al. (1989) present a approach which compensates for unknown loading and parameter uncertainty in computed torque manipulator control. Colbaugh et.al. (1993) present an adaptive scheme for controlling the end-effector impedance of robot manipulators in contact; however, an explicit control mode change is required for free motion. Hyde and Cutkosky (1994) experimentally evaluate several methods for controlling the transition from free motion to constrained motion, using a one-axis impact testbed. Yao and Tomizuka (1995) present an adaptive motion and force controller for manipulators with uncertainties in both the robot and contact surfaces. Vukobratovic et.al. (1996) consider the problem of simultaneous stabilization of both the robot motion and interaction force in Cartesian space after contact in robotic tasks. Tarn et.al. (1996) use an event-driven switching control strategy for robot impact control and force regulation where the instant of impact is required.

The current paper presents a manipulator control method for effective task performance, the Naturally-Transitioning Rate-to-Force Controller (*NTRFC*). In free motion the manipulator moves with rate control, while in contact with the environment the force/moment wrench exerted on the environment is controlled. No artificial control mode or gain parameter changes are required so the transition is termed natural. A wrist-mounted F/T sensor and Force/Moment Accommodation (*FMA*) algorithm are required. Rate and *FMA* are active on all Cartesian axes simultaneously so no hybrid scheme is necessary. Since there are no artificial mode changes required, the threshold of contact is unimportant.

The *NTRFC* was discovered through serendipity at NASA Langley Research Center (Williams et.al., 1996). It was implemented experimentally and proven very effective in completion of representative space telerobotics tasks (Willshire et.al., 1992). The authors are currently implementing the *NTRFC* in a different experimental system at Wright-Patterson AFB. Though the *NTRFC* has shown great promise in the lab, its previous development has been exclusively heuristic. Therefore, the goal of this paper is to demonstrate the *NTRFC* and provide methods for its design. Models and design methods are developed and evaluated. Considered are controller and manipulator dynamics, multiple degrees of freedom (dof), transient and steady-state response, and stability, which were ignored in previous *NTRFC* work.

2. *NTRFC* DESCRIPTION

This section presents the Naturally-Transitioning Rate-to-Force Controller (*NTRFC*) concept. It is applicable to control of any manipulator(s) with wrist-mounted force/torque (F/T) sensor, rate inputs, and contact with the environment. The system is presented for one manipulator, but dual-arm control has also been implemented (Williams et.al., 1997).

Figure 1 shows the coordinate frames for a manipulator. The *World*, *Base*, and *Wrist* frames are familiar. The Moving Reference Frame (*MRF*, denoted {*M*}) is the user-defined control frame. The Control Reference Frame (*CRF*) is the frame with respect to which the *MRF* is controlled. Figure 2 shows the *NTRFC* high-level control diagram. The two basic active ingredients are the resolved rate and force/moment accommodation (*FMA*) algorithms, described below.

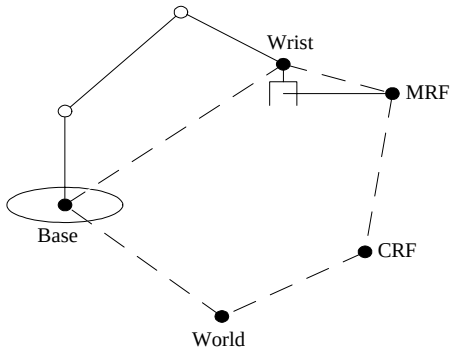

Figure 1. Manipulator Coordinate Frames

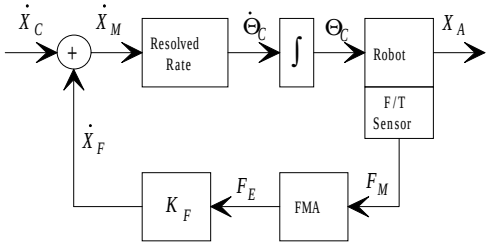

Figure 2. NTRFC Control Diagram

2.1 Resolved-Rate Control

The resolved-rate algorithm is based on Whitney (1969). The time-varying manipulator Jacobian matrix *J* maps the joint rates to Cartesian rates of the frame of interest, $\dot{X} = J\dot{\theta}$. The simplest symbolic Jacobian terms result when the frame of interest is *Wrist*. Rate inputs in {*M*}, $\dot{X}_M = \{v_M \quad \omega_M\}^T$, are achieved by rigid body velocity transformations back to *Wrist* (Craig, 1989):

$$ {}^k\dot{X}_W = {}^k\begin{Bmatrix} v_W \\ \omega_W \end{Bmatrix} = \begin{bmatrix} {}^k_jR & {}^k_WR\,{}^WP_M \times {}^k_jR \\ 0 & {}^k_jR \end{bmatrix}^j \begin{Bmatrix} v_M \\ \omega_M \end{Bmatrix} \tag{1} $$

$\dot{X}_M$ represents the six-dof Cartesian rates of {*M*} with respect to {*Base*}, but can be expressed in any coordinates {*j*} (e.g. *CRF*, *Base* or *World*). The translational and rotational velocity vectors are *v* and ω, ${}^i_mR$ is the orthonormal rotation matrix giving the orientation of {*m*} with respect to {*i*}, ${}^iP_m$ is the position vector from {*i*} to {*m*} expressed in {*i*}, and {*k*} is the Jacobian reference frame, Now the rate equation is inverted (alternatively, solved by Gaussian elimination) to calculate the instantaneous joint rates necessary to obtain the commanded Cartesian rate ${}^k\dot{X}_W$:

$$ \dot{\Theta}_C = {}^kJ(\Theta_A)^{-1}\,{}^k\dot{X}_W \tag{2} $$

The commanded joint rates are numerically integrated to commanded joint angles Θ_C. The manipulator attempts to achieve the currently commanded Θ_C; the actual joint angles Θ_A and Cartesian pose X_A result. The Jacobian is a function of configuration Θ_A.

The resolved-rate algorithm is sensitive to singularities, where the manipulator loses freedom to move in one or more Cartesian direction. In the neighborhood of singularities, extremely high joint rates are required to satisfy a finite Cartesian command. To deal with this problem, the determinant of the Jacobian matrix ${}^kJ$ must be monitored. When the determinant approaches zero, the matrix inverse (or Gaussian elimination) in Eq. 2 is replaced by a matrix pseudoinverse based on Singular Value Decomposition (*SVD*). Near singularities, the exact Cartesian command ${}^k\dot{X}_W$ cannot be satisfied, but the *SVD* yields bounded joint rates which moves the manipulator through the singular neighborhood until Eq. 2 can take over again.

Compared to an inverse position algorithm, the resolved-rate algorithm is attractive because it is a linearized, unique solution (assuming full rank for the Jacobian matrix). Also, control inputs from various sources are summed linearly to form the final command. An example for this is given below: Force/Moment Accommodation.

2.2 Force/Moment Accommodation

If the manipulator is in contact with its environment, there are constraints on X_A (the actual Cartesian pose in Fig. 2) and a Cartesian wrench exists. In this paper, wrench indicates a six-dof force/moment vector. An impedance controller (Hogan, 1985) with only the damping term has been implemented in the resolved-rate scheme to command forces to the environment with the manipulator. A six-dof wrist-mounted force/torque sensor reads the current contact wrench $F_S = \{f_S \quad m_S\}^T$ expressed in F/T sensor frame {*S*}. The weight and moment of the end-effector mounted outboard of the sensor must be subtracted from the sensor reading, accounting for manipulator configuration. The modified sensor reading is transformed to the *MRF* wrench, F_M (Craig, 1989):

$$ F_M = \begin{Bmatrix} f_M \\ m_M \end{Bmatrix} = \begin{bmatrix} {}^M_SR & 0 \\ {}^MP_S \times {}^M_SR & {}^M_SR \end{bmatrix} \begin{Bmatrix} f_S \\ m_S \end{Bmatrix} \tag{3} $$

An error vector $F_E = F_C - F_M$ is formed from the difference of the sensed and commanded wrenches in the *MRF* and converted to a rate $\dot{X}_F = K_F F_E$, sent to the summing junction in Fig. 2. This rate drives the manipulator motion so the desired force is achieved continuously. The diagonal gain matrix K_F has units *m*/*Ns* and *rad*/*Nms* for translational and rotational terms, respectively.

If a desired contact wrench F_C is commanded and the manipulator is in free motion, the rate input $\dot{X}_F$ will move the manipulator in the six-dof direction of F_C until the F/T sensor senses F_C through contact with the environment. Then F_C is maintained without any controller changes.

If zero wrench is commanded ($F_C = 0$) and the manipulator is in free motion, $\dot{X}_F = 0$ (assuming a perfect F/T sensor) because there is zero contact wrench F_M. If zero wrench is commanded and the manipulator contacts the environment, the $\dot{X}_F = -K_F F_M$ motion will automatically align the manipulator end-effector for minimal Cartesian contact wrench and misalignments. This is called *force/moment accommodation* (*FMA*).

2.3 Naturally-Transitioning Rate-to-Force Controller

In the Naturally-Transitioning Rate-to-Force Controller (*NTRFC*), the resolved-rate algorithm acts simultaneously with the *FMA* algorithm (see Fig. 2), for all Cartesian axes (no hybrid scheme is necessary). The overall resolved-rate input is the sum of the commanded rate and the *FMA* rate, $\dot{X}_M = \dot{X}_C + \dot{X}_F$. As the manipulator end-effector approaches a wall in the environment, the rate controller commands motion through the wall, but the *FMA* controller commands a reverse motion to exert zero wrench. Therefore, an equilibrium condition is entered, where the rate input is proportional to the exerted Cartesian contact wrench. The *NTRFC* automatically corrects misalignments so insertion tasks can be completed with minimum contact wrenches. If no force controller is used, it is difficult to complete tasks since the manipulator is "blind" in the wrench sense.

The system behaves as a rate controller in free motion and as a force controller in contact. The transition requires no mode changes, logical switches, or gain changes in the controller software or hardware and thus is termed a natural transition. The transition is a consequence of the physics of manipulator contact with the environment when using the control architecture of Fig. 2. Assuming a well-calibrated F/T sensor with minimal noise, the *NTRFC* does not care when the moment of contact occurs. The *FMA* algorithm is enabled continuously (on all Cartesian axes, simultaneously with rate control on all Cartesian axes), but only generates non-zero $\dot{X}_F$ in contact. The next section presents modeling and controller design for two 3-dof manipulators operating with the *NTRFC* .

3. *NTRFC* MODELING

This section presents dynamics and control modeling for a spatial 3*P* manipulator and a planar 3*R* manipulator in motion under the *NTRFC*. Since the free-motion to contact transition is a natural one, we must obtain desirable performance with only one set of gains and software control mode. Of interest is system stability, transient response, and steady-state response. Two control design procedures are presented, one in each of the following two subsections.

The *NTRFC* has been implemented heuristically in hardware at NASA Langley Research Center. The *NTRFC* is currently being implemented at Wright-Patterson AFB. The goal of this section is to provide an analytical (as opposed to heuristic) basis for *NTRFC* design. In hardware implementation the manipulator dynamics and environment characteristics are provided by the real world. In modeling, the control diagram in Fig. 2 must be expanded to model these real world effects, as shown in Fig. 3.

Figure 3 is the same as Fig. 2, with the following added: 1) The Resolved Rate block is identified as the inverse Jacobian mapping, which is a function of the actual (modeled) joint angles Θ_A ; 2) The commanded joint angles Θ_C are achieved using linear *PID* control for each joint independently; 3) The *PID* algorithms collectively yield the vector of input joint torques, from which the actual joint angles Θ_A are solved using forward dynamics, the manipulator equations of motion, and the contact wrench; 4) Forward Kinematics (pose and rate) are calculated to predict the current actual Cartesian pose X_A and rate (not shown); 5) The environment model predicts the contact wrench F_M (we assume perfect F/T sensor). In computer modeling of *NTRFC* motion, an artificial environment switch is used, but this is not necessary for hardware systems since the F/T sensor reading is continuously used; and 6) The *FMA* block is identified as the force error equation $F_E = F_C - F_M$. As in the Fig. 2 case, *FMA* is continuously enabled, but only generates non-zero $\dot{X}_F$ when the manipulator is in contact with the environment.

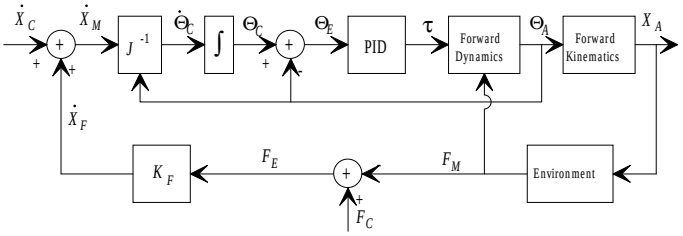

Figure 3. NTRFC Dynamics and Control Modeling Diagram

3.1 Spatial 3*P* Manipulator

We first model a 3-dof spatial serial manipulator consisting of three orthogonal prismatic joints (*P*) and present the first *NTRFC* design procedure. The 3*P* diagram is presented in Fig. 4a (front view, *YZ* plane) and Fig. 4b (right side view, *XZ* plane). Prismatic actuator 1 and its ground connection are not shown in Fig. 4a for clarity. The manipulator is modeled as three lumped masses m_i each with viscous dampers c_{Ai}, i=1,2,3. The relative manipulator/environment compliance is modeled as three spring/damper combinations k_{Ej}/c_{Ej}, j=*x,y,z*. Variable actuator lengths L_{A1}, L_{A2}, L_{A3} operate along the *X*, *Y*, *Z* axes, respectively. Fixed lengths L_{0j}, j=*x,y,z*, give the distance along each axis from the origin to the undisplaced environment location. Cartesian variables *x,y,z* are measured from the ends of L_{0j}.

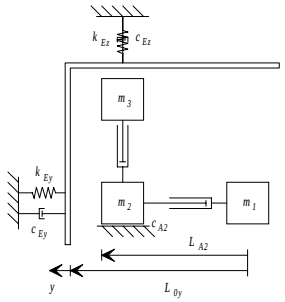

Figure 4a. 3P YZ Plane

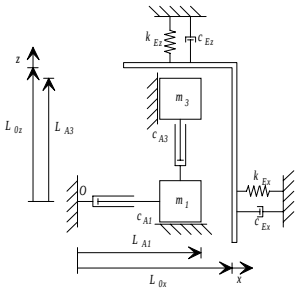

Figure 4b. 3P XZ Plane

The 3*P* manipulator has trivial resolved-rate ($J = J^{-1} = I_3$) and forward kinematics (L_{A1}, L_{A2}, L_{A3} are the total displacements along *X,Y,Z*) algorithms. No rotations are possible. The dynamics equations of motion are decoupled, each axis represented by:

$$f_{Ai} + f_{Mj} = m_i \ddot{L}_{Ai} + c_{Ai} \dot{L}_{Ai} \qquad i=1,2,3 \qquad (4)$$

where f_{Ai} is the i^{th} actuator force and f_{Mj} is the j^{th} Cartesian *MRF* contact force (j=*x,y,z* corresponds to i=1,2,3). Now, the first actuator must accelerate all three masses (the second two and the third just m_3)

but reaction forces are resisted by the structure and not by other actuators because they are orthogonal. Therefore, the *NTRFC* design can proceed for each axis independently. Figure 5 shows joint 1 and the *X* axis (*Y* and *Z* are similar).

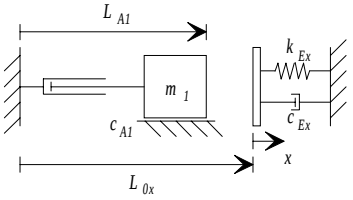

Figure 5. 3P Joint 1, X Axis

Given the 3*P* simplifications mentioned above, a linear SISO diagram can be obtained for each joint by simplifying Fig. 3, shown in Fig. 6. In Fig. 6, commanded rate $\dot{x}_C$ is the input. In free motion, L_A is the output, $-L_0$ is ignored, and f_M is zero so k_F has no effect (f_C is zero for *FMA*). In contact with the environment the input $\dot{x}_C$ no longer causes free motion but instead exerts a force on the environment (f_M is the F/T sensor reading, hence the force of the environment exerted back on the manipulator). Length L_0 (distance from the origin to the undisplaced environment) acts as a disturbance in this SISO system. The *PID* controller, joint dynamics, and environment transfer functions are:

$$G_C = k_P + \frac{k_I}{s} + k_D s \qquad G = \frac{1}{ms^2 + c_A s} \qquad G_E = -c_E s - k_E \quad (5)$$

Now *NTRFC* design is presented for the 3*P* manipulator using the *First NTRFC Design Procedure* (see below). Given desired contact force transient performance, calculate gains k_P, k_I, k_D, k_F for the three axes.

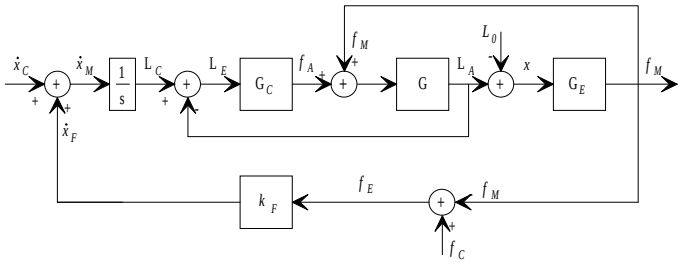

Figure 6. SISO Prismatic Joint NTRFC Block Diagram

First *NTRFC* Design Procedure

For simple systems with decoupled kinematics and dynamics and linear models. Derive the SISO transfer function for each independent axis; assume contact with the environment exists. Set desired (stable) transient performance characteristics of contact force given rate step input. Determine desired fourth-order characteristic polynomial based on dominant second-order system. Using parameter matching calculate *PID* and k_F gains for each joint separately. Determine stability ranges. Ensure same *PID* gains yield acceptable performance under free motion as well (k_F is always enabled but has no effect until contact). Simulate results.

***NTRFC* Design for One *P* Joint**

The linear superposition principle is used to find the total contact force output f_M given the rate input $\dot{x}_C$ (with L_0=0) and the disturbance input $-L_0$ (with $\dot{x}_C = 0$):

$$T_{\dot{X}} = \frac{f_{M1}}{\dot{x}_C} \qquad T_L = \frac{f_{M2}}{L_0} \qquad f_M = f_{M1} + f_{M2} = T_{\dot{X}}\dot{x}_C - T_L L_0 \quad (6)$$

The closed-loop transfer function $T_{\dot{X}}$ for Fig. 6 under contact is (with L_0=0):

$$T_{\dot{X}} = \frac{-\left[k_D c_E s^3 + (k_D k_E + k_P c_E)s^2 + (k_P k_E + k_I c_E)s + k_I k_E\right]}{a_4 s^4 + a_3 s^3 + a_2 s^2 + a_1 s + a_0} \quad (7)$$

where:

$$a_4 = m$$
$$a_3 = c_A + c_E + k_D - k_D c_E k_F$$
$$a_2 = k_E + k_P - (k_D k_E + k_P c_E)k_F$$
$$a_1 = k_I - (k_P k_E + k_I c_E)k_F$$
$$a_0 = -k_I k_E k_F$$

The closed-loop transfer function T_L for Fig. 6 under contact (with $\dot{x}_C = 0$) is different from Eq. 7, but has the same characteristic polynomial. Therefore, design for transient response affects both superposition components in the same manner.

Given a step input $\dot{x}_C$, we set two desired transient performance characteristics for f_M: 4% overshoot and 1 *sec* settling time ($\pm 2\%$), which leads to $\xi = 0.72$ and $\omega_n = 5.59$, for a dominant second-order characteristic polynomial $s^2 + 8s + 31.24$. Since we have four unknowns and a fourth-order characteristic polynomial, the dominant second-order polynomial is augmented with two negative real poles at least 10 times greater than the real part of the dominant poles to yield: $s^4 + 98s^3 + 2751s^2 + 18812s + 62482$ so $a_4 = 1$, $a_3 = 98$, $a_2 = 2751$, $a_1 = 18812$, and $a_0 = 62482$. The corresponding poles are $s_{1,2,3,4} = -4 \pm 3.90i, -40, -50$.

Now parameter matching is used to derive an analytical solution for the four unknown gains. First, the desired fourth-order polynomial must be uniformly scaled so the leading coefficient is $a_4 = m$; then the remaining coefficient equations from Eq. 7 are solved for the unknowns:

$$k_I = \frac{-a_0}{k_E k_F} \qquad k_D = \frac{a_3 - c_A - c_E}{1 - c_E k_F} \qquad k_P = \frac{(a_0 c_E - a_1 k_E)k_F - a_0}{(k_E k_F)^2}$$

$$b_3 k_F^3 + b_2 k_F^2 + b_1 k_F + b_0 = 0 \quad (8)$$

where:

$$b_3 = (a_0 c_E - a_1 k_E)c_E^2 + a_2 c_E k_E^2 + (c_A - a_3)k_E^3$$
$$b_2 = -a_0 c_E(1 + 2c_E) + 2a_1 c_E k_E - a_2 k_E^2 + k_E^3$$
$$b_1 = 3a_0 c_E - a_1 k_E$$
$$b_0 = -a_0$$

and a_i, i=0,1,2,3 are the desired fourth-order polynomial coefficients (to force the desired behavior in the Eq. 7 denominator). The third order polynomial in Eq. 8 is solved for k_F. Since coefficients b_i are real, at least one real root is guaranteed. Choose the real k_F value and then the first three expressions of Eq. 8 yield the unknowns k_P, k_I, k_D given k_F. The steady-state contact force and environment displacements are found using the final value theorem (to predict these steady-state values and validate the simulation).

$$f_{M1_{SS}} = \lim_{t\to\infty} f_{M1}(t) = \lim_{s\to 0} sf_{M1}(s) = \lim_{s\to 0} sT_{\dot{X}} \frac{\dot{x}_C}{s} = \frac{\dot{x}_C}{k_F} \qquad f_{M2_{SS}} = 0$$

$$f_{M_{SS}} = f_{M1_{SS}} + f_{M2_{SS}} = \frac{\dot{x}_C}{k_F} \qquad x_{SS} = \frac{-f_{M_{SS}}}{k_E} = \frac{-\dot{x}_C}{k_E k_F} \qquad (9)$$

3*P* Manipulator Simulation Example

The *First NTRFC Design Procedure* is applied, using Eqs. 8 three times independently, one for each *P* joint. Given the following parameters, solve for $k_{Pi}, k_{Ii}, k_{Di}, k_{Fj}$ (i=1,2,3 and j=x,y,z) and simulate *NTRFC* motion. Note that *i* represents joint space and *j* Cartesian space, identical for the 3*P*. Standard *SI* units are used.

$$\begin{array}{llllll} m_1 = 3 & c_{A1} = 0.3 & L_{0x} = 0.20 & \dot{x}_C = 0.1 & k_{Ex} = 100 & c_{Ex} = 0.7 \\ m_2 = 2 & c_{A2} = 0.5 & L_{0y} = 0.10 & \dot{y}_C = 0.2 & k_{Ey} = 100 & c_{Ey} = 0.7 \\ m_3 = 1 & c_{A3} = 0.4 & L_{0z} = 0.12 & \dot{z}_C = 0.3 & k_{Ez} = 100 & c_{Ez} = 0.7 \end{array}$$

Table 1 summarizes the 3*P* solution for *NTRFC* design. For joint 1 $m = m_1 + m_2 + m_3$, for joint 2 $m = m_2 + m_3$, and for joint 3 $m = m_3$. The desired characteristic polynomial for each case (derived above) is $s^4 + 98s^3 + 2751s^2 + 18812s + 62482$, uniformly scaled so the leading coefficient is $a_4 = m$ for each joint.

Table 1. 3P NTRFC Design Results

Gain	Axis 1	Axis 2	Axis 3
k_P	4063.0	2050.6	711.5
k_I	16197.0	8190.6	2867.2
k_D	505.1	252.4	84.1
k_F	-0.232	-0.229	-0.218

A MATLAB SIMULINK model was developed to simulate the 3*P* under *NTRFC* motion. In this simulation, the inputs $\dot{x}_C, \dot{y}_C, \dot{z}_C$ were each ramped up to their final value with a slope of 1. As shown in Fig. 7, the actuator lengths (also the global Cartesian displacements) each increase linearly under rate control in free motion, experience transient behavior (difficult to see at this scale), and assume their steady-state value after the natural transition from rate to force control. With the arbitrary simulation values chosen, each axis moves with different rates and the time of contact with the environment is different. The simulated *MRF* contact forces in Fig. 8 show these different times of contact. Also, though difficult to see at this scale, each force transient behavior satisfies the desired 4% overshoot and 1 *sec* settling time. These control goals are not met exactly due to the fourth-order approximation of the dominant second-order polynomial and the *PID* controller adds zeros to the system. Figure 8 shows that the contact forces are zero in free motion until each axis contacts the environment; they also experience transient behavior and assume steady-state (constant) force values after the transition *even though the rate commands are still applied*. Figure 9 shows the actuator forces required to achieve *NTRFC* motion. To initiate the constant-rate free motion, each force briefly approximates a step input. During the transition from rate to force control, each axis requires a sharp change in actuator force. The actuator force f_{A3} has a static offset $m_3 g$ to resist gravity loading.

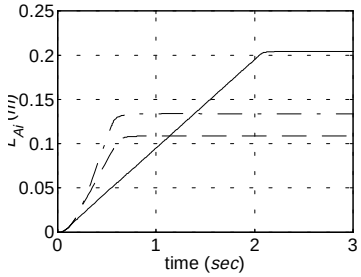

Figure 7. 3P Actuator Lengths
L_{A1} (solid), *L_{A2}* (dash), *L_{A3}* (dash-dot)

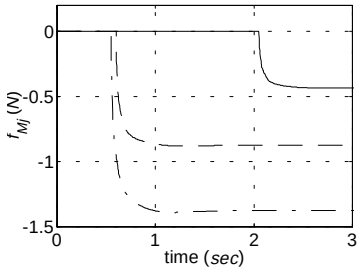

Figure 8. 3P Contact Forces
f_{Mx} (solid), *f_{My}* (dash), *f_{Mz}* (dash-dot)

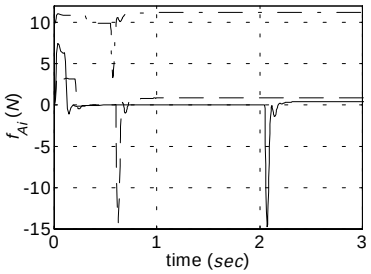

Figure 9. 3P Actuator Forces
f_{A1} (solid), *f_{A2}* (dash), *f_{A3}* (dash-dot)

The steady state values calculated from Eq. 9 are:

$$\begin{array}{ll} f_{Mx_{SS}} = -0.432 & x_{SS} = 0.00432 \\ f_{My_{SS}} = -0.874 & y_{SS} = 0.00874 \\ f_{Mz_{SS}} = -1.377 & z_{SS} = 0.01377 \end{array}$$

Given the *PID* gain values from Table 1, the gains k_{Fj} for each Cartesian axis were varied to investigate stability. The stability results shown in Fig. 10 are identical for each axis since the same characteristic polynomial (albeit scaled) was used for all three axes. Figure 10 reports the real part of the four poles for each axis. The full behavior is adequately represented by the range $-1 \le k_{Fj} \le 1$. The two complex conjugate poles (dash) have a relatively small negative real part over the entire k_{Fj} range. The two real poles are negative and identical up to k_{Fj} = -0.24 (close to the design values for each k_{Fj} in Table 1). After this point the real poles bifurcate. One becomes more negative. The other becomes zero at k_{Fj} = 0 and positive for positive k_{Fj}. Therefore, the 3*P* system under *NTRFC* motion is marginally stable for k_{Fj} = 0 and unstable for k_{Fj} > 0. This result makes sense physically since k_{Fj} > 0 would cause a rate to increase the contact force on axis *j*.

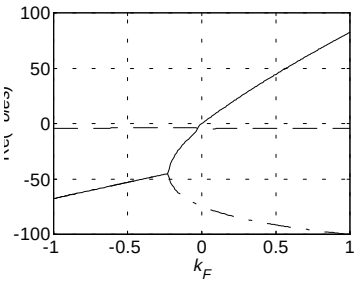

Figure 10. 3P Stability Results

Lastly, we complete the remaining step in the *First NTRFC Design Procedure*. The Table 1 gains were designed for the contact case; now we must analyze the free motion characteristics *using the same gains.* The motion looks fine in the free motion portions of Fig. 7, but we quantify this performance below. Referring to Fig. 6 and using G_C and G from Eq. 5, the transfer function representing *PID* control and joint dynamics is:

$$T_{L_A L_C} = \frac{L_A}{L_C} = \frac{k_D s^2 + k_P s + k_I}{ms^3 + (c_A + k_D)s^2 + k_P s + k_I} \qquad (10)$$

The three free-motion poles for all axes are nearly identical ($-4.23 \pm 4.21i, -75.8$, $-4.27 \pm 4.22i, -75.8$, $-4.46 \pm 4.25i, -75.6$ for joints 1, 2, and 3) and provide 7.9% overshoot and 0.26 *sec* settling time for output L_{Ai} with input L_{Ci}. All axes are stable for the designed *PID* gains since the poles all have strictly negative real parts.

3.2 Planar 3*R* Manipulator

We now model a 3-dof planar serial manipulator consisting of three parallel revolute joints (*R*) and present the second *NTRFC* design procedure. The 3*R* diagram is Fig. 11. The manipulator is modeled as three distributed masses m_i with inertia scalars I_i, i=1,2,3. The relative manipulator/environment compliance is modeled as three spring/damper combinations k_{Ej}/c_{Ej}, j=*x*,*y*,*r* (two translational and one rotational). The fixed lengths are L_1, L_2, L_3. The variable joint angles $\Theta_A = \{\theta_1 \quad \theta_2 \quad \theta_3\}^T$ are controlled by joint torques $\tau = \{\tau_1 \quad \tau_2 \quad \tau_3\}^T$. The rate inputs are $\dot{X}_C = \{\dot{x}_C \quad \dot{y}_C \quad \omega_z\}^T$, relative to *MRF* ($X_M$ is aligned with link 3). The Cartesian pose is $X_A = \{x \quad y \quad \phi\}^T$ and the Cartesian contact wrench is $F_M = \{f_x \quad f_y \quad m_z\}^T$. The 3*R* forward kinematics solution and Jacobian matrix reflect joint coupling and are straight-forward to derive. The three dynamics equations of motion are coupled and nonlinear, represented by:

$$\tau = M(\Theta_A)\ddot{\Theta}_A + V(\Theta_A, \dot{\Theta}_A) + G(\Theta_A) \qquad (11)$$

The terms in Eq. 11 are rather complex, but straight-forward to derive (Craig, 1989). The 3*R* system is coupled and nonlinear so the *First NTRFC Design Procedure* cannot be applied. Figure 3 represents the 3*R* case. Now the *Second NTRFC Design Procedure* is presented. The goal is still to calculate k_{Pi}, k_{Ii}, k_{Di} for the three *R* joints and k_{Fj} for the three Cartesian directions.

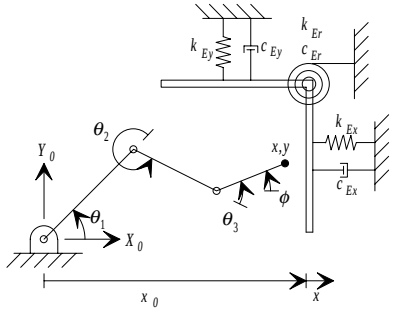

Figure 11. 3R Diagram

Second *NTRFC* Design Procedure

For more complex systems with coupled kinematics and dynamics and nonlinear models. Set desired (stable) transient performance characteristics of each joint angle given rate step input. Employ standard methods to design *PID* gains independently for each axis assuming decoupled dynamics and free motion. Using same *PID* gains, choose diagonal matrix gain K_F to ensure stable, desired performance of contact wrench given rate step inputs. Determine stability ranges. Simulate results.

3*R* Manipulator Simulation Example

Given the following parameters, solve for k_{Pi}, k_{Ii}, k_{Di} (i=1,2,3), determine K_F, and simulate *NTRFC* motion. Standard *SI* units are used and angle units are *deg*.

$L_1 = 0.3$	$m_1 = 2$	$I_1 = 0.5875$	$k_{Ex} = 100$	$c_{Ex} = 0.7$
$L_2 = 0.2$	$m_2 = 1.5$	$I_2 = 0.0775$	$k_{Ey} = 100$	$c_{Ey} = 0.7$
$L_3 = 0.1$	$m_3 = 1$	$I_3 = 0.0025$	$k_{Er} = 50$	$c_{Er} = 0$

Assuming decoupled dynamics and free motion, the *PID* gains are determined independently for each joint. The transfer function relating actual joint angle to commanded joint angle is T_θ, where the decoupled linearized joint dynamics plant G was used:

$$T_\theta = \frac{\theta_A}{\theta_C} = \frac{k_D s^2 + k_P s + k_I}{Is^3 + (c + k_D)s^2 + k_P s + k_I} \qquad G = \frac{1}{Is^2 + cs} \qquad (12)$$

Viscous damping coefficient c = 0.2 is included for each joint. Now we specify 3% overshoot and 0.5 *sec* settling time for each θ_A in free motion, yielding a desired characteristic polynomial (dominant second order augmented with real third pole ten times greater) $s^3 + 96s^2 + 1395s + 9230$. This polynomial must be scaled for each joint so $a_3 = I_i$ and then k_{Pi}, k_{Ii}, k_{Di} are calculated via parameter matching with the denominator of T_θ in Eq. 12. The results are:

Table 2. 3R Free Motion PID Design Results

Gain	Axis 1	Axis 2	Axis 3
k_P	819.8	108.1	3.5
k_I	5422.4	715.3	23.1
k_D	56.2	7.2	0.04

A MATLAB SIMULINK model was developed to determine *FMA* diagonal gain matrix K_F and simulate the 3*R* under *NTRFC* motion. In this example, the 3*R* manipulator is to make contact with the environment by moving along the horizontal *X* axis at a constant rate $\dot{X}_C = \{0.05 \quad 0 \quad 0\}^T$; this could simulate a peg-in-the-hole task.

However, assume there is an unknown and undesired angular misalignment: Given initial angles $\Theta_{A_{in}} = \{45 \quad -90 \quad 65\}^T$, the forward kinematics solution yields the initial Cartesian pose $X_{A_{in}} = \{0.448 \quad 0.105 \quad 20\}^T$ (similar to Fig. 11 pose), with an angular misalignment $\phi = 20$. Therefore, when commanding $\dot{X}_C$ in *MRF* coordinates, the resulting motion will not be horizontal, but along link 3, inclined at 20 *deg*. Free motion will continue until the environment has been contacted in the X_0 direction (at $x_0 = 0.50$, see Fig. 11). Assume perfect contact in all Cartesian directions from that point in time forward. Then the equilibrium point for *Y* environment motion is $y_0 = 0.124$ which is the point of initial contact, but the equilibrium angle is $\phi_0 = 0$ which represents an immediate angular misalignment upon contact. The point of this simulation is to demonstrate how the angular misalignment will automatically correct itself under *NTRFC* motion.

Using the Table 2 free motion *PID* gains and trial-and-error with SIMULINK , a "good" value for K_F in contact was found to be a diagonal matrix of dimension 3 with all three diagonal elements -0.1. "Good" is defined to be stable with reasonable transient performance. Figures 12 through 14 show simulated results for this example. Note in Figs. 13 and 14 the angular terms have separate scales on the right. Figure 12 shows the simulated joints angles Θ_A. As shown in Fig. 13, the Cartesian pose variables *x, y* in {0} coordinates each increase linearly under rate control in free motion, experience transient behavior (difficult to see at this scale), and assume their steady-state value after the natural transition from rate to force control. Since there is only an *X* rate command, the *x* steady-state value compresses the environment, beyond $x_0 = 0.50$, but the *y* steady-state value settles at $y_0 = 0.124$. Due to simulated forward dynamics, ϕ briefly exceeds the misalignment 20 *deg* but then maintains that value in free motion. Upon contact, the *NTRFC* drives the manipulator to rectify the angle misalignment, sending ϕ to $\phi_0 = 0$. Figure 14 shows the simulated Cartesian contact wrench. The component f_x behaves similarly to any axis of the 3*P* case (see Fig. 8), except there is a more interesting transient due to coupled dynamics. The component f_y also starts at zero in free motion, experiences a transient which gradually increases from zero, and then settles down to zero since $\dot{y}_C = 0$. The contact moment m_z is also zero in free motion, but experiences a step change on contact due to the angle misalignment. After the transient, the steady-state moment is zero since there is no ω_z rate command. When the moment step change occurs, the rotational term of the rate input $\dot{X}_F$ from *FMA* is no longer zero, but drives the manipulator in the direction to relieve the moment and hence the angular misalignment. When the angular misalignment has been eliminated, the *MRF* coordinates line up with {0} as desired.

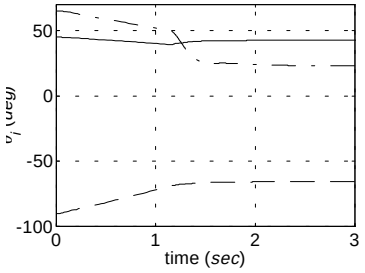

Figure 12. 3R Joint Angles
θ_1 *(*solid*),* θ_2 *(*dash*),* θ_3 *(*dash-dot*)*

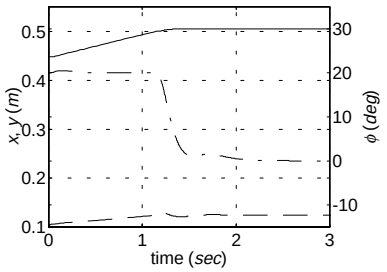

Figure 13. 3R Cartesian Pose
*x (*solid*), y (*dash*),* ϕ *(*dash-dot*)*

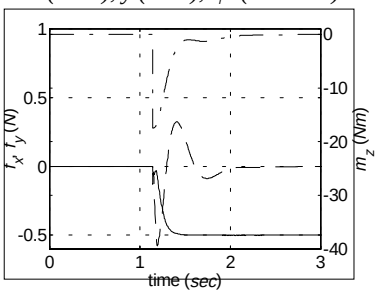

Figure 14. 3R Contact Wrench
f_x *(*solid*),* f_y *(*dash*),* m_z *(*dash-dot*)*

Even though the 3*R* system is coupled and nonlinear, the steady-state values can be calculated using Eq. 9. The result is $f_{x_{SS}} = -0.50$, $\Delta x_{SS} = 0.0050$, $f_{y_{SS}} = m_{z_{SS}} = \Delta y_{SS} = \Delta \phi_{SS} = 0$. Stability analysis was conducted by SIMULINK simulation. It was found that the 3*R* has the same stability conditions as the 3*P* case: marginally stable for any one $k_{Fj} = 0$ and unstable for any one $k_{Fj} > 0$. However, perhaps due to the joint coupling, the unstable behavior was different for the two cases. The 3*P* suffers exponential increases in X_{Aj} and f_{Mj} whenever $k_{Fj} > 0$. For the 3*R*, the unbounded outputs increase only linearly (with oscillations about the line) whenever any one $k_{Fj} > 0$.

Due to limited space, only the "good" case is reported for both 3*P* and 3*R* cases. Behavior varies widely for other K_F values. Unstable cases are discussed in the previous paragraph. For K_F values negative and larger than the design results reported, the systems are stable and achieve steady-state values as calculated in Eq. 9. However, significant and unacceptable transient oscillations can occur. For K_F values negative and smaller than the design results reported, the systems are stable but sluggish to reach steady-state. Finally, when all $k_{Fj} = 0$, our stability conclusions predict marginally stable systems. However, this case corresponds to turning off the *FMA* algorithm and hence no natural transition from rate to force control occurs. Given a constant rate command $\dot{X}_C$, the manipulator performs fine in free motion, but without *FMA* generates unacceptably high forces in contact. With properly-designed K_F, the *NTRFC* provides excellent contact characteristics with a rate controller and no mode changes.

4. TELEOPERATION

The *NTRFC* methods of this paper apply to any manner of commanding a robot. That is, the Cartesian rate command $\dot{X}_C$ can come from an automated path planning algorithm, real-time sensory feedback (such as machine vision or proximity sensing), and/or teleoperated human inputs via hand controller.

When a manipulator is controlled using the *NTRFC* and teleoperation, the following behavior results. In free motion, the displacement of the human's hand with the hand controller is proportional to the manipulator Cartesian rate. In contact, the displacement of the human's hand is proportional to the Cartesian

wrench exerted by the manipulator on the environment because the control has transitioned naturally from rate to force. Furthermore, if the hand controller enables force-reflection to the operator, the displacement of the hand controller is still proportional to the Cartesian rate in free motion. But in contact, the wrench of the human's hand reacting to the force-reflection provided by the hand controller is proportional to the Cartesian wrench exerted by the manipulator on the environment (assuming static equilibrium).

The *NTRFC* was implemented in manipulator teleoperation at NASA Langley Research Center, with and without force-reflecting hand controllers. The *NTRFC* significantly improved operator performance, both subjectively and with objective performance measurements. Task completion time and especially unwanted contact wrenches arising from misalignment were reduced significantly using the *NTRFC* compared to rate control without *FMA*. Performance with the *NTRFC* and force-reflecting hand controllers was even better. For details, see Willshire et.al. (1992).

5. CONCLUSION

This paper has presented a manipulator control algorithm, the Naturally-Transitioning Rate-to-Force Controller (*NTRFC*). Existing rate control methods are often preferable to inverse position control in free motion, but unacceptable in contact with the environment due to integration of commands while under motion constraints, which builds up high contact wrenches. The *NTRFC* provides a method by which rate control in free motion naturally transitions to force control in contact. A force/moment accommodation (*FMA*) algorithm is the key ingredient. A wrist-mounted F/T sensor is required on the manipulator. The transition occurs due to the combination of the rate and *FMA* algorithms acting simultaneously on all Cartesian axes. No artificial mode or gain changes are necessary, hence the method is insensitive to knowing the exact moment of contact in hardware implementation.

Two manipulators were simulated, including controller and manipulator dynamics, transient response, and stability determination. Two design methods were presented to determine *NTRFC* gains. The design methods require knowledge of the manipulator and environment models. Also, the design method for complex, coupled, nonlinear systems requires trial-and-error with a computer simulation. Future goals are: 1) Apply adaptive control techniques to lessen the dependence on models; and 2) Develop analytical techniques for *NTRFC* design in nonlinear systems.

ACKNOWLEDGMENTS

This research was supported in part by the AFOSR Summer Faculty Research Program. Many thanks to researchers in the Human Sensory Feedback Lab at Wright-Patterson AFB for support and facilities. Credit is due also to NASA LaRC researchers for implementation of the original experimental system.

REFERENCES

R. Colbaugh, H. Seraji, and K. Glass, 1993, "*Direct Adaptive Impedance Control of Robot Manipulators*", Journal of Robotic Systems, 10(2): 217-248.

J.J. Craig, 1989, *Introduction to Robotics: Mechanics and Control*, Addison Wesley Publishing Co., Reading, MA.

A.A. Goldenberg, J.A. Apkarian, and H.W. Smith, 1989, "*Approach to Adaptive Control of Robot Manipulators using the Computed Torque Technique*", ASME Journal of Dynamic Systems, Measurement and Control, 111(1): 1-8.

N. Hogan, 1985, "*Impedance Control: An Approach to Manipulation*", ASME Journal of Dynamic Systems, Measurement, and Control.

J.M. Hyde and M.R. Cutkosky, 1994, "*Controlling Contact Transition*", IEEE Control Systems Magazine, 14(1): 25-30.

M. Raibert and J.J. Craig, 1981, "*Hybrid Position/Force Control of Manipulators*", ASME Journal of Dynamic Systems, Measurement, and Control, 103(2): 126-133.

T.J. Tarn, Y. Wu, N. Xi, and A. Isidori, 1996, "*Force Regulation and Contact Transition Control*", IEEE Control Systems Magazine, 16(1): 32-40.

M. Vukobratovic and R. Stojic, 1996, "*On Position/Force Control of Robot Interacting with Dynamic Environment in Cartesian Space*", ASME Journal of Dynamic Systems, Measurement and Control, 118(1): 187-92.

D.E. Whitney, 1969, "*Resolved Motion Rate Control of Manipulators and Human Prostheses*", IEEE Transactions on Man-Machine Systems.

D.E. Whitney, 1985, *"Historical Perspective and State of the Art in Robot Force Control"*, IEEE International Conference on Automation and Robotics, 262-268.

R.L. Williams II, F.W. Harrison, and D.I. Soloway, 1997, *"Shared Control of Multiple-Manipulator, Sensor-Based Telerobotic Systems"*, IEEE International Conference on Automation and Robotics, Albuquerque, NM.

R.L. Williams II, F.W. Harrison, and D.I. Soloway, 1996, *"Naturally-Transitioning Rate-to-Force Controller for Manipulators"*, IEEE International Conference on Automation and Robotics, Minneapolis, MN.

K. Willshire, F.W. Harrison, E.F. Hogge, R.L. Williams II, and D.I. Soloway, 1992, *"Results of Telerobotic Hand Controller Study Using Force Information and Rate Control"*, AIAA Paper 92-1451, AIAA Space Programs Conference, Huntsville, AL.

B. Yao and M. Tomizuka, 1995, "*Adaptive Control of Robot Manipulators in Constrained Motion - Controller Design*", ASME Journal of Dynamic Systems, Measurement and Control, 117(3): 320-328.