Force Control of an RPR manipulator
EL 522 Final Project

Griswald Brooks

May 25, 2012

Abstract

While positioning of manipulator end effectors is an important task, often applying a particular force to the environment is needed. This paper describes a basic hybrid force control technique and presents simulation results.

Contents

Chapter 1

Introduction

1.1 Concept and Motivation

Many manipulator applications require specific amounts of force to be applied to the environment. Tasks include machining, object manipulation, and part assembly. Without force control, the robot would apply an arbitrary amount of force as a function of its joint error and interaction with the environment. This would lead to unpredictable results and possible manipulator instability, e.g. causing the end effector to bounce or damage the object or environment with which it's interacting.

One method of force control is a feed forward method called impedance control. In this method the environment is modeled as a mass-spring or a mass-spring-dampened system and the required displacement needed for a desired force is calculated (1.1). The manipulator is then commanded to its intended tracking plus this displacement. This can be thought of as the manipulator tracking an inner virtual wall.

$$\delta x = \frac{F_d}{k} \tag{1.1}$$

Where F_d is the desired force, k is the modeled wall spring constant,

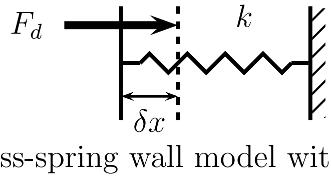

Figure 1.1: Mass-spring wall model with virtual wall.

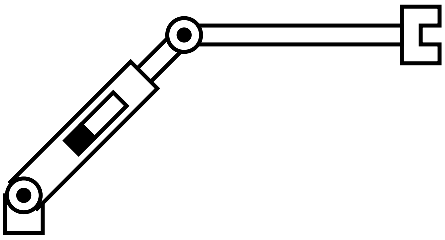

Figure 1.2: Schematic diagram of RPR manipulator

and δx is the wall displacement necessary to produce F_d. One drawback of this method is that it assumes perfect knowledge of the environment. If the wall stiffness varies the applied force will vary. To supplement this, a force/torque sensor can be added to the end effector which provides feedback to the controller allowing the applied force to be modulated. This method is known as Hybrid Force Control and is the technique used in this report.

1.2 Manipulator Model

The manipulator considered in this project is the RPR manipulator. It has a revolute joint at its base and wrist which are coplanar, and a prismatic joint in between, Figure 1.2.

1.2.1 Kinematics

The kinematic model of the manipulator is given by the transformation matrices (1.2), resulting in an end effector transformation of (1.3). The DH parameters for the manipulator are given in Table 1.1.

$$A_1 = \begin{bmatrix} cos\theta_1 & 0 & sin\theta_1 & 0 \\ sin\theta_1 & 0 & -cos\theta_1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.2a}$$

$$A_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.2b}$$

$$A_3 = \begin{bmatrix} cos\theta_3 & -sin\theta_3 & 0 & a_3cos\theta_3 \\ sin\theta_3 & cos\theta_3 & 0 & a_3sin\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.2c}$$

Joint	α_i	a_i	d_i	θ_i
1	0	90°	0	θ_1
2	0	−90°	d_2	0
3	a_3	0	0	θ_3

Table 1.1: DH parameters for RPR manipulator.

$$A_1A_2A_3 = \begin{bmatrix} c_{13} & -s_{13} & 0 & a_3c_{13}+d_2s_1 \\ s_{13} & c_{13} & 0 & a_3s_{13}-d_2c_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.3}$$

Using the standard input tranformation matrix seen in (1.4),the inverse kinematics of the manipulator are given by (1.5). These were used to calculate the desired joint positions, given an end effector reference trajectory.

$$H_{in} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & d_x \\ r_{21} & r_{22} & r_{23} & d_y \\ r_{31} & r_{32} & r_{33} & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.4}$$

$$\theta_1 + \theta_3 = atan2(r_{21}, r_{11}) \tag{1.5a}$$

$$\theta_1 = atan2(d_x - a_3c_{13}, a_3s_{13} - d_y) \tag{1.5b}$$

$$d_2 = \sqrt{(d_x - a_3c_{13})^2 + (d_y - a_3s_{13})^2} \tag{1.5c}$$

$$\theta_3 = atan2(r_{21}, r_{11}) - \theta_1 \tag{1.5d}$$

The desired joint velocities $\dot{q}$ were given using the inverse Jacobian, (1.6). This involved first calculating the Jacobian, (1.7), and numerically calculating the Moore-Penrose pseudoinverse.

$$\dot{q} = J^{\#}v \tag{1.6}$$

Where v is a 6×1 column vector containing the three end effector translational velocities and three angular velocities.

$$J = \begin{bmatrix} -a_3 s_{13} + d_2 c_1 & s_1 & -a_3 s_{13} \\ a_3 c_{13} + d_2 s_1 & -c_1 & a_3 c_{13} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix} \tag{1.7}$$

Following this, the desired joint accelerations $\ddot{q}$ were given by (1.9) and calculated using the aforementioned inverse Jacobian and the element-wise time derivative of the Jacobian (1.8).

$$\dot{J} = \begin{bmatrix} -a_3 c_{13}(\dot{\theta}_1 + \dot{\theta}_3) + \dot{d}_2 c_1 - d_2 s_1 \dot{\theta}_1 & c_1 \dot{\theta}_1 & -a_3 c_{13}(\dot{\theta}_1 + \dot{\theta}_3) \\ -a_3 s_{13}(\dot{\theta}_1 + \dot{\theta}_3) + \dot{d}_2 s_1 + d_2 c_1 \dot{\theta}_1 & s_1 \dot{\theta}_1 & -a_3 s_{13}(\dot{\theta}_1 + \dot{\theta}_3) \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{1.8}$$

$$\ddot{q} = J^{\#}(\dot{v} - \dot{J}\dot{q}) \tag{1.9}$$

1.2.2 Dynamics

Modeling the manipulator as a collection of point masses located at the revolute joints and end effector, the dynamics of the manipulator were given by (1.10).

$$D(q)\ddot{q} + C(q, \dot{q}) + G(q) = U + B(\dot{q}) + J(q)^T F \tag{1.10}$$

Where $D(q)$ is the inertia matrix (1.11), $C(q, \dot{q})$ is the centripetal vector (1.12), $G(q)$ are all the gravity terms (1.13), U is the control input, $B(\dot{q})$ are the dampening terms due to viscous joint friction (1.14), and F is the generalized force vector of forces/torques applied to the end effector (1.15).

$$D(q) = \begin{bmatrix} (m_2 + m_3)d_2^2 + m_3 a_3(1 - 2d_2 s_3) & -m_3 a_3 c_3 & m_3 a_3(1 - d_2 s_3) \\ -m_3 a_3 c_3 & m_2 + m_3 & -m_3 a_3 c_3 \\ m_3 a_3(1 - d_2 s_3) & -m_3 a_3 c_3 & m_3 a_3 \end{bmatrix} \tag{1.11}$$

$$C(q,\dot{q}) = \begin{bmatrix} 2d_2\dot{d}_2\dot{\theta}_1(m_2+m_3) + m_3a_3(\dot{d}_2s_3 - (\dot{d}_2s_3 + d_2c_3\dot{\theta}_3)(2\dot{\theta}_1+\dot{\theta}_3)) \\ m_3a_3s_3(\dot{\theta}_1+\dot{\theta}_3)^2 - (m_2+m_3)d_2\dot{\theta}_1^2 \\ m_3a_3(d_2\dot{\theta}_1\dot{\theta}_3c_3 - 2s_3\dot{d}_2\dot{\theta}_1) \end{bmatrix} \tag{1.12}$$

$$G(q) = g\begin{bmatrix} (m_2+m_3)d_2s_1 + m_3a_3c_{13} \\ -(m_2+m_3)c_1 \\ m_3a_3c_{13} \end{bmatrix} \tag{1.13}$$

Where m_2 and m_3 are the masses at the elbow (joint 3) and the end effector respectively and g is the acceleration due to gravity in $\frac{m}{s^2}$.

$$B(\dot{q}) = \begin{bmatrix} -b_1\dot{\theta}_1 \\ -b_2\dot{d}_2 \\ -b_3\dot{\theta}_3 \end{bmatrix} \tag{1.14}$$

For this simulation, the viscous dampening coefficients were $b_1 = b_2 = b_3 = 0.1\frac{Ns}{m}$.

$$F = \begin{bmatrix} f_x, f_y, f_z, \tau_x, \tau_y, \tau_z \end{bmatrix}^T \tag{1.15}$$

Where f_n represents a force in the n direction and τ_n represents a torque along the n axis. The details about the elements used for this simulation are discussed in Section 1.3.

1.3 Environmental Model

As mentioned in Section 1.1, hybrid force control is being simulated and as such, the environment with which the manipulator is going to interact, i.e. a wall, is modeled as a mass-spring system. In addition to a reactive force in the direction normal to the wall, a term for viscous friction c is added for when the manipulator is 'dragging' against the wall. This results in a force vector applied to the end effector when contacting the wall of (1.16).

$$F_w = \begin{bmatrix} k\delta x, -cv_y, 0, 0, 0, 0 \end{bmatrix}^T \tag{1.16}$$

Where δx in this case is always taken to be negative when the end effector is 'indenting' the wall. According to specification, k can take on any value from $6000\frac{N}{m}$ to $60,000\frac{N}{m}$, so in simulation the wall stiffness k is taken as a random number within this range.

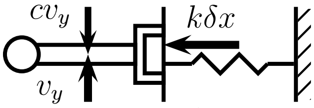

Figure 1.3: Wall modeled as a mass-spring system with viscous friction.

1.4 Feedback Linearization

For position control, a Feedback Linearizing control was employed. Feedback linearization is the concept of developing a control law that yields a linear input-output relationship within a non-linear system. In the context of the manipulator equation this can be seen under the following control law, (1.17).

$$U = D(q)V + C(q,\dot{q}) + G(q) \tag{1.17}$$

Where q and $\dot{q}$ are the joint positions and velocities currently read by the joint encoders, and V is a vector of inputs to the control input U. When applied to the manipulator equation (and ignoring friction and external force inputs for this example) yields (1.18).

$$\begin{aligned} D(q)\ddot{q} + C(q,\dot{q}) + G(q) &= U && \text{(1.18a)}\\ D(q)\ddot{q} + C(q,\dot{q}) + G(q) &= D(q)V + C(q,\dot{q}) + G(q) && \text{(1.18b)}\\ D(q)\ddot{q} &= D(q)V && \text{(1.18c)}\\ D(q)^{-1}D(q)\ddot{q} &= D(q)^{-1}D(q)V && \text{(1.18d)}\\ \ddot{q} &= V && \text{(1.18e)} \end{aligned}$$

The terms $C(q,\dot{q})$ and $G(q)$ are added in an attempt to cancel the torques on the manipulator due to the centripetal and gravity terms. $D(q)$ is a positive definite matrix and hence is always invertible, allowing us to accommodate for the inertia terms. Equation (1.18e) shows a linear relationship between joint accelerations $\ddot{q}$ and the new control input V. In this way we can now design a controller using linear feedback techniques.

A PD controller was designed for (1.18e). As this is a second order system, we can pick a dampening and natural frequency to accommodate our needs. Let us pick a dampening $\zeta = 1$ and a natural frequency of $\omega_n = 50\frac{rad}{sec}$. This yields the gain terms seen in (1.20). In order to make our equations linear in joint error, an additional $\ddot{q}_{ref}$ term must be added to our control law, (1.19), where $e = q - q_{ref}$ and q_{ref} is the vector of joint positions we wish to track.

$$\begin{aligned} \ddot{q} &= K_p e + K_d \dot{e} + \ddot{q}_{ref} && \text{(1.19a)}\\ \ddot{e} - K_d\dot{e} - K_p e &= 0 && \text{(1.19b)} \end{aligned}$$

$$K_p = -\omega_n^2 = -2500 \tag{1.20a}$$

$$K_d = -2\omega_n = -100 \tag{1.20b}$$

The final control law for position control is (1.21).

$$U = D(q)(K_p e + K_d \dot{e} + \ddot{q}_{ref}) + C(q, \dot{q}) + G(q) \tag{1.21}$$

1.5 Hybrid Force Control

The hybrid force controller has two parts, impedance control and force feedback. The impedance controller brings the end effector into the range of where the manipulator would have to be to apply a desired force and the feedback controller adjusts the joints as needed in accordance with measurements from a force/torque sensor.

1.5.1 Impedance Control

As in Section 1.3, the environment is modeled as a mass-spring system, in which case the depth needed for the end effector to produce the desired force f_d, is given by $\delta x = \frac{f_d}{k}$. In the context of the Hybrid Force Controller, the environmental stiffness k need not be known exactly as the feedback controller will make the necessary adjustments to bring the applied force to the desired force.

1.5.2 Force Feedback Control

Once the end effector is in contact with the wall, force/torque measurements of the end effector's interaction with the wall are taken. They effect the manipulator according to (1.10) through the vector F. In this manipulator, forces perpendicular to the wall are read along the end effector's x-axis, and drag from moving along the wall is read as force in the y direction (1.16). This manifests itself as (1.22).

$$F_w = [f_{w_x}, f_{w_y}, 0, 0, 0, 0]^T \tag{1.22}$$

In order to construct a linear controller the transpose of the Jacobian J^T will be applied to the control input of the force controller. This will feedback linearize the force relationship to be $F_w = F_u$, where F_u is the output by the force controller. As f_{w_x} will be negative when contacting the wall, force error takes the following form $e_{f_x} = f_{w_x} + f_{ref}$. While the drag force of the wall has an effect on the manipulator it is not significant, therefore the controller will attempt to directly compensate for it by applying the opposite force. The control law for the perpendicular force is given by (1.23).

$$F_{u_x} = f_{w_x} + K_{w_p} e_{f_x} + K_{w_d} \dot{e}_{f_x} + K_{w_i} \int e_{f_x} \tag{1.23}$$

The f_{w_x} term acts to cancel the reaction force from the wall while the PID controller then attempts to minimize the force error. As force in this case is modeled as proportional wall displacement by the end effector, the derivative of the force error term $\dot{e}_{f_x}$ is proportional the the end effector velocity. This term is then implemented as $\dot{e}_{f_x} = v_x$ as the reference velocity is zero.

Being a second order equation, the gains can be chosen according to (1.24). While setting ζ to one seems like the obvious choice, in practice this controller does not perform well. A better choice is to set $2\zeta\omega_n = \omega_n^2$. This brings the system under the tuning of ω_n and K_{w_d}. The derivative term acts to dampen end effector jitter, while ω_n reduces the systems sensitivity to noise manifested as variability in wall stiffness.

$$K_{w_d} s^2 + K_{w_p} s + K_{w_i} = s^2 + \frac{K_{w_p}}{K_{w_d}} s + \frac{K_{w_i}}{K_{w_d}} \quad = s^2 + 2\zeta\omega_n s + \omega_n^2 \tag{1.24a}$$

$$K_{w_p} = 2\zeta\omega_n K_{w_d} \tag{1.24b}$$

$$K_{w_i} = \omega_n^2 K_{w_d} \tag{1.24c}$$

A natural frequency of $\omega_n = 0.1\frac{rad}{sec}$ and a $K_{w_d} = 50$ were chosen yielding the final gain terms in (1.25).

$$K_{w_d} = 50 \tag{1.25a}$$

$$K_{w_p} = K_{w_i} = \omega_n^2 K_{w_d} = 0.5 \tag{1.25b}$$

Chapter 2

Results

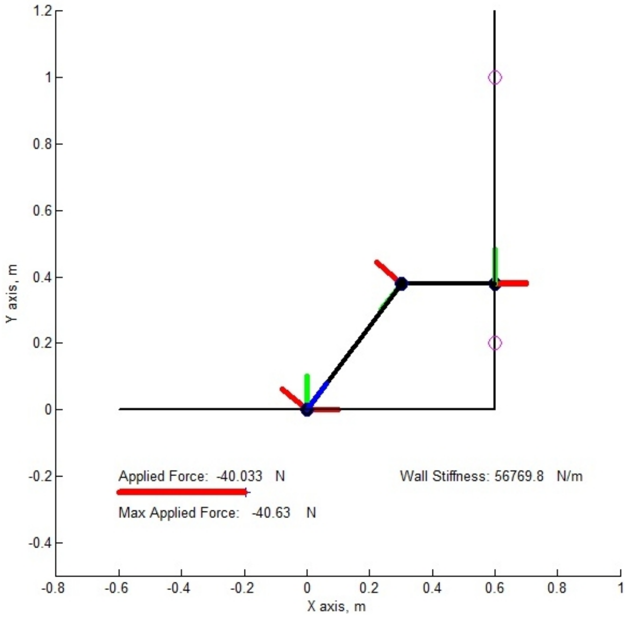

Figure 2.1: MATLAB Simulation of RPR manipulator contacting wall.

The simulation was programmed to randomly choose wall stiffnesses within the allowable range ($6000\frac{N}{m}$ to $60{,}000\frac{N}{m}$) which was then modulated by additive Gaussian noise in the range of 0 to $1000\frac{N}{m}$ for every force/torque sensor sample. The manipulator end effector was programmed to approach the wall in an exponential fashion in order to minimize force overshoot when colliding with the wall, and track between two points on the wall (0.2 m to 1 m). The manipulator used a sinusoid to track between the two points (2.1). The end effector orientation was set to be zero, perpendicular to the wall. These trajectories were differentiated and passed to the kinematics equations, Section 1.2.1, producing joint position, velocity, and acceleration commands which were input to the position and force controllers. The desired force to be applied to the wall was $F_d = 40N$.

$$\delta x = \frac{F_d}{60,000} = 6.6 \times 10^{-4} \tag{2.1a}$$

$$dx = (0.2 + \delta x)(1 - e^{-10t}) + 0.4005 \tag{2.1b}$$

$$dy = 0.4sint + 0.6 \tag{2.1c}$$

The simulation was evaluated every 0.01 seconds for 10 seconds using the RK4 method. A representative wall stiffness of $k = 56,769.7894\frac{N}{m}$ is used for the results of this section.

2.1 Position Error

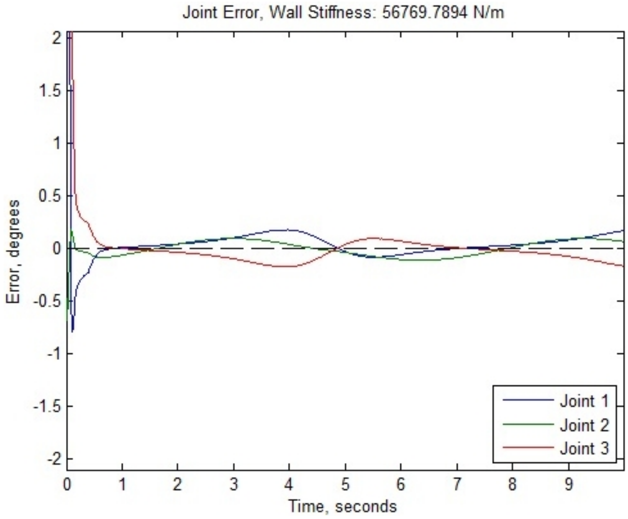

Figure 2.2: Graph of joint error over 10 seconds.

The slowest joint (joint 3) reached 0.1° joint position error in 0.5 seconds and all of the joints stayed within ±0.2° error for the length of the simulation.

2.2 Force Error

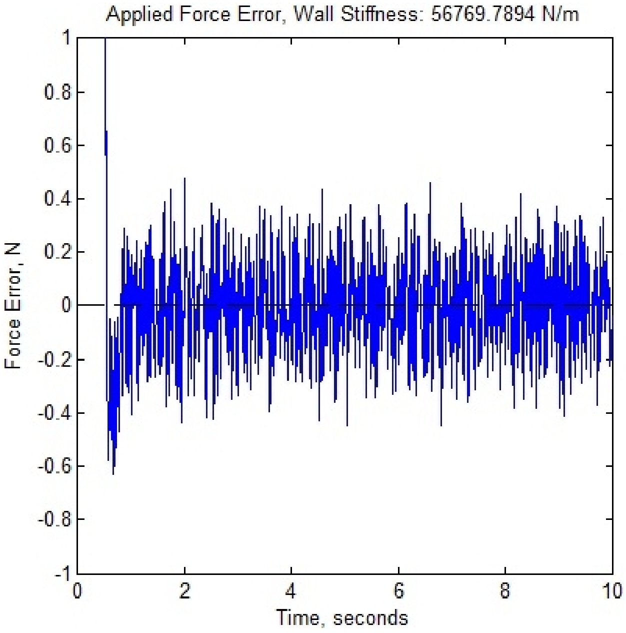

Figure 2.3: Graph of joint error over 10 seconds.

The end effector took 0.1 seconds to apply 36 N of force (90% of F_d) after wall contact. The maximum force applied to the wall was 40.63 N. The force error was $\pm 0.5N$ proceeding the initial overshoot of 0.63 N.

Chapter 3

Conclusion

Hybrid force control allows robust control over force applications in the presence of uncertain environmental conditions. It was generally tolerant of varying environmental stiffnesses and sensor noise. While the presented controller applies adequate force under the given specifications, a more in depth analysis of the stability of the force controller and the environment is necessary in order to improve its performance. Alternatively a Gain-Lead-Lag controller may provide increased performance over the PID controller at the expense of a small increase of controller memory.