January 1992

Multi-Arm Manipulation of Large Objects With Rolling Contacts

Eric D. Paljug
University of Pennsylvania

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-92-19.

First 16 pages missing

Multi-Arm Manipulation of Large Objects With Rolling Contacts

Comments

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-92-19.

First 16 pages missing

This method can be applied to the 1-D system of Equations 4.19 and 4.20. The weighting matrices V and Q are chosen to be diagonal thus decoupling the force and position error terms. The input d is the trajectory of the desired object position and the desired force F_1. This system can be simulated by first solving S and v by disregarding the minus sign on the left hand side of Equations 4.28 and 4.29 and forward integrating them from T to $T + t_0$. This information is then used in reverse. Thus, K and therefore u, which is a function of K and the state x, can be calculated at time t. A simulation of the system now can be run from t_0 through T.

In practical implementation, a suboptimal scheme can be employed. In this case, the steady state value of S is found as $T \to \infty$ off line. Although this value is not optimal, is normally a near optimal. Now, K is a constant and u is a function of the state x and the variable v (Equation 4.26). Expressing v as a function of time yields:

$$v(t) = e^{-(A-BK)^t(t-t_0)}v(t_0) - \int_{t_0}^{t} e^{-(A-BK)^t(t-\tau)}C^tQd(\tau)\,d\tau \tag{4.30}$$

Let $\hat{A} = (A - BK)^t$. A sampled version of v is now derived, where the sample period $T_s = t_{k+1} - t_k$. During this sample period, it is assumed that d is constant.

$$v(t_{k+1}) = e^{-\hat{A}(t_{k+1}-t_k)}v(t_k) - \int_{t_k}^{t_{k+1}} e^{-\hat{A}(t_{k+1}-\tau)}\,d\tau\;\; C^tQd(t_k) \tag{4.31}$$

$$v(t_{k+1}) = e^{-\hat{A}T}v(t_k) - \int_{0}^{T} e^{-\hat{A}\tau}\,d\tau\;\; C^tQ\; d(t_k) \tag{4.32}$$

This equation can be used by a computer at each sample period to update $v(t_k)$, but this requires $v(t_0)$, $e^{-\hat{A}T}$ and $\int_0^T e^{-\hat{A}\tau}\,d\tau$. The initial value of v can be calculated off-line by making use of Equation 4.30 as follows:

$$v(t_0) = e^{-(A-BK)^t(t_0-T)}v(T) - \int_{T}^{t_0} e^{-(A-BK)^t(t_0-\tau)}C^tQd(\tau)\,d\tau \tag{4.33}$$

Estimates of the remaining terms can be calculated as follows [6]:

$$e^{-\hat{A}T} = I - \hat{A}T + \frac{\hat{A}^2T^2}{2!} - \frac{\hat{A}^3T^3}{3!} + \cdots \tag{4.34}$$

$$e^{-\hat{A}T} \approx \sum_{i=0}^{N} \frac{(-1)^i\hat{A}^iT^i}{i!} \tag{4.35}$$

for sufficiently large N. Likewise,

$$\int_0^T e^{-\hat{A}\tau}\,d\tau = \int_0^T (I - \hat{A}T + \frac{\hat{A}^2T^2}{2!} - \frac{\hat{A}^3T^3}{3!} + \cdots)d\tau \tag{4.36}$$

$$\int_0^T e^{-\hat{A}\tau}\,d\tau = IT - \frac{\hat{A}T^2}{2!} + \frac{\hat{A}^2T^3}{3!} - \frac{\hat{A}^3T^4}{4!} + \cdots \tag{4.37}$$

$$\int_0^T e^{-\hat{A}\tau}\,d\tau \approx \sum_{i=0}^{N} \frac{(-1)^i\hat{A}^iT^{i+1}}{(i+1)!} \tag{4.38}$$

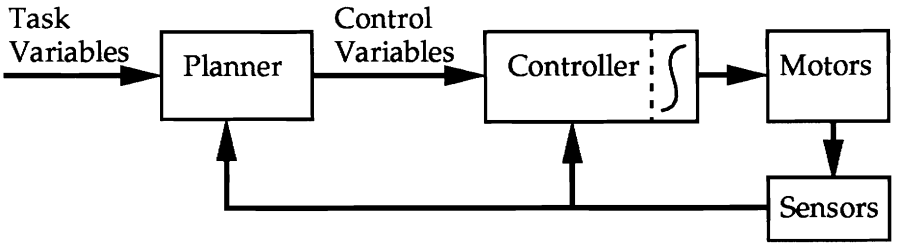

Figure 4.2: Complete Manipulation System.

again, for sufficiently large N. A computer can be programmed to make these calculations efficiently and to decide on-line the size of N. Thus, the above suboptimal controller can be physically implemented.

The complete manipulation system also has another major component, the planner (see Figure 4.2). As noted in the end of the previous section, the planner is an integral part of the system design. The planner has the responsibility to generate inputs that preserve the constraints on the system, namely forces that do not pull and contacts that do not separate. The inputs to the planner are the task variables, such as desired critical contact force and object position, velocity, and acceleration. The outputs of the planner, for the optimal controller developed above, are the desired object position and the desired force to be applied by the first manipulator (Equation 4.18).

4.2 Two dimensional case

The next case to be examined involves extending the task into two dimensions (2-D). The manipulators and the object are restricted to move within the same plane. The goal is to move the object within the plane along a desired trajectory while maintaining the grasp. The trajectory of the object is given in terms of 3 variables, translation in x and y directions and rotation ϕ about the plane normal (z). Analysis of the grasping force requires assumptions about the contact points. It is assumed that each manipulator makes a single point contact with the object. The point contact is characterized by a friction coefficient μ. Further, the contact curves are also constrained: the manipulators have straight line perimeters at and in the neighborhood of the contact point while the object's perimeter is circular. Finally, it is assumed that each robot can exercise all three independent degrees of freedom (DOF) at the contact point.

Notice that the task goal, manipulating the object, does not explicitly specify the be-

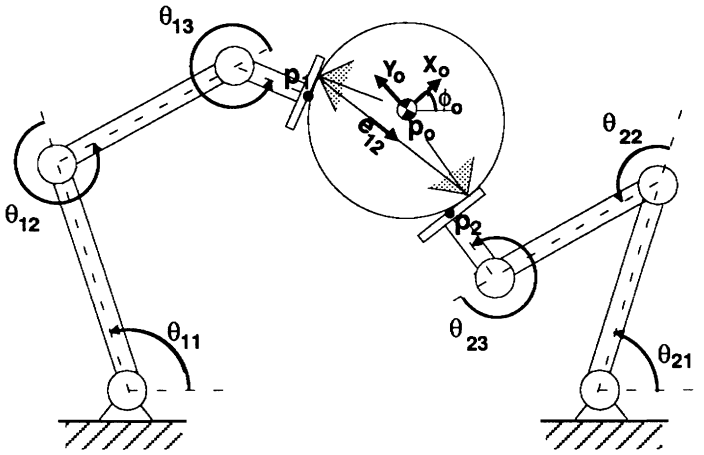

Figure 4.3: Two dimensional example.

havior of the robots at the contact points. In general, each contact point can be subject to separation, sliding and/or rolling. In this development, it is assumed that separation at the contact point implies the inability to manipulate the object and results in task failure. Similarly, sliding is not considered beneficial within this framework. However, rolling at the contacts is included in the analysis, and it is shown how rolling can be implemented to resolve the problems of sliding and separation.

In Figure 4.3, a typical 2 dimensional case is shown where two serial 3 link robots manipulate a circular object. This example we be referred to throughout this section.

4.2.1 System Equations

The object motion is an extension of Equation 4.1, given by:

$$M_o \ddot{p}_o = \sum_{i=1}^{n} \Gamma_i F_i + W \tag{4.39}$$

where $M_o = \text{diag}[m_o \; I_o]$ is the object inertia matrix with the mass term $m_o \in \mathcal{R}^{2\times 2}$ and inertia term $I_o \in \mathcal{R}$, $p_o = [x_o \; y_o \; \phi_o]^t$ is the position vector of the object's center of mass (CM) and $\omega_o = \dot{\phi}_o$, and $W = [w_f^t \; w_m]^t \in \mathcal{R}^3$ is a vector of disturbance forces/torques at the CM (and is comprised of force disturbance $w_f \in \mathcal{R}^2$ and moment disturbance $w_m \in \mathcal{R}$). Note that W can include the force due to gravity. The term $F_i \in \mathcal{R}^2$ is the force at the *ith* contact point. The matrix $\Gamma_i \in \mathcal{R}^{3\times 2}$ relates the contact force to an equivalent force/torque

at the CM and is of the form:

$$\Gamma_i(r_i) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -r_{i_y} & r_{i_x} \end{bmatrix} \tag{4.40}$$

where $r_i = [r_{i_x} \; r_{i_y}]^t$ is the vector from the CM to the *ith* contact point. Implicit in this equation is the fact there are rolling contacts, thus only forces (and no moments) are transmitted at the contact points. All of the above variables are referenced with respect to a fixed global coordinate frame.

The dynamic equation of the 3 DOF revolute joint robot is given by:

$$M_i(q_i)\ddot{q}_i + V_i(q_i, \dot{q}_i) = \tau_i - J_i^t \lambda_i F_i \tag{4.41}$$

where $M_i(q_i) \in \mathcal{R}^{3\times3}$ is the inertial matrix of the *ith* robot, $q_i \in \mathcal{R}^3$ is the joint angle vector (with time derivatives $\dot{q}_i$ and $\ddot{q}_i$), $V_i(q_i\dot{q}_i) \in \mathcal{R}^3$ is a vector that accounts for torques due to the velocity (and/or position) dependent terms of centrifugal, Coriolis, and gravitational force, $\tau_i \in \mathcal{R}^3$ is the vector of torques applied by the joint actuators, and $J_i^t \in \mathcal{R}^{3\times3}$ is the transpose of the robot Jacobian matrix relating the differential motion of a single reference point $p_i = [x_i \; y_i \; \phi_i]^t$ attached to the robot (effector) to the differential motion of the joints. F_i is defined with respect to Equation 4.39 and $\lambda_i \in \mathcal{R}^{3\times2}$ relates the forces/torques at the effector reference point p_i to the contact point p_{c_i} on the robot and has the form:

$$\lambda_i(d_i) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -d_{i_y} & d_{i_x} \end{bmatrix} \tag{4.42}$$

where d_i is the vector (in $\mathcal{R}^2$) from the *ith* robot's effector reference point to its contact point. Note that Equation 4.41 models only the physical dynamics of the arm, the actuator dynamics are not considered.

The velocity of the reference point on the robot effector is related to the velocity of the CM of the object by the relationship:

$$\lambda_i^t \dot{p}_i = \Gamma_i^t \dot{p}_o \tag{4.43}$$

$$\omega_i = \omega_o + \omega_{rel} \tag{4.44}$$

The translational velocities of the object and the robot at the contact point are equivalent as long as the contact does not slip. The rotational velocities of the object and arm at the

contact point are not equivalent in general ($\omega_{rel} \neq 0$), because the contact can roll. The acceleration relationship is of the form:

$$\lambda_i^t \ddot{p}_i + 2\dot{\lambda}_i^t \dot{p}_i - \omega_i^2 d_i = \Gamma_i^t \ddot{p}_o + 2\dot{\Gamma}_i^t \dot{p}_o - \omega_o^2 r_i + \frac{\dot{r}_i \cdot \dot{r}_i}{\mathcal{K}} n_i \tag{4.45}$$

where $\mathcal{K}$ is the radius of curvature of the circular object and n_i is the object surface normal (directed inward).

The relationship between the joint accelerations $\ddot{q}_i$ and the acceleration of the fixed reference point on the effector $\ddot{p}_i$ is given by:

$$\ddot{p}_i = J_i \ddot{q}_i + \dot{J}_i \dot{q}_i \tag{4.46}$$

The above equations can be combined into a complete system. Given that each robot has 3 DOF at its contact point, then the mobility of the 2 planar robots manipulating a planar object with rolling contacts is five. The rolling contacts are instantaneously modeled as a revolute joints and the system is equivalent to an eight bar serial linkage. The system can be specified by the position variable $\rho = [x_o \; y_o \; \phi_o \; \phi_1 \; \phi_2]^t$ where the first three terms represent the object's position (including orientation) and the last 2 terms represent the orientation of the surface normal of each robot at its contact point. (It is assumed that the orientation of the robot reference point p_i and the robot contact point are the same and equivalent to the surface normal of the robot at the contact point. Thus ϕ_i is the third element of the robot effector reference vector $p_i = [x_i \; y_i \; \phi_i]^t$.) The complete system with respect to the state variable $X = [X_1^t \; X_2^t]^t = [\rho^t \; \dot{\rho}^t]^t \in \mathcal{R}^{10}$ would have the form:

$$\dot{X} = \begin{bmatrix} \dot{X}_1 \\ \dot{X}_2 \end{bmatrix} = \begin{bmatrix} X_2 \\ F(X_1, X_2) \end{bmatrix} + \begin{bmatrix} 0 \\ G(X_1) \end{bmatrix} \mathcal{T} \tag{4.47}$$

where $F(\cdot) \in \mathcal{R}^5$ is a function of position and velocity, $G(\cdot) \in \mathcal{R}^{5\times 6}$ is a function of position, and $\mathcal{T} = [\tau_1^t \; \tau_2^t]^t \in \mathcal{R}^6$ is the vector of joint torques for both robots. The system output defined as $Y = [Y_1^t \; Y_2]^t = [\rho^t \;\; (F_1 \cdot e_{12})]^t \in \mathcal{R}^6$ where the force quantity F_1 is projected upon e_{12}, the unit vector of the line joining the contact point of the first robot to the contact point made by the second robot. The importance of this quantity is shown in the critical contact force analysis that follows. The output has the functional form:

$$Y = \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} X_1 \\ a(X_1, X_2) \end{bmatrix} + \begin{bmatrix} 0 \\ b(X_1) \end{bmatrix} \mathcal{T} \tag{4.48}$$

where $a(\cdot) \in \mathcal{R}$ is a function of the system position and velocity while $b(\cdot) \in \mathcal{R}^6$ is a function only of the system position.

The full derivation of the functions $F(X_1, \; X_2)$, $G(X_1)$, $a(X_1, \; X_2)$, and $b(X_1)$ are developed in Appendix A.

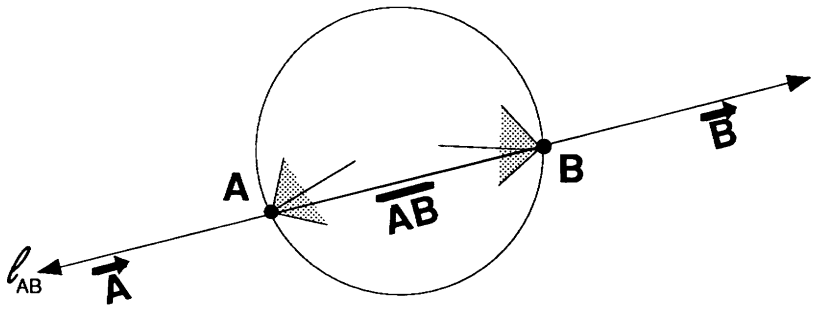

Figure 4.4: Critical contact force.

4.2.2 Critical Contact Force

The critical contact force again describes the force exerted by the robots that does not contribute to the object motion. For example, in the case of two robots as shown in Figure 4.3 where the line segment joining the two contact points is within each contact point's friction cone, the critical contact force is defined as:

$$F_{cc} = \min\{F_1 \cdot e_{12},\ -(F_2 \cdot e_{12})\} \geq 0 \tag{4.49}$$

where $F_i \in \mathcal{R}^2$ is the force applied by the *ith* robot at its contact point, and $e_{12} \in \mathcal{R}^2$ is defined above as the unit vector of the line joining the contact point of the first robot to the contact point made by the second robot.

Note that this definition is valid when both terms are non-negative, that is to say both $F_1 \cdot e_{12} \geq 0$ and $-(F_2 \cdot e_{12}) \geq 0$. This condition is satisfied when realistic coefficient of friction values are considered and the constraint of no contact separation is imposed. This is shown geometrically below with friction cones. A friction cone designates all possible forces applied at a contact point that do not violate the friction constraint ($\|F_{tangential}\| \leq \mu \|F_{normal}\|$). For example, define l_{12} as the line in the plane that includes the two contact points (A and B) on the object. Let $\overline{AB}$ be the line segment of l_{12} joining the two points. Let $\vec{A}$ be the ray with open endpoint A and extending along l_{12} in the direction opposite B and define $\vec{B}$ similarly. The non-negative condition excludes the possibility that the projection of the force applied at the either contact point onto the contact line l_{12} lies on either $\vec{A}$ or $\vec{B}$. If motion, gravity, and disturbance forces are not considered, then the definition of friction cones implies that the line segment $\overline{AB}$ must be within the friction cones of both contacts in order to apply a force along $\overline{AB}$. (See Figure 4.4.) For the common case of the coefficient of friction being less that one, if the line segment $\overline{AB}$ is within the friction cones of both

contact points, then the projection of any force applied within the friction cone onto the line l_{12} must lie on the segment $\overline{AB}$. Thus, the condition that $F_1 \cdot e_{12} \geq 0$ and $-(F_2 \cdot e_{12}) \geq 0$ holds.

As in the 1-D case, the critical contact force itself is not easily incorporated into the controller. However a bound can be found for the error in the critical contact force. This results in the controller managing the system trajectory and the force applied by one robot, since these variables are differentiable, and the non-differentiable function resides in the planner.

From Equation 4.49 we can write:

$$F_{cc} = \frac{F_1 \cdot e_{12} - (F_2 \cdot e_{12}) - |F_1 \cdot e_{12} + (F_2 \cdot e_{12})|}{2} \tag{4.50}$$

Taking the projection of the translational object motion equation (top two rows of Equation 4.39) upon e_{12} yields:

$$(m_o \ddot{x}_t - w_f) \cdot e_{12} = (F_1 + F_2) \cdot e_{12} \tag{4.51}$$

where $x_t = [x_o \; y_o]$. This expression can be combined with Equation 4.50 such that the $F_2 \cdot e_{12}$ term is eliminated. The resulting expression solved for $F_1 \cdot e_{12}$ is given as:

$$F_1 \cdot e_{12} = F_{cc} + \frac{1}{2}|(m_o \ddot{x}_t - w_f) \cdot e_{12}| + \frac{1}{2}(m_o \ddot{x}_t - w_f) \cdot e_{12} \tag{4.52}$$

This expression also serves as a planner for the desired $F_1 \cdot e_{12}$ which is written as:

$$F_1^d \cdot e_{12} = F_{cc}^d + \frac{1}{2}|(m_o^d \ddot{x}_t^d - w_f^d) \cdot e_{12}| + \frac{1}{2}(m_o^d \ddot{x}_t^d - w_f^d) \cdot e_{12} \tag{4.53}$$

The error terms are defined as:

$$e_{cc} = F_{cc}^d - F_{cc} \tag{4.54}$$

$$e_1 = (F_1^d - F_1) \cdot e_{12} \tag{4.55}$$

$$e_x = \ddot{x}_t^d - \ddot{x}_t \tag{4.56}$$

$$e_w = (w_f^d - w_f) \cdot e_{12} \tag{4.57}$$

and the modeled mass as $m_o^d = m_o + \Delta m_o$ where $\Delta m_o \in \mathcal{R}^{2\times 2}$ represents any model discrepancies. The expression for the error in the critical contact force is then written as:

$$\begin{aligned} e_{cc} = \; & F_1^d \cdot e_{12} - \frac{1}{2}|(m_o^d \ddot{x}_t^d - w_f^d) \cdot e_{12}| - \frac{1}{2}(m_o^d \ddot{x}_t^d - w_f^d) \cdot e_{12} \\ & - \left[F_1 \cdot e_{12} - \frac{1}{2}|(m_o \ddot{x}_t - w_f) \cdot e_{12}| - \frac{1}{2}(m_o \ddot{x}_t - w_f) \cdot e_{12}\right] \end{aligned} \tag{4.58}$$

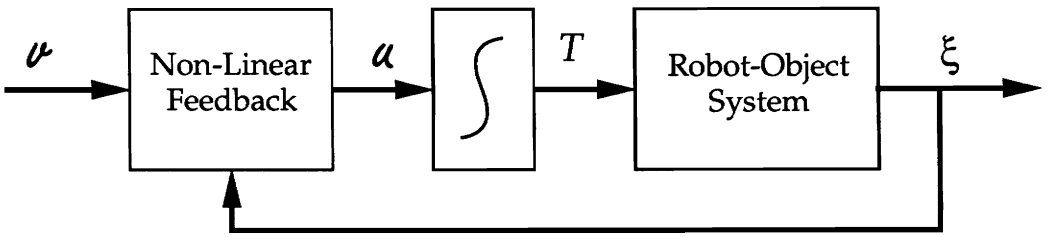

Figure 4.5: Nonlinear feedback preceding input integration.

By expanding m_o^d and gathering common terms the above equation can be written as:

$$\begin{aligned} e_{cc} \quad = \quad & F_1^d \cdot e_{12} - F_1 \cdot e_{12} \; - \; \frac{1}{2} m_o \ddot{x}_t^d \cdot e_{12} + \frac{1}{2} m_o \ddot{x}_t \cdot e_{12} \\ & - \frac{1}{2} w_f^d \cdot e_{12} - \frac{1}{2} w_f \cdot e_{12} - \; \frac{1}{2} | m_o \ddot{x}_t^d \cdot e_{12} - w_f^d \cdot e_{12} + \Delta m_o \ddot{x}_t^d \cdot e_{12} | \\ & + \frac{1}{2} | m_o \ddot{x}_t \cdot e_{12} - w_f \cdot e_{12} | - \frac{1}{2} \Delta m_o \ddot{x}^d \cdot e_{12} \end{aligned} \tag{4.59}$$

By taking the norm of both sides, a bound is found for the error in the critical contact force:

$$\| e_{cc} \| \quad \leq \quad \| e_1 \| + \| m_o \| \, \| e_x \| \, \| e_{12} \| + \| e_w \| + \| \Delta m_o \| \, \| \ddot{x}^d \| \, \| e_{12} \| \tag{4.60}$$

Thus, a controller that has bounds on the object trajectory error, the error in the component of force F_1 along e_{12}, the disturbance error, and the desired object acceleration will have a bounded critical contact force error. The planner, by making use of Equation 4.53, handles the non-differentiable function instead of the control system.

4.2.3 Controller Design

The controller for the 2-D case no longer regulates a linear time-invariant system as in the previous 1-D case. However, the 2-D system equations have the same force characteristic as the 1-D case: in Equation 4.48 the input torques are algebraicly related to the output force. The addition of integrators to the input channels resolves this pathology. This system is both highly non-linear and coupled. A non-linear feedback is found that both linearizes the entire system with respect to a new state and decouples the inputs to control one output. Two types of nonlinear feedback are developed below, with the introduction of feedback before and after the input integrators as the primary difference.

First, we develop the controller shown in Figure 4.5 with the nonlinear feedback applied before the input integrators. The new system is written with respect to the state $\xi =$

$[\xi_1^t\ \xi_2^t\ \xi_3^t]^t = [\rho^t\ \dot{\rho}^t\ \mathcal{T}^t]^t$ and has the form:

$$\dot{\xi} = \begin{bmatrix} \dot{\xi}_1 \\ \dot{\xi}_2 \\ \dot{\xi}_3 \end{bmatrix} = \begin{bmatrix} \xi_2 \\ F(\xi_1,\ \xi_2) + G(\xi_1)\xi_3 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ I_{(6)} \end{bmatrix} u \tag{4.61}$$

where $I_{(6)}$ is the 6×6 identity matrix and the input $u = [\dot{\mathcal{T}}]$. The output equation is then restated as:

$$Y = \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} \xi_1 \\ a(\xi_1,\ \xi_2) + b(\xi_1)\xi_3 \end{bmatrix} \tag{4.62}$$

Techniques from nonlinear control involving differential geometry are employed to find a feedback of the form [10]:

$$u\ =\ \alpha(\xi) + \beta(\xi)v \tag{4.63}$$

where $\alpha(\cdot)$ and $\beta(\cdot)$ are given as:

$$\Phi(\xi)\alpha(\xi) = -\begin{bmatrix} L_f^3 Y_1 \\ L_f Y_2 \end{bmatrix}\ ;\quad \Phi(\xi)\beta(\xi) = I_{(6)} \tag{4.64}$$

and $\Phi(\xi)$ is referred to as the system decoupling matrix. This matrix is given as:

$$\Phi(\xi) = \begin{bmatrix} L_g L_f^2 Y_1 \\ L_g Y_2 \end{bmatrix} \tag{4.65}$$

where the operator $L_g f$ is the Lie derivative of vector space f along vector space g defined as:

$$L_g f\ =\ \frac{\partial g}{\partial x} f\ -\ \frac{\partial f}{\partial x} g \tag{4.66}$$

The application of this feedback linearizes the system with respect to a new state z related to x by the:

$$z = T(\xi) = [z_1^t\ z_2^t\ z_3^t\ z_4]^t = [Y_1^t\ \ L_f Y_1^t\ \ L_f^2 Y_1^t\ \ Y_2]^t \tag{4.67}$$

The requirement for this technique to work is the invertibility of the decoupling matrix.

The final form of the system with nonlinear feedback before the integration is given below:

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \\ \dot{z}_4 \end{bmatrix} = \begin{bmatrix} 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ I & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \tag{4.68}$$

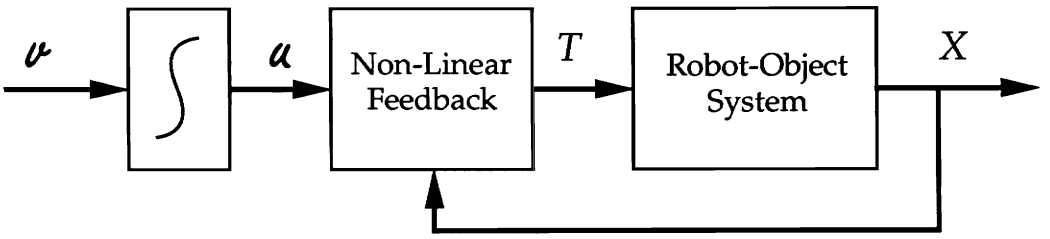

Figure 4.6: Nonlinear feedback following input intergators.

$$\begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & 0 & 0 & I \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} \tag{4.69}$$

An alternative to the approach above is to apply the nonlinear decoupling feedback after the input integrators, as shown in Figure 4.6. In this case, the feedback is applied as the input to the system of Equations 4.47 and 4.48 and has the form:

$$\mathcal{T} = \psi(X_1)^{-1}(u - \eta(X_1, X_2)) \tag{4.70}$$

where u is the new input and ψ and η are given by:

$$\psi(X_1) = \begin{bmatrix} G(X_1) \\ b(X_1) \end{bmatrix} \quad \eta(X_1, X_2) = \begin{bmatrix} F(X_1, X_2) \\ a(X_1, X_2) \end{bmatrix} \tag{4.71}$$

Now the system is linearized and decoupled at the level of the input u since

$$\begin{bmatrix} \ddot{X}_1 \\ F_1 \cdot e_{12} \end{bmatrix} = u \tag{4.72}$$

The system is completed by inserting the integrators preceding u. The system input is $v = \dot{u}$ and now the system can be written with respect to the state $\zeta = [\zeta_1^t \ \zeta_2^t \ \zeta_3^t \ \zeta_4]^t = [\rho^t \ \dot{\rho}^t \ \ddot{\rho}^t \ (F_1 \cdot e_{12})]^t$ as:

$$\dot{\zeta} = \begin{bmatrix} \dot{\zeta}_1 \\ \dot{\zeta}_2 \\ \dot{\zeta}_3 \\ \dot{\zeta}_4 \end{bmatrix} = \begin{bmatrix} \zeta_2 \\ \zeta_3 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ I_{(6)} \end{bmatrix} v \tag{4.73}$$

The output equation is simply:

$$Y = \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} \zeta_1 \\ \zeta_4 \end{bmatrix} \tag{4.74}$$

4.2.4 Planner

The previous section resulted in a system that was linear and decoupled, however the inputs to this system are not *task variables*, i.e. variables that are explicit in the task specification. For instance, the output vector $Y_1 = \rho = [x_o \; y_o \; \phi_o \; \phi_1 \; \phi_2]^t$ includes the orientation of the robots (ϕ_1 and ϕ_2) yet the task goal of manipulating an object does not specify anything about these orientations. The other output variable $Y_2 = F_1 \cdot e_{12}$ may also appear unrelated to the task goal. Thus, a planner is designed to relate *task variables* to *controller variables* in such a way that can satisfy the task goals and possibly improve the system.

We assume that implicit in the goal of successfully manipulating an object are certain the sub-goals:

1. To reduce the possibility of slipping at the contact points within the grasp.
2. To avoid loss of the grasp of the object.
3. To prevent crushing the object.

These goals require additional information to be provided along with the task variables besides the desired object trajectory. We show below that by simply adding to this set the desired critical contact force, these goals can be met.

The relationship of the force vector applied by the robot to the orientation at the contact point directly effect Case 1 above. The force vector is a result of the desired motion of the object and the desired critical contact force. If this force is not within the friction cone of the contact point, slipping will occur. Thus, to prevent slipping, the following equation must hold:

$$| F_i - (F_i \cdot n_i) n_i | \; \leq \; \mu (F_i \cdot n_i) \tag{4.75}$$

where μ is the coefficient of friction (Coulomb's Law), F_i is the force applied by the ith robot ($i = 1, 2$), and n_i is the normal at the ith contact point (pointing into the object). The coefficient of friction is often quite small, on the order of $0.15 - 0.6$ for metal-on-metal contacts [2], thus this constraint is very susceptible to being violated.

A more stable way to apply the force is to closely aligned it with the contact surface normal, which minimizes the left hand side of the inequality of Equation 4.75. Consider two extreme cases of grasping, as shown in Figure 4.7. In the first case, the robot contact perimeters face each other. The critical contact force applied by both robots will be normal to the surface and within the friction cone. However, the force applied by the robots to counteract gravity and produce motion can be in any direction, and the resultant force

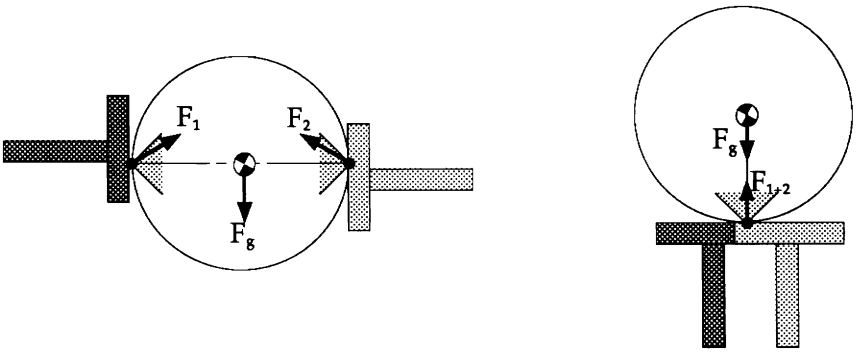

Figure 4.7: Extreme conditions.

may extend beyond the friction cone. A larger force could be applied along the contact line by both robots in order to keep the resultant force within the friction cone with out effecting the object motion. However, this may lead to the crushing of the object (and/or the violation of robot arm torque limits).

In the second extreme case, both robots contact the object at the same point which is directly opposite the required gravity/motion force. Thus the forces applied by the robots are centered within the friction cone, but since the contact points overlap, there is no notion of critical contact force. Indeed, this configuration is quite unstable with respect to directional disturbances.

Between these two extremes resides a grasp solution which considers both the critical contact force and the gravity/motion force and results in a force that is aligned with the surface normal. This can be realized by changing the location of the contact as the desired gravity/motion force changes. Once an object is grasped, the only way to change the contact position without separation or slipping is through rolling. Rolling can be accomplished by controlling the object trajectory and the robot effector orientation. This justifies the inclusion of ϕ_1 and ϕ_2 in the system trajectory.

The complete planner is now specified: its inputs are the desired trajectory of the object (p_o^d, $\dot{p}_o^d$, $\ddot{p}_o^d$, $p_o^{(3)^d}$) and the desired critical contact force. Its outputs are the desired system state trajectory (p^d, $\dot{p}^d$, $\ddot{p}^d$, $p^{(3)^d}$) and the desired component of F_1 along the unit vector e_{12} and its derivative. The system trajectory output includes the object and robot orientation, thus the rolling of the contact is stipulated. The planner calculates these outputs based on the inputs and the system state and characteristics of the object (mass, shape, CM) and the robot effectors (shape, size).

Roll Algorithm

The planner's roll algorithm is developed below for the case of a circular object manipulated by two 3 link serial manipulators within the plane. It is assumed the object mass is known and centered at the center of the object. The object is described as a circle of radius r. The desired trajectory of the object is specified as the object position p_o and its higher order derivatives. Further, it is assumed that the object only translates, its desired trajectory does not include any rotations (see the Section 4.3.4 for the analysis with rotations).

Based on the desired trajectory and the (known) mass of the object, the desired equivalent force vector $F_e = m_o\ddot{x}_t - w_t$ is calculated such that when it is applied to the object's CM, the desired motion object motion is produced. Here W can represent the force of gravity of the object as well as any other known disturbance forces. F_e is restricted to be a continuous, smooth function.

Let the rotational transformation to the Σ coordinate system ${}^{\Sigma}R \in \mathcal{R}^{2\times 2}$ be defined such that ${}^{\Sigma}F_e = {}^{\Sigma}RF_e$ has a zero ${}^{\Sigma}x$ component and a non-negative ${}^{\Sigma}y$ component. In other words, ${}^{\Sigma}F_e = [0,\ |F_e|\,]$ is the vector with the magnitude of F_e but in the positive ${}^{\Sigma}y$ direction. A symmetric solution is then pursued in the Σ coordinate system. Each arm will contribute equally in the positive ${}^{\Sigma}y$ direction to produce ${}^{\Sigma}F_e$ and each arm will contribute in equal magnitudes along the ${}^{\Sigma}x$ axis direction to produce the critical contact force. Thus, the desired forces by each arm in the Σ system are given as:

$$ {}^{\Sigma}F_1^d = \frac{{}^{\Sigma}F_e}{2} + [F_{cc}\ \ 0] = [F_{cc}\ \ \frac{|F_e|}{2}] \tag{4.76} $$

$$ {}^{\Sigma}F_2^d = \frac{{}^{\Sigma}F_e}{2} + [-F_{cc}\ \ 0] = [-F_{cc}\ \ \frac{|F_e|}{2}] \tag{4.77} $$

Now, these forces are transformed back to the original coordinate system to yield the desired forces:

$$ F_1^d = ({}^{\Sigma}R)^{-1}\ {}^{\Sigma}F_1^d \tag{4.78} $$

$$ F_2^d = ({}^{\Sigma}R)^{-1}\ {}^{\Sigma}F_2^d \tag{4.79} $$

The final step of the planner is to determine the orientation of the the robots and the projection of F_1 along e_{12}. The desired orientation of the ith arm is given by:

$$ \phi_i = \arctan2(F_{i_y}^d, F_{i_x}^d) \tag{4.80} $$

Thus, the desired force F_i^d coincides with the desired contact normal and is centered within the friction cone. The direction e_{12} is the first column of the inverse rotational transformation and the calculation $F_1 \cdot e_{12}$ can be performed. Higher order derivatives of ϕ_i and $F_1 \cdot e_{12}$ are solvable as well and involve the higher order derivatives of the object motion.

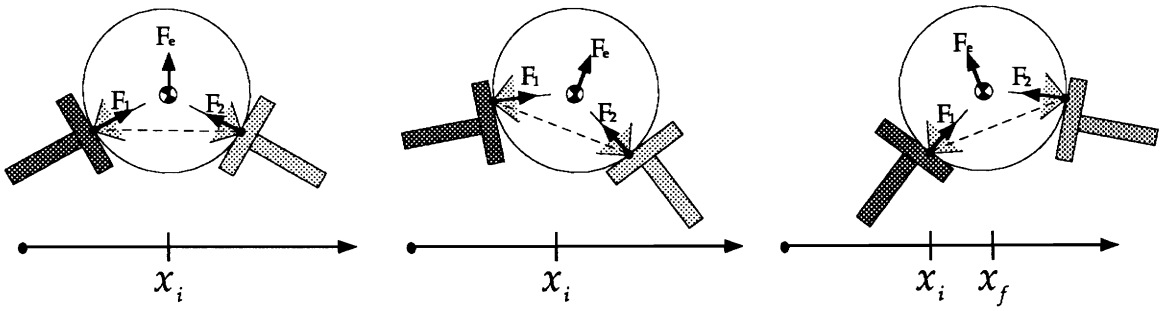

Figure 4.8: Rolling.

This rolling behavior is illustrated by a series of pictures in Figure 4.8. In these frames, a circular object is moved from an initial point at rest $\mathcal{X}_i$ to a final point at rest $\mathcal{X}_f$, and the orientation of the object does not change at any time. Notice that the first frame the robots is merely counteracting the force of gravity and applying the critical contact force. The second frame the robots are collectively moving the object to the right. Note that they have rolled to new contact points so that the force from each robot is along the contact normal. In the final frame, the force of the robots is de-accelerating the object. Again, they have rolled to new contact points so that the force is along the surface normal.

To achieve the goal of not crushing the object (Case 3), it is assumed that the given desired critical contact force is well below the object's ultimate tensile strength. Referencing the critical contact force analysis, we have show that the error in the critical contact force is bounded by the error in trajectory, a force component, and the modeling error.

The prevention of separation (Case 2) is guaranteed by the above algorithm since the force applied by the object is closely aligned with the surface normal and the force is non-negative.

In summary, the planner transforms task variables (p_o, $\dot{p}_o$, $\ddot{p}_o$, $p_o^{(3)}$, F_{cc}) into controller variables (p, $\dot{p}$, $\ddot{p}$, $p_o^{(3)}$, $F_1 \cdot e_{12}$) while maintaining a grasp of the object that neither separates nor excessively compresses the object. Further, the planner calculates the trajectory of the roll at the contact points to complete the task successfully.

4.3 Extensions to the 2-D Case

This section will discuss extensions of the 2-D case by relaxing some of the assumptions of the previous sections. In particular, the following scenarios are developed: contact by more than two robots, less restrictive object shapes and mass distributions, and effectors with curved (not straight line) contact perimeters. Additionally, the treatment of moments

applied to the object are discussed. These topics are investigated to find modifications to the previous analysis. The complete solution to the new problem is not stated, but instead methods are presented that extend the previous analysis to resolve the new problem.

4.3.1 Multi-robot contacts

Previously, the 2-D case involved two robots manipulating an object. Additional robots can positively impact the task execution in two ways: the re-distribution of heavy loads and improved disturbance rejection. The manipulation of heavy objects may exceed the force limit of the individual robot arms involved, however additional robot arms, strategically placed, can reduce the burden of any single arm.

The 2-D case planner showed how two arm manipulation can keep the force applied by each arm centered within the friction cone. This algorithm required that any disturbance forces applied to the object are known *and* vary smoothly. In practice, they must vary with small changes due to the small bandwidth of the rolling of the physical system. Any large change in the disturbance force requires the system to respond with a large amount of rolling. By introducing a third arm such that the three contact points made by the robots form the inscribed equilateral triangle of the circular object, then any pair of robots can become the *active* pair in countering a force disturbance without a large amount of roll.

An additional robot effects the formulation of the system equations. In the planar case, where each robot has 3 DOF, then every additional robot adds 3 new inputs to the system– the three motor torques. Given the initial contact point of the added robot and the constraint that it can only roll at the contact point, then the position of all three robot joints is determined by its current orientation. Thus each additional robot increases the system's mobility by one. If the system is to have the same number of inputs as outputs (which enables it to be input-output decoupled), then for each new robot there are two new outputs to be determined.

For example, if three robots contact a circular object, then the system has a mobility of six. The six generalized coordinates can be chosen to be the object location ($p_o = [x_o\ y_o\ \phi_o]^t$) and the robot orientations (ϕ_1, ϕ_2, and ϕ_3). The system has nine inputs corresponding to the three motors of each robot. If the system is restricted to nine outputs, then a natural choice for six of them is the set of generalized coordinates. The other three can be defined in relation to the critical contact forces. For example, Figure 4.9 depicts three critical contact forces between each pair of contacts where the critical contact point is defined as:

$$F_{cc_{ij}} = \min\{F_{i_{eij}},\ -F_{j_{eij}}\} \qquad i \in \{1,2,3\},\ j \in \{1,2,3\},\ i \neq j$$

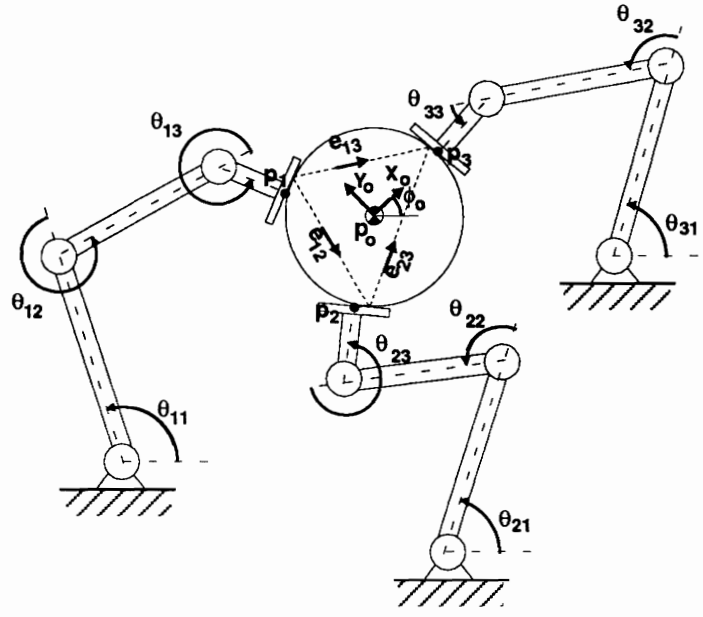

Figure 4.9: Three arms manipulating an object.

$$F_{cc_{ij}} = \frac{F_{i_{eij}} - F_{j_{eij}} - |F_{i_{eij}} + F_{j_{eij}}|}{2} \tag{4.81}$$

In this notation, the forces applied by the kth robot is written as a combination of two components

$$\begin{aligned} F_k &= \alpha e_{kj} + \beta e_{kl} \\ &= F_{k_{ekj}} + F_{k_{ekl}} \end{aligned} \tag{4.82}$$

where α, β are scalars. Here e_{kj} (e_{kl}) is the unit vector joining the kth (kth) contact point to the jth (lth). Thus, the $F_{k_{ekj}}$ term from the above equation is equal to αe_{kj}.

As in the previous cases, formulating a controller with the critical contact force is difficult due to the non-differentiable function. So, instead we design a controller which regulates the system position and three force components. We show below that errors in the controller variables bound the errors in the critical contact force. Consider the motion equation of the object:

$$M_o \ddot{p}_o = \tilde{\Gamma}_1 \begin{bmatrix} F_{1_{e12}} \\ F_{1_{e13}} \end{bmatrix} + \tilde{\Gamma}_2 \begin{bmatrix} F_{2_{e23}} \\ F_{2_{e21}} \end{bmatrix} + \tilde{\Gamma}_3 \begin{bmatrix} F_{3_{e31}} \\ F_{3_{e32}} \end{bmatrix} \tag{4.83}$$

where $F_{i_{ejk}}$ is expressed in terms of its components along the lines joining the contact points, and M_o, p_o, w_o, and I_o are defined in Equation 4.39. The 3×2 matrix $\tilde{\Gamma}_i$ converts

the forces to the cartesian frame *and* and translates the force applied at the contact point to an equivalent force/moment at the object CM. For example, $\tilde{\Gamma}_1$ is given by:

$$\tilde{\Gamma}_1 = \begin{bmatrix} e_{12_x} & e_{13_x} \\ e_{12_y} & e_{13_y} \\ (e_{12_x} r_{1_y} - e_{12_y} r_{1_x}) & (e_{13_x} r_{1_y} - e_{13_y} r_{1_x}) \end{bmatrix} \tag{4.84}$$

where $r_1 = [r_{1_x}\ r_{1_y}]$ is the vector from the object CM to the contact point and e_{12_x} is the x component of the unit vector of the line joining the first contact point to the second. We can choose three forces from Equation 4.83 to be output from the controller such as $F_o = [F_{1_{e12}}\ F_{2_{e23}}\ F_{3_{e31}}]^t$. In that case, the three remaining forces can be expressed as a function of these three forces and the object acceleration.

$$\begin{bmatrix} F_{1_{e13}} \\ F_{2_{e21}} \\ F_{3_{e32}} \end{bmatrix} = \mathcal{B}^{-1} M_o \ddot{p}_o - \mathcal{B}^{-1} \mathcal{A} F_o \tag{4.85}$$

$$\mathcal{B} = \begin{bmatrix} \tilde{\Gamma}_{1_{12}} & \tilde{\Gamma}_{2_{12}} & \tilde{\Gamma}_{3_{12}} \\ \tilde{\Gamma}_{1_{22}} & \tilde{\Gamma}_{2_{22}} & \tilde{\Gamma}_{3_{22}} \\ \tilde{\Gamma}_{1_{32}} & \tilde{\Gamma}_{2_{32}} & \tilde{\Gamma}_{3_{32}} \end{bmatrix} \quad \mathcal{A} = \begin{bmatrix} \tilde{\Gamma}_{1_{11}} & \tilde{\Gamma}_{2_{11}} & \tilde{\Gamma}_{3_{11}} \\ \tilde{\Gamma}_{1_{21}} & \tilde{\Gamma}_{2_{21}} & \tilde{\Gamma}_{3_{21}} \\ \tilde{\Gamma}_{1_{31}} & \tilde{\Gamma}_{2_{31}} & \tilde{\Gamma}_{3_{31}} \end{bmatrix}$$

Now, all three critical contact forces are given by combining Equation 4.81 and 4.85:

$$2F_{cc} = 2 \begin{bmatrix} F_{cc_{12}} \\ F_{cc_{23}} \\ F_{cc_{31}} \end{bmatrix} = F_o - \mathcal{B}^{-1} \left(M_o \ddot{p}_o - \mathcal{A} F_o \right) - \left| F_o + \mathcal{B}^{-1} \left(M_o \ddot{p}_o - \mathcal{A} F_o \right) \right| \tag{4.86}$$

Likewise, the desired critical contact force is described by:

$$2F_{cc}^d = F_o^d - \mathcal{B}^{-1} \left(M_o \ddot{p}_o^d - \mathcal{A} F_o^d \right) - \left| F_o^d + \mathcal{B}^{-1} \left(M_o \ddot{p}_o^d - \mathcal{A} F_o^d \right) \right| \tag{4.87}$$

By defining the error terms as follows:

$$e_{cc} = F_{cc}^d - F_{cc}; \qquad e_f = F_o^d - F_o; \qquad e_p = \ddot{p}_o^d - \ddot{p}_o \tag{4.88}$$

then a bound for the error in the critical contact force is expressed as:

$$\|e_{cc}\| \leq \|e_f\| + \|\mathcal{B}^{-1}\| \|\mathcal{A}\| \|e_f\| + \|\mathcal{B}^{-1}\| \|M_o\| \|e_p\| \tag{4.89}$$

The desired values for the output forces can be calculated by solving the desired critical contact force expression (Equation 4.87) for the output forces, although this is not trivial

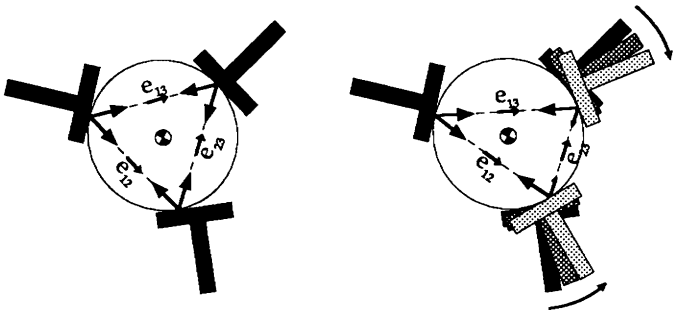

Figure 4.10: Three arm rolling.

due to the absolute value function. This calculation is performed in the control system planner.

Rolling can still be used to center the applied force at each contact force within the friction cone. For instance, neglecting gravity and other disturbances, it is easy to see that for a circular object with no desired motion and equal desired critical contact forces then the system will be rolled such that the lines joining the contact points form an equilateral triangle. (See Figure 4.10.) If instead $F_{cc_{12}} = F_{cc_{23}} > F_{cc_{31}}$ then the system would roll to the acute triangle configuration.

4.3.2 Arbitrary objects

Most of the previous detailed analysis assumed that the object was circular and of uniform density (and therefore the CM was located at the circle's center). Here we discuss the relaxing of these two assumptions.

The circular assumption is relaxed to the assumption that the object is a simple closed convex figure. For example, ellipses and quadrilaterals qualify as acceptable objects. In these cases, the curvature of the object perimeter will dictate what part rolling has in task implementation. Since the robot effectors are straight lines, straight object perimeters do not allow rolling at all. Near straight perimeters are very sensitive to rolling. Rolling is not highly desired in this case because this motion may require very long robot effectors. If rolling is not possible or desirable, then given the desired object trajectory and the critical contact forces, the stability of the grasp can not be improved once the object is grasped.

On the other hand, when objects have a *useful* curvature, the contact points can be rolled to grasp configurations where the applied forces are less likely to cause slipping or separation. The algorithm for rolling with non-circular objects is more complex because it must consider the effects of the induced moments. This was not the case with circular

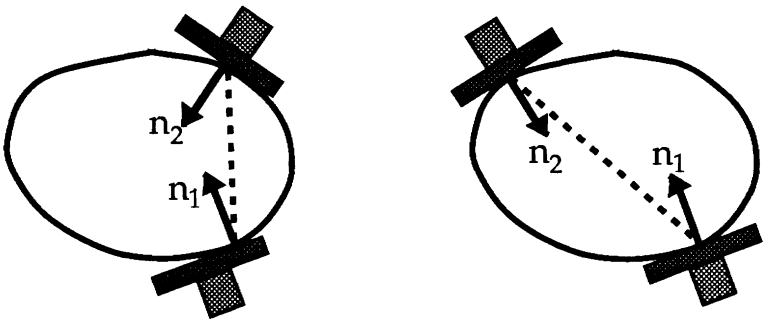

Figure 4.11: Arbitrary shaped object manipulated by two manipulators.

objects because the desired force was always applied from the contact point to the object CM, thus producing no moment. In the general case, moving the contact point will change the moment that the force applied at the contact point produces at the object CM.

Additionally, desired tasks may be impossible due to the object shape and effector lengths. The planner must determine if the exact solution (where the applied forces are centered within the friction cone) exists, and in the case that it does not, it must check if any solution exists that would prevent slipping and separation.

The relaxation of the uniform density requirement effects the object inertial matrix and its CM. Whether the object CM is centered or not is less important than whether the location of the CM is known. First, consider the static case where no motion is desired (and there is no gravity or disturbance forces). Here, the rolling is such that the applied forces attempt to be balanced along the lines joining the contact points while the applied force at each contact point is normal to the surface. Once this configuration is resolved, then the motion generating and disturbance countering force (F_e) is considered. This technique is reasonable in practice if one assumes that F_e is much smaller than the critical contact forces. In other words, the grasp is firm and the motion is slow. For two contacts, the it can be seen that if the both surface normals are on the same side of the line joining the contact points, then each contact should be rolled to reduce the angle between the between the normal that contact line (see Figure 4.11). If the surface normals are on opposite sides of the line joining the contact points, then the contacts can both be rolled in the direction of positive rotation or both rolled in the direction of negative rotation.

The general problem of arbitrary objects and multiple manipulators becomes very complex. Restrictions due to the the surface curvatures and the effector size limit the amount of rolling possible. None the less, rolling at a particular contact does permit an adjust-

ment between the angle of applied force and the surface normal. A useful general strategy involves locating the robots initial contact in an area which permits effective rolling. For example, given a superquadric model of the object, then the sections of the object where the curvature is conducive toward rolling can be located. This information is then used as one of the decision variables in the grasping planner.

4.3.3 Curved effectors

The robot effector has been characterized as a straight line in the preceding analysis. Here we discuss the impact of curved effectors, restricting the curve to be convex and smooth. In the case of the circular object, the relationship of the acceleration of the effector reference point to the object CM (Equation 4.45) must be changed to account for the curved effector. It now becomes:

$$\lambda_i^t \ddot{p}_i + 2\dot{\lambda}_i^t \dot{p}_i - \omega_i^2 d_i - \frac{\dot{d}_i \cdot \dot{d}_i}{\mathcal{K}_i} n_i = \Gamma_i^t \ddot{p}_o + 2\dot{\Gamma}_i^t \dot{p}_o - \omega_o^2 r_i + \frac{\dot{r}_i \cdot \dot{r}_i}{\mathcal{K}_o} n_i \tag{4.90}$$

where $\mathcal{K}_o$ and $\mathcal{K}_i$ are the radii of curvature of the object and the effector at the *ith* contact point, respectively, and n_i is the object surface normal (directed inward).

Additionally, the planner becomes more complex. Now, the function of the perimeters of both the object and the effector are involved in the calculation of the effector's desired orientation.

4.3.4 The treatment of moments

The desire to rotate the object or to counter-act disturbance rotations of the object requires producing a moment about its CM. Producing a moment for the circular object case destroys the symmetry of the problem. At each contact point, a tangential force is applied to produce the moment.

$$M_z = r(F_{1_t} + F_{2_t}) \tag{4.91}$$

Here, F_{i_t} is tangent to the object perimeter at the *ith* contact point and is positive when its cross product with the surface normal (pointing from the contact point to the center of the circle) is in the positive z direction out of the plane. The planner of 2-D case resulted in applied forces that were centered within the friction cone. This is used as the starting point for the inclusion of moments. For the circular object with two contacts, the moment can be produced by having both contacts making equal contributions:

$$F_{i_t} = \frac{M_z}{2r} \tag{4.92}$$

The friction cone constraint must now be verified to avoid slipping. This is simply a comparison of the tangential force from Equation 4.92 and the magnitude of the applied force from the planner F_i, since the applied force is along the contact norm.

$$\mu \geq \frac{\mid F_{i_t} \mid}{\parallel F_i \parallel} \tag{4.93}$$

If this constraint is violated, then the task will fail. Otherwise, the force applied at the contact point is given by combining the tangential and normal components. Since the applied force starts in the center of the friction cone, this algorithm will result with an applied force that is closest to the center of the friction cone for both contacts.

4.4 Simulation and Experimentation

This section presents the proposed simulation and experimentation of the two robot example. They are used to verify the theory and to learn more about the system and the problems involved in its practical implementation.

In both cases, the simulation and experimentation will focus on the manipulation of a circular object by two robot arms, demonstrating the performance of the planner algorithm and the controllers described in Section 4.2. In particular, the performance in controlling the position of the system and the critical contact force will be investigated. Two tasks we be implemented. The first task will involve moving the circular object back and forth. In this case the contacts will roll slightly, depending on the speed of motion, the weight of the object and the desired critical contact force. The second task will involve countering disturbance forces. The robots will roll the contact points in order to keep the applied forces due to disturbance response centered within the contact friction cone.

4.4.1 Simulation

The configuration of Figure 4.3 is investigated where two 3 DOF serial, planar robot arms manipulate a circular object whose CM is located at its center. The formulation of the simulation incorporates many of the equations of Section 4.2. These equations are reformed into a single (large) differential equation system. (See Appendix A.) The non-linear feedback is calculated for both cases presented in Section 4.2.3, feedback before the input integrators and feedback after the input integrators. The first implementation results in the differentiation of the entire system. This process is a very complex endeavor, and requires the use of the symbolic manipulating program *Mathematica* [30]. This package not only assists in symbolic differentiation, but can also be programmed to formulate the result in usable

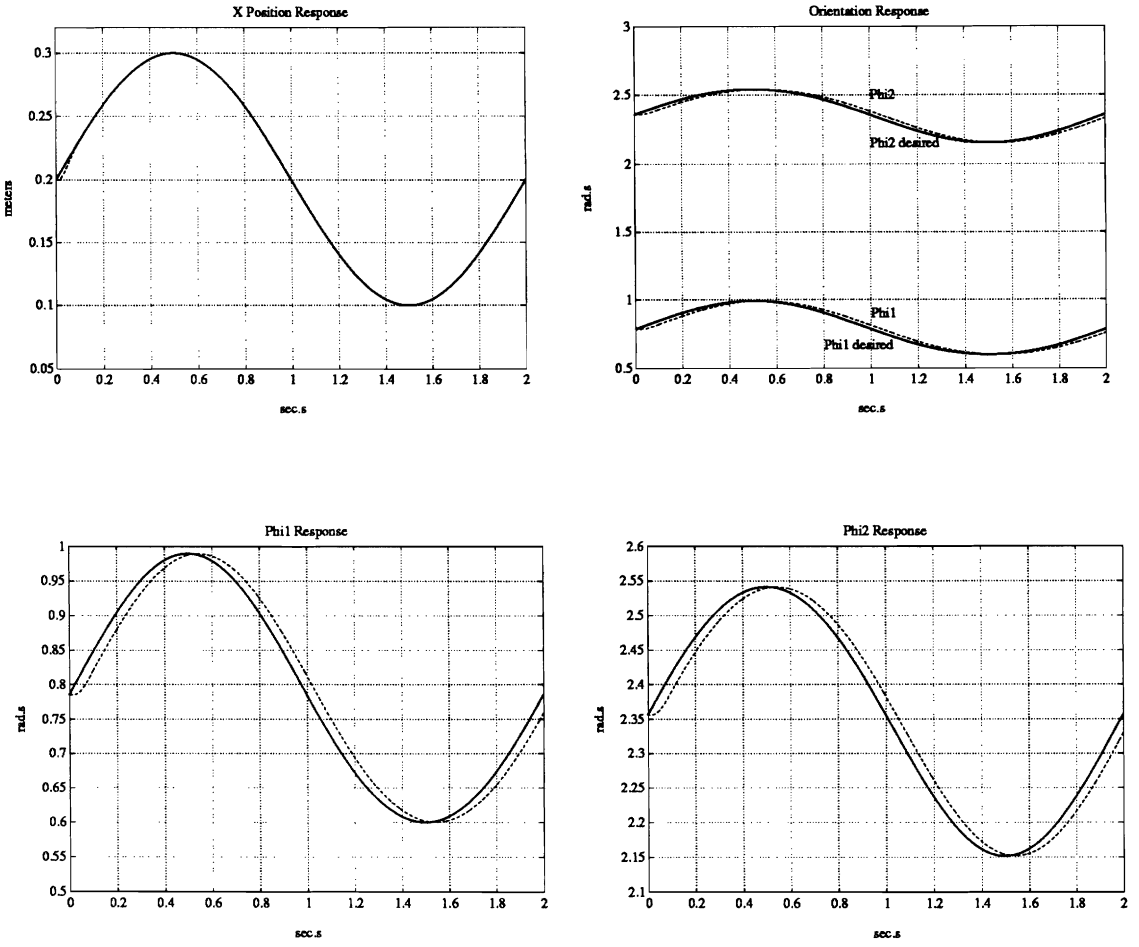

Figure 4.12: Simulation of two effector manipulation with rolling contacts.

computer code. The alternate method of applying the feedback after the integrators does not require differentiating the entire system, so it is computationally less complex, however since the feedback is after the integrators, it may prove more sensitive. Both systems use a forth order Runge Kutta integration routine and the entire system is programmed in the C language on Sun Microsystem Sparc II's.

In Figure 4.12, the preliminary results of the simulation using the feedback before the integration are shown.

Further analysis is capable with the simulation. Below is listed the issues that are to be investigated with both methods of feedback. The results of this investigation will be qualitative in nature in order to determine an understanding of the effect of each issue.

- **Discretizing the controller in time.** Since the actual experimental system is

controlled by a digital computer, the effects of discretizing the control law are investigated. It is thought that this should only present a small disturbance to the system, depending on the simulated sampling frequency. Additionally, the experimental system in use at the lab has a measurement delay of one sample period and these effects are also simulated.

- **Mis-matched coefficients.** The theory assumes exact knowledge about a number of parameters, including object mass and robot inertia, mass, and link lengths. By incorporating small errors between the control law values and the system values, the sensitivity of these parameters can be qualified.

- **Simplified control matrices.** In order to implement the theory in real-time, the controller may need to be considerably simplified. Through simulation, one can verify the important elements of key matrices and the elements that have a negligible effect.

4.4.2 Experimentation

The experimentation will continue the study of the issues discussed in the simulation analysis. Experimental results are to be compared to the simulation results and any differences explored. Additionally, the experimentation introduces new aspects. For example, the motor dynamics have not been modeled in the preceding development, under the assumption of the availability of (near) ideal torque sources. The friction of the gear train is a trade-off to large amounts of backlash, and neither effects are in the theoretical development. Only position feedback is available with the setup. Velocity is obtained by numerically differentiation when it is absolutely required, although this is not a desirable method for obtaining this quantity. Joint acceleration is unknown. The effects of these and other implementation specific problems will be investigated.

The experimental setup is named the Two Robotic Arm Coordination System (TRACS) [37, 23, 17, 22]. (See Figure 4.13). It is designed and built to investigate the issues presented in this proposal. It is comprised of two Puma 250 robots, each with 6 DOF. The end effector of the robots are flat plates to represent any flat surface on the robot. One of the robots contains a 6 DOF force/torque sensor between the plate and wrist. Additionally, both plates are equipped with Interlink contact/force sensors which provide feedback on contact location and normal contact force. These robots are configured such that each models a serial 3 DOF robot and their effectors operate in the same plane.

The system is controlled by a PC-AT based computer that contains an AMD 29000 based high speed floating point coprocessor board. This coprocessor board and the PC-AT

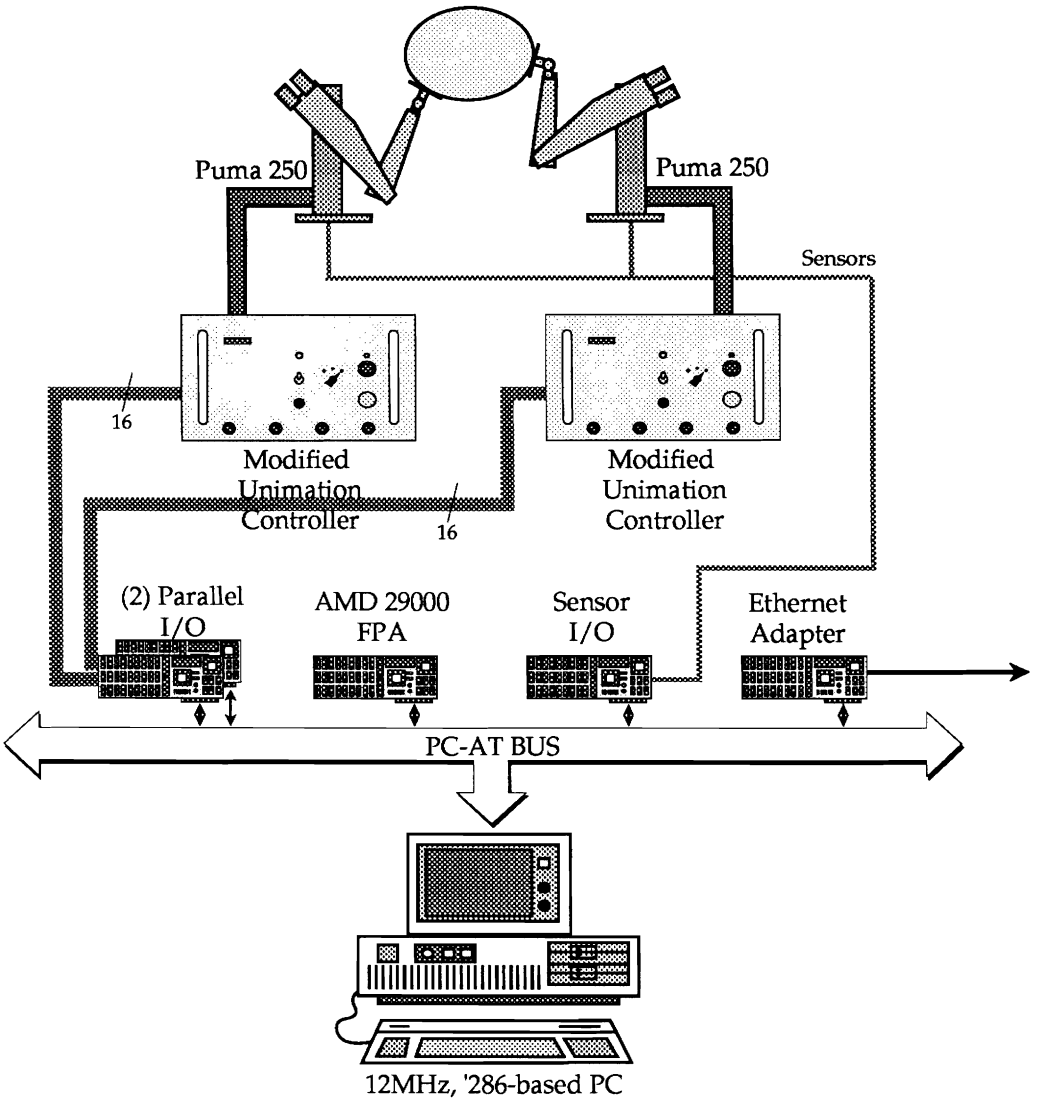

Figure 4.13: TRACS hardware architecture.

80286 processor combine to yield sample rates for control of all twelve joints in excess of 200 Hz. The system is organized such that the processor of the PC performs robot sensor and actuator I/O while the coprocessor board calculates the control. This coprocessor board is programmed in the C computer language, thus much of the simulation code can be used with the experimental setup.

Chapter 5

Summary

The proposed research is centered about a relatively new problem within the area of multi-arm robot systems: *whole arm manipulation* (WAM) [24]. This involves the manipulation of objects that are large relative to the size of the robot. In general, this task requires more than one manipulator and each manipulator is permitted the use of any link surface. Many robots arms can not accommodate manipulation on their link surfaces due cable routing and actuator placement. New manipulators are being design for these types of tasks [1]. These manipulators will require new control algorithms. This proposal investigates the control of robot arms for whole arm manipulation.

The following two sections summarize the current status of the research effort and details the work yet to be completed. The last section then lists the contribution of this proposed work.

5.1 Current status

The following work of the proposed research is completed:

- **Modeling.** The formulation of the differential equations governing the 2-D example of two arms manipulating a circular object is complete and included in Appendix A.
- **Design.** The controller employing nonlinear feedback before the input integrators has been solved for the same example with the use of Mathematica. The equations of the planner which calculates the desired roll and enforces the unilateral constraints have also been derived.
- **Simulation.** Simulation of the system and the controller with feedback preceding the input integrators is near completion. This simulation demonstrates the decoupled

position and force control.

- **Experimentation.** The experimental system is functional. It consists of two Puma 250's and a PC-AT based controller. It is currently capable of implementing simple control schemes with both arms.

5.2 Proposed work plan

The following work is involved in completing the proposed research:

- **Modeling.** The extensions of the 2-D example will be formalized and examples will be provided for illustration.
- **Design.** The controller implementing nonlinear feedback after the input integrators will be developed.
- **Simulation.** The simulation of the system with nonlinear feedback preceding the input integrators will be completely debug. The simulations for the case of feedback following the integrators must be developed. Finally, simulation data will be gathered for the tasks of moving a circular object and rejecting a force disturbance.
- **Experimentation.** The experimental setup will be overhauled and undergo various calibration procedures in order to obtain the best performance possible. A 6 DOF force/torque sensor will be incorporated into the system. Then the proposed tasks will be performed for both types of controller feedback and data will be gathered from these experiments.
- **Analysis.** The simulation and experimental data will be analyzed, compared, and contrasted. The control techniques will be evaluated and practical implementation issues will be addressed. Lastly, the information will be written up and documented in the final thesis.

5.3 Contribution

The contribution of this research is divided into two distinct categories, *scientific* and *engineering*. The scientific contributions are the basic fundamental results of this work. The engineering contributions refer to the significance of the work in regard to practical issues, including physical implementation.

- **Scientific Research Issues:**
  - **Controlled Dynamic Rolling.** In general, the contact between the object and link surface is not a fixed grasp. Rolling, sliding and separation can occur at these points. This proposal considers rolling at the contact point. This rolling is not only part of the kinematic model of the system, but it is also actively controlled. The controller goal is the prevention of sliding and separation, thus improving the stability of the grasp. This is accomplished by attempting to keep the applied force vector centered within the friction cone. The trajectory of the roll is based on the object dynamics and the critical contact force, while the controller also considers the manipulator dynamics.
  - **Controller Design with Unilateral Constraints.** The general approach to the control design involves formulating the system as a set on nonlinear differentiable equations. Unilateral constraints, such as the restriction that each manipulator must push at the contact point point and cannot affect a pull, introduce non-differential functions into the problem. Our approach is to design the controller for the set of differential equations that represent the system. The unilateral constraints are delegated to the system planner. The error in the planner-controller system is shown to be bounded by the error term (and other terms) of the controlled system.
  - **Critical Contact Force.** The internal force of the grasp in this proposal is represented by the notion of a critical contact force. Unlike internal force representations that are defined by the null space of a grasping matrix, the critical contact force is easily visualized by the geometry of the contact points. Additionally, the critical contact force enforces the unilateral constraint of pushing at the contact point. The subsequent analysis and control is built upon this construct.
- **Engineering Research Issues:**
  - **Simulation.** Developing complex simulations of cooperating arms will yield insight to the important features of the controller, such as its principal coefficients and its parameter sensitivity. The performance obtained from the application of nonlinear compensation before and after integration will be qualified.
  - **TRACS.** A new test bed is developed to experimentally investigate the coordinated control of two robot arms. The system is inexpensive and fast enough for dynamic experimentation.

- **Experimentation.** There has been little experimental data published regarding the coordination of robots arms, especially in the case of WAM with rolling contacts. Through experimentation, the partial benefits and hardships will be determined.
- **Practical Implementation of Force Control.** Many publications that do propose force controllers seldom provide simulation and experimental data from complex robotic systems for evaluation. The result of this proposal is a controller that regulates both force and position simultaneously through decoupled subsystems. The force controller explicitly introduces integrator on the input channels to create a casual relationship between the inputs and the outputs.

Appendix A

Derivation of State Space Equations

This appendix defines the matrices of the system of equations given in Section 4.2 and restated here:

$$\dot{X} = \begin{bmatrix} \dot{X}_1 \\ \dot{X}_2 \end{bmatrix} = \begin{bmatrix} X_2 \\ F(X_1,\ X_2) \end{bmatrix} + \begin{bmatrix} 0 \\ G(X_1) \end{bmatrix} \mathcal{T} \tag{A.1}$$

$$Y = \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} X_1 \\ a(X_1,\ X_2) \end{bmatrix} + \begin{bmatrix} 0 \\ b(X_1) \end{bmatrix} \mathcal{T} \tag{A.2}$$

Note that $X_1 = [p_o^t\ \phi_1\ \phi_2]^t$, $X_2 = \dot{X}_1$, and $\mathcal{T} = [\tau_1^t\ \tau_2^t]^t$.

First, we begin with the manipulator dynamics (Equation 4.41) restated below:

$$M_i(q_i)\ddot{q}_i\ +\ V_i(q_i, \dot{q}_i)\ =\ \tau_i\ -\ J_i^t \lambda_i F_i \tag{A.3}$$

In the above equation, the acceleration is expressed in joint variables $\ddot{q}_i$. The relationship of the acceleration of the joint variables to the acceleration of the effector reference point was given in Equation 4.46 and restated here:

$$\ddot{p}_i\ =\ J_i\ddot{q}_i\ +\ \dot{J}_i\dot{q}_i \tag{A.4}$$

Expressing the arm dynamics with respect to the reference point of the effector gives us:

$$M_{r_i}\ddot{p}_i\ +\ V_{r_i}\ =\ J_i^{-t}\tau_i\ -\ \lambda_i F_i \tag{A.5}$$

where $M_{r_i} = J_i^{-t} M_i(q_i) J_i^{-1}$, $V_{r_i} = J_i^{-t} V_i(q_i, \dot{q}_i) - J_i^{-t} M_i(q_i) J_i^{-1} \dot{J}_i \dot{q}_i$, and $p_i = [x_i\ y_i\ \phi_i]^t$

For convenience, the three equations of Equation A.5 are partitioned into a two sets, the translational equations (top two) and the rotational equations (bottom one).

$$\begin{bmatrix} M_{rT_i} \\ M_{rB_i} \end{bmatrix} \ddot{p}_i + \begin{bmatrix} V_{rT_i} \\ V_{rB_i} \end{bmatrix} = \begin{bmatrix} J_{T_i}^{-t} \\ J_{B_i}^{-t} \end{bmatrix} \tau_i - \begin{bmatrix} \lambda_{T_i} \\ \lambda_{B_i} \end{bmatrix} F_i \tag{A.6}$$

Note that λ_{T_i} is simply the 2×2 identity matrix, thus the translational equations can be solved for F_i.

$$F_i = J_{T_i}^{-t}\tau_i - M_{rT_i}\ddot{p}_i - V_{rT_i} \tag{A.7}$$

The acceleration relationship of the effector reference point to the circular object's CM was given by Equation 4.45 and restated below:

$$\lambda_i^t \ddot{p}_i + 2\dot{\lambda}_i^t \dot{p}_i - \omega_i^2 d_i = \Gamma_i^t \ddot{p}_o + 2\dot{\Gamma}_i^t \dot{p}_o - \omega_o^2 r_i + \frac{\dot{r}_i \cdot \dot{r}_i}{\mathcal{K}} n_i \tag{A.8}$$

λ_i and Γ_i are given in Equation 4.42 and 4.40, respectively.

Note that the state variable X_1 of the desired system is comprised of p_o and the last elements of p_1 and p_2, namely ϕ_1 and ϕ_2. We partition $\ddot{p}_i = [\ddot{\bar{p}}_i^t \ \ddot{\phi}_i]^t$ where $\ddot{\bar{p}}_i = [\ddot{x}_i \ \ddot{y}_i]^t$ and solve Equation A.8 for $\ddot{\bar{p}}_i$.

$$\ddot{\bar{p}}_i = \begin{bmatrix} 1 & 0 & -r_{i_y} & d_{i_y} & 0 \\ 0 & 1 & r_{i_x} & -d_{i_x} & 0 \end{bmatrix} \ddot{X}_1 + 2\dot{\Gamma}_i^t \dot{p}_o - \omega_o^2 r_1 + \frac{\dot{r}_i \cdot \dot{r}_i}{\mathcal{K}} n_i - 2\dot{\lambda}_i^t \dot{p}_i + \omega_i^2 d_i \tag{A.9}$$

$$\ddot{\bar{p}}_i = \begin{bmatrix} 1 & 0 & -r_{i_y} & 0 & d_{i_y} \\ 0 & 1 & r_{i_x} & 0 & -d_{i_x} \end{bmatrix} \ddot{X}_1 + 2\dot{\Gamma}_i^t \dot{p}_o - \omega_o^2 r_1 + \frac{\dot{r}_i \cdot \dot{r}_i}{\mathcal{K}} n_i - 2\dot{\lambda}_i^t \dot{p}_i + \omega_i^2 d_i \tag{A.10}$$

This equation is restated with the simplified notation:

$$\ddot{\bar{p}}_i = B_i \ddot{X}_1 + C_i \tag{A.11}$$

where the expressions for B_i and C_i are easily obtained from Equations A.9 and A.10. The matrix B_i is a function of position (X_1) only while the matrix C_i is a function of both position and velocity (X_1 and X_2).

Now, the expression for F_i (Equation A.7) is written with the partitioned $\ddot{p}_i$ and Equation A.11 is substituted in for $\ddot{\bar{p}}_i$.

$$F_i = J_{T_i}^{-t}\tau_i - M_{rT_i} \begin{bmatrix} \ddot{\bar{p}}_i \\ \ddot{\phi}_i \end{bmatrix} - V_{rT_i}$$

$$F_i = J_{T_i}^{-t}\tau_i - M_{rT_i} \begin{bmatrix} B_i \ddot{X}_1 + C_i \\ \ddot{\phi}_i \end{bmatrix} - V_{rT_i} \tag{A.12}$$

Note that $\ddot{\phi}_i$ is itself a member of $\ddot{X}_1$, and now F_i is written with respect to the system state.

$$F_i = E_i\ddot{X}_1 + G_i + J_{T_i}^{-t}\tau_i \tag{A.13}$$

where the matrix E_i is a function of position, the matrix G_i is a function of both position and velocity, and they are given as:

$$E_1 = -M_{rT_1}\begin{bmatrix} B_1 \\ 0\,0\,0\,1\,0 \end{bmatrix} \qquad E_2 = -M_{rT_2}\begin{bmatrix} B_2 \\ 0\,0\,0\,0\,1 \end{bmatrix}$$
$$G_1 = -V_{rT_1} - M_{rT_1}\begin{bmatrix} C_1 \\ 0 \end{bmatrix} \qquad G_2 = -V_{rT_2} - M_{rT_2}\begin{bmatrix} C_2 \\ 0 \end{bmatrix}$$

The object motion equation was given in Equation 4.39 and restated here for the case of two manipulators contacting the object:

$$M_o\ddot{p}_o = \Gamma_1 F_1 + \Gamma_2 F_2 + W \tag{A.14}$$

Substituting in Equation A.13 for F_1 and F_2 gives:

$$\begin{aligned} M_o\ddot{p}_o &= \Gamma_1(E_1\ddot{X}_1 + G_1 + J_{T_1}^{-t}\tau_1) + \Gamma_2(E_2\ddot{X}_1 + G_2 + J_{T_2}^{-t}\tau_2) + W \\ M_o\ddot{p}_o &= D\ddot{X}_1 + S + UT \end{aligned} \tag{A.15}$$

where

$$D = \Gamma_1 E_1 + \Gamma_2 E_2 \qquad S = \Gamma_1 G_1 + \Gamma_2 G_2 + W \qquad U = \left[\Gamma_1 J_{T_1}^{-t} \;\; \Gamma_2 J_{T_2}^{-t}\right] \qquad \mathcal{T} = [\tau_1^t \; \tau_2^t]^t$$

The matrices D and U are both functions of position while S is a function of position and velocity. Equation A.15 will be used later in the final steps of formulating the system equations.

The rotational part of Equation A.6 is given as:

$$M_{rB_i}\ddot{p}_i + V_{rB_i} = J_{B_i}^{-t}\tau_i - \lambda_{B_i}F_i \tag{A.16}$$

Substituting in the partitioned p_i and isolating ϕ_i yields:

$$M_{rB_{i_3}}\ddot{\phi}_i = J_{B_i}^{-t}\tau_i - \left[M_{rB_{i_1}} \;\; M_{rB_{i_2}}\right]\ddot{\tilde{p}}_i - V_{rB_i} - \lambda_{B_i}F_i \tag{A.17}$$

Note that M_{rB_i} is a 1×3 vector written as $[M_{rB_{i_1}} \; M_{rB_{i_2}} \; M_{rB_{i_3}}]$. Substituting in the state form of $\ddot{\tilde{p}}_i$ from Equation A.11 and the state form of F_i from Equation A.13 gives:

$$M_{rB_{i_3}}\ddot{\phi}_i = J_{B_i}^{-t}\tau_i - \left[M_{rB_{i_1}} \;\; M_{rB_{i_2}}\right](B_i\ddot{X}_1 + C_i) - V_{rB_i} - \lambda_{B_i}(E_i\ddot{X}_1 + G_i + J_{T_i}^{-t}\tau_i) \tag{A.18}$$

Finally, this equation is solved for $\ddot{\phi}_i$ noting that $M_{rB_{i_3}}$ is a scalar that represents the cartesian inertia of the arm at the effector reference point and is thus non-zero. This equation has the form:

$$\ddot{\phi}_i = L_i \ddot{X}_1 + H_i \tau_i + N_i \tag{A.19}$$

where

$$\begin{aligned} L_i &= \left(\frac{-1}{M_{rB_{i_3}}}\right)\left(\left[M_{rB_{i_1}}\ M_{rB_{i_2}}\right] B_i + \lambda_{B_i} E_i\right) \\ H_i &= \left(\frac{-1}{M_{rB_{i_3}}}\right)[\lambda_{B_i} - 1]\, J_i^{-t} \qquad \text{(A.20)} \\ N_i &= \left(\frac{1}{M_{rB_{i_3}}}\right)[\lambda_{B_i} - 1]\left(M_{r_i}\begin{bmatrix} C_i \\ 0 \end{bmatrix} + V_{r_i}\right) \qquad \text{(A.21)} \end{aligned}$$

The matrices L_i and H_i are functions of position and N_i is a function of both position and velocity.

The system equations are now constructed from Equations A.15 and A.19. They are written together as follows:

$$\begin{bmatrix} M_o & 0 \\ 0 & I_{(2)} \end{bmatrix}\begin{bmatrix} \ddot{p}_o \\ \ddot{\phi}_1 \\ \ddot{\phi}_2 \end{bmatrix} = \begin{bmatrix} D \\ L_1 \\ L_2 \end{bmatrix}\ddot{X}_1 + \begin{bmatrix} S \\ N_1 \\ N_2 \end{bmatrix} + \begin{bmatrix} U & \\ H_1 & 0 \\ 0 & H_2 \end{bmatrix}\mathcal{T} \tag{A.22}$$

The 2×2 identity matrix is denoted as $I_{(2)}$. Note the vector $[\ddot{p}_o\ \ddot{\phi}_1\ \ddot{\phi}_2]^t = \ddot{X}_1$. Thus the system can be written in the simplified expression:

$$\mathcal{M}\ddot{X}_1 = Q + P\mathcal{T} \tag{A.23}$$

where the matrices $\mathcal{M}$, Q, and P are easily obtained from Equation A.22. Again, note that the matrices $\mathcal{M}$ and P are functions of position and Q is a function of position and velocity. If $\mathcal{M}$ is non-singular, then the equation is solved for $\ddot{X}_1$ to yield:

$$\ddot{X}_1 = \mathcal{M}^{-1}Q + \mathcal{M}^{-1}P\mathcal{T} \tag{A.24}$$

This expression defines the matrices $F(X_1, X_2)$ and $G(X_1)$ of Equation 4.47 rewritten here:

$$\dot{X} = \begin{bmatrix} \dot{X}_1 \\ \dot{X}_2 \end{bmatrix} = \begin{bmatrix} X_2 \\ F(X_1, X_2) \end{bmatrix} + \begin{bmatrix} 0 \\ G(X_1) \end{bmatrix}\mathcal{T} \tag{A.25}$$

where $F(X_1, X_2) = \mathcal{M}^{-1}Q$ and $G(X_1) = \mathcal{M}^{-1}P$.

The output $Y_2 = F_1 \cdot e_{12}$. Equation A.13 defines F_i in terms of the state variable $\ddot{X}_1$. Substituting in Equation A.24 for the state yields:

$$F_1 = E_1(\mathcal{M}^{-1}Q + \mathcal{M}^{-1}P\mathcal{T}) + G_1 + J_{T_1}^{-t}\tau_1 \quad \text{(A.26)}$$

$$F_1 = E_1\mathcal{M}^{-1}Q + G_1 + \left(E_1\mathcal{M}^{-1}P + \left[J_{T_1}^{-t} \; 0\right]\right)\mathcal{T} \quad \text{(A.27)}$$

This expression defines the matrices $a(X_1, X_2)$ and $b(X_1)$ of Equation 4.48 rewritten here:

$$Y = \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} X_1 \\ a(X_1, X_2) \end{bmatrix} + \begin{bmatrix} 0 \\ b(X_1) \end{bmatrix}\mathcal{T} \quad \text{(A.28)}$$

where $a(X_1, X_2) = e_{12} \cdot (E_1\mathcal{M}^{-1}Q + G_1)$ and $b(X_1) = e_{12} \cdot (E_1\mathcal{M}^{-1}P + \left[J_{T_1}^{-t} \; 0\right])$.

This concludes the derivation of the matrices of the system. This formulation is used in the theoretical development as well as the simulation and experimentation sections.

Bibliography

[1] *WAM-1 Maintenance and User Manual.* Barrett Technology, Inc, 1990.

[2] F. Beer and R. Johnson Jr. *Vector Mechanics for Engineers.* McGraw-Hill, 1984.

[3] C. Cai. Control of a manipulator under multiple contacts between rigid curved surfaces. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 1990.

[4] A. Cole, J. Hauser, and S. Sastry. Kinematics and control of multifingered hands with rolling contact. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 228–233, Philadelphia, PA, April 1988.

[5] Pierre Dauchez. *Task Descriptions for the Symmetric Hybrid Control of a Two-Arm Robot Manipulator.* PhD thesis, University of Montpellier, 1990.

[6] G. F. Franklin, J. D. Powell, and M. L. Workman. *Digital Control of Dynamic Systems.* Addison-Wesley Publishing Co., Inc., second edition, 1990.

[7] G. Gou and W. Gruver. Fingertip force planning for multifingered robot hands. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 665–672, Sacramento, CA, April 1991.

[8] P. Hsu, Z. Li, and S. Sastry. On grasping and coordinated manipulation by a multifingered robot hand. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 384–389, Philadelphia, PA, April 1988.

[9] T. Ishida. Force control in coordination of two arms. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence*, pages 717–722, August 1977.

[10] A. Isidori. *Nonlinear Control Systems: An Introduction.* Springer-Verlag, Berlin, New York, 1985.

[11] J. Kerr and B. Roth. Analysis of multifingered hands. *Int. J. of Robotics Research*, 4(4), 1986.

[12] Kab I. Kim and Yuan F. Zheng. Two strategies of position and force control for two industrial robots handling a single object. *Robotics and Autonomous Systems*, 5(4):395–403, 1989.

[13] Kab Il Kim and Yuan F. Zheng. Unknown load distribution of two industrial robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 992–997, Sacramento, CA, April 1991.

[14] Antti J. Koivo and George A. Bekey. Report of workshop on coordinated multiple robot manipulators: planning, control, and application. *IEEE Journal of Robotics and Automation*, 4(1):91–93, February 1988.

[15] Carl D. Kopf and Tetsuro Yabuta. Experimental comparison of master/slave and hybrid two arm position/force control. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1633–1637, Philadelphia, PA, April 1988.

[16] V. Kumar and K. J. Waldron. Suboptimal algorithms for force distribution in multifingered grippers. *IEEE transactions on Robotics and Automation*, 5(4), August 1989.

[17] V. Kumar, X. Yun, E. Paljug, and N. Sarkar. Control of contact conditions for manipulation with multiple robotic systems. In *Proceedings of the IEEE International Conference on Robotics and Automation*, May 1991.

[18] Frank L. Lewis. *Optimal Control.* John Wiley & Sons, Inc., 1986.

[19] Zexiang Li and John Canny. Motion of two rigid bodies with rolling constraint. *IEEE Transactions on Robotics and Automation*, 6(1):62–72, February 1990.

[20] David J. Montana. The kinematics of contact and grasp. *The International Journal of Robotics Research*, 7(3):17–32, June 1988.

[21] Y. Nakamura, K. Nagai, and T. Yoshikawa. Dynamics and stability in coordination of multiple robotic mechanisms. *The International Journal of Robotics Research*, 8(2):44–61, 1989.

[22] E. Paljug and X. Yun. Tracs: two robotic arm coordination system. In *Video Proceedings of the IEEE International Conference on Robotics and Automation*, May 1991.

[23] E. Paljug, X. Yun, and F. Fuma. *TRACS: The Hardware and Software Architecture of a New Two Robotic Arm Coordination System.* Technical Report MS-CIS-90-70, University of Pennsylvania, 1990. GRASP Lab 236.

[24] J. K. Salisbury, W. Townsend, B. Eberman, and D. DiPietro. Preliminary design of a whole arm manipulation system. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 254–260, Philadelphia, PA, April 1988.

[25] K. Salisbury and B. Roth. Kinematics and force analysis of articulate mechanical hands. *ASME Journal of Mechanisms, Transmissions and Automation in Design*, 105:35–41, March 1983.

[26] T. J. Tarn, A. K. Bejcsy, and X. Yun. Design of dynamic control of two cooperating robot arms: closed chain formulation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 7–13, Raleigh, North Carolina, March 1987.

[27] T. J. Tarn, A. K. Bejczy, and X. Yun. New nonlinear control algorithms for multiple robot arms. *IEEE Transactions on Aerospace and Electronic Systems*, 24(5):571–583, September 1988.

[28] Masaru Uchiyama and Pierre Dauchez. A symmetric hybrid position/force control scheme for the coordination of two robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 350–356, Philadelphia, PA, April 1988.

[29] M. A. Unseren and A. J. Koivo. Reduced order model and decoupled control architecture for two manipulators holding an object. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1240–1245, Scottsdale, Arizona, May 1989.

[30] S. Wolfram. *Mathematics: A System for Doing Mathematics by Computer.* Addison Wesley Publishing Company, 1988.

[31] T. Yoshikawa, T. Sugie, and M. Tanaka. Dynamic hybrid position/force control of robot manipulators—controller design and experiment. *IEEE Journal of Robotics and Automation*, 4(6):699–705, 1988.

[32] Tsuneo Yoshikawa. Dynamic hybrid position/force control of robot manipulators—description of hand constraints and calculation of joint driving force. *IEEE Journal of Robotics and Automation*, RA-3(5):386–392, 1987.

[33] Tsuneo Yoshikawa and Xinzhi Zheng. Coordinated dynamic control for multiple robot manipulators handling an object. In *Proceedings of the International Conference on Advanced Robotics*, pages 579–584, 1991.

[34] Tsuneo Yoshikawa and Xinzhi Zheng. Coordinated dynamic hybrid position/force control for multiple robot manipulators handling one constrained object. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1178–1183, 1990.

[35] X. Yun. Coordination of two-arm pushing. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 182–187, Sacramento, CA, April 1991.

[36] X. Yun and V. Kumar. An approach to simultaneous control of trajectory and interaction forces in dual arm configurations. *IEEE Transaction on Robotics and Automation*, October 1991.

[37] X. Yun, E. Paljug, and R. Bajcsy. *TRACS: An Experimental Multiagent Robotic System.* Technical Report MS-CIS-90-60, University of Pennsylvania, 1990. GRASP Lab 230.

[38] Y. F. Zheng and J. Y. S. Luh. Control of two coordinated robots in motion. In *Proceedings of IEEE Conference on Decision and Control*, pages 1761–1765, Ft. Lauderdale, Florida, December 1985.

[39] Y. F. Zheng and J. Y. S. Luh. Optimal load distribution for two industrial robots handling a single object. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 344–359, 1988.