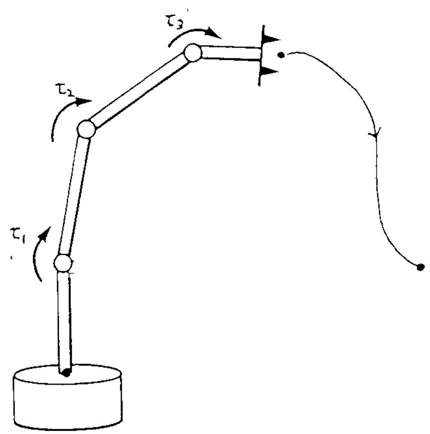
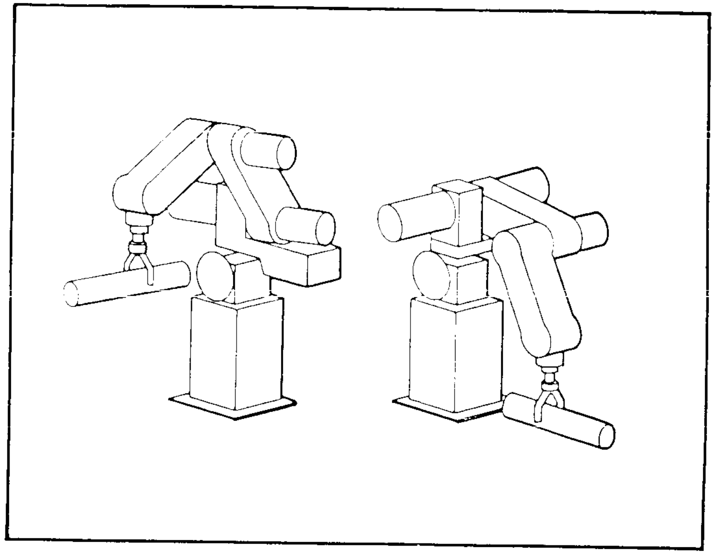
ROBOT CONTROL

The **control problem** for robot manipulator is to determine the joint inputs required to cause the robot to follow a commanded motion.



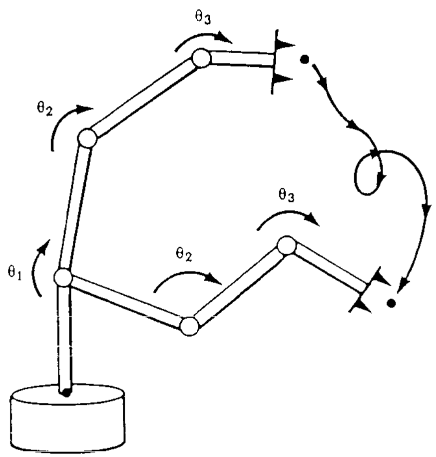
The joint inputs may be joint forces and torques, or inputs to the actuators, for example, voltage inputs to the motors.



The basic problem is to move the robot from an initial position to some desired final position. This control problem is called **Point to Point Control**.

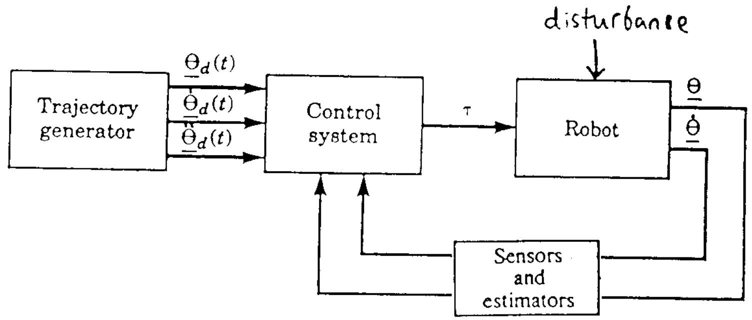
Sometimes, it is necessary to specify the motion in much more detail than simply stating the desired final position. Thus, a desired path or trajectory should be specified. Such control problem is called **Path Control.**

Usually, it is desirable for the motion of the robot to be *smooth*. Rough, jerky motions tend to cause increased wear on the mechanism.



Feedback Control System

The following figure shows a block diagram of feedback control system for robot manipulator. Such system uses feedback from **sensors** to keep the manipulator on course.



In order to cause the manipulator to follow the desired motion, we must specify a **control algorithm** which sends control inputs to the joints.

Note that **disturbances**, which are inputs that we cannot control, also influence the behaviour of the system.

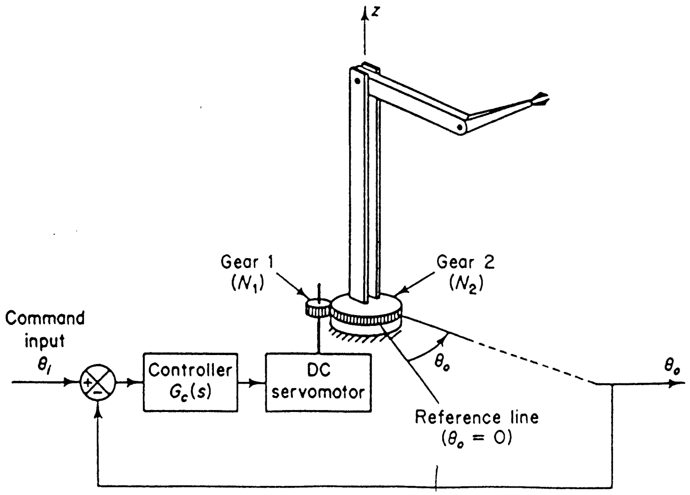
Therefore, the controller must also be designed in such a way that the effects of the disturbances are reduced. That is, to **reject the disturbances**.

2. Motion Control Schemes

Several **control techniques** can be applied to the control of manipulators. The simplest method is the direct applications of **linear control theory**. Other methods include some **non-linear control schemes** such as computed torque control.

2.1 Classical Joint Control

In this method, each joint of the robot is treated as a simple joint servomechanism.



In the early days of robotics, **independent joint control** was popular since it allows a **decoupled analysis** of the closed-loop system using single-input/single-output (SISO) classical techniques.

2.2 Non-linear Control Methods

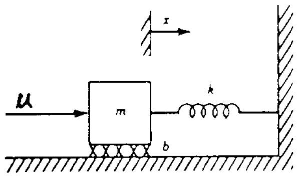
Robot dynamics are actually highly **non-linear with coupling** exists between joints. In the classical control approach, we made several approximations that allow a linear analysis of the control problem. This approximation result in limiting precision and speed of the end-effector making it appropriate only for limited precision tasks.

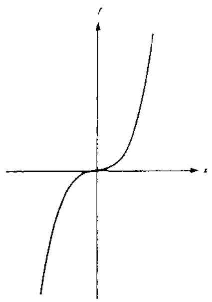
To improve the performance, we need to consider the use of more efficient **dynamic model** and **non-linear design methods**.

Example of non-linear mechanical system

Before considering the non-linear techniques for control of robot manipulator, let us start by considering a simple nonlinear mechanical system to illustrate the concept.

The following figure shows a block of mass m attached to a spring and subject to a friction of coefficient b. The spring relationship is described by the nonlinear equation $f = kx^3$.





Suppose that an actuating force u is applied to the block. The equation of motion can be derived from Newton's law of motion as follows:

$$m\ddot{x} + b\dot{x} + kx^3 = u$$

One way of controlling the system is to linearize the system by restricting the motion of the mechanical system to a linear region. However, this approximation results in limiting performance of the system. We will introduce a more advanced control technique for which this assumption will not have to be made.

2.2.1 Control Law Partitioning

In this method, we will partition the controller into a **model-based portion** and a **servo portion**. The **basic concept** of this method is to cancel the non-linearity of system using the model-based portion so that the overall closed-loop system is linear. For example, considering the control of the above nonlinear mechanical system, the model-based portion of the control appears in a control law of the form:

$$u = \alpha v + \beta$$

where α and β are functions to be chosen so that if v is taken as the *new input* to the system, *the system appears to be a unit mass.*

Substituting the control into the equation of motion, we have

$$m\ddot{x} + b\dot{x} + kx^3 = \alpha\, v + \beta$$

Clearly, in order to make the system appears as a unit mass with input v, for this particular system we should choose α and β as:

$$\alpha = m$$

$$\beta = b\dot{x} + kx^3.$$

Making these assignments and plugging them back into the closed-loop equation, we have

$$\ddot{x} = v$$

which is a equation of motion of a unit mass. We can now design a control law to compute v as follows:

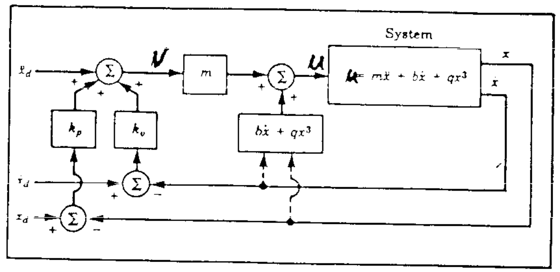
$$v = \ddot{x}_d + k_v\dot{e} + k_p e$$

where $e = x_d - x$ and x_d denote a desired position. Combining the above two equations give the error equation:

$$\ddot{e} + k_v\dot{e} + k_p e = 0$$

Recall from the study of differential equation and E301, the form of the solution depends on the roots of the characteristic equation.

The following figure shows a block diagram of the system.



Performance Specifications

The characteristic equation of a second order system can be expressed in the standard form,

$$s^2 + 2\zeta w_n s + w_n^2 = 0$$

where

ς is the damping ratio

w_n is the undamped natural frequency.

For the reason of safety, the robot manipulator cannot have an underdamped response. That is,

$$\varsigma \geq 1$$

In practice, we must be careful not to excite structural oscillation and resonance of the joint.

A rule of thumb is to choose w_n such that,

$$w_n \leq 0.5\, w_{res}$$

where w_{res} is the lowest structural resonant frequency.

Compare with the closed-loop characteristics equation,

$$s^2 + k_v s + k_p = 0,$$

the feedback gains should be chosen as,

$$k_v = 2\varsigma\, w_n,$$

$$k_p = w_n^2.$$

Disturbance Rejection

One of the purposes of a control system is to maintain good performance (small errors) even in the presence of some external disturbance. In the previous example, if a disturbance force d is presence in the system so that:

$$m\ddot{x} + b\dot{x} + kx^3 + d = u,$$

the error equation becomes

$$\ddot{e} + k_v\dot{e} + k_p e = d/m.$$

Taking Laplace transformation gives,

$$\frac{E(s)}{D(s)} = \frac{1}{m(s^2 + k_v s + k_p)}$$

The steady state error is therefore given as

$$e_{ss} = \frac{d}{mk_p}$$

Thus it is clear that the higher the position gain, the smaller the steady state error.

In order **to eliminate the steady state error,** an integral term is added to the control law:

$$v = \ddot{x}_d + k_v\dot{e} + k_p e + k_i \int e\, dt.$$

This result in the error equation

$$\ddot{e} + k_v\dot{e} + k_p e + k_i \int e\, dt = d/m$$

Taking Laplace transformation gives,

$$\frac{E(s)}{D(s)} = \frac{s}{m(s^3 + k_v s^2 + k_p s + k_i)}$$

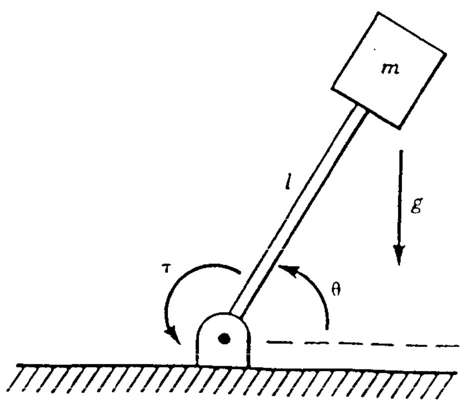
For a constant disturbance such that $D(s) = \frac{d}{s}$, we have

$$e_{ss} = \lim_{t\to\infty} e(t) = \lim_{s\to 0} sE(s)$$

$$= \lim_{s\to 0} \frac{s}{m(s^3 + k_v s^2 + k_p s + k_i)} = 0$$

2.2.2 Robot Dynamic Model

Robot dynamics can be derived from the Langrange equation of motion (classical mechanics) by considering the energies of the mechanical system. Let us begin our discussion by looking at the dynamic model of a simple single-link robot as shown in the figure.



Model of a single link robot

The mass is considered to be located at the end of the link. The kinetic energy is,

$$K = \frac{1}{2}mv^2 = \frac{1}{2}ml^2\dot{\theta}^2$$

and the potential energy is

$$P = mgy = mgl\sin(\theta)$$

L is the Lagrangian which is the difference between the kinetic and potential energies,

$$L = K - P$$

The dynamic model can be derived from the Langrange's equation of motion as:

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}} = \tau + \frac{\partial L}{\partial \theta},$$

which is just a restatement of Newton's law:

$$\frac{d}{dt}(momentum) = force$$

Since

$$L = K - P = \frac{1}{2} ml^2\dot{\theta}^2 - mgl\sin(\theta)$$

$$\frac{\partial L}{\partial \dot{\theta}} = ml^2\dot{\theta}$$

$$\frac{\partial L}{\partial \theta} = -mgl\cos(\theta)$$

From the langrange equation, we have

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}} = \tau + \frac{\partial L}{\partial \theta}$$

$$\Rightarrow \frac{d}{dt}(ml^2\dot{\theta}) = \tau - mgl\cos(\theta)$$

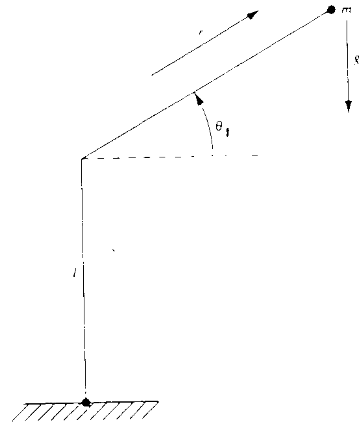
Therefore, the dynamic model of the manipulator can be derived as:

$$ml^2\ddot{\theta} + m\lg\cos(\theta) = \tau$$

which is nonlinear. The moment of inertia is ml^2 and there is a gravitational force $m\lg\cos(\theta)$ due to load.

Model of a two-link polar robot

Next, let us consider a two-link polar robot as shown in the figure. Assume that the link mass is concentrated at the end of the link.



The total kinetic energy is,

$$K = \frac{1}{2} m\, r^2\dot{\theta}_1^2 + \frac{1}{2} m\, \dot{r}^2$$

and the potential energy is

$$P = mgr\sin(\theta_1)$$

The Lagrangian is

$$L = K - P = \frac{1}{2} m\, r^2 \dot{\theta}_1^2 + \frac{1}{2} m\, \dot{r}^2 - mgr\sin(\theta_1)$$

If the joint angles, velocities and control torque are expressed into a vector form as:

$$\theta = \begin{bmatrix} \theta_1 \\ r \end{bmatrix}, \; \dot{\theta} = \begin{bmatrix} \dot{\theta}_1 \\ \dot{r} \end{bmatrix}, \quad \tau = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}.$$

Now, we have

$$\frac{\partial L}{\partial \dot{\theta}} = \begin{bmatrix} \frac{\partial L}{\partial \dot{\theta}_1} \\ \frac{\partial L}{\partial \dot{r}} \end{bmatrix} = \begin{bmatrix} mr^2 \dot{\theta}_1 \\ m\dot{r} \end{bmatrix}$$

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}} == \begin{bmatrix} mr^2 \ddot{\theta}_1 + 2mr\dot{r}\dot{\theta}_1 \\ m\ddot{r} \end{bmatrix}$$

$$\frac{\partial L}{\partial \theta} = \begin{bmatrix} \frac{\partial L}{\partial \theta_1} \\ \frac{\partial L}{\partial r} \end{bmatrix} = \begin{bmatrix} -mgr\cos(\theta_1) \\ mr\dot{\theta}^2 - mg\sin(\theta_1) \end{bmatrix}$$

From the langrange equation, we have

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}} = \tau + \frac{\partial L}{\partial \theta}$$

which yields the dynamics equation,

$$mr^2\ddot{\theta}_1 + 2mr\dot{r}\dot{\theta}_1 + mgr\cos(\theta_1) = \tau_1$$

$$m\ddot{r} - mr\dot{\theta}_1^2 + mg\sin(\theta_1) = \tau_2$$

Writing the dynamics in vector form yields:

$$M(\theta)\ddot{\theta} + C(\theta,\dot{\theta})\dot{\theta} + g(\theta) = \tau$$

where $\theta = \begin{bmatrix} \theta_1 \\ r \end{bmatrix}$ and $\tau = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}$.

$M(\theta)$ is called the **manipulator mass matrix.**

$$M(\theta) = \begin{bmatrix} mr^2 & 0 \\ 0 & m \end{bmatrix}$$

Any manipulator mass matrix is symmetric and positive definite, and therefore, always invertable.

The term $C(\theta,\dot{\theta})\dot{\theta}$ contains all the terms that are dependent on joint velocity. Therefore, we have,

$$C(\theta,\dot{\theta})\dot{\theta} = \begin{bmatrix} 2mr\dot{r}\dot{\theta}_1 \\ -mr\dot{\theta}_1^2 \end{bmatrix}$$

A term like $mr\dot{\theta}_1^2$ is caused by a **centrifugal force** and is organized as such because it depends on the square of a joint velocity. A term like $2m\,r\dot{r}\dot{\theta}_1$ is caused by a **Coriolis force** and will always contain the product of two different joint velocities.

The **gravity term** contain all those terms in which the gravitational constant, g, appears. Therefore we have

$$g(\theta) = \begin{bmatrix} mgr\cos(\theta_1) \\ mg\sin(\theta_1) \end{bmatrix}.$$

Model of a n-link planar robot

In general, the vector equation of motion of a robot with n degree of freedom can be written as:

$$M(\theta)\ddot{\theta} + C(\theta,\dot{\theta})\dot{\theta} + g(\theta) = \tau$$

where

$$\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix}$$ is the *n x 1* vector of joint positions,

$$\tau = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \vdots \\ \tau_n \end{bmatrix}$$ is the *n x 1* vector of joint inputs,

M(θ) is an *n x n* matrix called the **manipulator mass matrix**, $C(\theta,\dot{\theta})\dot{\theta}$ represents torque arising from **centrifugal and Coriolis** forces and *g(θ)* represents torque due to **gravity**.

2.2.3 Nonlinear Control of Robot

In the previous section, we have seen that robot dynamics are actually highly non-linear with coupling exists between joints. We now return to the partitioned control law (or **Computed Torque control law**) and will see that it can be applied to the control of robot with nonlinear dynamics.

Recall that the **basic concept** of the partitioned control law or computed torque control is to cancel the non-linearity of the robot dynamics using the model-based portion so that the overall closed-loop system is linear.

Let us review the concept again by considering the single-link robot with one rotational joint.

The dynamic model of the manipulator is given as:

$$ml^2\ddot{\theta} + m\lg\cos(\theta) = \tau$$

The model-based portion of the control appears in a control law of the form:

$$\tau = \alpha v + \beta$$

where α and β are functions to be chosen so that if v is taken as the *new input* to the system, *the system appears to be a unit mass.*

Substituting the control into the equation of motion, we have

$$ml^2\ddot{\theta} + m\lg\cos(\theta) = \alpha\ v + \beta$$

Clearly, in order to make the system appears as a unit mass with input v, for this particular system we should choose α and β as:

$\alpha =$

$\beta =$

Making these assignments and plugging them back into the closed-loop equation, we have

$$\ddot{\theta} = v$$

We can now design a control law to compute v as follows:

$$v = \ddot{\theta}_d + k_v \dot{e} + k_p e$$

where $e = \theta_d - \theta$ and θ_d denote a desired joint angle. Combining the above two equations give the error equation:

$$\ddot{e} + k_v \dot{e} + k_p e = 0$$

Multi-input, multi-output Control Systems

The dynamic model of the robot manipulator is given as:

$$M(\theta)\ddot{\theta} + C(\theta,\dot{\theta})\dot{\theta} + g(\theta) = \tau,$$

which is a multi-input, multi-output systems. That is, we have a vector of joint torque $\tau = [\tau_1, \cdots, \tau_n]^T \in R^n$ and a vector of joint angles $\theta = [\theta_1, \cdots, \theta_n]^T \in R^n$ for a robot with n degree of freedom. Given this system, the partitioned control law still takes the form

$$\tau = \alpha v + \beta$$

but now v and β are $n \times 1$ vectors and α is an $n \times n$ matrix. This gives

$$M(\theta)\ddot{\theta} + C(\theta,\dot{\theta})\dot{\theta} + g(\theta) = \alpha v + \beta$$

Therefore,

$$\alpha = M(\theta)$$

$$\beta = C(\theta,\dot{\theta})\dot{\theta} + g(\theta)$$

gives the closed-loop equation:

$$\ddot{\theta} = v$$

PD-Computed Torque

The servo law for the multi-input, multi-output robot system becomes

$$v = \ddot{\theta}_d + K_v \dot{E} + K_p E$$

where θ_d, $\dot{\theta}_d$ and $\ddot{\theta}_d$ are vectors of desired joint angles, velocity and accelerations respectively, K_v and K_p are now $n \times n$ matrices which are in general chosen to be diagonal. $E = \theta_d - \theta$, $\dot{E} = \dot{\theta}_d - \dot{\theta}$ are $n \times 1$ vectors of errors in position and velocity respectively.

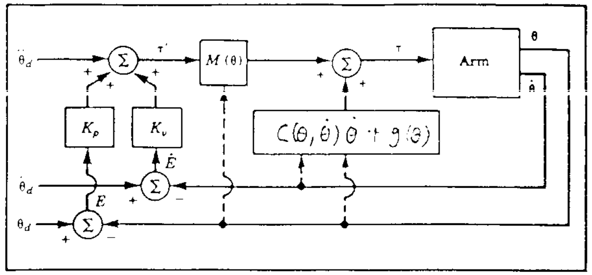
Now, the error equation becomes:

$$\ddot{E} + K_v \dot{E} + K_p E = 0$$

Note that the error vector equation can be decoupled since the matrices K_p and K_v can

be chosen as diagonal matrices so that

$$\ddot{e}_i + k_{vi} \dot{e}_i + k_{pi} e_i = 0$$



PID-Computed Torque

With the addition of an integral term in the servo control law:

$$v = \ddot{\theta}_d + K_v \dot{E} + K_p E + K_I \int E \, dt$$

the closed-loop system becomes:

$$\ddot{E} + K_v \dot{E} + K_p E + K_I \int E \, dt = 0$$

Review: What is the purpose of adding the integral term ?

Example

Consider the dynamic equation of the 2-link polar robot with mass $m = 2kg$. Design a PD Computed torque controller for the robot. The resonance frequency of the robot is 12 rad per second.

If m = 2 kg, from page 21, we have

$$M(\theta)\ddot{\theta} + C(\theta,\dot{\theta})\dot{\theta} + g(\theta) = \tau$$

where $\theta = \begin{bmatrix} \theta_1 \\ r \end{bmatrix}$, $\tau = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}$,

$$M(\theta) = \begin{bmatrix} 2r^2 & 0 \\ 0 & 2 \end{bmatrix}, \quad C(\theta,\dot{\theta})\dot{\theta} = \begin{bmatrix} 4r\dot{r}\dot{\theta}_1 \\ -2r\dot{\theta}_1^2 \end{bmatrix},$$

$$g(\theta) = \begin{bmatrix} 19.6r\cos(\theta_1) \\ 19.6\sin(\theta_1) \end{bmatrix}.$$

The partitioned control law still takes the form

$$\tau = \alpha v + \beta$$

This gives

$$M(\theta)\ddot{\theta} + C(\theta,\dot{\theta})\dot{\theta} + g(\theta) = \alpha v + \beta$$

Therefore,

$$\alpha = M(\theta) = \begin{bmatrix} 2r^2 & 0 \\ 0 & 2 \end{bmatrix}$$

$$\beta = C(\theta,\dot{\theta})\dot{\theta} + g(\theta) = \begin{bmatrix} 4r\dot{r}\dot{\theta}_1 + 19.6r\cos(\theta_1) \\ -2r\dot{\theta}_1^2 + 19.6\sin(\theta_1) \end{bmatrix}$$

gives the closed-loop equation:

$$\ddot{\theta} = v$$

or

$$\begin{bmatrix} \ddot{\theta}_1 \\ \ddot{r} \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$$

The servo law becomes

$$v = \ddot{\theta}_d + K_v\dot{E} + K_pE$$

or

$$\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} \ddot{\theta}_{1d} \\ \ddot{r}_d \end{bmatrix} + \begin{bmatrix} k_{v1} & 0 \\ 0 & k_{v2} \end{bmatrix} \begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \end{bmatrix} + \begin{bmatrix} k_{p1} & 0 \\ 0 & k_{p2} \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \end{bmatrix}$$

where K_v and K_p are now 2x2 diagonal matrices,

$\theta_d = \begin{bmatrix} \theta_{1d} \\ r_d \end{bmatrix}$ is the desired joint trajectory, $E = \theta_d - \theta$ $= \begin{bmatrix} \theta_{1d} - \theta_1 \\ r_d - r \end{bmatrix}$, $\dot{E} = \dot{\theta}_d - \dot{\theta} = \begin{bmatrix} \dot{\theta}_{1d} - \dot{\theta}_1 \\ \dot{r}_d - \dot{r} \end{bmatrix}$ are 2×1 vectors.

Now, the error equation becomes:

$$\begin{bmatrix} \ddot{e}_1 \\ \ddot{e}_2 \end{bmatrix} + \begin{bmatrix} k_{v1} & 0 \\ 0 & k_{v2} \end{bmatrix} \begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \end{bmatrix} + \begin{bmatrix} k_{p1} & 0 \\ 0 & k_{p2} \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = 0$$

or

$\ddot{e}_i + k_{vi}\dot{e} + k_{pi}e = 0$ where $i=1,2$.

Hence, the closed-loop characteristics equation is,

$$s^2 + k_{vi}s + k_{pi} = 0.$$

For the reason of safety, the robot manipulator cannot have an underdamped response. That is,

$$\varsigma \geq 1$$

We choose

$$\varsigma = 1$$

for a critically damped response.

In practice, we must be careful not to excite structural oscillation and resonance of the joint.

$w_n \leq 0.5\ w_{res} = 0.5 \times 12 = 6\ rad\ s^{-1}$

The characteristic equation can be expressed in the standard form as,

$$s^2 + 2\varsigma\ w_n s + w_n^2 = s^2 + 12s + 36 = 0$$

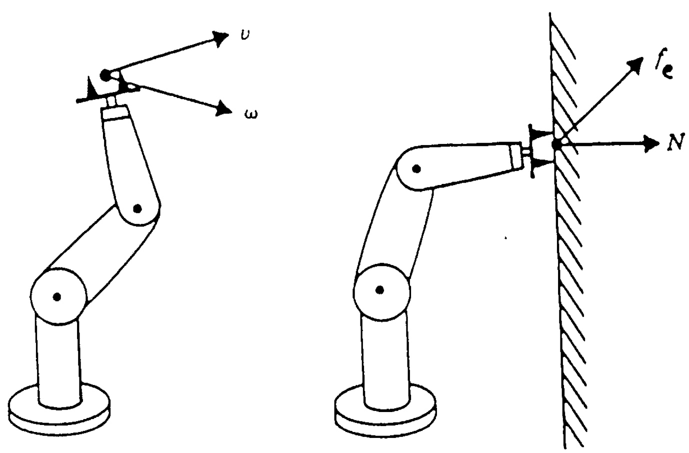
The feedback gains should be chosen as,

$k_v = 12$

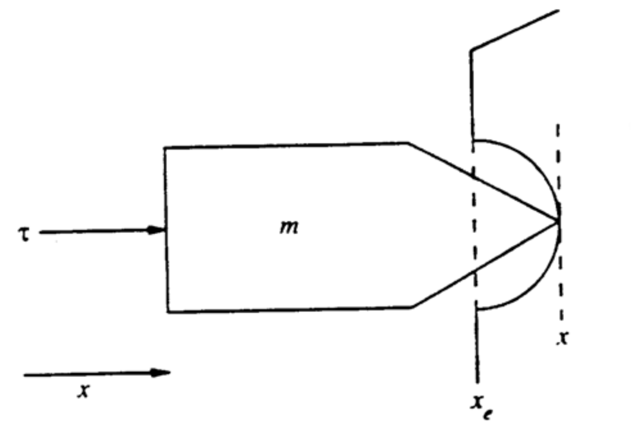
$k_p = 36.$

3 Force control

In many applications, such as deburring, spot welding and assembly parts, the manipulator end effector is in contact with the environments. Therefore, position as well as interaction forces need to be controlled.



To illustrate the concept, let us consider the force control problem for the system as follows:



The overall system is governed by the equation:

$$m\ddot{x} + f = \tau$$

Assuming that the environmental stiffness can be modeled as a spring, the force exerted on the environment is given by:

$f = k_e(x - x_e)$

where k_e is the environmental stiffness and x_e is the static location of the environment.

Let

$\tau = \alpha v + \beta$

The closed-loop system is described by

$m\ddot{x} + f = \alpha v + \beta$

Choose

$\alpha = m$

$\beta = f$

gives

$\ddot{x} = v$

Since $\ddot{f} = k_e \ddot{x}$, the dynamics equation can be written as

$$\frac{1}{k_e}\ddot{f} = v$$

Let $e_{ff} = f_d - f$ where f_d is a desired force, the force servo controller is:

$$v = \frac{1}{k_e}(f_d + k_v \dot{e}_f + k_{1p} e_f)$$

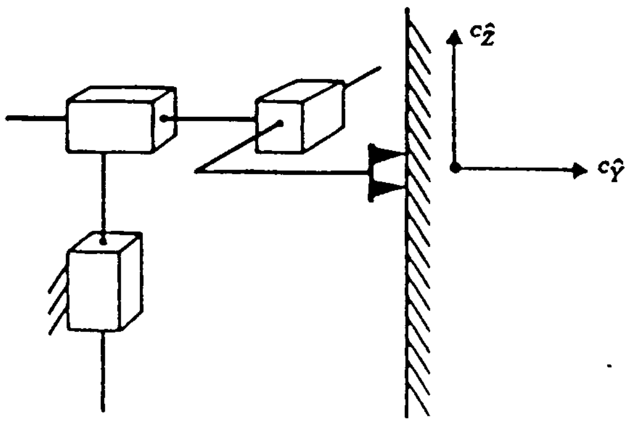
Thus

$$\ddot{e}_f + k_v \dot{e}_f + k_p e_f = 0$$

3.1 Hybrid Position/Force Control

In force control applications, we normally need to control the motions in certain directions (**motion control subspace**) and the force in the direction orthogonal to the motions **(force control subspace**).

For example, consider the following robot with three degrees of freedom contacting a surface.



The Cartesian space vector for this robot is:

$$X = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

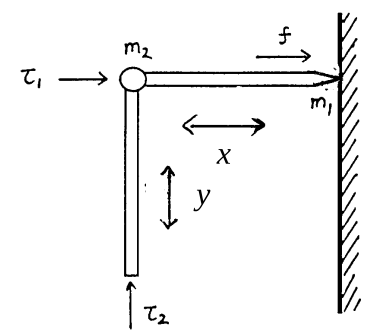
The end effector is free to move in x and z directions but the motion in the y direction is constrained by a surface.

Therefore, the x and z components are to be **position controlled** and the y component is to be **force controlled**.

The **basic concept** of Hybrid Position/Force control method is to decouple the position and force control problems into two sub-tasks so that motion and force controllers can be designed separately.

Example

To illustrate the concept, consider the following robot with two degree of freedom.



The

Cartesian space vector for this robot is:

$$X = \begin{bmatrix} x \\ y \end{bmatrix}$$

The position y along the surface should be **motion controlled** while x should be **force controlled.**

The dynamics of the robot is given as:

$$m_1\ddot{x} + f = \tau_1$$

$$(m_1 + m_2)\ddot{y} + (m_1 + m_2)g = \tau_2$$

or

$$\begin{bmatrix} m_1 & 0 \\ 0 & m_1 + m_2 \end{bmatrix}\begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} + \begin{bmatrix} 0 \\ (m_1 + m_2)g \end{bmatrix} + \begin{bmatrix} f \\ 0 \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}$$

The hybrid position and force controller is:

$$\tau = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \alpha v + \beta$$

where

$$\alpha = \begin{bmatrix} m_1 & 0 \\ 0 & m_1 + m_2 \end{bmatrix}, \quad \beta = \begin{bmatrix} 0 \\ (m_1 + m_2)g \end{bmatrix} + \begin{bmatrix} f \\ 0 \end{bmatrix}.$$

Therefore, we have

$$\ddot{X} = \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = v$$

Let $v = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$, we have

$$\ddot{x} = v_1$$

$$\ddot{y} = v_2$$

Position Control

The servo control law is

$$v_2 = \ddot{x}_d + k_{2v}\dot{e} + k_{2p}e$$

where $e = x_d - x$. Therefore,

$$\ddot{e} + k_{2v}\dot{e} + k_{2p}e = 0$$

Force Control

The normal force exerted on the surface is given by:

$$f = k_e(x_1 - x_e)$$

where k_e is the surface stiffness. Since $\ddot{f} = k_e\ddot{x}_1$, the dynamics equation can be written as

$$\frac{1}{k_e}\ddot{f} = v_1$$

Let $e_f = f_d - f$, the force servo controller is:

$$v_1 = \frac{1}{k_e}(f_d + k_{1v}\dot{e}_f + k_{1p}e_f)$$

Thus

$$\ddot{e}_f + k_{1v}\dot{e}_f + k_{1p}e_f = 0$$

The force controller requires f and $\dot{f}$, which can be calculated from the position and velocity:

$$f = k_e(x_1 - x_e)$$

$$\dot{f} = k_e\dot{x}_1$$

In general, suppose the robot dynamics is given as:

$$M_x\ddot{X} + C_x + g_x + f_e = \tau$$

The hybrid position and force controller takes the form:

$$\tau = \alpha v + \beta$$

$$\alpha = M_x$$

$$\beta = C_x + g_x + f_e$$

Substitute into the robot dynamics, we have

$$M_x \ddot{X} + C_x + g_x + f_e = \alpha v + \beta$$

which gives:

$$\ddot{X} = v$$

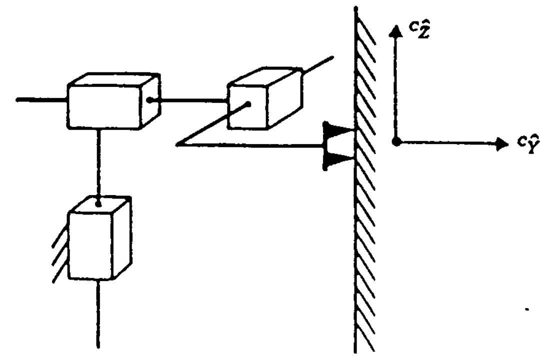
Now, let

$$X = \begin{bmatrix} x_T \\ x_N \end{bmatrix}$$

where x_T is the motion vector along the environment and x_N is the motion vector normal to the environment. One must determine (select) which variables should be force controlled and which should be position controlled. For example,

$$x_T = \begin{bmatrix} x \\ z \end{bmatrix}, \quad x_N = y$$

in the previous example.



Similarly, the velocity and acceleration vectors can be written as

$$\dot{X} = \begin{bmatrix} \dot{x}_T \\ \dot{x}_N \end{bmatrix}, \quad \ddot{X} = \begin{bmatrix} \ddot{x}_T \\ \ddot{x}_N \end{bmatrix}$$

If we partition the servo control law as

$$F' = \begin{bmatrix} F_T \\ F_N \end{bmatrix}$$

we have

$\ddot{x}_T = F_T$ (motion control)

$\ddot{x}_N = F_N$ (force control)

The motion and force controller can now be designed separately based on the above equations.

Motion control

The motion servo law:

$$F_T = \ddot{x}_{Td} + K_{Tv}\dot{E}_T + K_{Tp}E_T$$

results in the error equation:

$$\ddot{E}_T + K_{Tv}\dot{E}_T + K_{Tp}E_T = 0$$

where $E_T = x_{Td} - x_T$ and x_{Td} is a desired position of the end-effector on the environment.

Force control

Assume that the surface can be modeled as a spring, the force exerted on the environment is modeled as:

$$f_e = K_e(x_N - x_e)$$

where K_e is a diagonal matrix used to denote the environmental stiffness and x_e is a vector that is used to denote the static location of the environment. Taking second derivative with respect to time gives:

$$\ddot{x}_N = K_e^{-1}\ddot{f}_e$$

Hence

$$K_e^{-1}\ddot{f}_e = F_N$$

We define a force tracking error as $E_N = f_{ed} - f_e$ where f_{ed} is a desired force normal to the environment. The force servo law:

$$F_N = K_e^{-1}(\ddot{f}_{ed} + K_{Nv}\dot{E}_N + K_{Np}E_N)$$

gives the closed loop error equation:

$$\ddot{E}_N + K_{Nv}\dot{E}_N + K_{Np}E_N = 0.$$