Contents

1 Coordinate Frames

1.1 Descriptions

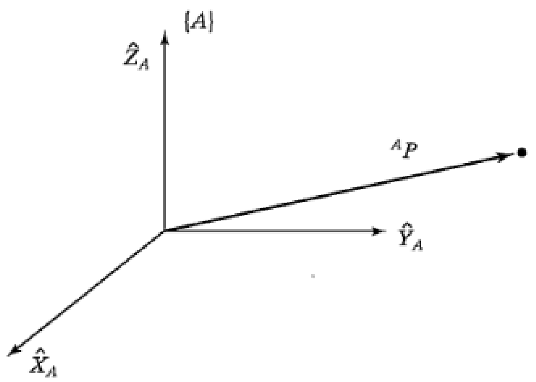
- used to specify attributes of various objects with which the manipulation system deals (parts, tools, manipulator)

1.1.1 Description of a position

Let A be a defined coordinate system in addition to the universe coordinate system and P a position vector. Then

$$^{A}P = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}$$

is a point represented as a vector in the coordinate system $\{A\}$ (p_x, p_y and p_z indicate distances along the axes of $\{A\}$)



1.1.2 Description of an orientation

- Is a description of a coordinate system relative to the reference system

Let $\{B\}$ be a coordinate system and
the unit vectors giving the principal direction of the coordinate system $\{B\}$ be: $\hat{X}_B$, $\hat{Y}_B$, $\hat{Z}_B$ and
let $\{A\}$ be a coordinate system (reference coordinate system). The unit vectors giving the principal direction of $\{B\}$ relative to $\{A\}$ are:

$$^{A}\hat{X}_B, {}^{A}\hat{Y}_B \text{ and } {}^{A}\hat{Z}_B$$

The rotation matrix which describes $\{B\}$ relative to $\{A\}$ is defined as:

$$^{A}_{B}R := \begin{bmatrix} ^{A}\hat{X}_B & ^{A}\hat{Y}_B & ^{A}\hat{Z}_B \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = \begin{bmatrix} \hat{X}_B \cdot \hat{X}_A & \hat{Y}_B \cdot \hat{X}_A & \hat{Z}_B \cdot \hat{X}_A \\ \hat{X}_B \cdot \hat{Y}_A & \hat{Y}_B \cdot \hat{Y}_A & \hat{Z}_B \cdot \hat{Y}_A \\ \hat{X}_B \cdot \hat{Z}_A & \hat{Y}_B \cdot \hat{Z}_A & \hat{Z}_B \cdot \hat{Z}_A \end{bmatrix} = \begin{bmatrix} ^{B}\hat{X}_A^T \\ ^{B}\hat{Y}_A^T \\ ^{B}\hat{Z}_A^T \end{bmatrix} = {}^{A}_{B}R^T$$

1.1.2.1 Inverse of Rotation matrix

$$^{A}_{B}R = {}^{B}_{A}R^T = {}^{B}_{A}R^{-1}$$

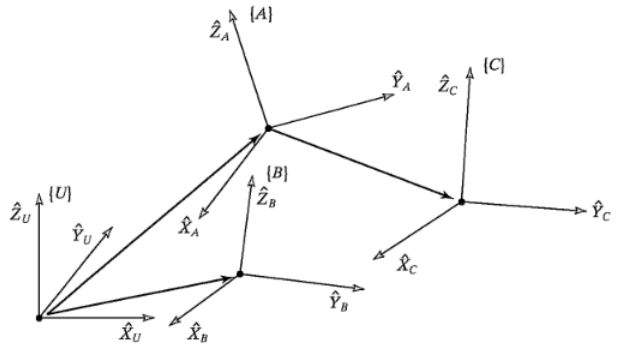
1.1.3 Description of a frame

- is a set of four vectors giving position and orientation information
- is a coordinate system with a position vector which locates its origin relative to some other embedding frame

Let $\{A\}$ be a coordinate system. The frame $\{B\}$ is defined as

$$\{B\} := \{\, ^{A}_{B}R,\ ^{A}P_{BORG}\}$$

where $^{A}_{B}R$ is a rotation matrix and $^{A}P_{BORG}$ is the vector that locates the origin of the frame $\{B\}$



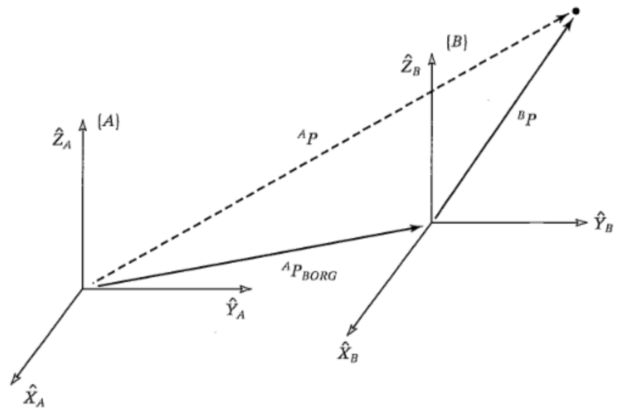
1.2 Mappings

- Changing descriptions from frame to frame

1.2.1 Mappings involving translated frames

Let $\{A\}$ and $\{B\}$ be frames with the same orientation.
Let ${}^{A}P_{BORG}$ be a vector that locates the origin of $\{B\}$ relative to $\{A\}$
and ${}^{B}P$ a vector that locates a point P relative to $\{B\}$.
Then this point relative to $\{A\}$ is defined as:

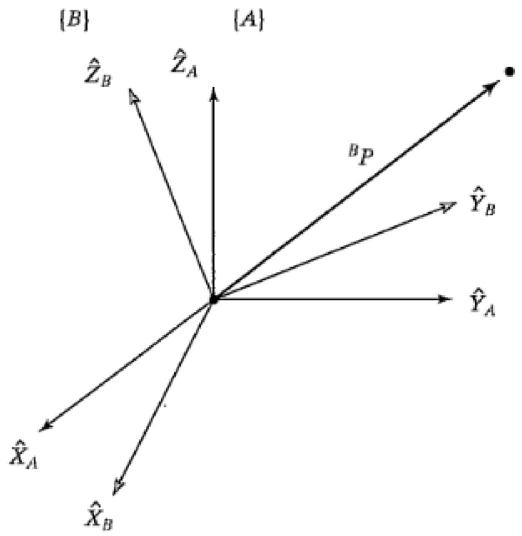
$$ {}^{A}P := {}^{B}P + {}^{A}P_{BORG} $$



1.2.2 Mappings involving rotated frames

Let $\{A\}$ and $\{B\}$ be frames with the same origin and
${}^{A}_{B}R$ a rotation matrix that describes $\{B\}$ relative to $\{A\}$.
Let ${}^{B}P$ be a vector that locates a point P relative to $\{B\}$.
Then this point relative to $\{A\}$ is defined as:

$$ {}^{A}P := {}^{A}_{B}R \cdot {}^{B}P $$



1.2.3 Mappings involving general frames

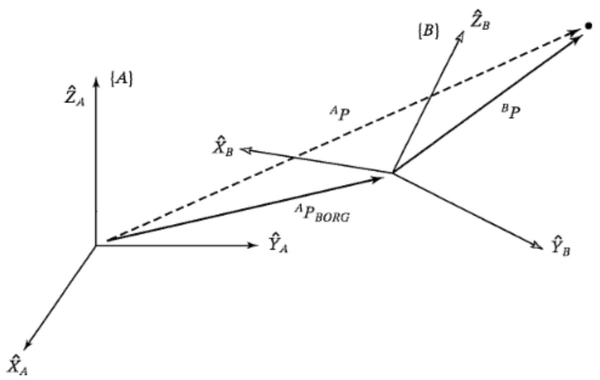
Let $\{A\}$ and $\{B\}$ be frames and
^A_BR a rotation matrix that describes $\{B\}$ relative to $\{A\}$.
Let $^AP_{BORG}$ be a vector that locates the origin of $\{B\}$ relative to $\{A\}$.
Let ^BP be a vector that locates a point P relative to $\{B\}$.
Then this point relative to $\{A\}$ is defined as:

$$^AP := {^A_BR} \cdot {^BP} + {^AP_{BORG}}$$

$$\begin{bmatrix} ^AP \\ 1 \end{bmatrix} = \left[\begin{array}{c|c} ^A_BR & ^AP_{BORG} \\ \hline 0\ 0\ 0 & 1 \end{array}\right] \cdot \begin{bmatrix} ^BP \\ 1 \end{bmatrix}$$

$$^AP = {^A_BT} \cdot {^BP}$$

where $\begin{bmatrix} ^AP \\ 1 \end{bmatrix}$ is a 4×1 position vector and the 4×4 matrix is called a homogeneous transform



1.3 Operators

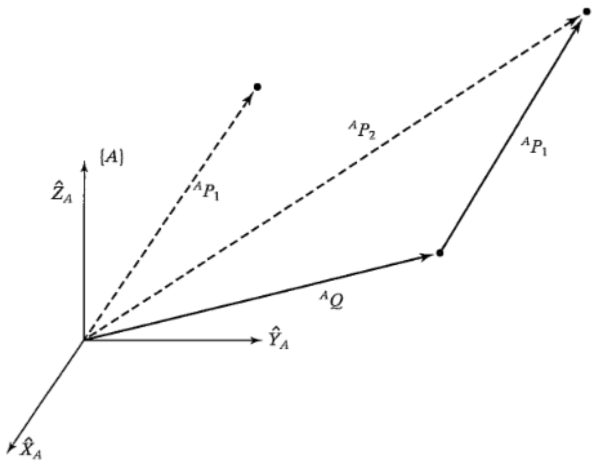
1.3.1 Translational operators

Let $^AP_1$ be a vector that is translated by a vector ^AQ.
The result of the operation is a new vector $^AP_2$ calculated as:

$$^AP_2 = {^AP_1} + {^AQ}$$

$$^AP_2 = D_Q(q) \cdot {^AP_1}$$

where $D_Q(q) = \begin{bmatrix} 1 & 0 & 0 & q_x \\ 0 & 1 & 0 & q_y \\ 0 & 0 & 1 & q_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$ and q_x, q_y and q_z are the components of the translation vector Q and $q = \sqrt{q_x^2 + q_y^2 + q_z^2}$



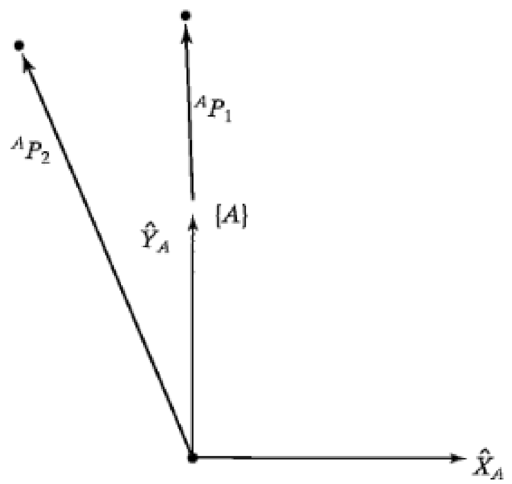
1.3.2 Rotational operators

Let $^AP_1$ be a vector that is rotated by the rotation matrix R.
The result of the operation is a new vector $^AP_2$ calculated as:

$$^AP_2 = R \cdot \ ^AP_1$$

$$^AP_2 = R_K(\theta) \cdot \ ^AP_1$$

where $R_K(\theta)$ is a rotational operator that performs a rotation about the axis direction $\hat{K}$ by θ degrees.



1.3.3 Transformation operators

Let $^AP_1$ be a vector that is rotated and translated by the operator T.
The result of the operation is a new vector $^AP_2$ calculated as:

$$^AP_2 = T \cdot \ ^AP_1$$

where

$$T = \left[\begin{array}{c|c} R_K(\theta) & Q \\ \hline 0\ 0\ 0 & 1 \end{array}\right]$$

and $R_K(\theta)$ is a rotational operator and Q is a translational vector

1.4 Transformation arithmetic

1.4.1 Compound transformations

Let ${}^A_BT$ and ${}^B_CT$ be homogeneous transforms.
Then ${}^A_CT$ is defined as:

$$
{}^A_CT := {}^A_BT \cdot {}^B_CT = \left[\begin{array}{c|c} {}^A_BR \cdot {}^B_CR & {}^A_BR \cdot {}^BP_{CORG} + {}^AP_{BORG} \\ \hline 0\ 0\ 0 & 1 \end{array}\right]
$$

1.4.2 Inverting a transform

Let ${}^A_BT$ be a transform.
Then ${}^B_AT$ is defined as:

$$
{}^B_AT := {}^A_BT^{-1} = \left[\begin{array}{c|c} {}^A_BR^T & -{}^A_BR^T \cdot {}^AP_{BORG} \\ \hline 0\ 0\ 0 & 1 \end{array}\right]
$$

1.5 More on representation of orientation

1.5.1 X-Y-Z fixed angles

Let ${}^A_BR_{XYZ}(\gamma, \beta, \alpha)$ be a rotation matrix that describes a frame $\{B\}$ relative to a frame $\{A\}$ by a rotation about $\hat{X}_A$ by an angle γ, then about $\hat{Y}_A$ by an angle β and finally about $\hat{Z}_A$ by an angle α.
Then ${}^A_BR_{XYZ}(\gamma, \beta, \alpha)$ is defined as:

$$
{}^A_BR_{XYZ}(\gamma, \beta, \alpha) = R_Z(\alpha) \cdot R_Y(\beta) \cdot R_X(\gamma) = \begin{bmatrix} \cos\alpha\cos\beta & \cos\alpha\sin\beta\sin\gamma - \sin\alpha\cos\gamma & \cos\alpha\sin\beta\cos\gamma + \sin\alpha\sin\gamma \\ \sin\alpha\cos\beta & \sin\alpha\sin\beta\sin\gamma + \cos\alpha\cos\gamma & \sin\alpha\sin\beta\cos\gamma - \cos\alpha\sin\gamma \\ -\sin\beta & \cos\beta\sin\gamma & \cos\beta\cos\gamma \end{bmatrix}
$$

Let ${}^A_BR_{XYZ}(\gamma, \beta, \alpha) = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$ be a rotation matrix (same conditions as above).
Then α, β and γ are calculated as:

$$
\beta = \text{atan2}(-r_{31}, \sqrt{r_{11}^2 + r_{21}^2})
$$

$$
\alpha = \text{atan2}(\frac{r_{21}}{\cos\beta}, \frac{r_{11}}{\cos\beta})
$$

$$
\gamma = \text{atan2}(\frac{r_{32}}{\cos\beta}, \frac{r_{33}}{\cos\beta})
$$

1.5.2 Z-Y-X Euler angles

Let ${}^A_BR_{Z'Y'X'}(\alpha, \beta, \gamma)$ be a rotation matrix that describes a frame $\{B\}$ relative to a frame $\{A\}$ by a rotation about $\hat{Z}_B$ by an angle α, then about $\hat{Y}_B$ by an angle β and finally about $\hat{X}_B$ by an angle γ.
Then ${}^A_BR_{Z'Y'X'}(\alpha, \beta, \gamma)$ is defined as:

$$
{}^A_BR_{Z'Y'X'}(\alpha, \beta, \gamma) = R_Z(\alpha) \cdot R_Y(\beta) \cdot R_X(\gamma) = \begin{bmatrix} \cos\alpha\cos\beta & \cos\alpha\sin\beta\sin\gamma - \sin\alpha\cos\gamma & \cos\alpha\sin\beta\cos\gamma + \sin\alpha\sin\gamma \\ \sin\alpha\cos\beta & \sin\alpha\sin\beta\sin\gamma + \cos\alpha\cos\gamma & \sin\alpha\sin\beta\cos\gamma - \cos\alpha\sin\gamma \\ -\sin\beta & \cos\beta\sin\gamma & \cos\beta\cos\gamma \end{bmatrix}
$$

1.5.3 Z-Y-Z fixed angles

Let ${}^A_BR_{Z'Y'Z'}(\alpha, \beta, \gamma)$ be a rotation matrix that describes a frame $\{B\}$ relative to a frame $\{A\}$ by a rotation about $\hat{Z}_B$ by an angle α, then about $\hat{Y}_B$ by an angle β and finally about $\hat{Z}_b$ by an angle γ.
Then ${}^A_BR_{Z'Y'Z'}(\alpha, \beta, \gamma)$ is defined as:

$$
{}^A_BR_{Z'Y'Z'}(\alpha, \beta, \gamma) = \begin{bmatrix} \cos\alpha\cos\beta\cos\gamma - \sin\alpha\sin\gamma & -\cos\alpha\cos\beta\sin\gamma - \sin\alpha\cos\gamma & \cos\alpha\sin\beta \\ \sin\alpha\cos\beta\cos\gamma + \cos\alpha\sin\gamma & -\sin\alpha\cos\beta\sin\gamma + \cos\alpha\cos\gamma & \sin\alpha\sin\beta \\ -\sin\beta\cos\gamma & \sin\beta\sin\gamma & \cos\beta \end{bmatrix}
$$

Let ${}^A_BR_{Z'Y'Z'}(\alpha,\beta,\gamma) = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$ be a rotation matrix (same conditions as above).
Then α, β and γ are calculated as:

$$\beta = \text{atan2}(\sqrt{r_{31}^2 + r_{32}^2}, r_{33})$$
$$\alpha = \text{atan2}(\frac{r_{23}}{\sin\beta}, \frac{r_{13}}{\sin\beta})$$
$$\gamma = \text{atan2}(\frac{r_{32}}{\sin\beta}, -\frac{r_{31}}{\sin\beta})$$

1.5.4 Equivalent angle-axis

1.5.4.1 Same origin

Let $\{A\}$ and $\{B\}$ be two frames with the same origin, where $\{B\}$ is rotated relative to $\{A\}$ about a vector $\hat{K} = \begin{bmatrix} k_x \\ k_y \\ k_z \end{bmatrix}$ by θ degrees . (K goes through the origin)
Then the rotation matrix ${}^A_BR(\hat{K},\theta)$ is defined as:

$${}^A_BR(\hat{K},\theta) = R_K(\theta) = \begin{bmatrix} k_xk_x(1-\cos\theta)+\cos\theta & k_xk_y(1-\cos\theta)-k_z\sin\theta & k_xk_z(1-\cos\theta)+k_y\sin\theta \\ k_xk_y(1-\cos\theta)+k_z\sin\theta & k_yk_y(1-\cos\theta)+\cos\theta & k_yk_z(1-\cos\theta)-k_x\sin\theta \\ k_xk_z(1-\cos\theta)-k_y\sin\theta & k_yk_z(1-\cos\theta)+k_x\sin\theta & k_zk_z(1-\cos\theta)+\cos\theta \end{bmatrix}$$

Let ${}^A_BR_K(\theta) = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$ be a rotation matrix (same conditions as above).
Then θ and $\hat{K}$ are calculated as:

$$\theta = \arccos\left(\frac{r_{11}+r_{22}+r_{33}-1}{2}\right)$$
$$\hat{K} = \frac{1}{2\sin\theta}\begin{bmatrix} r_{32}-r_{23} \\ r_{13}-r_{31} \\ r_{21}-r_{12} \end{bmatrix} \text{, if } \theta \neq 0^\circ \text{ and } \theta \neq 180^\circ$$

1.5.4.2 Different origins

Let $\{A\}$ and $\{B\}$ be two frames, where $\{B\}$ is rotated relative to $\{A\}$ about a vector $\hat{K} = \begin{bmatrix} k_x \\ k_y \\ k_z \end{bmatrix}$ by θ degrees. $\hat{K}$ passes through the point ${}^AP$
Then the translation matrix ${}^A_BR(\hat{K},\theta)$ is defined as:

$${}^A_BR(\hat{K},\theta) = {}^A_{A'}T \cdot {}^{A'}_{B'}T \cdot {}^{B'}_BT$$

where ${}^A_{A'}T = \left[\begin{array}{c|c} I_3 & {}^AP \\ \hline 0\ 0\ 0 & 1 \end{array}\right]$, ${}^B_{B'}T = \left[\begin{array}{c|c} I_3 & -{}^AP \\ \hline 0\ 0\ 0 & 1 \end{array}\right]$
and $\{A'\}$ and $\{B'\}$ are frames with the same origin as $\hat{K}$ and the same orientation as $\{A\}$ and $\{B\}$.

1.5.5 Euler parameters

Let $\hat{K} = \begin{bmatrix} k_x \\ k_y \\ k_z \end{bmatrix}$ be a equivalent axis and θ be the equivalent angle. Then the Euler parameters are given by:

$$\epsilon_1 = k_x \sin\frac{\theta}{2}$$
$$\epsilon_2 = k_y \sin\frac{\theta}{2}$$
$$\epsilon_3 = k_z \sin\frac{\theta}{2}$$
$$\epsilon_4 = \cos\frac{\theta}{2}$$

It follows:

$$\epsilon_1^2 + \epsilon_2^2 + \epsilon_3^2 + \epsilon_4^2 = 1$$

The rotation matrix R_ϵ is defined as:

$$R_\epsilon = \begin{bmatrix} 1 - 2\epsilon_2^2 - 2\epsilon_3^2 & 2(\epsilon_1\epsilon_2 - \epsilon_3\epsilon_4) & 2(\epsilon_1\epsilon_3 + \epsilon_2\epsilon_4) \\ 2(\epsilon_1\epsilon_2 + \epsilon_3\epsilon_4) & 1 - 2\epsilon_1^2 - 2\epsilon_3^2 & 2(\epsilon_2\epsilon_3 - \epsilon_1\epsilon_4) \\ 2(\epsilon_1\epsilon_3 - \epsilon_2\epsilon_4) & 2(\epsilon_2\epsilon_3 + \epsilon_1\epsilon_4) & 1 - 2\epsilon_1^2 - 2\epsilon_2^2 \end{bmatrix}$$

Given a rotation matrix, the equivalent Euler parameters are:

$$\epsilon_1 = \frac{r_{32} - r_{23}}{4\epsilon_4}$$

$$\epsilon_2 = \frac{r_{13} - r_{31}}{4\epsilon_4}$$

$$\epsilon_3 = \frac{r_{21} - r_{12}}{4\epsilon_4}$$

$$\epsilon_4 = \frac{1}{2}\sqrt{1 + r_{11} + r_{22} + r_{33}}$$

1.6 Vectors

1.6.1 Equality

Two vectors are equal if they have the same dimension, magnitude and direction

1.6.2 Equivalence

Two vectors are equivalent in a certain capacity if each produces the very same effect in this capacity

1.6.3 line vector

A line vector refers to a vector that is dependent on its line of action, along with direction and magnitude, for causing effects

1.6.4 Free vector

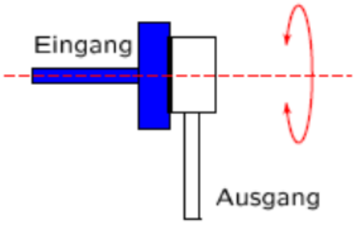
A free vector is a vector that may be positioned anywhere in space without loss or change of meaning, provided that magnitude and direction are preserved

2 Forward Kinematics

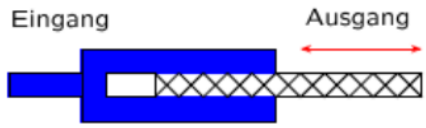
- Science of motion that treats the subject without regard to the forces that cause it.
- compute the position and orientation of the manipulator's end-effector relative to the base of the manipulator as a function of the joint variables

2.1 Link description

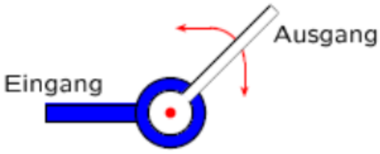
2.1.1 Joints



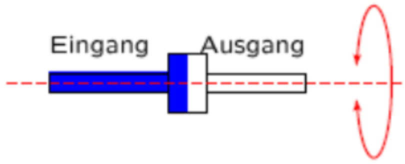
revolving joint



linear joint

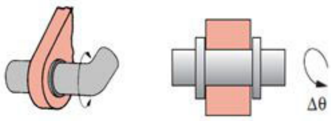


rotational joint

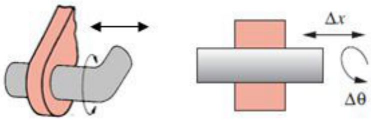


twistingjoint

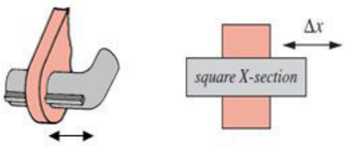
2.1.2 Lower pair joints



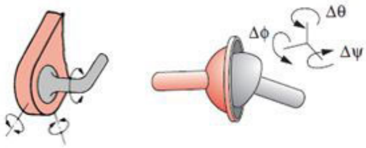
Revolute (R) joint



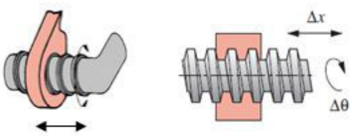
Cylindrical (C) joint



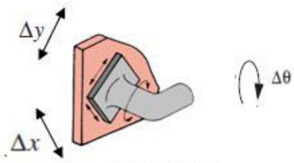
Prismatic (P) joint



Spherical (S) joint



Helical (H) joint
Screw joint



Flat (F) joint
Planar joint

2.1.3 Link

- body connected in a chain by joints

- a link is considered only as a rigid body that defines the relationship between two neighboring joint axes of a manipulator

2.1.4 Numbering of links

- Starting from the base: link 0, link 2, ..., link n
- First moving body is link 1

2.1.5 Link length

Let link $i-1$ be a link with two axes Axis $i-1$ and Axis i.
The measure of distance of a line that is mutually orthogonal to both axes is called link length a_{i-1}

2.1.6 Link twist

Let link $i-1$ be a link with two axes Axis $i-1$ and Axis i and the link length a_{i-1}.
The angle that is measured from axis $i-1$ to axis i in the right-hand sense about a_{i-1} is called link twist α_{i-1}

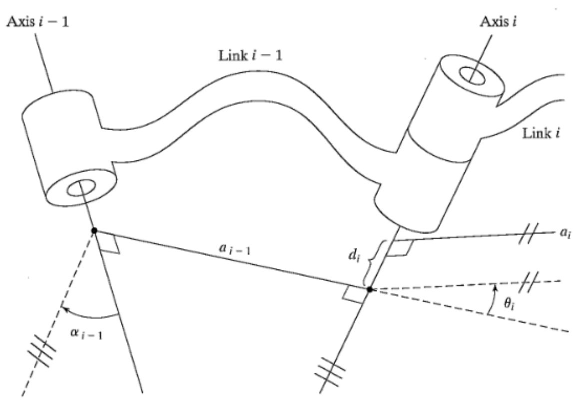
2.2 Link-connection description

2.2.1 Link offset

Let link $i-1$ and link i be two links with the common joint axis Axis i.
The distance along this common axis from the link $i-1$ to the link i is called link offset d_i

2.2.2 Joint angle

Let link $i-1$ and link i be two links with the common joint axis Axis i.
The amount of rotation about this common axis between the link $i-1$ and link i is called joint angle θ_i (right-hand sense)



2.2.3 First and last links in the chain

Let link 0 be the first link and link n be the last link. It follows:

$$a_0 = a_n = 0$$
$$\alpha_0 = \alpha_n = 0$$

2.2.4 Revolute/Rotational Joints

Let joint i be a revolute joint. It follows:
$d_i = 0$ and θ_i is called the joint variable

2.2.5 Prismatic/Linear Joints

Let joint i be a prismatic joint. It follows:
$\theta_i = 0$ and d_i is called the joint variable

2.2.6 Denavit-Hartenberg notation

- Description of any robot kinematically by giving the values of four quantities for each link. Two describing the link itself ($a_0, \dots, a_{n-1}$ and $\alpha_0, \dots, \alpha_{n-1}$) and two describing the link's connection to a neighboring link ($d_0, \dots, d_n$ and $\theta_0, \dots, \theta_i$):

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	α_0	a_0	d_1	θ_1
2	α_1	a_1	d_2	θ_2
		$\vdots$		
n	α_{n-1}	a_{n-1}	d_n	θ_n

2.3 Convention for affixing frames to links

2.3.1 Attach a frame to a link

Let link i be a link with two axes Axis i and Axis $i+1$ and the link length a_i.
The frame $\{i\}$ will be located on the link as follows:

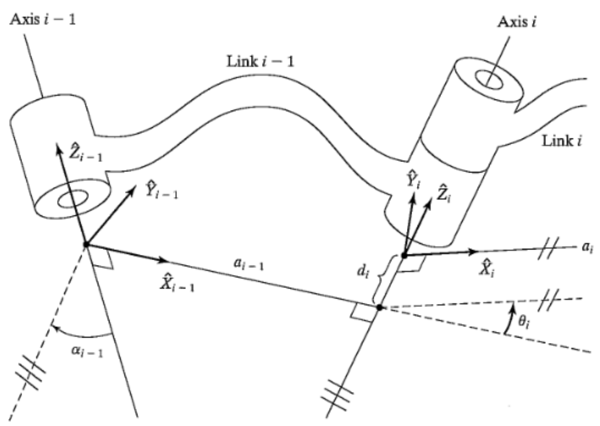
The $\hat{Z}$-axis of frame $\{i\}$ is called $\hat{Z}_i$ and is coincident with the Axis i

The origin of frame $\{i\}$ is located where the a_i orthogonal intersects the Axis i

The $\hat{X}$-axis of frame $\{i\}$ is called $\hat{X}_i$ and points along a_i in the direction from Axis i to Axis $i+1$

The link twist α_i is measured in the right-hand sense about $\hat{X}_i$

The $\hat{Y}$-axis of frame $\{i\}$ is called $\hat{Y}_i$ and is formed by the right-hand rule



2.3.2 Base frame

The frame attached to the base of the robot, or link 0, is called frame $\{0\}$ and doesn't move.

2.3.3 Link parameters in terms of the link frames

$$a_i = \text{the distance from } \hat{Z}_i \text{ to } \hat{Z}_{i+1} \text{ measured along } \hat{X}_i$$
$$\alpha_i = \text{the angle from } \hat{Z}_i \text{ to } \hat{Z}_{i+1} \text{ measured about } \hat{X}_i$$
$$d_i = \text{the distance from } \hat{X}_{i-1} \text{ to } \hat{X}_i \text{ measured along } \hat{Z}_i$$
$$\theta_i = \text{the angle from } \hat{X}_{i-1} \text{ to } \hat{X}_i \text{ measured about } \hat{Z}_i$$

2.3.4 Link-Frame Attachment Procedure

The following is a summary of the procedure to follow when faced with a new mechanism, in order to properly attach the link frames:

1. Identify the joint axes and imagine (or draw) infinite lines along them. steps 2 through 5 below, consider two of these neighboring lines (at axes i and $i+1$).
2. Identify the common perpendicular between them, or point of intersection. At the point of intersection, or at the point where the common perpendicular meets the i-th axis, assign the link-frame origin.
3. Assign the $\hat{Z}_1$ axis pointing along the i-th joint axis.
4. Assign the $\hat{X}_i$ axis pointing along the common perpendicular, or, if the axes intersect, assign $\hat{X}_i$ to be normal to the plane containing the two axes.
5. Assign the $\hat{Y}_i$ axis to complete a right-hand coordinate system.
6. Assign $\{0\}$ to match $\{1\}$ when the first joint variable is zero. For $\{N\}$, choose an origin location and $\hat{X}_N$ direction freely, but generally so as to cause as many linkage parameters as possible to become zero.

2.4 Forward Kinematics

2.4.1 Link transformations

Let ${}^{i-1}_{i}T$ be the transform that defines frame $\{i\}$ relative to frame $\{i-1\}$.
Let $\{P\}$, $\{Q\}$ and $\{R\}$ be frames where
frame $\{R\}$ differs from frame $\{i-1\}$ by a rotation of α_{i-1},
frame $\{Q\}$ differs from frame $\{R\}$ by a translation a_{i-1},
frame $\{P\}$ differs from frame $\{Q\}$ by a rotation θ_i and
frame $\{i\}$ differs from frame $\{P\}$ by a translation d_i.
${}^{i-1}_{i}T$ is calculated as:

$$ {}^{i-1}_{i}T = {}^{i-1}_{R}T \cdot {}^{R}_{Q}T \cdot {}^{Q}_{P}T \cdot {}^{P}_{i}T = R_X(\alpha_{i-1}) \cdot D_X(a_{i-1}) \cdot R_Z(\theta_i) \cdot D_Z(d_i) $$

$$ = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & a_{i-1} \\ \sin\theta_i\cos\alpha_{i-1} & \cos\theta_i\cos\alpha_{i-1} & -\sin\alpha_{i-1} & -\sin\alpha_{i-1}d_i \\ \sin\theta_i\sin\alpha_{i-1} & \cos\theta_i\sin\alpha_{i-1} & \cos\alpha_{i-1} & \cos\alpha_{i-1}d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

2.4.2 Concatenating link transformations

Let ${}^{i-1}_{i}T$ be the transform that defines frame $\{i\}$ relative to frame $\{i-1\}$ and let frame $\{0\}$ be the first frame and $\{N\}$ be the last frame.
${}^{0}_{N}T$ is calculated as:

$$ {}^{0}_{N}T = {}^{0}_{1}T \cdot {}^{1}_{2}T \cdot \ldots \cdot {}^{N-1}_{N}T $$

2.5 Frames With Standard Names

2.5.1 Base Frame {B}

- is located at the base of the manipulator
- Another name for frame $\{0\}$
- Affected to a non-moving part of the robot (sometimes called link 0)

2.5.2 Station Frame {S}

- is located in a task-relevant location (e.g at the corner of a table)
- if the user of this robot is concerned, $\{S\}$ is the universe frame
- Also called: task/world/universe frame
- specified relative to the base frame: ${}^{B}_{S}T$

2.5.3 Wrist Frame {W}

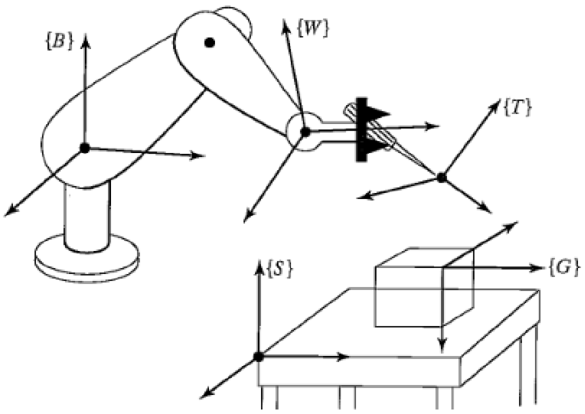
- is affixed to the last link of the manipulator
- Another name for frame $\{N\}$
- Often $\{W\}$ has its origin fixed at a point called the wrist of the manipulator
- specified relative to the base frame: ${}^{B}_{W}T = {}^{0}_{N}T$

2.5.4 Tool Frame {T}

- is affixed to the end of any tool the robot happens to be holding
- When the hand is empty $\{T\}$ is usually located with its origin between the fingertips of the robot
- specified relative to the wrist frame: ${}^{W}_{T}T$

2.5.5 Goal Frame {G}

- is a description of the location to which the robot is to move the tool
- specified relative to the station frame: ${}^{S}_{G}T$



2.6 WHERE is the tool

Let $\{B\}$ be the base frame, $\{S\}$ be the station frame, $\{W\}$ be the wrist frame and $\{T\}$ be the tool frame.
The WHERE function in some robot system computes where the tool is relative to the station. That is computed as:

$${}^{S}_{T}T = {}^{B}_{S}T^{-1} \cdot {}^{B}_{W}T \cdot {}^{W}_{T}T$$

3 Inverse Kinematics

3.1 Solvability

- All Systems with revolute and prismatic joints having a total of six degrees of freedom in a single series chain are solvable

3.1.1 One Solution

- If the desired position and orientation of the wrist frame is in the workspace, then at least one solution exists.

3.1.2 Multiple Solutions

- The chosen solution depends on several criteria e.g:
  - Nearest solution
  - Obstacles
  - Link lengths and weights
- Number of solutions of a manipulator with six rotational joints:

a_i	Number of solutions
$a_1 = a_3 = a_5 = 0$	≤ 4
$a_3 = a_5 = 0$	≤ 8
$a_3 = 0$	≤ 16
all $a_i \neq 0$	≤ 16

3.1.3 Methods of Solution

- **numerical solutions:** generally much slower than closed-form solution
- **closed-form solutions:**
  - based on analytic expressions or on the solution of a polynomial of degree 4 or less
  - Distinction between algebraic and geometric solution

3.2 The Notion of Manipulator Subspace when $n < 6$

Let ${}^B_W T$ be the transformation matrix of the wrist frame relative to the base frame of a n-degree-of-freedom manipulator, where $n < 6$. The reachable workspace can be thought of as a portion of an n-degree-of-freedom subspace given by:

$${}^B_W T$$

3.2.1 Specifying a general Goal for a manipulator when $n < 6$

Let a general goal frame ${}^S_G T$ be given

1. Compute a modified goal frame ${}^S_{G'} T$, such that ${}^S_{G'}$ lies in the manipulator's subspace and is as near to ${}^S_G T$ as possible
2. Compute the inverse kinematics to find joint angles using ${}^S_{G'}$ as the desired goal

3.3 Algebraic solution

Let ${}^B_W T$ be the Transformation matrix of a 3R planar manipulator with

$${}^B_W T = {}^0_3 T = \begin{bmatrix} c_{123} & -s_{123} & 0 & l_1 c_1 + l_2 c_{12} \\ s_{123} & c_{123} & 0 & l_1 s_1 + l_2 s_{12} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

It follows:

$$\theta_2 = \operatorname{atan2}(s_2, c_2)$$

$$\theta_1 = \operatorname{atan2}(y, x) - \operatorname{atan2}(k_2, k_1)$$

$$\theta_1 + \theta_2 + \theta_3 = \operatorname{atan2}(s_{123}, c_{123})$$

, where

$$x = k_1 c_1 - k_2 s_1$$

$$y = k_1 s_1 + k_2 c_1$$

$$k_1 = l_1 + l_2 c_2$$

$$k_2 = l_2 s_2$$

3.4 Geometric solution

Let ${}^B_WT$ be the Transformation matrix of a 3R planar manipulator with
${}^B_WT = {}^0_3T = \begin{bmatrix} c_\phi & -s_\phi & 0 & x \\ s_\phi & c_\phi & 0 & y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
It follows:

$$\cos\theta_2 = \frac{x^2+y^2-l_1^2-l_2^2}{2l_1l_2}$$
$$\theta_2' = -\theta_2$$
$$\theta_1 = \operatorname{atan2}(y,x) \pm \psi$$
$$\theta_1+\theta_2+\theta_3 = \operatorname{atan2}(s_\phi, c_\phi)$$

,where

$$\cos\psi = \frac{x^2+y^2+l_1^2-l_2^2}{2l_1\sqrt{x^2+y^2}}$$

3.5 Algebraic Solution by Reduction to Polynomial

Make the following substitution:

$$u = \tan\frac{\theta}{2}$$
$$\cos\theta = \frac{1-u^2}{1+u^2}$$
$$\sin\theta = \frac{2u}{1+u^2}$$

This substitution convert transcendental equations into polynomial equations in u

Let the following transcendental equation be given:
$a\cdot\cos\theta + b\cdot\sin\theta = c$
θ is calculated as:

$$\theta = \operatorname{atan2}(b,a) \pm \operatorname{atan2}(\sqrt{a^2+b^2-c^2}, c)$$

3.6 Pieper's solution when three axes intersect

Let a manipulator be given with the following properties:

- 6 degree of freedom
- all six joints are revolute
- the last three axes intersect

Let the origins of the link frames $\{4\}, \{5\}$ and $\{6\}$ be:
${}^0P_{4ORG} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$
Let further definitions be given as:
$f_1 = a_3c_3 + d_4s\alpha_3s_3 + a_2$
$f_2 = a_3c\alpha_2s_3 - d_4s\alpha_3c\alpha_2c_3 - d_4s\alpha_2c\alpha_3 - d_3s\alpha_2$
$f_3 = a_3s\alpha_2s_3 - d_4s\alpha_3s\alpha_2c_3 + d_4c\alpha_2c\alpha_3 + d_3c\alpha_2$
$k_1 = f_1$
$k_2 = -f_2$
$k_3 = f_1^2 + f_2^2 + f_3^2 + a_1^2 + d_2^2 + 2d_2f_3$
$k_4 = f_3c\alpha_1 + d_2c\alpha_1$
and $r := x^2+y^2+z^2$
It follows:

$$r = (k_1c_2 + k_2s_2)2a_1 + k_3$$
$$z = (k_1s_2 - k_2c_2)s\alpha_1 + k_4$$

θ_3 can be computed as follows:

- If $a_1 = 0 \implies r = k_3$. Use substitution of section 3.5
- If $s\alpha_1 = 0 \implies z = k_4$. Use substitution of section 3.5
- Otherwise: $\frac{(r-k_3)^2}{4a_1^2} + \frac{(z-k_4)^2}{s^2\alpha_1} = k_1^2 + k_2^2$

θ_2 can be computed using:

$$r = (k_1c_2 + k_2s_2)2a_1 + k_3$$
$$z = (k_1s_2 - k_2c_2)s\alpha_1 + k_4$$

θ_1 can be computed using:

$$^0P_{4ORG} = \begin{bmatrix} c_1g_1 - s_1g_2 \\ s_1g_1 + c_1g_2 \\ g_3 \end{bmatrix}$$

, where
$g_1 = c_2f_1 - s_2f_2 + a_1$
$g_2 = s_2c\alpha_1f_1 + c_2c\alpha_1f_2 - s\alpha_1f_3 - d_2s\alpha_1$
$g_3 = s_2s\alpha_1f_1 + c_2s\alpha_1f_2 + c\alpha_1f_3 + d_2c\alpha_1$
and

$$^4_6R|_{\theta_4=0} = \ ^0_4R^{-1}|_{\theta_4=0} \cdot \ ^0_6R$$

3.7 Solve-ing a manipulator

The SOLVE function implements Cartesian transformations and calls the inverse kinematic function. It calculates the location of $\{W\}$ relative to $\{B\}$:

$$^B_WT = \ ^B_ST \cdot \ ^S_TT \cdot \ ^W_TT^{-1}$$

Then, the inverse kinematics take ^B_WT as an input and calculate θ_1 through θ_n

4 Jacobians: velocities and static forces

4.1 Notation for time-varying Position and Orientation

4.1.1 Differentiation of position vectors

Let ^BQ be a position vector relative to frame $\{B\}$.
The velocity of Q relative to frame $\{B\}$ is computed as the derivative of ^BQ w.r.t. time and defined as:

$$^BV_Q = \frac{d}{dt} \ ^BQ = \lim_{\Delta t \to 0} \frac{^BQ(t+\Delta t) - \ ^BQ(t)}{\Delta t}$$

Let ^BQ be a position vector relative to $\{B\}$
The velocity vector of ^BQ expressed in terms of frame $\{A\}$ is calculated as:

$$^A(^BV_Q) = \ ^A_BR \cdot \ ^BV_Q = \frac{^Ad}{dt} \cdot \ ^BQ$$

, where the velocity (done by the derivation) is relative to frame $\{B\}$ but the velocity vector itself is expressed in terms of frame $\{A\}$ and

$$^B(^BV_Q) = \ ^BV_Q$$

Let ^UCORG be the origin of frame $\{C\}$ relative to frame $\{U\}$
The velocity of ^UCORG relative to the universe frame $\{U\}$ is defined as:

$$v_C = \ ^UV_{CORG}$$

and the velocity vector of ^UCORG expressed int terms of frame $\{A\}$ (though differentiation done relative to $\{U\}$) is defined as:

$$^Av_C = \ ^A_UR \cdot v_C$$

4.1.2 Angular velocity vector

Let $\{A\}$ and $\{B\}$ be two frames.
The angular velocity of a rotation of frame $\{B\}$ relative to $\{A\}$ is defined as:

$$^{A}\Omega_{B}$$

, where the direction of $^{A}\Omega_{B}$ indicates the axis of rotation
and the magnitude of $^{A}\Omega_{B}$ indicates the speed of rotation

Let $^{A}\Omega_{B}$ be an angular velocity vector
$^{A}\Omega_{B}$ expressed in terms of a frame $\{C\}$ is defined as:

$$^{C}(^{A}\Omega_{B})$$

, where the rotation of frame $\{B\}$ is relative to frame $\{A\}$ but the angular velocity vector itself is expressed in terms of frame $\{C\}$

Let $\{C\}$ be a frame and $\{U\}$ be the universe frame
The angular velocity of a rotation of frame $\{C\}$ relative to $\{U\}$ is defined as:

$$\omega_C = \, ^{U}\Omega_{C}$$

and the angular velocity vector of $\{C\}$ relative to $\{U\}$ expressed in terms of frame $\{A\}$ is defined as:

$$^{A}\omega_{C}$$

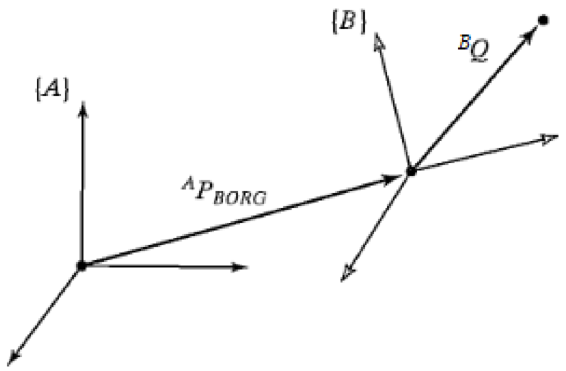
4.2 Linear and Rotational Velocity of rigid bodies

4.2.1 Linear Velocity

Let $\{A\}$ and $\{B\}$ be two frames
and ^{B}Q a position vector relative to $\{B\}$ which may change with time
Let the location of $\{B\}$ relative to $\{A\}$ be described by a position vector $^{A}P_{BORG}$ which may change with time
and a rotation matrix $^{A}_{B}R$ which is not changing with time
The velocity of Q relative to $\{A\}$ is computed as:

$$^{A}V_{Q} = \, ^{A}V_{BORG} + \, ^{A}_{B}R \cdot \, ^{B}V_{Q}$$

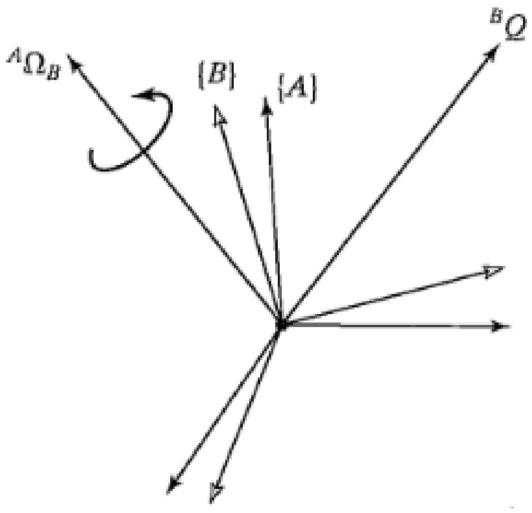
, where $^{A}V_{BORG}$ is the linear velocity of frame $\{B\}$ relative to $\{A\}$



4.2.2 Rotational velocity

Let $\{A\}$ and $\{B\}$ be two frames with the same origins and with zero linear relative velocity
Let ^{B}Q be a position vector relative to $\{B\}$ which may change with time
Let $^{A}\Omega_{B}$ be a vector describing the rotational velocity of $\{B\}$ relative to $\{A\}$
Let $^{A}_{B}R$ a rotation matrix describing $\{B\}$ relative to $\{A\}$
The velocity of Q relative to $\{A\}$ is computed as:

$$^{A}V_{Q} = \, ^{A}_{B}R \cdot \, ^{B}V_{Q} + \, ^{A}\Omega_{B} \times \, ^{A}_{B}R \cdot \, ^{B}Q$$



4.2.3 Simultaneous linear and rotational velocity

Let $\{A\}$ and $\{B\}$ be two frames
Let ${}^{B}Q$ be a position vector relative to $\{B\}$ which may change with time
Let the location of $\{B\}$ relative to $\{A\}$ be described by a position vector ${}^{A}P_{BORG}$ which may change with time
Let ${}^{A}\Omega_B$ be a vector describing the rotational velocity of $\{B\}$ relative to $\{A\}$
Let ${}^{A}_{B}R$ a rotation matrix describing $\{B\}$ relative to $\{A\}$
The velocity of Q relative to $\{A\}$ is computed as:

$$ {}^{A}V_Q = {}^{A}V_{BORG} + {}^{A}_{B}R \cdot {}^{B}V_Q + {}^{A}\Omega_B \times {}^{A}_{B}R \cdot {}^{B}Q $$

4.3 More on Angular Velocity

4.3.1 Other representations of angular velocity

4.3.1.1 Z-Y-Z Euler angles

$$ \dot{\Theta}_{Z'Y'Z'} = \begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix} $$

4.4 Velocity "Propagation" from link to link

4.4.1 Rotational joints

Let joint $i+1$ be rotational:

Let ${}^{i+1}_{i}R$ a rotation matrix describing $\{i\}$ relative to $\{i+1\}$
Let ${}^{i}\omega_i$ be the angular velocity vector of link $\{i\}$ relative to $\{B\}$ expressed in terms of frame $\{i\}$
The angular velocity of link $i+1$ with respect to frame $\{i+1\}$ is calculated as:

$$ {}^{i+1}\omega_{i+1} = {}^{i+1}_{i}R \cdot {}^{i}\omega_i + \dot{\theta}_{i+1} \cdot {}^{i+1}\hat{Z}_{i+1} $$

, where $\dot{\theta}_{i+1} \cdot {}^{i+1}\hat{Z}_{i+1} = {}^{i+1}\begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_{i+1} \end{bmatrix}$

Let ${}^{i+1}_{i}R$ a rotation matrix describing $\{i\}$ relative to $\{i+1\}$
Let ${}^{i}v_i$ be the linear velocity of the origin of link $\{i\}$ relative to $\{B\}$ expressed in terms of frame $\{i\}$
Let ${}^{i}\omega_i$ be the angular velocity vector of link $\{i\}$ relative to $\{B\}$ expressed in terms of frame $\{i\}$
Let ${}^{i}P_{i+1}$ be the origin of $\{i+1\}$ relative to $\{i\}$ The linear velocity of link $i+1$ with respect to frame $\{i+1\}$ is calculated as:

$$ {}^{i+1}v_{i+1} = {}^{i+1}_{i}R({}^{i}v_i + {}^{i}\omega_i \times {}^{i}P_{i+1}) $$

4.4.2 Prismatic joints

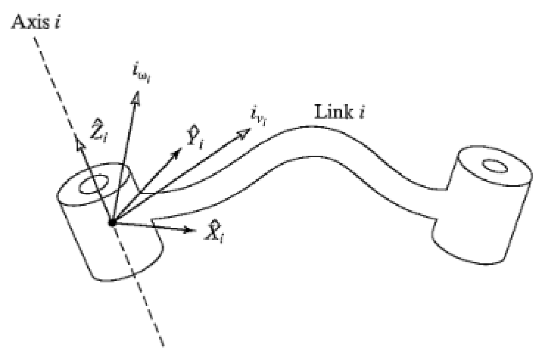
Let joint $i+1$ be prismatic:

Let ${}^{i+1}_{i}R$ a rotation matrix describing $\{i\}$ relative to $\{i+1\}$
Let ${}^{i}\omega_i$ be the angular velocity vector of link $\{i\}$ relative to $\{B\}$ expressed in terms of frame $\{i\}$
The angular velocity of link $i+1$ with respect to frame $\{i+1\}$ is calculated as:

$$^{i+1}\omega_{i+1} = {}^{i+1}_{i}R \cdot {}^{i}\omega_i$$

Let ${}^{i+1}_{i}R$ a rotation matrix describing $\{i\}$ relative to $\{i+1\}$
Let ${}^{i}v_i$ be the linear velocity of the origin of link $\{i\}$ relative to $\{B\}$ expressed in terms of frame $\{i\}$
Let ${}^{i}\omega_i$ be the angular velocity vector of link $\{i\}$ relative to $\{B\}$ expressed in terms of frame $\{i\}$
Let ${}^{i}P_{i+1}$ be the origin of $\{i+1\}$ relative to $\{i\}$ The linear velocity of link $i+1$ with respect to frame $\{i+1\}$ is calculated as:

$$^{i+1}v_{i+1} = {}^{i+1}_{i}R({}^{i}v_i + {}^{i}\omega_i \times {}^{i}P_{i+1}) + \dot{d}_{i+1} \cdot {}^{i+1}\hat{Z}_{i+1}$$



4.5 Jacobians

4.5.1 Jacobian-Matrix

$$\mathcal{J}(\Theta) = \begin{bmatrix} \frac{\partial f_1}{\partial \theta_1} & \cdots & \frac{\partial f_1}{\partial \theta_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial \theta_1} & \cdots & \frac{\partial f_m}{\partial \theta_n} \end{bmatrix}$$

, with $\Theta = \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_n \end{bmatrix}$

4.5.2 Relate Joint Velocities to Cartesian Velocities

$$^{i}\nu = \begin{bmatrix} ^{i}v \\ ^{i}\omega \end{bmatrix} = {}^{i}\mathcal{J}(\Theta) \cdot \dot{\Theta}$$

, where ${}^{i}\nu$ is a vector of Cartesian velocities (here: of the end-effector frame relative to $\{B\}$): A linear velocity vector expressed in terms of $\{i\}$ and a rotational velocity vector expressed in $\{i\}$. and

$\Theta = \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_n \end{bmatrix}$

Dimension of the Jacobian Matrix:

- Number of rows $\hat{=}$ Number of degrees of freedom in the Cartesian space ($\times 2$ for velocity and angular velocity)
- Number of columns $\hat{=}$ Number of joints of the manipulator

4.5.3 Changing a Jacobian's frame of reference

Let ${}^{B}\mathcal{J}(\Theta)$ be a Jacobian written in frame $\{B\}$ (with ${}^{B}\nu = {}^{B}\mathcal{J}(\Theta)\dot{\Theta}$)
The Jacobian written in $\{A\}$ is computed as:

$$ {}^{A}\mathcal{J}(\Theta) = \left[\begin{array}{c|c} {}^{A}_{B}R & 0 \\ \hline 0 & {}^{A}_{B}R \end{array}\right] \cdot {}^{B}\mathcal{J}(\Theta) $$

4.5.3.1 Example

Let ${}^{n}v_n = \begin{bmatrix} \alpha_{11}\dot{\theta}_1 + & \dots & +\alpha_{1n}\dot{\theta}_n \\ \alpha_{21}\dot{\theta}_1 + & \dots & +\alpha_{2n}\dot{\theta}_n \\ \alpha_{31}\dot{\theta}_1 + & \dots & +\alpha_{3n}\dot{\theta}_n \end{bmatrix}$ be a velocity vector of frame $\{n\}$ relative to $\{B\}$ expressed in $\{n\}$ and
let ${}^{n}\omega_n = \begin{bmatrix} \beta_{11}\dot{\theta}_1 + & \dots & +\beta_{1n}\dot{\theta}_n \\ \beta_{21}\dot{\theta}_1 + & \dots & +\beta_{2n}\dot{\theta}_n \\ \beta_{31}\dot{\theta}_1 + & \dots & +\beta_{3n}\dot{\theta}_n \end{bmatrix}$ be an angular velocity vector of frame $\{n\}$ relative to $\{B\}$ expressed in $\{n\}$
and
let $\dot{\Theta} = \begin{bmatrix} \dot{\theta}_1 \\ \vdots \\ \dot{\theta}_n \end{bmatrix}$
It follows:

$$ {}^{n}\nu = \begin{bmatrix} {}^{n}v \\ {}^{n}\omega \end{bmatrix} = {}^{n}\mathcal{J}(\Theta) \cdot \dot{\Theta} = \begin{bmatrix} \alpha_{11} & \dots & \alpha_{1n} \\ \alpha_{21} & \dots & \alpha_{2n} \\ \alpha_{31} & \dots & \alpha_{3n} \\ \beta_{11} & \dots & \beta_{1n} \\ \beta_{21} & \dots & \beta_{2n} \\ \beta_{31} & \dots & \beta_{3n} \end{bmatrix} \cdot \begin{bmatrix} \dot{\theta}_1 \\ \vdots \\ \dot{\theta}_n \end{bmatrix} $$

4.6 Singularity

4.6.1 Categories

- **Workspace-boundary singularities** occur when the manipulator is fully stretched out or folded back on itself in such a way that the end-effector is at or very near the boundary of the workspace
- **Workspace interior singularities** occur away from the workspace boundary; they generally are caused by a lining up of two or more joint axes

4.6.2 Calculation

Let $\mathcal{J}(\Theta)$ be a Jacobian (with $v = \mathcal{J}(\Theta)\dot{\Theta}$)
The Jacobian is singular (a singularity of the mechanism exists) if:

$$ \det[\mathcal{J}(\Theta)] = 0 $$

4.6.3 Calculate joint rates from given Cartesian velocities

Let $\mathcal{J}(\Theta)$ be a Jacobian (with $\nu = \mathcal{J}(\Theta)\dot{\Theta}$) and
let ν be a vector of Cartesian velocities: A linear velocity vector and a rotational velocity vector and
The joint rates $\dot{\theta}_1, \dots, \dot{\theta}_n$ are calculated as:

$$ \dot{\Theta} = \mathcal{J}^{-1}(\Theta)\nu $$

, where $\dot{\Theta} = \begin{bmatrix} \dot{\theta}_1 \\ \vdots \\ \dot{\theta}_n \end{bmatrix}$

4.7 Static Forces in Manipulators

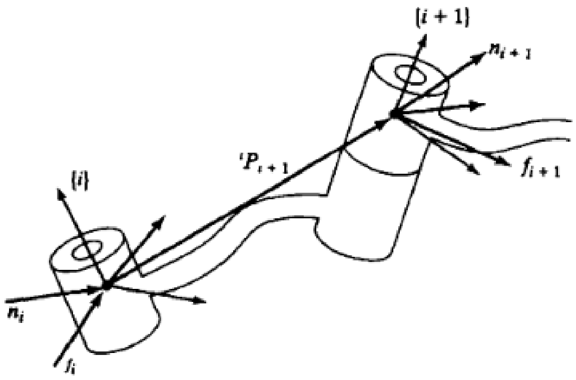
Let ${}^{i+1}_{i}R$ be a rotation matrix describing $\{i\}$ relative to $\{i+1\}$ and
let ${}^{i+1}f_{i+1}$ be a force exerted on link $i+1$ by link i relative to link $i+1$

The force exerted on link i by link $i-1$ relative to link i is defined as:

$$^{i}f_i = {}^{i}_{i+1}R \cdot {}^{i+1}f_{i+1}$$

Let $^{i+1}_{i}R$ be a rotation matrix describing $\{i\}$ relative to $\{i+1\}$ and
let $^{i+1}n_{i+1}$ be a torque exerted on link $i+1$ by link i relative to link $i+1$ and
let $^{i}P_{i+1}$ be the origin of frame $\{i+1\}$ relative to $\{i\}$
The torque exerted on link i by link $i-1$ relative to link i is defined as:

$$^{i}n_i = {}^{i}_{i+1}R \cdot {}^{i+1}n_{i+1} + {}^{i}P_{i+1} \times {}^{i}f_i$$



4.7.1 Calculate the joint Torque required to maintain the static equilibrium

4.7.1.1 Rotational joints:

Let $^{i}n_i$ be the torque exerted on link i by link $i-1$ relative to link i and
let $^{i}\hat{Z}_i$ be the joint axis vector.
The joint torque required to maintain the static equilibrium is computed as:

$$\tau_i = {}^{i}n_i^T \cdot {}^{i}\hat{Z}_i$$

4.7.1.2 Prismatic joints:

Let $^{i}f_i$ be the force exerted on link i by link $i-1$ relative to link i and
let $^{i}\hat{Z}_i$ be the joint axis vector.
The joint torque required to maintain the static equilibrium is computed as:

$$\tau_i = {}^{i}f_i^T \cdot {}^{i}\hat{Z}_i$$

4.8 Jacobians in the Force Domain

Let $\mathcal{J}$ be a Jacobian (with $\nu = \mathcal{J}(\Theta)\dot{\Theta}$) expressed in terms of $\{0\}$ and
let $\mathcal{F}$ be a 6×1 Cartesian force-moment vector acting at the end-effector expressed in terms of $\{0\}$ ($\mathcal{F} = [F, N]^T$ and F is the force acting on the center of the mass of the link and N is the moment acting on the center of the mass of the link)
The 6×1 vector of torques at the joints is calculated as:

$$\tau = {}^{0}\mathcal{J}^T \cdot {}^{0}\mathcal{F}$$

4.9 Cartesian Transformation of Velocities and static Forces

Let $^{B}_{A}R$ a rotation matrix describing $\{A\}$ relative to $\{B\}$
Let $^{A}v_A$ be the linear velocity of the origin of $\{A\}$ relative to $\{U\}$ expressed in terms of frame $\{A\}$
Let $^{A}\omega_A$ be the angular velocity vector of $\{A\}$ relative to $\{U\}$ expressed in terms of frame $\{A\}$
Let $^{A}P_{BORG}$ be the origin of $\{B\}$ relative to $\{A\}$
The vector of Cartesian velocities of B with respect to frame $\{B\}$ is calculated as:

$$^{B}\nu_B = \begin{bmatrix} ^{B}v_B \\ ^{B}\omega_B \end{bmatrix} = \begin{bmatrix} ^{B}_{A}R & -^{B}_{A}R \cdot {}^{A}P_{BORG}\times \\ 0 & ^{B}_{A}R \end{bmatrix} \begin{bmatrix} ^{A}v_A \\ ^{A}\omega_A \end{bmatrix} = {}^{B}_{A}T_v \cdot {}^{A}\nu_A$$

, where $P\times = \begin{bmatrix} 0 & -p_z & p_y \\ p_z & 0 & -p_x \\ -p_y & p_x & 0 \end{bmatrix}$ and T_v is called a velocity transformation

and Let ${}^A_BR$ a rotation matrix describing $\{B\}$ relative to $\{A\}$
It follows:

$$^A\nu_A = \begin{bmatrix} ^Av_A \\ ^A\omega_A \end{bmatrix} = \begin{bmatrix} ^A_BR & ^AP_{BORG} \times \, ^A_BR \\ 0 & ^A_BR \end{bmatrix} \begin{bmatrix} ^Bv_B \\ ^B\omega_B \end{bmatrix} = \, ^A_BT_v \cdot \, ^B\nu_B$$

and let ^BF_B be a force vector and
let ^BN_B be a moment vector
$^A\mathcal{F}_A = \begin{bmatrix} ^AF_A \\ ^AN_A \end{bmatrix}$ is computed as:

$$^A\mathcal{F}_A = \begin{bmatrix} ^AF_A \\ ^AN_A \end{bmatrix} = \begin{bmatrix} ^A_BR & 0 \\ ^AP_{BORG} \times ^A_B R & ^A_BR \end{bmatrix} \begin{bmatrix} ^BF_B \\ ^BN_B \end{bmatrix} = \, ^A_BT_f \cdot \, ^F\mathcal{F}_B$$

, where T_f is called a force-moment transformation

5 Dynamics

5.1 Acceleration of a rigid body

Let ^BV_Q the velocity of a point Q relative to $\{B\}$
The linear acceleration of Q relative to $\{B\}$ is defined as:

$$^B\dot{V}_Q = \frac{d}{dt} \, ^BV_Q = \lim_{\Delta t \to 0} \frac{^BV_Q(t+\Delta t) - \, ^BV_Q(t)}{\Delta t}$$

Let $^A\Omega_B$ be the angular velocity of a rotation of frame $\{B\}$ relative to $\{A\}$
The angular acceleration of $\{B\}$ relative to $\{A\}$ is defined as:

$$^A\dot{\Omega}_B = \frac{d}{dt} \, ^A\Omega_B = \lim_{\Delta t \to 0} \frac{^A\Omega_B(t+\Delta t) - \, ^A\Omega_B(t)}{\Delta t}$$

Let ^UAORG be the origin of frame $\{A\}$ relative to frame $\{U\}$
The acceleration of ^UAORG relative to the universe frame $\{U\}$ is defined as:

$$\dot{v}_A = \, ^U\dot{V}_{AORG}$$

Let $\{A\}$ be a frame and $\{U\}$ be the universe frame
The angular acceleration of a rotation of frame $\{A\}$ relative to $\{U\}$ is defined as:

$$\dot{\omega}_A = \, ^U\dot{\Omega}_A$$

5.1.1 Linear acceleration

Let ^BQ be a position vector relative to frame $\{B\}$ and
let $^B\dot{V}_Q$ be the linear acceleration of Q relative to $\{B\}$ and
let ^BV_Q be the linear velocity of Q relative to $\{B\}$ and
let $^A\dot{\Omega}_B$ be a vector describing the angular acceleration of $\{B\}$ relative to $\{A\}$ and
let $^A\Omega_B$ be a vector describing the angular velocity of $\{B\}$ relative to $\{A\}$
The linear acceleration of a position vector Q relative to $\{A\}$ is calculated as:

$$^A\dot{V}_Q = \, ^A\dot{V}_{BORG} + \, ^A_BR \cdot \, ^B\dot{V}_Q + 2 \cdot \, ^A\Omega_B \times \, ^A_BR \cdot \, ^BV_Q + \, ^A\dot{\Omega}_B \times \, ^A_BR \cdot \, ^BQ + \, ^A\Omega_B \times (^A\Omega_B \times \, ^A_BR \cdot \, ^BQ)$$

in the case of $^BV_Q = ^B\dot{V}_Q = 0$:

$$^A\dot{V}_Q = \, ^A\dot{V}_{BORG} + \, ^A\dot{\Omega}_B \times \, ^A_BR \cdot \, ^BQ + \, ^A\Omega_B \times (^A\Omega_B \times \, ^A_BR \cdot \, ^BQ)$$

5.1.2 Angular acceleration

Let ${}^{A}\Omega_B$ be a vector describing the angular velocity of $\{B\}$ relative to $\{A\}$ and
let ${}^{A}\dot{\Omega}_B$ be the vector describing the angular acceleration of $\{B\}$ relative to $\{A\}$ and
let ${}^{B}\Omega_C$ be a vector describing the angular velocity of $\{C\}$ relative to $\{B\}$ and
let ${}^{B}\dot{\Omega}_C$ be the vector describing the angular acceleration of $\{C\}$ relative to $\{B\}$
The angular acceleration of a rotation of $\{C\}$ relative to $\{A\}$ is defined as:

$$ {}^{A}\dot{\Omega}_C = {}^{A}\dot{\Omega}_B + {}^{A}_{B}R \cdot {}^{B}\dot{\Omega}_C + {}^{A}\Omega_B \times {}^{A}_{B}R \cdot {}^{B}\Omega_C $$

5.2 Acceleration from link to link

5.2.1 Rotational joints

Let joint $i+1$ be rotational:

Let ${}^{i}\omega_i$ be the angular velocity vector of link $\{i\}$ relative to $\{B\}$ expressed in terms of frame $\{i\}$
Let ${}^{i}\dot{\omega}_i$ be the angular acceleration vector of link $\{i\}$ relative to $\{B\}$ expressed in terms of frame $\{i\}$
The angular acceleration of link $i+1$ with respect to frame $\{i+1\}$ is calculated as:

$$ {}^{i+1}\dot{\omega}_{i+1} = {}^{i+1}_{i}R \cdot {}^{i}\dot{\omega}_i + {}^{i+1}_{i}R \cdot {}^{i}\omega_i \times \dot{\theta}_{i+1} \cdot {}^{i+1}\hat{Z}_{i+1} + \ddot{\theta}_{i+1} \cdot {}^{i+1}\hat{Z}_{i+1} $$

, where $\ddot{\theta}_{i+1} \cdot^{i+1} \hat{Z}_{i+1} = {}^{i+1}\begin{bmatrix} 0 \\ 0 \\ \ddot{\theta}_{i+1} \end{bmatrix}$

Let ${}^{i}\dot{v}_i$ be the linear acceleration of the origin of link $\{i\}$ relative to $\{B\}$ expressed in terms of frame $\{i\}$
Let ${}^{i}\omega_i$ be the angular velocity vector of link $\{i\}$ relative to $\{B\}$ expressed in terms of frame $\{i\}$
Let ${}^{i}\dot{\omega}_i$ be the angular acceleration vector of link $\{i\}$ relative to $\{B\}$ expressed in terms of frame $\{i\}$
Let ${}^{i}P_{i+1}$ be the origin of $\{i+1\}$ relative to $\{i\}$ The linear acceleration of link $i+1$ with respect to frame $\{i+1\}$ is calculated as:

$$ {}^{i+1}\dot{v}_{i+1} = {}^{i+1}_{i}R({}^{i}\dot{\omega}_i \times {}^{i}P_{i+1} + {}^{i}\omega_i \times ({}^{i}\omega_i \times {}^{i}P_{i+1}) +^{i} \dot{v}_i) $$

5.2.2 Prismatic joints

Let joint $i+1$ be prismatic:

Let ${}^{i}\dot{\omega}_i$ be the angular acceleration vector of link $\{i\}$ relative to $\{B\}$ expressed in terms of frame $\{i\}$
The angular acceleration of link $i+1$ with respect to frame $\{i+1\}$ is calculated as:

$$ {}^{i+1}\dot{\omega}_{i+1} = {}^{i+1}_{i}R \cdot {}^{i}\dot{\omega}_i $$

Let ${}^{i}v_i$ be the linear velocity of the origin of link $\{i\}$ relative to $\{B\}$ expressed in terms of frame $\{i\}$
Let ${}^{i}\omega_i$ be the angular velocity vector of link $\{i\}$ relative to $\{B\}$ expressed in terms of frame $\{i\}$
Let ${}^{i}\dot{\omega}_i$ be the angular acceleration vector of link $\{i\}$ relative to $\{B\}$ expressed in terms of frame $\{i\}$
Let ${}^{i}P_{i+1}$ be the origin of $\{i+1\}$ relative to $\{i\}$ The linear velocity of link $i+1$ with respect to frame $\{i+1\}$ is calculated as:

$$ {}^{i+1}\dot{v}_{i+1} = {}^{i+1}_{i}R({}^{i}\dot{\omega}_i \times {}^{i}P_{i+1} + {}^{i}\omega_i \times ({}^{i}\omega_i \times {}^{i}P_{i+1}) + {}^{i}\dot{v}_i) + 2 \cdot {}^{i+1}\omega_{i+1} \times \dot{d}_{i+1} \cdot {}^{i+1}\hat{Z}_{i+1} + \ddot{d}_{i+1} \cdot {}^{i+1}\hat{Z}_{i+1} $$

5.3 Mass Distribution

Let ρ be the density of the material and
let the rigid body be composed of differential volume elements dv where each is located with a vector ${}^{A}P = [xyz]^T$
The inertia tensor (which can be thought of as a generalization of the scalar moment of inertia of an object) relative to frame $\{A\}$ is defined as:

$$ {}^{A}I = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & -I_{zz} \end{bmatrix} $$

, with

$$ I_{xx} = \iiint_V (y^2 + z^2)\rho dv $$

$$I_{yy} = \iiint_V (x^2 + z^2)\rho dv$$
$$I_{zz} = \iiint_V (x^2 + y^2)\rho dv$$
$$I_{xy} = \iiint_V xy\rho dv$$
$$I_{xz} = \iiint_V xz\rho dv$$
$$I_{yz} = \iiint_V yz\rho dv$$

The elements I_{xx}, I_{yy} and I_{zz} are called the mass moments of inertia
The elements with mixed indices are called the mass products of intertia

5.3.1 Parallel-axis theorem

- A way of computing how the inertia tensor changes under translations of the reference coordinate system

Let $\{C\}$ be located at the center of mass of the body and
let $\{A\}$ be an arbitrarily translated frame.
The inertia tensor related in a frame with origin at the center of mass to the inertia tensor with respect to another reference frame is defined as:

$$^{A}I_{zz} = {}^{C}I_{zz} + m(x_c^2 + y_c^2)$$
$$^{A}I_{xy} = {}^{C}I_{xy} - m x_c y_c$$
$$^{A}I = {}^{C}I + m[P_c^T P_c I_3 - P_c P_c^T]$$

5.3.2 Facts about inertia tensors

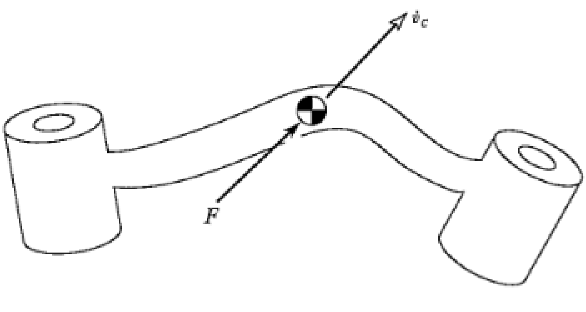
- If two axes of the reference frame form a plane of symmetry for the mass distribution of the body, the products of inertia having as an index the coordinate that is normal to the plane of symmetry will be zero.
- Moments of inertia must always be positive. Products of inertia may have either sign.
- The sum of the three moments of inertia is invariant under orientation changes in the reference frame.
- The eigenvalues of an inertia tensor are the principal moments for the body. The associated eigenvectors are the principal axes.

5.4 Newton's Equation, Euler's Equation

5.4.1 Newton's Equation

Let m be the total mass of a body (e.g. a link) and
let $\dot{v}_C$ be the acceleration with which the center of mass is accelerating.
The force acting at the center of mass and causing this acceleration is defined as:

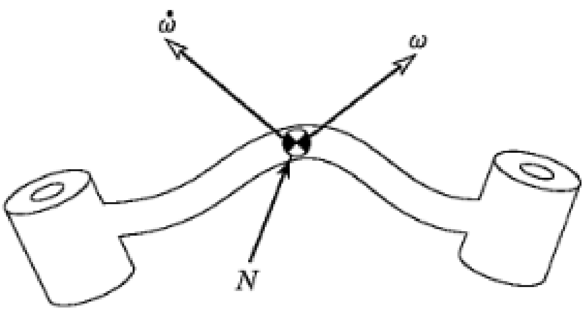
$$F = m\dot{v}_C$$



5.4.2 Euler's Equation

Let ω be the angular velocity with which a rigid body is rotating and
let $\dot{\omega}$ be the corresponding acceleration and
let ${}^{C}I$ be the inertia tensor of the body written in frame $\{C\}$, whose origin is located at the center of mass
The moment N, which must be acting on the body to cause this motion is defined as:

$$N = {}^{C}I\dot{\omega} + \omega \times {}^{C}I\omega$$



5.5 Iterative Newton-Euler Dynamic Formulation

5.5.1 Outward iterations to compute velocities and acceleration

Let $\{C_i\}$ be a frame attached to each link, having its origin located at the center of mass of the link and having the same orientation as the link frame $\{i\}$ and
let ${}^{i}\dot{v}_i$ be the linear acceleration of the origin of link $\{i\}$ relative to $\{B\}$ expressed in terms of frame $\{i\}$
Let ${}^{i}\omega_i$ be the angular velocity vector of link $\{i\}$ relative to $\{B\}$ expressed in terms of frame $\{i\}$
Let ${}^{i}\dot{\omega}_i$ be the angular acceleration vector of link $\{i\}$ relative to $\{B\}$ expressed in terms of frame $\{i\}$
Let ${}^{i}P_{C_i}$ be the origin of $\{C_i\}$ relative to $\{i\}$.
The linear acceleration of the center of mass of each link is computed as:

$${}^{i}\dot{v}_{C_i} = {}^{i}\dot{\omega}_i \times {}^{i}P_{C_i} + {}^{i}\omega_i \times ({}^{i}\omega_i \times {}^{i}P_{C_i}) + {}^{i}\dot{v}_i$$

5.5.2 Force and Torque acting on a link

Let $\{C_i\}$ be a frame attached to each link, having its origin located at the center of mass of the link and having the same orientation as the link frame $\{i\}$ and
The force and torque acting at the center of mass of each link expressed in $\{i\}$ is computed as:

$${}^{i}F_i = m_i \cdot {}^{i}\dot{v}_{C_i}$$

$${}^{i}N_i = {}^{C_i}I_i \cdot {}^{i}\dot{\omega}_i + {}^{i}\omega_i \times {}^{C_i}I_i \cdot {}^{i}\omega_i$$

5.5.3 Inward iterations to compute forces and torques

let ${}^{i+1}f_{i+1}$ be a force exerted on link $i+1$ by link i relative to link $i+1$ and
let ${}^{i}F_i$ be the force acting at the center of mass and causing this acceleration relative to $\{i\}$.
The force exerted on link i by link $i-1$ relative to link i is defined as:

$${}^{i}f_i = {}^{i}_{i+1}R \cdot {}^{i+1}f_{i+1} + {}^{i}F_i$$

Let ${}^{i+1}n_{i+1}$ be a torque exerted on link $i+1$ by link i relative to link $i+1$ and
let ${}^{i}P_{i+1}$ be the origin of frame $\{i+1\}$ relative to $\{i\}$ and
let ${}^{i}P_{C_i}$ be the origin of frame $\{C_i\}$ relative to $\{i\}$ and
let $\{C_i\}$ be a frame attached to link $\{i\}$, having its origin located at the center of mass of the link and having the same orientation as the link frame $\{i\}$ and
let ${}^{i}N_i$ be the torque acting at the center of mass of link $\{i\}$ relative to $\{i\}$ and
let ${}^{i}F_i$ be the force acting at the center of mass relative to $\{i\}$.
The torque exerted on link i by link $i-1$ relative to link i is defined as:

$${}^{i}n_i = {}^{i}N_i + {}^{i}_{i+1}R \cdot {}^{i+1}n_{i+1} + {}^{i}P_{C_i} \times {}^{i}F_i + {}^{i}P_{i+1} \times {}^{i}_{i+1}R \cdot {}^{i+1}f_{i+1}$$

The required joint torques that will result in the net forces and torques being applied to each link are computed as:
Rotational joint:

$$\tau_i = {}^i n_i^T \cdot {}^i\hat{Z}_i$$

Prismatic joint:

$$\tau_i = {}^i f_i^T \cdot {}^i\hat{Z}_i$$

5.5.4 The iterative Newton-Euler dynamics algorithm

for i = 0 to 5
do
  calculate: ${}^{i+1}\omega_{i+1}$, ${}^{i+1}\dot{\omega}_{i+1}$, ${}^{i+1}\dot{v}_{i+1}$, ${}^{i+1}\dot{v}_{C_{i+1}}$, ${}^{i+1}F_{i+1}$ and ${}^{i+1}N_{i+1}$
done

for i = 6 downto 1
do
  calculate: ${}^i f_i$, ${}^i n_i$ and τ_i
done

5.5.5 Inclusion of gravity forces in the dynamics algorithm

Set

$${}^0\dot{v}_0 = G$$

, where G has the magnitude of the gravity vector but points in the opposite direction

5.6 The Structure of a Manipulator's Dynamic Equations

5.6.1 State-space Equation

Let $\tau = [\tau_1, \ldots, \tau_n]$ be a vector of actuator torques and
let $M(\Theta)$ be the $n \times n$ mass matrix of the manipulator and
let $V(\Theta, \dot{\Theta})$ be an $n \times 1$ vector of centrifugal and Coriolis terms and
let $G(\Theta)$ be an $n \times 1$ vector of gravity terms.
The state-space equation is defined as:

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

5.6.2 Configuration-space Equation

Let $\tau = [\tau_1, \ldots, \tau_n]$ be a vector of actuator torques and
let $M(\Theta)$ be the $n \times n$ mass matrix of the manipulator and
let $B(\Theta)$ be a matrix of dimensions $n \times n(n-1)/2$ of Coriolis coefficient and
let $[\dot{\Theta}\dot{\Theta}] = [\dot{\theta}_1\dot{\theta}_2, \dot{\theta}_1\dot{\theta}_3, \ldots, \dot{\theta}_{n-1}\dot{\theta}_n]^T$ with dimension $n(n-1)/2 \times 1$ and
let $C(\Theta)$ be an $n \times n$ matrix of centrifugal coefficients and
let $[\dot{\Theta}^2] = [\dot{\theta}_1^2, \ldots, \dot{\theta}_n^2]^T$ of dimension $n \times 1$ and
let $G(\Theta)$ be an $n \times 1$ vector of gravity terms.
The configuration-space equation is defined as:

$$\tau = M(\Theta)\ddot{\Theta} + B(\Theta)[\dot{\Theta}\dot{\Theta}] + C(\Theta)[\dot{\Theta}^2] + G(\Theta)$$

5.7 Lagrangian Formulation of Manipulator Dynamics

5.7.1 Kinematic energy

Let m_i be the mass of the link i and
let v_{C_i} be the linear velocity of the center of mass of link i and
Let ${}^i\omega_i$ be the angular velocity vector of link $\{i\}$ relative to $\{B\}$ expressed in terms of frame $\{i\}$
let ${}^{C_i}I$ be the inertia tensor of the body written in frame $\{C_i\}$, whose origin is located at the center of mass of link

i.
The kinematic energy of the ith link is defined as:

$$k_i = \frac{1}{2} m_i \cdot v_{C_i}^T \cdot v_{C_i} + \frac{1}{2}\, {}^i\omega_i^T \cdot {}^{C_i}I_i \cdot {}^i\omega_i$$

The total kinematic energy of the manipulator is computed as:

$$k = \sum_{i=1}^{n} k_i$$

$$k(\Theta, \dot{\Theta}) = \frac{1}{2} \dot{\Theta}^T M(\Theta) \dot{\Theta}$$

, where $M(\Theta)$ is the $n \times n$ mass matrix of the manipulator and $\Theta = [\theta_1, \ldots, \theta_n]^T$

5.7.2 Potential energy

Let m_i be the mass of the link i and
let ${}^0g$ be the 3×1 gravity vector and
let ${}^0P_{C_i}$ be the vector locating the center of mass of the ith link and
let u_{ref_i} be a constant chosen so that the minimum values of u_i is zero.
The potential energy of the ith link is defines as:

$$u_i = -m_i \cdot {}^0g^T \cdot {}^0P_{C_i} + u_{ref_i}$$

The total potential energy stored in the manipulator is computed as:

$$u = u(\Theta) = \sum_{i=1}^{n} u_i$$

5.7.3 Lagrangian formulation

Let $k(\Theta, \dot{\Theta})$ be the total kinematic energy of a manipulator and
let $u(\Theta)$ be the total potential energy of the manipulator.
The Lagrangian of the manipulator is:

$$\mathcal{L}(\Theta, \dot{\Theta}) = k(\Theta, \dot{\Theta}) - u(\Theta)$$

The equations of motions for the manipulator are given by:

$$\frac{d}{dt} \frac{\partial k(\Theta, \dot{\Theta})}{\partial \dot{\Theta}} - \frac{\partial k(\Theta, \dot{\Theta})}{\partial \Theta} + \frac{\partial u(\Theta)}{\partial \Theta} = \tau$$

5.8 Formulating Manipulator Dynamics in Cartesian Space

5.8.1 Cartesian state-space equation

Let $\mathcal{J}(\Theta)$ be the Jacobian with $\tau = \mathcal{J}^T(\Theta)\mathcal{F}$ and
let $M(\Theta)$ be the $n \times n$ mass matrix of the manipulator and
let $V(\Theta, \dot{\Theta})$ be an $n \times 1$ vector of centrifugal and Coriolis terms and
let $G(\Theta)$ be an $n \times 1$ vector of gravity terms.

The Cartesian mass matrix of the manipulator is defined as:

$$M_X(\Theta) = \mathcal{J}^{-T}(\Theta) M(\Theta) \mathcal{J}^{-1}(\Theta)$$

The Vector of velocity terms in Cartesian space is defines as:

$$V_X(\Theta, \dot{\Theta}) = \mathcal{J}^{-T}(\Theta)(V(\Theta, \dot{\Theta}) - M(\Theta)\mathcal{J}^{-1}(\Theta)\dot{\mathcal{J}}(\Theta)\dot{\Theta})$$

The vector of gravity terms in Cartesian space is defined as:

$$G_X(\Theta) = \mathcal{J}^{-T}(\Theta) G(\Theta)$$

The force-torque vector is defined as:

$$\mathcal{F} = M_X(\Theta)\ddot{\chi} + V_X(\Theta, \dot{\Theta}) + G_X(\Theta)$$

, where $\ddot{\chi} = \dot{\mathcal{J}}\dot{\Theta} + \mathcal{J}\ddot{\Theta}$

5.8.2 Cartesian configuration space torque equation

Let $\tau = [\tau_1, \dots, \tau_n]$ be a vector of actuator torques and
let $M_X(\Theta)$ be the $n \times n$ Cartesian mass matrix of the manipulator and
let $B_X(\Theta)$ be a matrix of dimensions $n \times n(n-1)/2$ of Coriolis coefficient and
let $[\dot{\Theta}\dot{\Theta}] = [\dot{\theta}_1\dot{\theta}_2, \dot{\theta}_1\dot{\theta}_3, \dots, \dot{\theta}_{n-1}\dot{\theta}_n]^T$ with dimension $n(n-1)/2 \times 1$ and
let $C_X(\Theta)$ be an $n \times n$ matrix of centrifugal coefficients and
let $[\dot{\Theta}^2] = [\dot{\theta}_1^2, \dots, \dot{\theta}_n^2]^T$ of dimension $n \times 1$ and
let $G(\Theta)$ be an $n \times 1$ vector of gravity terms.
The configuration-space equation is defined as:

$$\tau = \mathcal{J}^T(\Theta)M_X(\Theta)\ddot{\chi} + B_X(\Theta)[\dot{\Theta}\dot{\Theta}] + C_X(\Theta)[\dot{\Theta}^2] + G(\Theta)$$

, where in general $B_X(\Theta) \neq B(\Theta)$ and $C_X(\Theta) \neq C(\Theta)$

5.9 Inclusion of Nonrigid Body Effects

The viscous friction is defined as:

$$\tau_{friction} = v\dot{\theta}$$

The Coulomb-friction is defined as:

$$\tau_{friction} = c \cdot \operatorname{sgn}(\dot{\theta})$$

, where c is a Coulomb-friction constant
The viscous and Coulomb friction is defined as:

$$f(\theta, \dot{\theta}) = \tau_{friction} = c \cdot \operatorname{sgn}(\dot{\theta}) + v\dot{\theta}$$

The state-space equation with inclusion of friction is defined as:

$$\tau = M(\Theta)\ddot{\Theta} + V(\Theta, \dot{\Theta}) + G(\Theta) + F(\Theta, \dot{\Theta})$$

6 Trajectory Generation

- **Trajectory:** a time history of position, velocity and acceleration for each degree of freedom
- **Path-update rate:** the rate, the trajectory points are computed (typically 60 Hz - 2000 Hz)
- **Motions of a manipulator:** motions of the tool frame $\{T\}$ relative to the station frame $\{S\}$
- **Via points:** intermediate frames between the initial and final position
- **Path points:** includes all the via points plus the initial and final points (Here: points are frames)

6.1 Joint-Space Schemes

- methods of path generation in which the path shapes (in space and time) are described in terms of function of joint angles
- θ describes the joint angle, $\dot{\theta}$ the velocity and $\ddot{\theta}$ the acceleration

6.1.1 Cubic polynomials

- Smooth function
- Velocity at initial and goal position is 0

Let θ be the a joint angle and
let θ_0 the angle of the initial position and
let θ_f the angle of the goal position and
let $\theta(t)$ be a function that describes the angle relative to the time t where $\theta(t_0)$ is the initial position and $\theta(t_f)$ is the goal position.
It follows:

$$\theta(t_0 = 0) = \theta_0$$

$$\theta(t_f) = \theta_f$$
$$\dot{\theta}(0) = 0$$
$$\dot{\theta}(t_f) = 0$$
$$\theta(t) = \theta_0 + \left(\frac{3}{t_f^2}(\theta_f - \theta_0)\right) \cdot t^2 + \left(-\frac{2}{t_f^3}(\theta_f - \theta_0)\right) \cdot t^3$$
$$\dot{\theta}(t) = \left(\frac{6}{t_f^2}(\theta_f - \theta_0)\right) \cdot t + \left(-\frac{6}{t_f^3}(\theta_f - \theta_0)\right) \cdot t^2$$
$$\ddot{\theta}(t) = \left(\frac{6}{t_f^2}(\theta_f - \theta_0)\right) + \left(-\frac{12}{t_f^3}(\theta_f - \theta_0)\right) \cdot t$$

6.1.2 Cubic polynomials for a path with via points

- Velocity at initial and goal position is ≥ 0

Let θ be the a joint angle with θ_0 the angle of the position of the via point and θ_f the angle of the position of the next via point.
Let $\theta(t)$ be a function that describes the angle relative to the time t where $\theta(t_0)$ is the position of the via point and $\theta(t_f)$ is the position of the next via point. It follows:

$$\theta(t_0 = 0) = \theta_0$$
$$\theta(t_f) = \theta_f$$
$$\dot{\theta}(0) = \dot{\theta}_0$$
$$\dot{\theta}(t_f) = \dot{\theta}_f$$
$$\theta(t) = \theta_0 + \dot{\theta}_0 \cdot t + \left(\frac{3}{t_f^2}(\theta_f - \theta_0) - \frac{2}{t_f}\dot{\theta}_0 - \frac{1}{t_f}\dot{\theta}_f\right) \cdot t^2 + \left(-\frac{2}{t_f^3}(\theta_f - \theta_0) + \frac{1}{t_f^2}(\dot{\theta}_f + \dot{\theta}_0)\right) \cdot t^3$$
$$\dot{\theta}(t) = \dot{\theta}_0 + \left(\frac{6}{t_f^2}(\theta_f - \theta_0) - \frac{4}{t_f}\dot{\theta}_0 - \frac{2}{t_f}\dot{\theta}_f\right) \cdot t + \left(-\frac{6}{t_f^3}(\theta_f - \theta_0) + \frac{3}{t_f^2}(\dot{\theta}_f + \dot{\theta}_0)\right) \cdot t^2$$
$$\ddot{\theta}(t) = \left(\frac{6}{t_f^2}(\theta_f - \theta_0) - \frac{4}{t_f}\dot{\theta}_0 - \frac{2}{t_f}\dot{\theta}_f\right) + \left(-\frac{12}{t_f^3}(\theta_f - \theta_0) + \frac{6}{t_f^2}(\dot{\theta}_f + \dot{\theta}_0)\right) \cdot t^2$$

6.1.3 High-order polynomials

- Smooth function
- Velocity at initial and goal position is ≥ 0
- Acceleration at initial and goal position is specified

Let θ be the a joint angle, and
let θ_0 be the angle of the initial position, and
let θ_f be the angle of the goal position, and
let $\theta(t)$ be a function that describes the angle relative to the time t, and
let $\theta(t_0)$ be the initial position, and
let $\theta(t_f)$ be the goal position.
It follows:

$$\theta(t_0 = 0) = \theta_0$$
$$\theta(t_f) = \theta_f$$
$$\dot{\theta}(0) = \dot{\theta}_0$$
$$\dot{\theta}(t_f) = \dot{\theta}_f$$
$$\ddot{\theta}(0) = \ddot{\theta}_0$$

$$\ddot{\theta}(t_f) = \ddot{\theta}_f$$

$$\theta(t) = \theta_0 + \dot{\theta}_0 \cdot t + \frac{\ddot{\theta}_0}{2} \cdot t^2 + \frac{20\theta_f - 20\theta_0 - (8\dot{\theta}_f + 12\dot{\theta}_0)t_f - (3\ddot{\theta}_0 - \ddot{\theta}_f)t_f^2}{2t_f^3} \cdot t^3$$
$$+ \frac{30\theta_0 - 30\theta_f + (14\dot{\theta}_f + 16\dot{\theta}_0)t_f + (3\ddot{\theta}_0 - 2\ddot{\theta}_f)t_f^2}{2t_f^4} \cdot t^4 + \frac{12\theta_f - 12\theta_0 - (6\dot{\theta}_f + 6\dot{\theta}_0)t_f - (\ddot{\theta}_0 - \ddot{\theta}_f)t_f^2}{2t_f^5} \cdot t^5$$

6.1.4 Linear function with parabolic blends

- Linear function with smooth start and end (parabolic blends)

Let θ_0 be the angle of the initial position and
let θ_f be the angle of the goal position.
The time at the end of the blend region is computed as:

$$t_b = \frac{t}{2} - \frac{\sqrt{\ddot{\theta}^2 t^2 - 4\ddot{\theta}(\theta_f - \theta_0)}}{2\ddot{\theta}}$$

, where

$$\ddot{\theta} \geq \frac{4(\theta_f - \theta_0)}{t^2}$$

The value of θ at the end of the blend region is computed as:

$$\theta_b = \theta_0 + \frac{1}{2}\ddot{\theta}t_b^2$$

- $\ddot{\theta} = \frac{4(\theta_f - \theta_0)}{t^2} \implies$ the linear portion length is 0 and the path is composed of two blends that connect
- $\ddot{\theta} \to \infty \implies$ the length of the blend region $\to 0$

6.1.5 Linear function with parabolic blends for a path with via points

Let j, k, l be 3 neighboring path points and
let the duration of the blend region at path point k be t_k and
let the duration of the linear portion between j and k be t_{jk} and
let the overall duration of the segment connecting j and k be t_{djk} and
let the velocity during the linear portion be $\dot{\theta}_{jk}$ and
let the acceleration during the blend at point j be $\ddot{\theta}_j$ and
let the angle at the path point k be θ_k
It follows:

$$\dot{\theta}_{jk} = \frac{\theta_k - \theta_j}{t_{djk}}$$

$$\ddot{\theta}_k = \operatorname{sgn}(\dot{\theta}_{kl} - \dot{\theta}_{jk})|\ddot{\theta}_k|$$

$$t_k = \frac{\dot{\theta}_{kl} - \dot{\theta}_{jk}}{\ddot{\theta}_k}$$

$$t_{jk} = t_{djk} - \frac{1}{2}t_j - \frac{1}{2}t_k$$

Let 1 be the initial path point
It follows:

$$\ddot{\theta}_1 = \operatorname{sgn}(\theta_2 - \theta_1)|\ddot{\theta}_1|$$

$$t_1 = t_{d12} - \sqrt{t_{d12}^2 - \frac{2(\theta_2 - \theta_1)}{\ddot{\theta}_1}}$$

$$\dot{\theta}_{12} = \frac{\theta_2 - \theta_1}{t_{d12} - \frac{1}{2}t_1}$$

$$t_{12} = t_{d12} - t_1 - \frac{1}{2}t_2$$

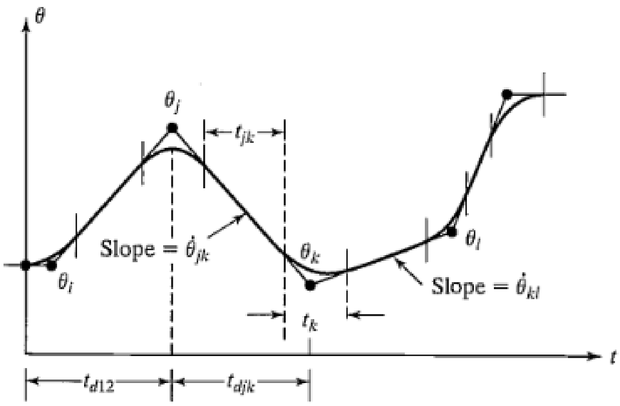
Let n be the final path point
It follows:

$$\ddot{\theta}_n = \mathrm{sgn}(\theta_{n-1} - \theta_n)|\ddot{\theta}_n|$$

$$t_n = t_{d(n-1)n} - \sqrt{t^2_{d(n-1)n} - \frac{2(\theta_n - \theta_{n-1})}{\ddot{\theta}_n}}$$

$$\dot{\theta}_{(n-1)n} = \frac{\theta_n - \theta_{n-1}}{t_{d(n-1)n} - \frac{1}{2}t_n}$$

$$t_{(n-1)n} = t_{d(n-1)n} - t_n - \frac{1}{2}t_{n-1}$$



6.2 Cartesian-Space Schemes

6.2.1 Geometric Problems with Cartesian Paths

- **Intermediate points unreachable:** It is possible that some points lying on the path are outside of the workspace.
- **High joint rates near singularity:** If a manipulator is following a Cartesian straight-line path and approaches a singular configuration of the mechanism, one or more joint velocities might increase toward infinity. Because velocities of the mechanism are upper bounded, this situation usually results in the manipulator's deviating from the desired path.
- **Start and Goal reachable in different solutions:** The goal point cannot be reached in the same physical solution as the robot is at the start point.

6.3 Path Generation at Runtime

6.3.1 Generation of joint-space paths

6.3.1.1 Linear Splines with Parabolic blends

In the linear portion:

$$\theta = \theta_j + \dot{\theta}_{jk}t$$

$$\dot{\theta} = \dot{\theta}_{jk}$$

$$\ddot{\theta} = 0$$

, where t is the time since the jth via point
In the blend region:

$$t_{inb} = t - (\frac{1}{2}t_j + t_{jk})$$

$$\theta = \theta_j + \dot{\theta}_{jk}(t - t_{inb}) + \frac{1}{2}\ddot{\theta}_k t_{inb}^2$$
$$\dot{\theta} = \dot{\theta}_{jk} + \ddot{\theta}_k t_{inb}$$
$$\ddot{\theta} = \ddot{\theta}_k$$

, where t is being reset to $\frac{1}{2}t_k$ when a linear segment is entered

7 Manipulator-mechanism design

7.1 Elements of a Robot System

- The manipulator, including its internal or proprioceptive sensors
- The end-effector, or end-of-arm tooling
- External sensors and effectors, such as vision systems and part feeders
- The controller.

7.2 Steps

- Choose general kinematic structure
- Choose actuation (Actuator, Reduction, Transmission)
- Select sensors

7.3 Task Requirements

- **Number of degrees of freedom:** The number of degrees of freedom in a manipulator should match the number required by the task
- **Workspace**
- **Load capacity:** Depends upon the sizing of its structural members, power-transmission system and actuators
- **Speed:** High speed offers advantages in many applications; For some applications the process itself limits the speed rather than the manipulator
- **Repeatability and accuracy:** High repeatability and accuracy are expensive to achieve

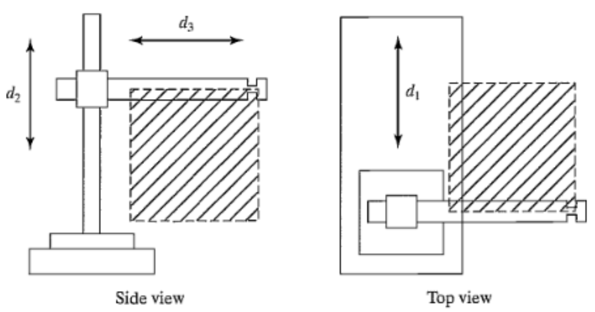
7.4 Kinematic configuration

Often used kinematic design:

- **Position structure:** Position the wrist (first 3 joints)
- **Orienting structure/Wrist:** Orient the end-effector and have axes that intersect at the wrist point (last 3 joints)

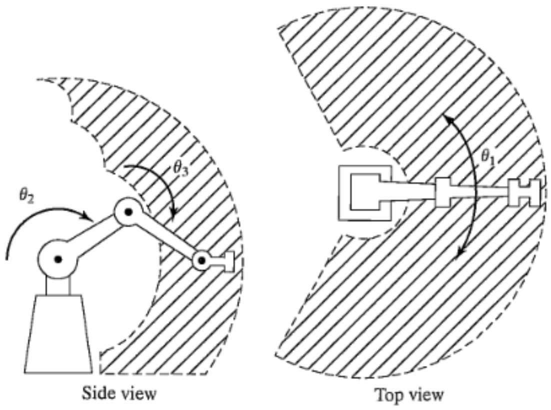
7.4.1 Cartesian

- Structure:
  - First 3 joints are prismatic, mutually orthogonal and correspond to the $\hat{X}, \hat{Y}$ and $\hat{Z}$ Cartesian directions
- Often used for gantry robots



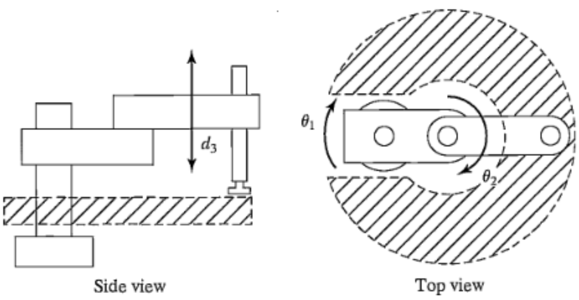
7.4.2 Articulated

- Structure:
  - 2 "shoulder" joints (one for rotation about a vertical axis and one for elevation out of the horizontal plane)
  - 1 "elbow" joint (usually parallel to the shoulder joint)
  - 2 or 3 wrist joints at the end of the manipulator
- Also called jointed, elbow or anthropomorphic manipulator



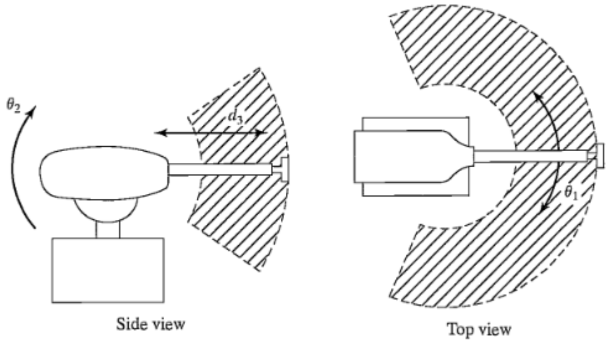
7.4.3 SCARA

- Structure:
  - 3 parallel revolute joints (for moving and orient in a plane)
  - 1 prismatic joint (for moving the end effector normal to the plane)
- First 3 joints don't have to support any of the weight of the manipulator or the load
- Link 0 can house the actuators for the first 2 joints
- Can move very fast



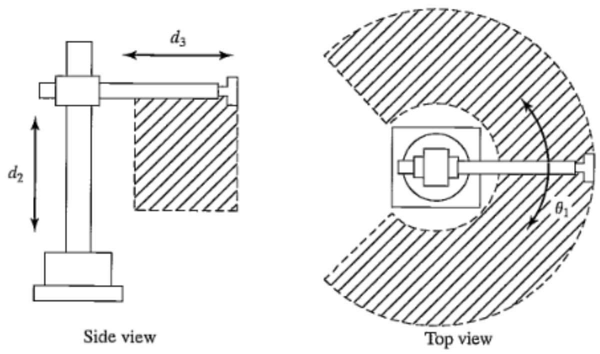
7.4.4 Spherical

- Structure:
  - 2 "shoulder" joints (one for rotation about a vertical axis and one for elevation out of the horizontal plane)
  - 1 prismatic joint
  - may (2 or 3 wrist joints at the end of the manipulator)



7.4.5 Cylindrical

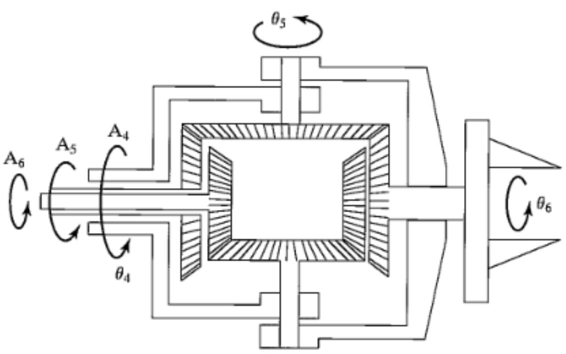
- Structure:
  - 1 revolute joint (with a vertical axis)
  - 1 prismatic joint (for translating vertically)
  - 1 prismatic joint (for translating horizontally)



7.4.6 Wrist

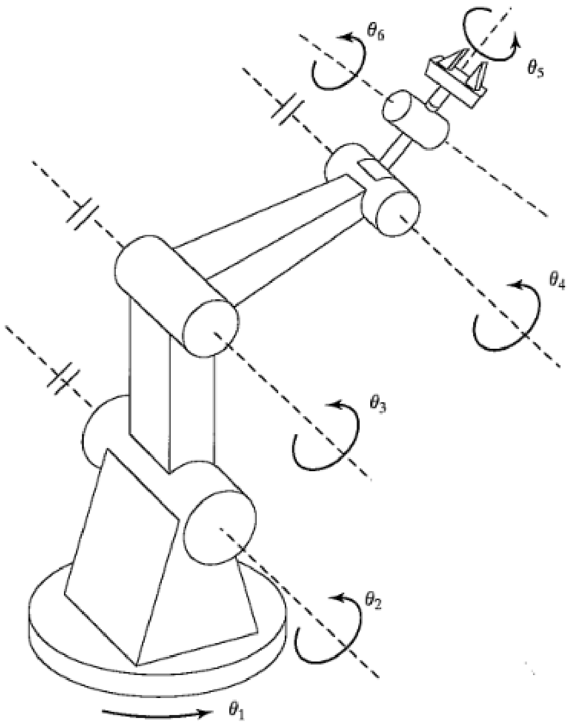
7.4.6.1 Most common wrist configuration

- Structure:
  - 2 or 3 revolute joints with orthogonal intersecting axes
- Any orientation can be reached
- Closed-form kinematic solution exist



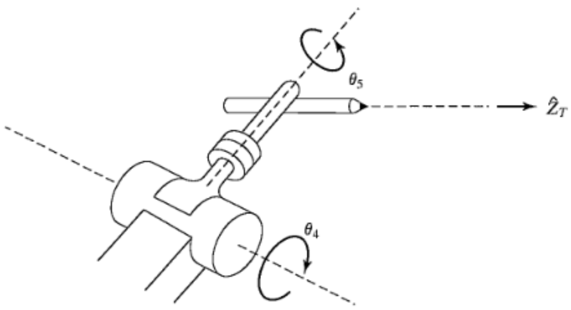
7.4.6.2 Parallel joints

- Structure:
  - joint-2, joint-3 and joint-4 axes are parallel
- Closed-form kinematic solution exist



7.4.6.3 5-DOF welding robot

- Structure:
  - 2 revolute joints with orthogonal intersecting axes



7.5 Quantitative Measures of Workspace Attributes

7.5.1 Efficiency of design in terms of generating workspace

Let $a_i - 1$ be the link length of link $i - 1$ and
let d_i be the link offset of joint i and
let N be the number of joints.
The length sum of a manipulator is computed as:

$$L = \sum_{i=1}^{N}(a_{i-1} + d_i)$$

Let L be the length sum of a manipulator and
let W be the volume of the manipulator's workspace
The structural length index is defined as:

$$Q_L = \frac{L}{\sqrt[3]{W}}$$

, where good designs have a low Q_L

7.5.2 Designing well-conditioned workspaces

- When the manipulator is near a singular point, actions of the manipulator are said to be poorly conditioned

Let $\mathcal{J}(\Theta)$ be the Jacobian of a manipulator.
The manipulability measure is defined as:

$$w = \sqrt{\det(\mathcal{J}(\Theta)\mathcal{J}^T(\Theta)}$$

The manipulability measure for a nonredundant manipulator is defined as:

$$w = |\det(\mathcal{J}(\Theta))|$$

, where good designs have high values of w

Let $M(\Theta)$ be the mass matrix of the manipulator and
let $M_X(\Theta)$ be the Cartesian mass matrix of the manipulator and
let $\mathcal{J}(\Theta)$ be the Jacobian of a manipulator.
The manipulability measure based on acceleration analysis or force-application capability is defined as:

$$M_X(\Theta) = \mathcal{J}^{-T}(\Theta)M(\Theta)\mathcal{J}^{-1}(\Theta)$$

7.6 Redundant and Closed-Chain Structures

7.6.1 Micromanipulators and other redundancies

- Avoidance of singular configurations
- Avoidance of collisions while operating in cluttered work environments

7.6.2 Closed-loop structures

- Structure with parallel links
- Increase the stiffness of the mechanism
- Reduces the allowable range of motion of the joints and thus decrease the workspace size

Let l be the number of links (including the base) and
let n be the total number of joints and
let f_i be the number of degrees of freedom associated with the ith joint.
The Grübler's formula which computes the total number of degrees of freedom is defined as

$$F = 6(l - n - 1) + \sum_{i=1}^{n} f_i$$

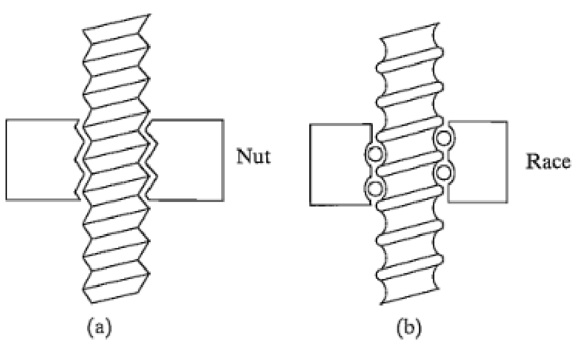
7.7 Actuation Schemes

7.7.1 Actuator location

- **Direct-drive**
  - Placed at the joint
  - Simple in design
  - High controllability
  - No transmission or reduction elements
- **Speed-reduction system**
  - Many actuators are suited to high speeds and low torques
  - Could be placed either at the actuator or at the joint
  - Lowers speed and increases the torque
  - Adds complexity, friction and flexibility
- **Transmission system**
- Actuators tend to be rather heavy. By locating them from the joint to the base, the overall inertia can be reduced.
- Transfers the motion from the actuator to the joint
- Adds complexity, friction and flexibility

7.7.2 Reduction and transmission systems

- **Gears**
  - Most common element used for reduction
  - Can produce large reduction in compact configurations
  - But adds backlash and friction
- **Cables, flexible bands, belts, roller chains**
  - Able to combine transmission with reduction
- **Lead screws (a), ball-bearing screws (b)**
  - Are very stiff
  - Can support very large loads
  - Able to transform rotary motion into linear motion
  - Ball-bearing screwa have very low friction



7.7.2.1 Gear ratio

Let η be speed-reducing and torque-increasing effect of a gear pair ($\eta > 1$) and
let $\dot{\theta}_i$ be the input speed.
The output speed is calculated as:

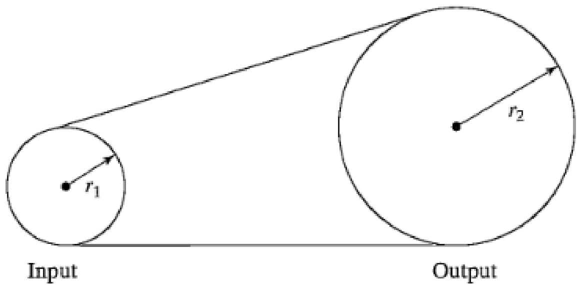
$$\dot{\theta}_o = \frac{\dot{\theta}_i}{\eta}$$

Let η be speed-reducing and torque-increasing effect of a gear pair ($\eta > 1$) and
let τ_i be the input torque.
The output torque is calculated as:

$$\tau_o = \eta \tau_i$$

Let $r2$ be the radius of the output pulley(using bends, cables, ...)/gear and
let $r1$ be the radius of the input pulley/gear.
The gear ratio is defined as:

$$\eta = \frac{r_2}{r_1}$$



7.8 Stiffness and Deflections

7.8.1 Flexible elements in parallel and in series

Let k_1 and k_2 the stiffness of 2 flexible members connected in parallel.
The produced net stiffness is calculated as:

$$k_{parallel} = k_1 + k_2$$

Let k_1 and k_2 the stiffness of 2 flexible members connected in series.
The produced net stiffness is calculated as:

$$\frac{1}{k_{series}} = \frac{1}{k_1} + \frac{1}{k_2}$$

7.8.2 Shafts

Let G be the shear modulus of elasticity (e.g. $7.5 \times 10^{10} Nt/m^2$ for steel) and
let d be the shaft diameter and
let l be the shaft length.
Th torsional stiffness of a round shaft can be calculated as:

$$k = \frac{G\pi d^4}{32l}$$

7.8.3 Gears

Let $C_g = 1.34 \times 10^{10} Nt/m^2$ for steel and
let b be the face width of the gears and
let r be the radius of the output gear.
The stiffness of the output gear is calculated as:

$$k = C_g b r^2$$

Let η be the gear ratio and
let k_i be the stiffness of the input gear.
The stiffness of the output gear is computed as:

$$k_o = \frac{\tau_o}{\delta\theta_o} = \frac{\eta k_i \delta\theta_i}{\frac{1}{\eta}\delta\theta_i} = \eta^2 k_i$$

7.8.4 Belts

Let A be the cross-sectional area of the belt and
let E be the modulus of elasticity of the belt and
let l be the length of the free belt between pulleys + $\frac{1}{3}$ of the length of the belt in contact with the pulleys.
The stiffness of the belt drive is calculated as:

$$k = \frac{AE}{l}$$

7.8.5 Links

Let E be the modulus of elasticity (e.g. $2 \times 10^{11} Nt/m^2$ for steel) and
let d_i and d_o be the inner and outer diameters of the tubular beam and
let w_i and w_o be the inner and outer widths of the beam (i.e. wall thickness is $(w_o - w_i)/2$) and
let l be the length.

The stiffness of a round hollow beam is computed as:

$$k = \frac{3\pi E(d_o^4 - d_i^4)}{64 l^3}$$

The stiffness of a square-cross-section hollow beam is computed as:

$$k = \frac{E(w_o^4 - w_i^4)}{4 l^3}$$

7.9 Actuators

7.9.1 Hydraulic cylinders

- High forces; No need of a reduction system
- Speed depends upon the pump and accumulator system
- Position control is well understood and straightforward
- Require a lot of equipment (pumps, accumulators, hoses, servo valves)
- Tend to be inherently messy

7.9.2 Pneumatic cylinders

- High forces; No need of a reduction system
- Speed depends upon the pump and accumulator system
- Cleaner than hydraulics
- Difficult to control accurately

7.9.3 Brush motors

- High peak torques but only much lower torques over long periods of time
- windings gain high temperature
- Problems of Brush wear and friction

7.9.4 Brushless motors

- No brush wear and friction
- Possibility to attach the windings outside to the motor case → Better cooling

7.10 Position Sensing

7.10.1 Rotary Optical Encoder

- As the encoder shaft turns, a disk containing a pattern of fine lines interrupts a light beam. The shaft angle is determined by counting the number of pulses.
- To determine the direction, there are two such channels, with wave pulse trains 90 degrees out of phase. The direction of rotation is determined by the relative phase of the two square waves

7.10.2 Resolvers

- Devices that output 2 signals (the sine of the shaft angel and the the cosine)
- The shaft angle is computed from the relative magnitude of the two signals
- Often more reliable than optical encoders
- Lower resolution than optical encoders

7.10.3 Potentiometers

- Produce a voltage proportional to the shaft position
- Limited by resolution, linearity and noise susceptibility

7.10.4 Tachometers

- Provide an analog signal proportional to the shaft velocity

7.11 Force Sensing

7.11.1 Placements of Force sensors

- **At the joint actuators:** measure the torque/force output of the actuator itself
- **Between the end-effector and the last joint of the manipulator:** referred to as wrist sensors; measures the force/torques acting on the end-effector
- **At the fingertips:** measure from one to four components of force acting at each fingertip

8 Linear control of manipulators

- the system can be modeled mathematically by linear differential equations

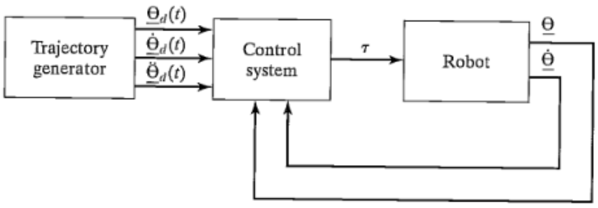
8.1 Feedback and Closed-loop Control

8.1.1 Open-loop scheme

- Calculation of the torques for the actuators without made of use of the feedback from the sensors (position, velocity sensors)

8.1.2 Closed-loop scheme

- Calculation of the torques for the actuators with made of use of the feedback from the sensors (position, velocity sensors)
- A bad designed control system can result in unstable performance (Servo Error increasing instead of decreasing)



8.1.3 Servo Error

- Presence of disturbances (noise) leads to deviant position

Let Θ_d and $\dot{\Theta}_d$ be the desired position and velocity of the trajectory and
let Θ and $\dot{\Theta}$ be actual position and velocity.
The difference between the desired and the actual position is computed as:

$$E = \Theta_d - \Theta$$

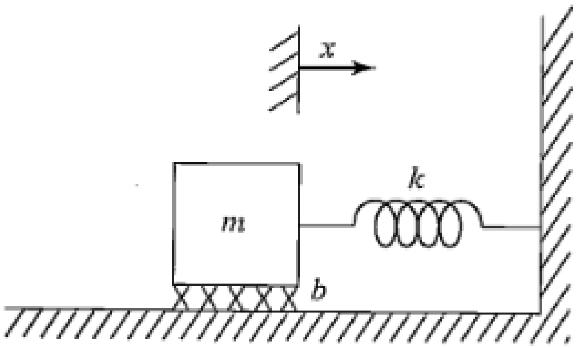
The difference between the desired and the actual velocity is computed as:

$$\dot{E} = \dot{\Theta}_d - \dot{\Theta}$$

8.1.4 Independent Joint Control

- Uses an independent single-input, single-output (SISO) control system for each joint
- Adopted by most industrial robot suppliers

8.2 Second-Order Linear Systems



Let be given a spring-mass system with friction where
m be the mass of a block attached to the spring and
k be the stiffness of the spring and
b a coefficient of friction.
It follows:

$$m\ddot{x} + b\dot{x} + kx = 0$$

The corresponding characteristic equation is defined as:

$$ms^2 + bs + k = 0$$

with the roots (poles of the system)

$$s_{1,2} = -\frac{b}{2m} \pm \frac{\sqrt{b^2 - 4mk}}{2m}$$

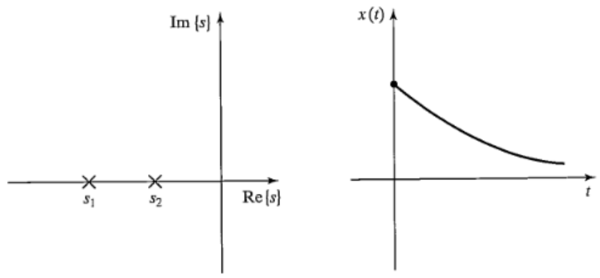
8.2.1 Real and Unequal Roots

- If the roots are real and negative → Overdamped (non-oscillatory exponential decay)
- If the roots are real and positive → Not BIBO stable (non-oscillatory exponential increase)

Let $b^2 > 4mk$
It follows:

$$x(t) = c_1 e^{s_1 t} + c_2 e^{s_2 t}$$

, where c_1 and c_2 are constants which can be computed for any given set of initial conditions



8.2.2 Complex Roots

- If the roots are complex with negative real components → Underdamped (oscillatory decay)
- If the roots are complex with positive real components → Not BIBO stable (oscillatory increase)
- If the roots are purely imaginary → Undamped (oscillatory behavior without increase/decay)

Let $b^2 < 4mk$
It follows:

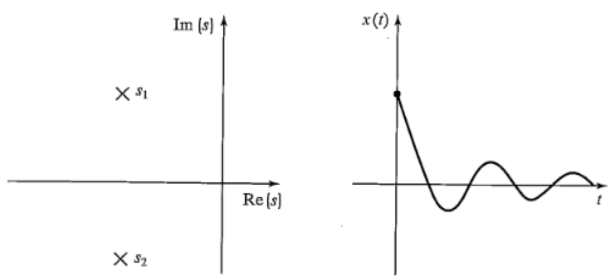
$$x(t) = c_1 e^{\lambda t} \cos(\mu t) + c_2 e^{\lambda t} \sin(\mu t)$$

, where $s_{1,2} = \lambda \pm \mu i$

Let $c_1 = r \cos \delta$ and
let $c_2 = r \sin \delta$.
It follows

$$x(t) = r e^{\lambda t} \cos(\mu t - \delta)$$

, where
$r = \sqrt{c_1^2 + c_2^2}$
$\delta = \mathrm{atan2}(c_2, c_1)$

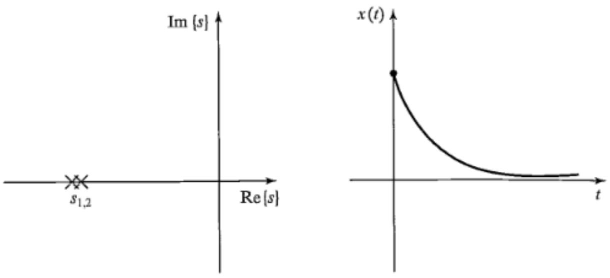


8.2.3 Real and Equal Roots

- If the roots are real, equal and negative → Critically damped (fastest non-oscillatory exponential decay)

Let $b^2 = 4mk$
It follows:

$$x(t) = c_1 e^{-\frac{b}{2m}t} + c_2 t e^{-\frac{b}{2m}t}$$



8.2.4 Damping Ratio

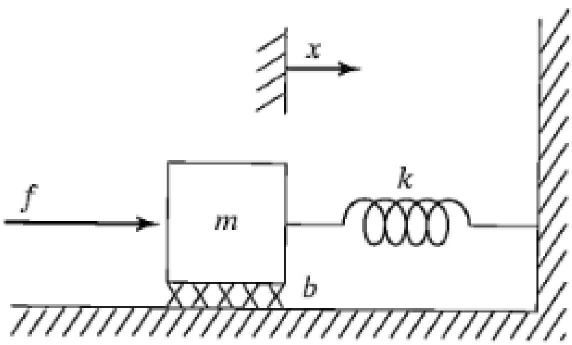
Let $\zeta = \frac{b}{2\sqrt{km}}$ be the damping ratio and
let $\omega_n = \sqrt{k/m}$ be the natural frequency.
It follows:

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

and

- $\zeta < 1$: underdamped
- $\zeta = 1$: critically damped
- $\zeta > 1$: overdamped

8.3 Control of Second-Order Systems



Let be given a spring-mass system with friction and an actuator where
m be the mass of a block attached to the spring and
k be the stiffness of the spring and
b a coefficient of friction and
f be a force applied by an actuator to the block
It follows:

$$m\ddot{x} + b\dot{x} + kx = f$$

Let x be the position of the block detected by a sensor and
let $\dot{x}$ be the velocity of the block detected by a sensor and
The force that should be applied by the actuator is computed as:

$$f = -k_p x - k_v \dot{x}$$

, where k_p is the position gain and k_v is the velocity gain

8.3.1 Closed-loop Stiffness

Let the closed-loop stiffness be $k' = k + k_p$ and
let $b' = b + k_v$.
It follows:

$$m\ddot{x} + b'\dot{x} + k'x = 0$$

- If b' or k' is negative → Unstable Control System
- For a critical damping: $b' = 2\sqrt{mk'}$

8.4 Control-Law Partitioning

- Partitioning of the controller into
  - a model-based portion and
  - a servo portion

8.4.1 Model-based portion

Let f be the force applied by an actuator.
The model-based potion of the control is defined as:

$$f = \alpha f' + \beta$$

, where α and β are function or constants and
are chosen so that if f' is taken as the new input to the system, the system appears to be a unit mass

8.4.2 Model-based portion of a spring-mass system with friction

Let be given a spring-mass system with friction and an actuator where
m be the mass of a block attached to the spring and
k be the stiffness of the spring and
b a coefficient of friction and
α and β are chosen as follows:

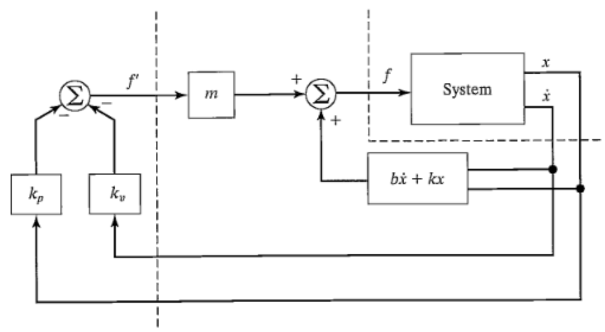
$$\alpha = m$$

$$\beta = b\dot{x} + kx$$

For a critical damping follows:

$$k_v = 2\sqrt{k_p}$$

, where $f' = \ddot{x} = -k_v\dot{x} - k_p x$



8.5 Trajectory-Following Control

8.5.1 Servo Error

Let the trajectory be given as a function of time with
$x_d(t)$ be the desired position of the block and
$\dot{x}_d(t)$ be the desired velocity of the block and
$\ddot{x}_d(t)$ be the desired acceleration of the block and
The servo error is defined as:

$$e = x_d - x$$

It follows:

$$\dot{e} = \dot{x}_d - \dot{x}$$

and

$$\ddot{e} = \ddot{x}_d - \ddot{x}$$

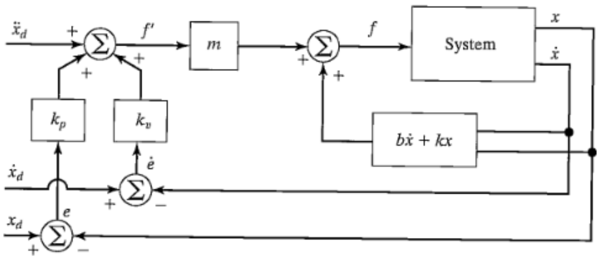
8.5.2 Trajectory following Servo-control Law (Servo Portion)

The servo-control law that will cause trajectory following is defined as:

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

It follows:

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



8.6 Disturbance Rejection

Let e be the servo error with its derivatives $\dot{e}$ and $\ddot{e}$
The disturbance force is computed as:

$$f_{dist} = m(\ddot{e} + k_v \dot{e} + k_p e)$$

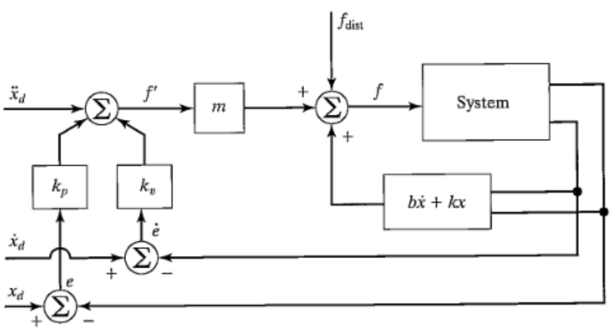
If f_{dist} is bounded $\implies$ the solution of the differential equation $e(t)$ is also bounded
$\implies$ BIBO (bounded-input, bounded-output) stability

8.6.1 Steady-state Error

Let the disturbance force f_{dist} be constant.
The steady-state error is computed as:

$$e = \frac{f_{dist}}{k_p}$$

$\implies$ The higher the position gain k_p, the smaller will be the steady-state error.



8.6.2 PID control law (proportional, integral, derivative control law)

By adding an integral term to the control law to eliminate steady-state error.
It follows:

$$f' = \ddot{x}_d + k_v\dot{e} + k_p e + k_i \int e \, \mathrm{dt}$$

The disturbance force is computed as:

$$f_{dist} = \ddot{e} + k_v\dot{e} + k_p e + k_i \int e \, \mathrm{dt}$$

If $e(t) = 0$ for $t < 0$ it follows:

$$\dot{f}_{dist} = \dddot{e} + k_v\ddot{e} + k_p\dot{e} + k_i e$$

It follows that the steady-state error $e = 0$

8.7 Continuous vs. Discrete Time Control

- Force f of discussed control system is a continuous function of time, but in reality it is a discrete function of time
- Assumptions are valid if the computations are quickly enough
- Considerations in choosing a sufficiently fast sample rate:
  - **Tracking reference time:** The sample rate must be at least twice the bandwidth of reference inputs
  - **Disturbance rejection:** Sample period should be 10 times shorter than the correlation time of the noise
  - **Antialiasing:** Sample rate should be chosen such that the amount of energy that appears in the aliased signal is small
  - **Structural resonances:** Sample rate should be chosen at least twice the natural frequency of the resonances of mechanics

9 Nonlinear Control of manipulators

9.1 Nonlinear and Time-varying Systems

9.1.1 Linearization

- Make a nonlinear time-invariant system linear by using a nonlinear control term to cancel the nonlinearity in the controlled system (Linearization):
  - Compute a nonlinear model-based control law that "cancels" the nonlinearities of the system to be controlled.
  - Reduce the system to a linear system that can be controlled with the simple linear servo law developed for the unit mass.

9.1.2 Control-Law Partitioning of a nonlinear system

Let be given a spring-mass system (with friction and an actuator) with
a nonlinear spring relationship $f = qx^3$ and
the open-loop equation $m\ddot{x} + b\dot{x} + qx^3 = f$
The model-based portion of the control is:

$$f = \alpha f' + \beta$$

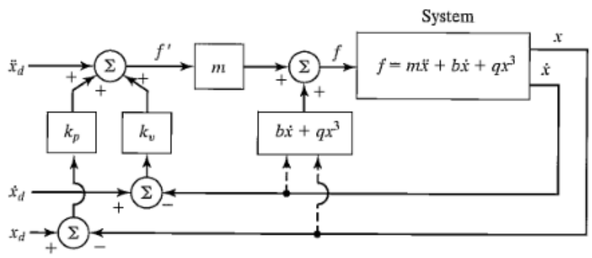
, where

$$\alpha = m$$

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

The servo portion is

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



9.2 Control-Law Partitioning of a (MIMO) Multi-Input, Multi-Output Control System

9.2.1 Model-Based Portion

Let $F = [f_1, \ldots, f_n]^n$ be a vector of forces applied by n actuators.
The Model-based portion of the control is defined as:

$$F = \alpha F' + \beta$$

, where α is a $n \times n$ matrix and F' and β are $n \times 1$ vectors and the model-based portion of the control law is called a linearizing and decoupling control law.

9.2.2 Servo-Portion

Let $E = X_d - X$ a $n \times 1$ vector of the errors in position and
Let $\dot{E}$ a $n \times 1$ vector of the errors in velocity and
let X_d be a vector of the desired positions and
let X be a vector of the current positions.
The servo law is defined as:

$$F' = \ddot{X}_d + K_v\dot{E} + K_p E$$

, where K_v and K_p are $n \times n$ matrices, which are generally chosen to be diagonal with constant gains

9.3 The Control Problem for Manipulators

9.3.1 Manipulator's Dynamic Equation with friction

Let $\tau = M(\Theta)\ddot{\Theta} + V(\Theta, \dot{\Theta}) + G(\Theta)$ be the dynamic equation of a manipulator and
let $F(\Theta, \dot{\Theta})$ be a model of friction dependent on the joint position Θ,
The dynamic equation with friction is defined as:

$$\tau = M(\Theta)\ddot{\Theta} + V(\Theta, \dot{\Theta}) + G(\Theta) + F(\Theta, \dot{\Theta})$$

9.3.2 Partitioning of a Dynamics Equation with Friction

Let $\tau = M(\Theta)\ddot{\Theta} + V(\Theta, \dot{\Theta}) + G(\Theta) + F(\Theta, \dot{\Theta})$ be the dynamic equation of a manipulator with friction
The model-based portion is defined as:

$$\tau = \alpha\tau' + \beta$$

, where

$$\alpha = M(\Theta)$$

$$\beta = V(\Theta, \dot{\Theta}) + G(\Theta) + F(\Theta, \dot{\Theta})$$

The servo law is defined as:

$$\tau' = \ddot{\Theta}_d + K_v\dot{E} + K_p E$$

, where $E = \Theta_d - \Theta$

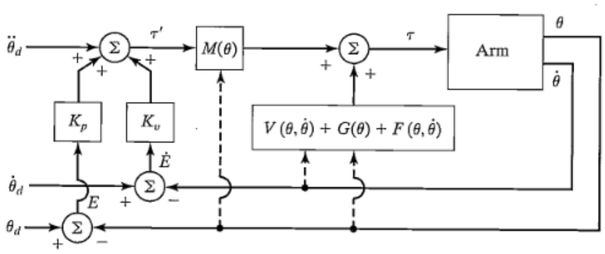
It follows:

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

and on a joint-by-joint basis:

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

, where this equation is possible because the vector equation is decoupled



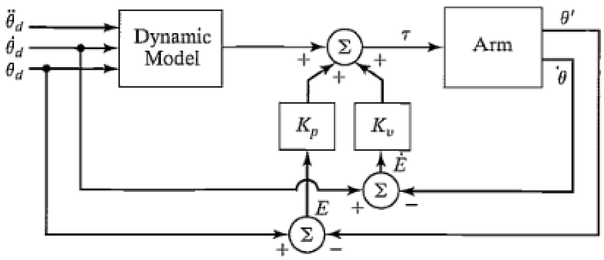
9.4 Practical Considerations

9.4.1 Time required to compute the model

- Equations are valid if system is running in continuous time but system run in discrete time
- Computations in discrete time are often difficult to apply to the case of nonlinear systems

9.4.2 Feedforward Nonlinear Control

- Place the model-based control outside the servo loop
- Results in: $\ddot{E} + M^{-1}(\Theta)K_v\dot{E} + M^{-1}(\Theta)K_pE = 0$
- Doesn't provide complete decoupling



9.4.3 Dual-rate computed-torque implementation

- Express the dynamic model in its configuration space form ($\tau = M(\Theta)\ddot{\Theta} + B(\Theta)[\dot{\Theta}\dot{\Theta}] + C(\Theta)[\dot{\Theta}^2] + G(\Theta)$) so that the dynamic parameters of the manipulator appear as function of manipulator position only
- Express the dynamic model in its configuration space form ($\tau = M(\Theta)\ddot{\Theta} + B(\Theta)[\dot{\Theta}\dot{\Theta}] + C(\Theta)[\dot{\Theta}^2] + G(\Theta)$) so that the dynamic parameters of the manipulator appear as function of manipulator position only
- These functions might then be computed by a background process or by a second computer or be looked up
- The dynamic parameters can be updated at a rate slower than the rate of the closed-loop servo

9.4.4 Lack of knowledge of parameters

- Manipulator dynamic model often not known accurately
- Mass properties of objects picked up by the manipulator often not known

9.5 Current Industrial-Robot Control Systems

- Because of problems with having good knowledge of parameters, present-day manipulators are usually controlled with very simple control laws without decoupling
- Most simple robot controllers do not use a model-based component at all
- Impossible to select fixed gains ($k_p, k_v, \dots$), instead average gain, which approximate critical damping in the center of the robot's workspace
- Gravity terms tend to cause static positioning errors → sometimes a gravity model is included in the control law

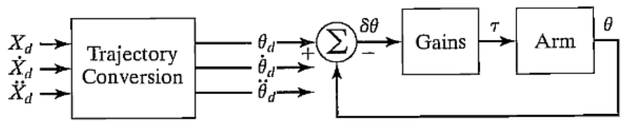
9.6 Lyapunov Stability Analysis

- Applicable to linear and nonlinear systems
- Is concerned with determining the stability of a differential equation $\dot{X} = f(X)$, where X is $m \times 1$ and $f(\cdot)$ could be nonlinear
- The idea is that a positive definite "energy-like" function of state is shown to always decrease or remain constant $\rightarrow$ the system is stable in the sense that the size of the state vector is bounded.
- A system is stable, if a general energy function of the system has the following properties:
  - $v(X)$ has continuous first partial derivatives and $v(X) > 0$ for all X except $v(0) = 0$
  - $\dot{v}(X) \leq 0$. Here, $\dot{v}(X)$ means the change in $v(X)$ along all system trajectories
- If $\dot{v}(x) < 0 \rightarrow$ The system is asymptotically stable

9.7 Cartesian-Based Control Systems

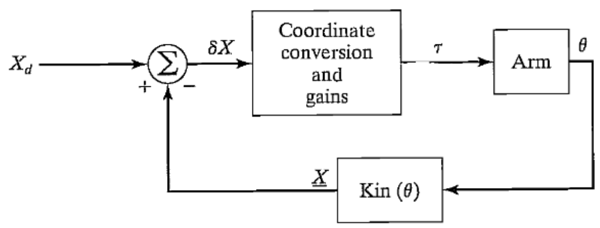
9.7.1 Joint-based control scheme with Cartesian-path input

- Uses the trajectory-conversion process to compute the joint trajectory from the Cartesian-based trajectory
- Afterwards a joint-based control scheme is used
- Usually just the solution for Θ_d is performed by inverse kinematics and then the joint velocities and accelerations are computed numerically by first and second differences.



9.7.2 Concept of a Cartesian-based control scheme

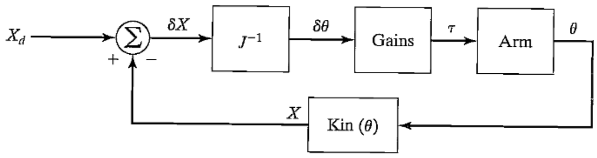
- The sensed position of the manipulator is transformed by means of the kinematic equations into a Cartesian description of position. This Cartesian description is then compared to the desired Cartesian position in order to form errors in Cartesian space
- The kinematics and other transformations are now inside the loop $\rightarrow$ lower sampling frequency than joint-based control



9.7.3 Inverse-Jacobian controller

Scheme:

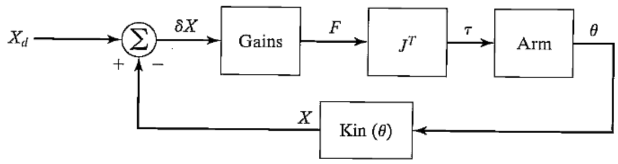
1. Cartesian position is compared to the desired position to form an error, δX, in Cartesian space.
2. This error may be mapped into a small displacement in joint space by means of the inverse Jacobian.
3. The resulting errors in joint space, $\delta\Theta$, are then multiplied by gains to compute torques that will tend to reduce these errors.
4. The sensed position of the manipulator is transformed by means of the kinematic equations into a Cartesian description of position.



9.7.4 Transpose-Jacobian controller

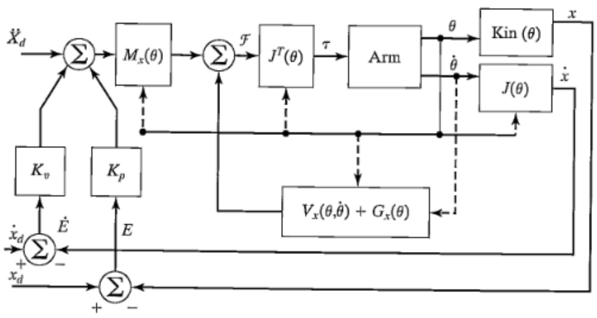
Scheme:

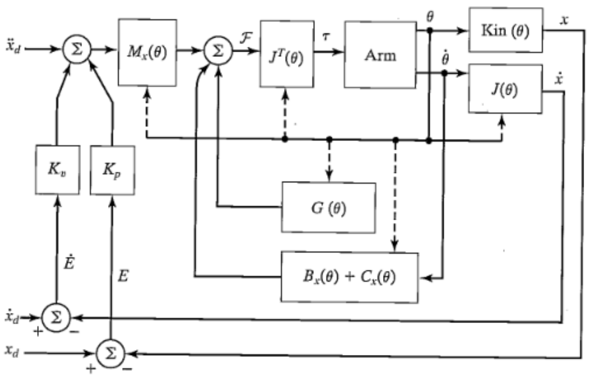
1. Cartesian position is compared to the desired position to form an error, δX, in Cartesian space.
2. This Cartesian error vector is multiplied by a gain to compute a Cartesian force vector.
3. This Cartesian force vector (actually a force—moment vector) is then mapped through the Jacobian transpose in order to compute the equivalent joint torques that would tend to reduce the observed errors.
4. The sensed position of the manipulator is transformed by means of the kinematic equations into a Cartesian description of position.



9.7.5 Cartesian decoupling scheme

- Linearizing and decoupling by using $\mathcal{F} = M_X(\Theta)\ddot{\chi} + V_X(\Theta, \dot{\Theta}) + G_X(\Theta)$, where $\mathcal{F}$ is force-moment vector acting on the end-effector and χ is an Cartesian vector representing position and orientation of the end-effector
- Computing of the joint torques by using $\tau = \mathcal{J}^T(\Theta)\mathcal{F}$





9.8 Adaptive Control

- When the parameters in the model do not match the parameters of the real device, servo errors will result
- These servo errors could be used to drive some adaptation scheme that attempts to update the values of the model parameters until the errors disappear

10 Force control of Manipulators

- By measuring and control contact forces generated at the hand, absolute errors in the position of the manipulator and the manipulated objects are not as important as they would be in a purely position-controlled system.

10.1 A Framework for Control in Partially Constrained Tasks

- Tasks can be broken down into subtasks that are defined by a particular contact situation occurring between the end-effector and the work environment

10.1.1 Natural constraints

- Constraints that result from the particular mechanical and geometric characteristics of the task configuration
- Natural constraint means that these constraints arise naturally from the particular contacting situation.
- Defined by a
  - natural position (position or orientation) constraint (or often as a "velocity equals zero" constraint) ${}^{C}\nu$ (constraint reachable positions by the environment)
  - natural force (force or moment) constraint ${}^{C}\mathcal{F}$ (constraint forces that would cause derivative position change)
- Described in terms of frame $\{C\}$, the constraint frame, which is located in the task-relevant location.
- $\{C\}$ could be fixed in the environment or could move with the end-effector

10.1.2 Artificial constraint

- Introduced in accordance with the natural constraint to specify desired motions of force application.
- Each time the user specifies a desired trajectory in either position or force, an artificial constraint is defined.

10.1.3 Assembly Strategy

- Term that refers to a sequence of planned artificial constraints that will cause the task to proceed in a desirable manner.
- Such strategies must include methods by which the system can detect a change in the contacting situation.

10.2 The Hybrid Position/Force Control Problem

The hybrid position/force controller must solve three problem:

- Position control of a manipulator along directions in which a natural force constraint exists (due to no forces to react against).
- Force control of a manipulator along directions in which a natural position constraint exists.
- A scheme to implement the arbitrary mixing of these modes along orthogonal degrees of freedom of an arbitrary frame $\{C\}$.

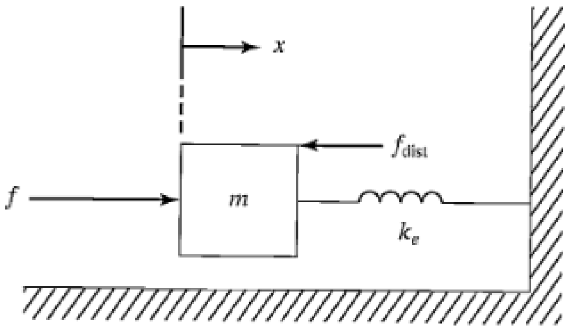
10.3 Force Control of a Mass-Spring System

Let be given a spring-mass system, where
m be the mass of the block and
k_e be the stiffness of the spring and
f_{dist} be an unknown disturbance force.
The equation describing this physical system is:

$$f = m\ddot{x} + k_e x + f_{dist}$$

or with $f_e = k_e x$

$$f = m k_e^{-1} \ddot{f}_e + f_e + f_{dist}$$



10.3.1 Control-Law Partitioning

Model-based portion:
Let be given a spring-mass system with the describing function $f = m k_e^{-1} \ddot{f}_e + f_e + f_{dist}$ The model-based portion of the control is:

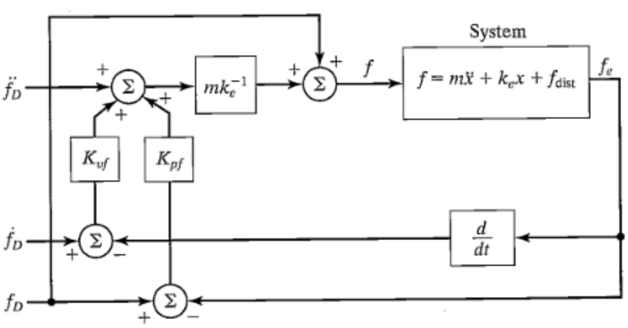
$$f = m k_e^{-1} [\ddot{f}_d + k_{vf} \dot{e}_f + k_{pf} e_f] + f_e + f_{dist}$$

, where

$$\alpha = m k_e^{-1}$$
$$\beta = f_e + f_{dist}$$
$$e_f = f_d - f_e$$

Steady-state Error:
The steady-state error is computed as:

$$e_f = \frac{f_{dist}}{\alpha}$$

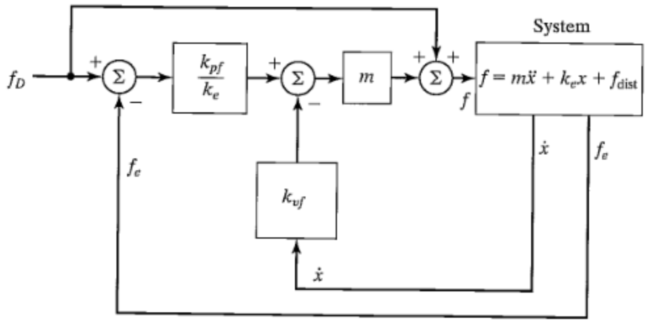


Using f_d instead of $f_e + f_{dist}$:
The model-based portion of the control is:

$$f = m k_e^{-1} [\ddot{f}_d + k_{vf} \dot{e}_f + k_{pf} e_f] + f_d$$

The steady-state error is computed as:

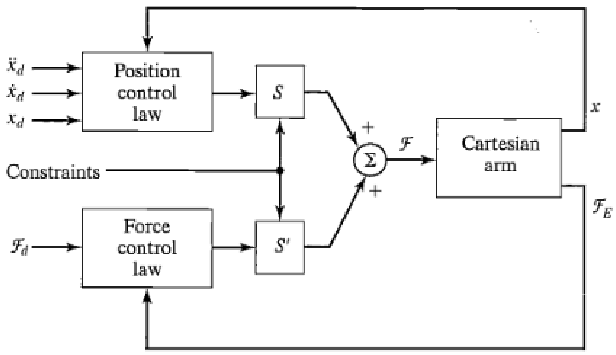
$$e_f = \frac{f_{dist}}{1+\alpha}$$



10.4 The Hybrid Position/Force Control Scheme

10.4.1 A Cartesian manipulator aligned with $\{C\}$

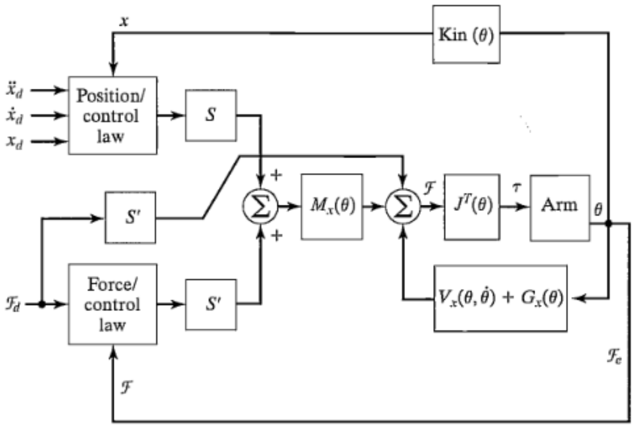
- Let be given a 3 DOF manipulator with prismatic joints acting in the $\hat{X}, \hat{Y}$ and $\hat{Z}$ directions
- If we wish to be able to switch the nature of the constraint surface, we specify a complete position trajectory in all 3 DOF and a force trajectory in 3 DOF and set modes to indicate which components of which trajectory to follow.
- A mode uses the matrices S and S' to control if position or force is used to control each joint of the Cartesian arm, where S and S' are diagonal and a one in S means, that the position control is in effect on this joint (otherwise a zero is in S) and a one in S' means, that the force control is in effect on this joint (otherwise a zero is in S').



E.g. $S = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$, $S' = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

10.4.2 A general manipulator

- Through use of a dynamic model written in Cartesian space, it is possible to control so that the combined system of the actual manipulator and computed model appear as a set of independent, uncoupled unit masses.
- For use in hybrid control scheme, the dynamics, the Jacobian and the kinematics are computed with respect to $\{C\}$.



10.5 Current Industrial-Robot Control Schemes

10.5.1 Passive compliance

- E.g. RCC:
  - inserted between wrist and end-effector
  - is essentially a spring with 6 DOF
  - Allows the end-effector to have a lower stiffness
- Used in industrial applications of manipulators

11 Other Basics

11.1 Screw notation

$\text{Screw}_Q(r, \phi)$ stands for the combination of a translation along an axis Q by a distance r and a rotation about the same axis by an angle ϕ

11.2 Joint axis vectors

$$\hat{X} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \qquad \hat{Y} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \qquad \hat{Z} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

11.3 three-link planar arm

A manipulator build of three parallel revolute joints is called a three-link planar arm.

11.4 Workspace

- Volume of space that the robot end-effector of the manipulator can reach.

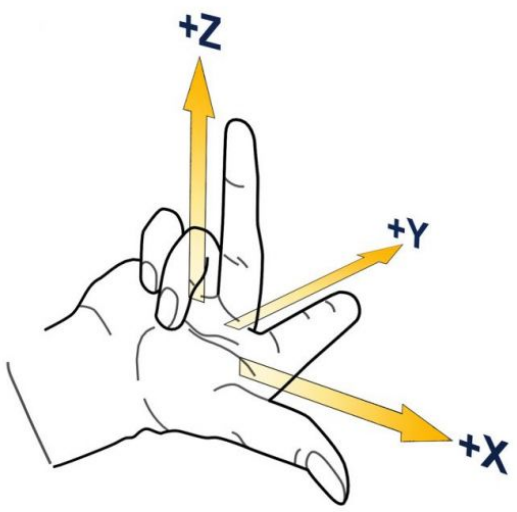
11.4.1 Dextrous workspace

- Volume of space that the robot end-effector can reach with all orientations.

11.4.2 Reachable workspace

- Volume of space that the robot can reach in at least one orientation.

11.5 Right-Hand Coordinate System



11.6 Schematic notation

11.6.1 Parallel axes

Double hash marks on a simple schematic notation indicate that the axis are mutually parallel

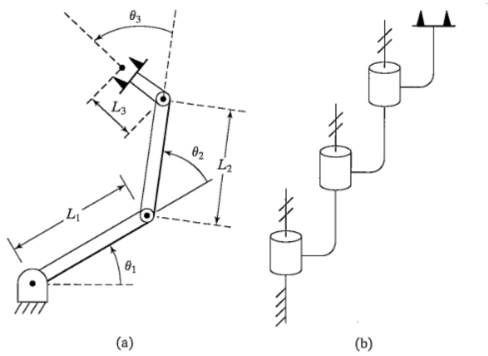


FIGURE 3.6: A three-link planar arm. On the right, we show the same manipulator by means of a simple schematic notation. Hash marks on the axes indicate that they are mutually parallel.

11.6.2 orthogonal axes

12 Mathematical Basics

12.1 Rotation Matrix

Rotation about x-axis: $R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}$

Rotation about y-axis: $R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$

Rotation about z-axis: $R_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Determinant: $\det R = +1$

12.2 Trigonometric Functions

12.2.1 Arcustangens 2

$$\operatorname{atan2}(y,x) = \begin{cases} \arctan(\frac{y}{x}) & \text{if } x > 0 \\ \arctan(\frac{y}{x}) + \pi & \text{if } x < 0 \text{ and } y > 0 \\ \pm\pi & \text{if } x < 0 \text{ and } y = 0 \\ \arctan(\frac{y}{x}) - \pi & \text{if } x < 0 \text{ and } y < 0 \\ +\frac{\pi}{2} & \text{if } x = 0 \text{ and } y > 0 \\ -\frac{\pi}{2} & \text{if } x = 0 \text{ and } y < 0 \\ \text{undefined} & \text{if } x = 0 \text{ and } y = 0 \end{cases}$$

12.2.2 Simplifications

$$\cos(\alpha + \beta) = \cos\alpha \cdot \cos\beta - \sin\alpha \cdot \sin\beta$$
$$\sin(\alpha + \beta) = \sin\alpha \cdot \cos\beta + \cos\alpha \cdot \sin\beta$$

13 Notations

13.1 CoSinus

$$\cos\theta_i \equiv c\theta_i \equiv c_i \qquad \text{for } i \in \mathbb{N}$$
$$\sin\theta_i \equiv s\theta_i \equiv s_i \qquad \text{for } i \in \mathbb{N}$$

13.2 CoSinus2

$$c_{12} = c_1 c_2 - s_1 s_2 = \cos(\theta_1 + \theta_2)$$
$$s_{12} = c_1 s_2 + s_1 c_2 = \sin(\theta_1 + \theta_2)$$

13.3 CoSinus3

$$c_{123} = c_1 s_2 s_3 + s_1 s_2 c_3 - s_1 c_2 s_3 - c_2 c_3 c_4$$
$$s_{123} = s_1 c_2 c_3 + c_1 c_2 s_3 - c_1 s_2 c_3 - s_1 s_2 s_3$$

Note

This is a summary of the lecture Robotics of the Technical University Munich. This lecture was presented by Burschka D. in the winter semester 2018/19. This summary was created by Gaida B. All provided information is without guarantee.

References

John J. Craig. *Introduction to Robotics Mechanics and Control*. Pearson Education, Inc. New York. 2005