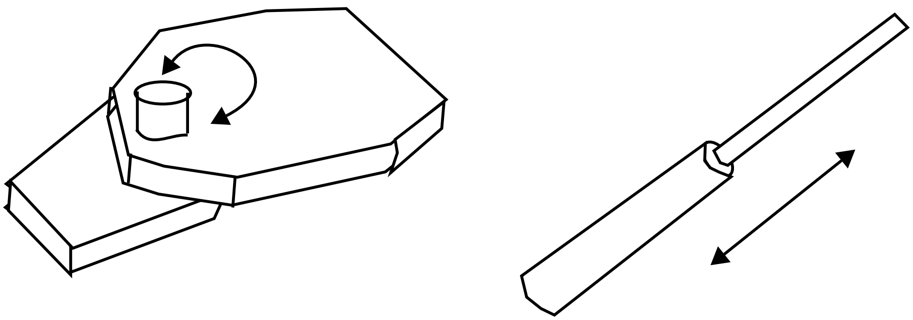


a) Revolute joint.

b) Prismatic joint.

FIGURE 127. Typical Joints used in computer animation

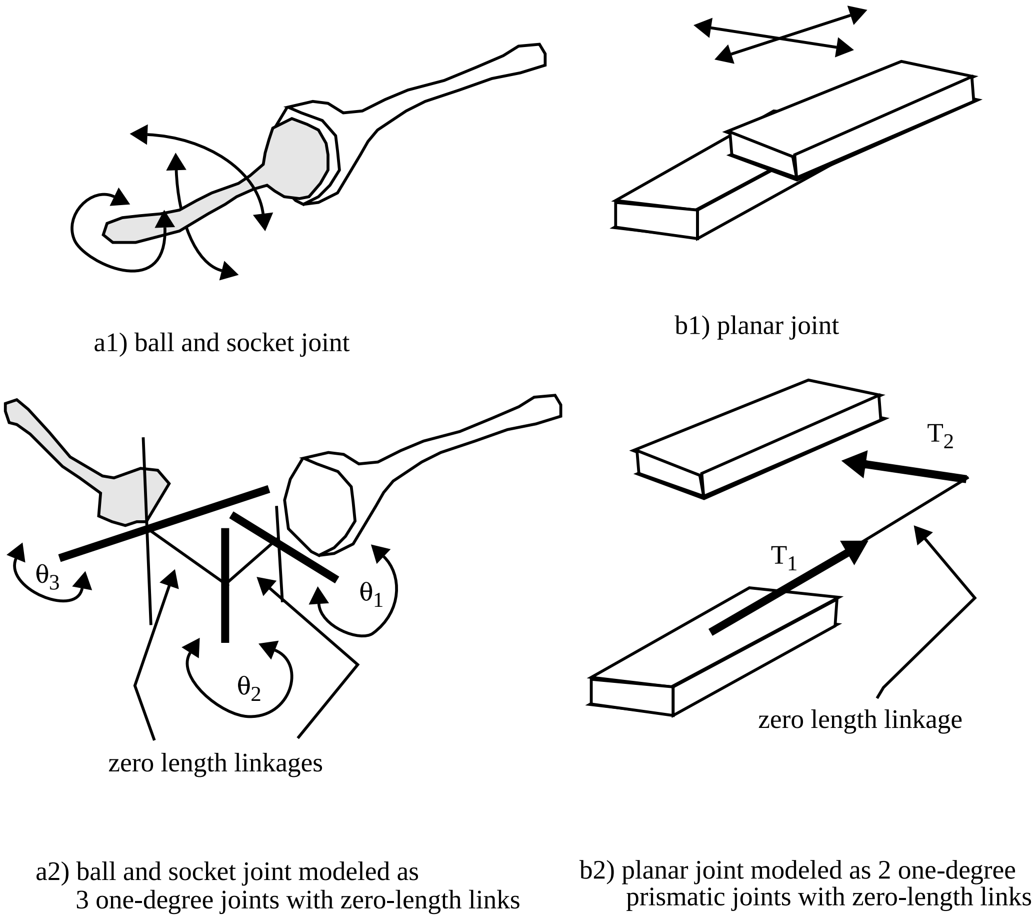


FIGURE 128. Modeling complex joints

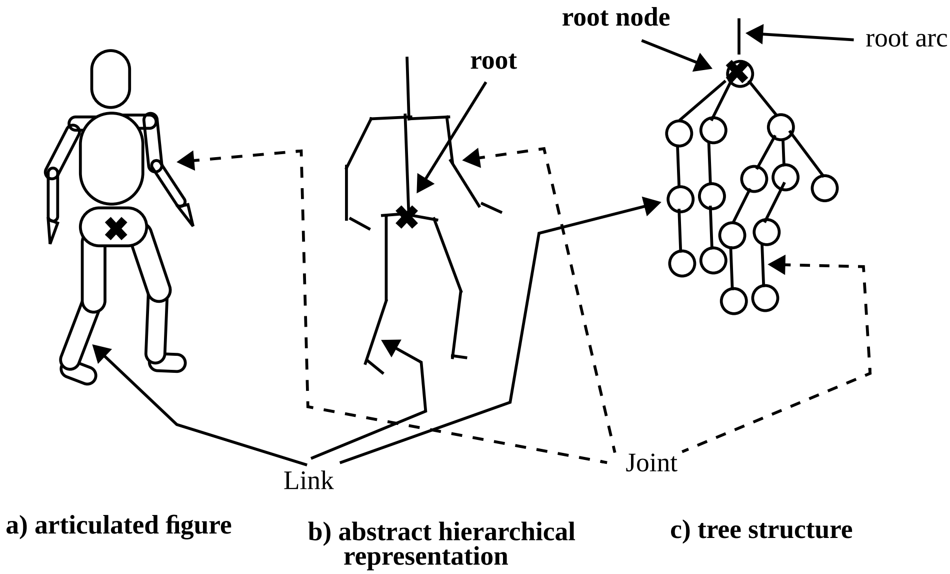


FIGURE 129. **Example tree structure representing a hierarchical structure.**

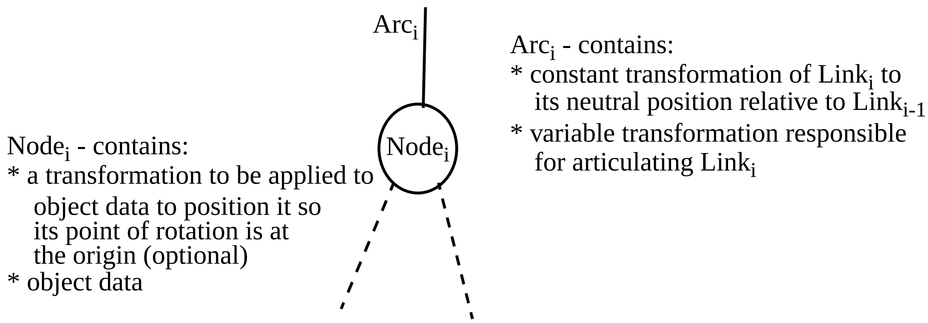


FIGURE 130. **Arc and node definition.**

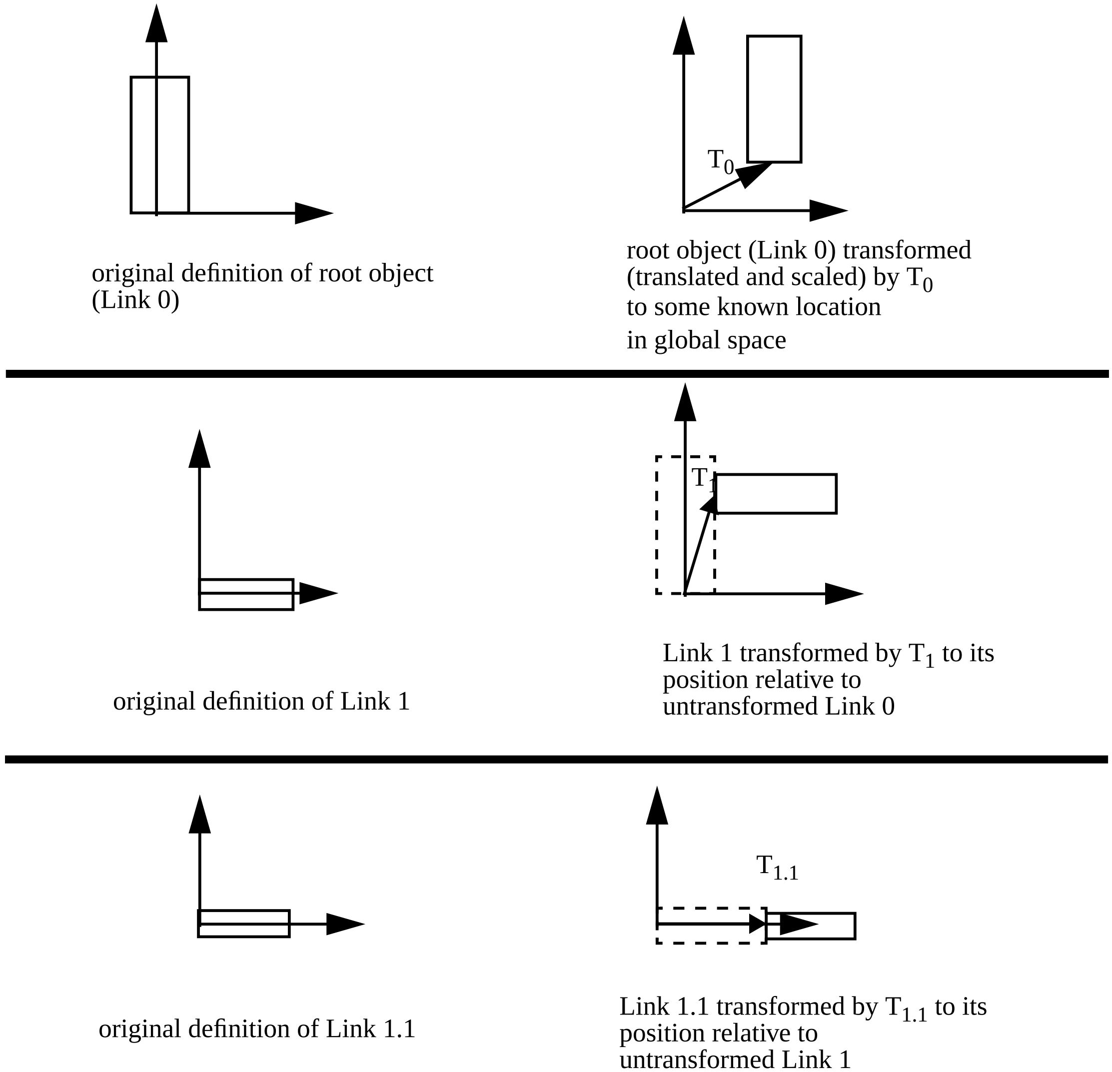


FIGURE 131. Example hierarchical model.

.

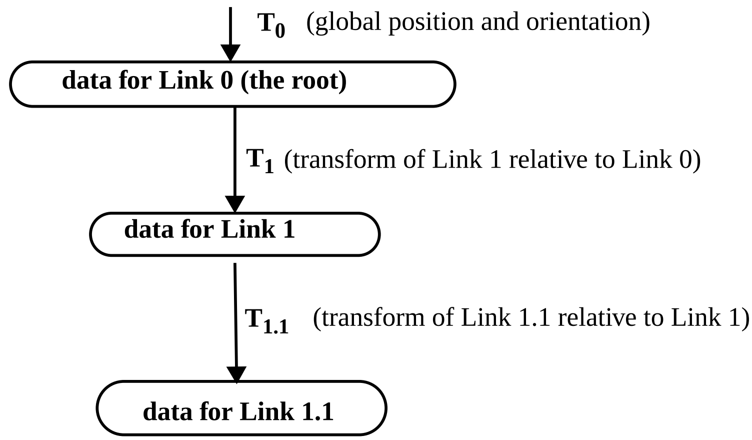


FIGURE 132. Example tree structure.

$$V_0' = T_0 \cdot V_0 \quad \text{(EQ 75)}$$

$$V_1' = T_0 \cdot T_1 \cdot V_1 \quad \text{(EQ 76)}$$

$$V_{1.1}' = T_0 \cdot T_1 \cdot T_{1.1} \cdot V_{1.1} \quad \text{(EQ 77)}$$

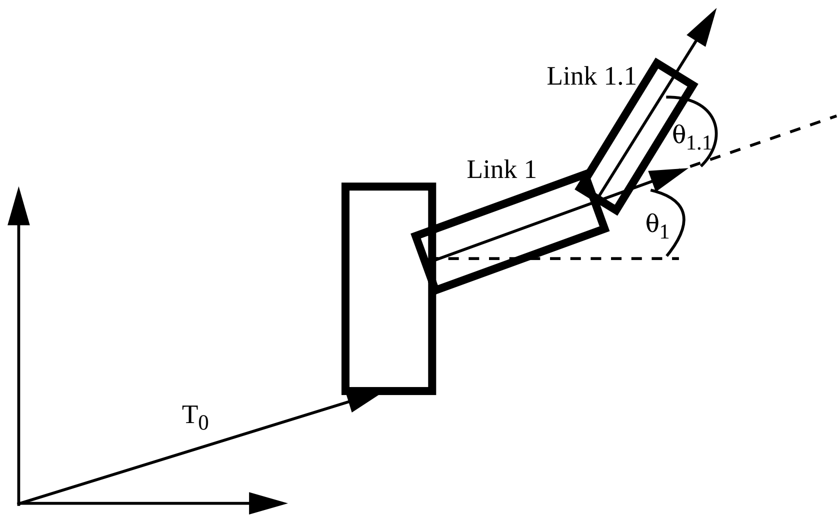


FIGURE 133. **Variable rotations at the joints.**

.

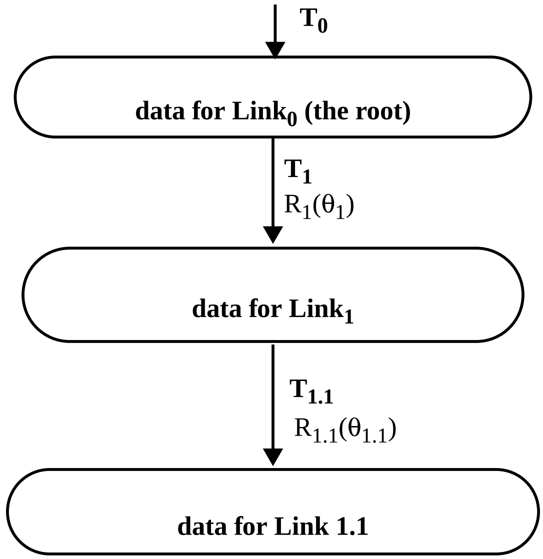


FIGURE 134. **Hierarchy showing joint rotations.**

$$V_1' = T_0 \cdot T_1 \cdot R_1(\theta_1) \cdot V_1 \quad \text{(EQ 78)}$$

$$V'_{1.1} = T_0 \cdot T_1 \cdot R_1(\theta_1) \cdot T_{1.1} \cdot R_{1.1}(\theta_{1.1}) \cdot V_{1.1} \quad \text{(EQ 79)}$$

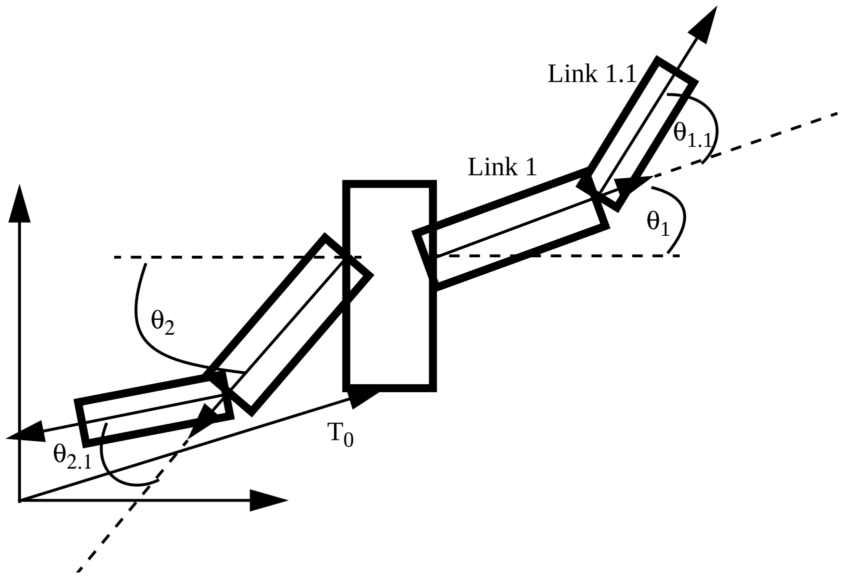


FIGURE 135. **Hierarchy with two appendages.**

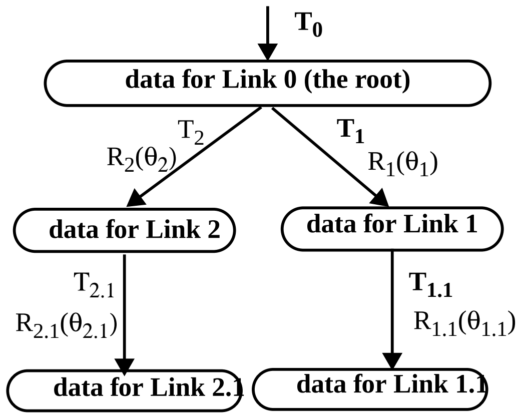


FIGURE 136. **Tree structure corresponding to hierarchy with two appendages.**

```
/* the node structure */
typedef struct node {
        object      *           obj;        /*  pointer to object data structure */
        struct arc  *           arc_array;  /* array of arcs emanating downward from node */
        int                     num_arc;    /* number of arcs in array */
} node_struct;

/* the arc structure */
typedef struct arc {
        trans_mat               rot;        /* joint rotation matrix */
        trans_mat               m;          /* orientation and position matrix */
        node_struct             *nptr;      /* pointer to node below arc */
} arc_struct;

/* the highest structure of the tree is the root arc holding the global transforms */
/* the high level routine simply calls for the (recursive) evaluation of the root node */
eval_tree(struct arc rootArc)
{
        eval_node(rootArc->m,rootArc->node);        /* recursively evaluate the root node */
}

/* the recursive evaluation routine */
eval_node(trans_mat m,node_struct node);
{
        trans_mat               temp_m;         /* temporary transformation */

        concat_tm(node->m,m,&temp_m);           /* concatenate current and node transform */
        transf_obj(obj,temp_m,&temp_obj);       /* transform object */
        display_obj(temp_obj);                  /* display transformed object */

        /* loop over each arc emanating from node and recursively evaluate
        /* the attached node */
        for (l=0; l<node->num_arc; l++) {
                premul_tm(node->arc_array[l]->m,temp_m);
                premul_tm(node->arc_array[l]->rot,temp_m);
                eval_node(temp_m,node->arc_array[l]->node);
        }
}
```

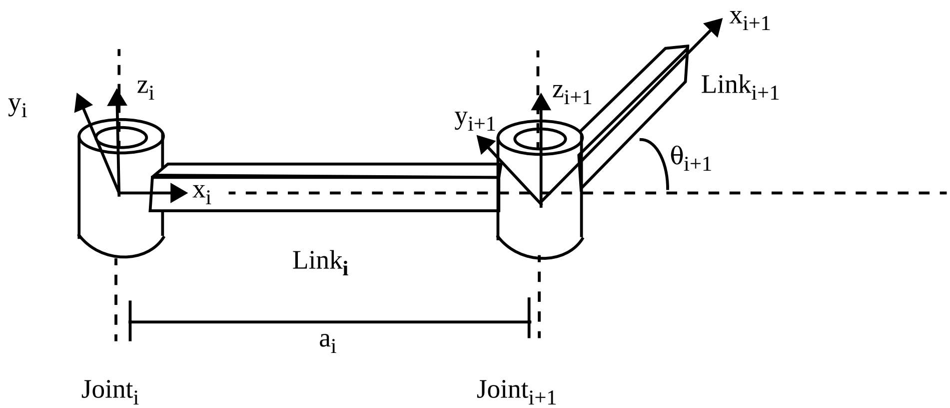


FIGURE 137. Denavit-Hartenberg parameters for planar joints.

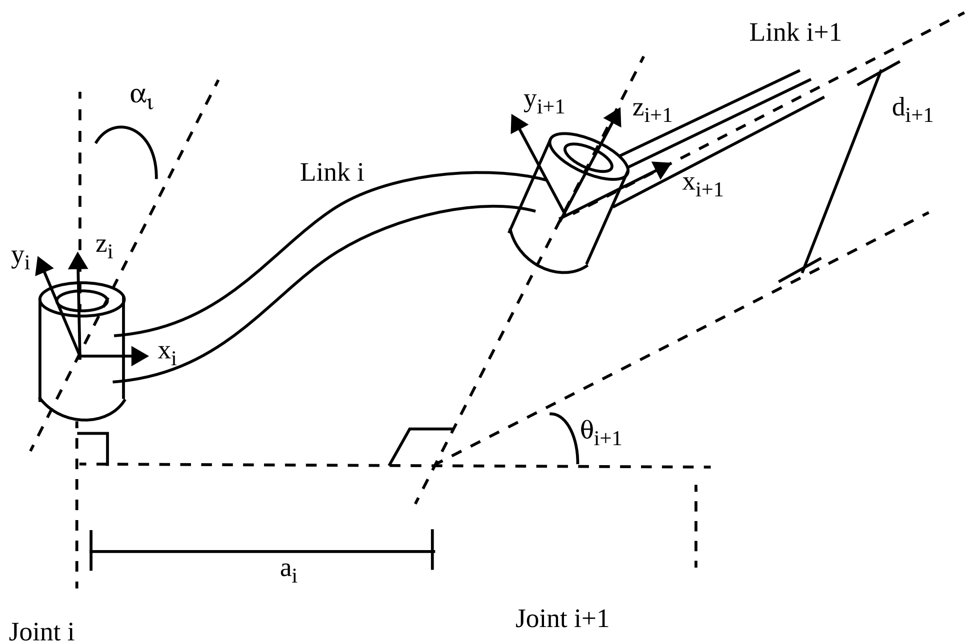


FIGURE 138. Denavit-Hartenberg Parameters

name	symbol	description
link offset	d_i	the distance from x_{i-1} to x_i along z_i
joint angle	θ_ι	the angle between x_{i-1} and x_i about z_i
link length	a_i	the distance from z_i to z_{i+1} along x_i
link twist	α_ι	the angle between z_i and z_{i+1} about x_i

TABLE 1. Denavit-Hartenberg joint parameters for joint i.

.

name	symbol	description	screw transformation
link offset	d_{i+1}	the distance from x_i to x_{i+1} along z_{i+1}	relative to z_{i+1}
joint angle	$\theta_{\iota+1}$	the angle between x_i and x_{i+1} about z_{i+1}	“
link length	a_i	the distance from z_i to z_{i+1} along x_i	relative to x_i
link twist	α_ι	the angle between z_i and z_{i+1} about x_i	“

TABLE 2. Parameters which relate the ith frame and the i+1st frame.

$$V_i = T_X(a_i)R_X(\alpha_i)T_Z(d_{i+1})R_Z(\theta_{i+1})V_{i+1}$$

$$R_Z(\theta_{i+1}) = \begin{bmatrix} \cos(\theta_{i+1}) & -\sin(\theta_{i+1}) & 0 & 0 \\ \sin(\theta_{i+1}) & \cos(\theta_{i+1}) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_Z(d_{i+1}) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_{i+1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \text{(EQ 80)}$$

$$R_X(\alpha_i) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha_i) & -\sin(\alpha_i) & 0 \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_X(a_i) = \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$V_i = M_i^{i+1}V_{i+1}$$

$$M_i^{i+1} = \begin{bmatrix} \cos(\theta_{i+1}) & -\sin(\theta_{i+1}) & 0 & a_i \\ \cos(\alpha_i)\cdot\sin(\theta_{i+1}) & \cos(\alpha_i)\cdot\cos(\theta_{i+1}) & -\sin(\alpha_i) & -d_{i+1}\cdot\sin(\alpha_i) \\ \sin(\alpha_i)\cdot\sin(\theta_{i+1}) & \sin(\alpha_i)\cdot\cos(\theta_{i+1}) & \cos(\alpha_i) & d_{i+1}\cdot\cos(\alpha_i) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

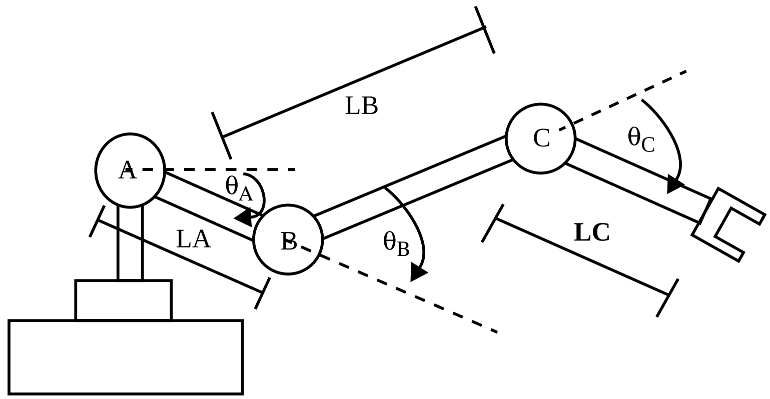


FIGURE 139. Simple Manipulator Using Three Revolute Joints

Joint \ Parameter	Link Displacement	Joint Angle	Link Length	Link Twist
A	0	θ_A	0	0
B	0	θ_B	LA	0
C	0	θ_C	LB	0

TABLE 3. Parameters for three revolute joint armature

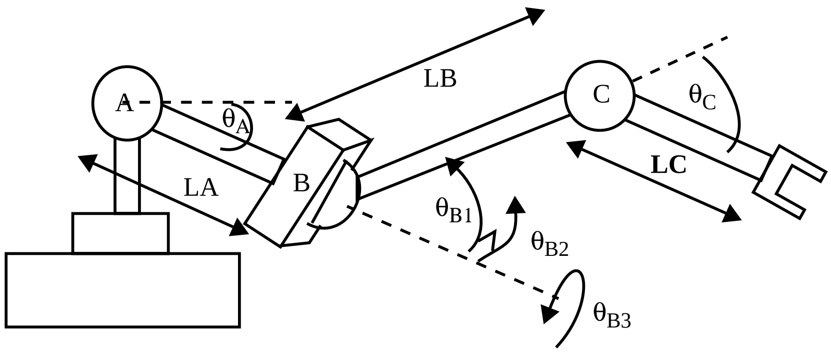


FIGURE 140. Incorporating a Ball-and-Socket Joint

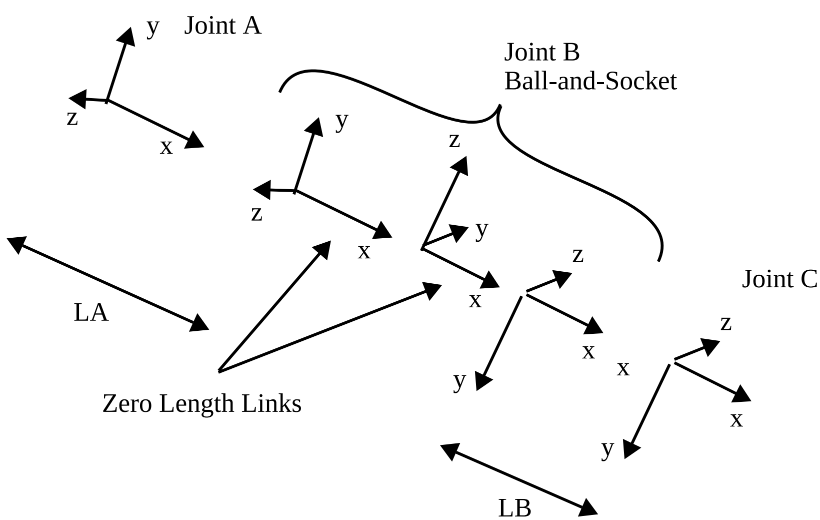


FIGURE 141. Coordinate axes induced by the DH representation of a ball-and-socket joint.

.

Joint \ Parameter	Link Displacement	Joint Angle	Link Length	Link Twist
A	0	θ_A	0	0
B1	0	θ_{B1}	LA	90
B2	0	$90+\theta_{B2}$	0	90
B3	0	θ_{B3}	0	0
C	0	θ_C	LB	0

TABLE 4. Joint parameters for ball-and-socket joint.

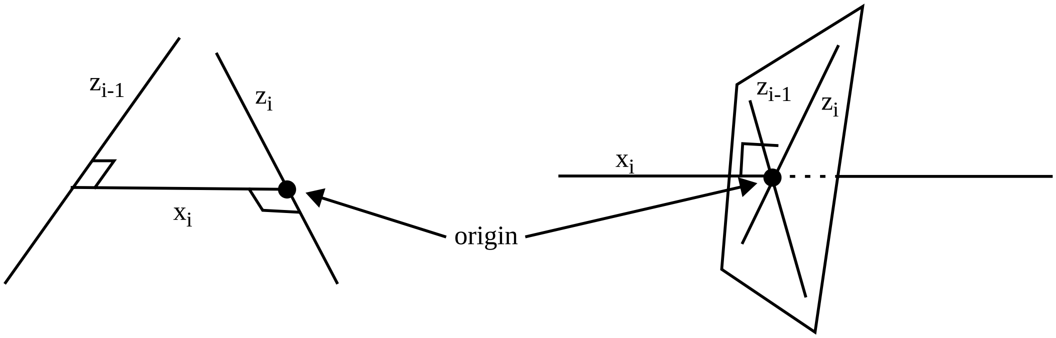


FIGURE 142. Determining the origin and x-axis of the ith frame

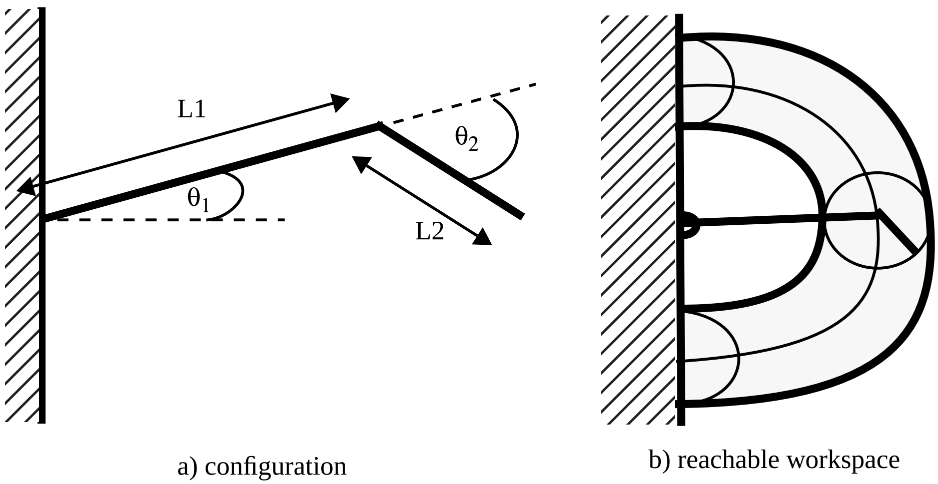
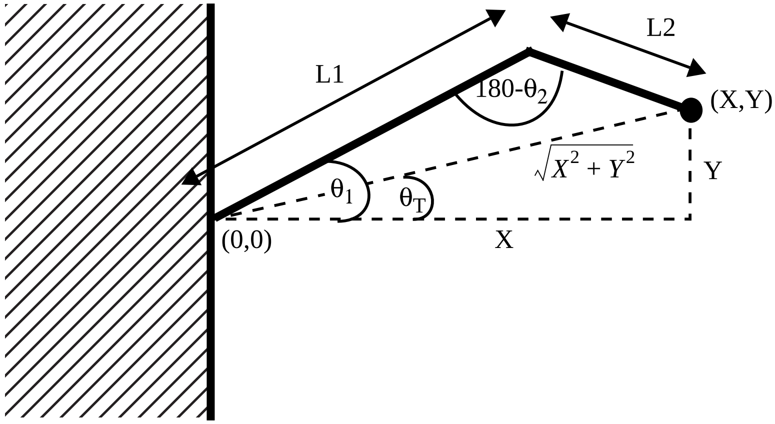


FIGURE 143. Simple linkage.



$$\cos(\theta_T) = \frac{X}{\sqrt{X^2 + Y^2}}$$

$$\theta_T = \operatorname{acos}\left(\frac{X}{\sqrt{X^2 + Y^2}}\right)$$

$$\cos(\theta_1 - \theta_T) = -\frac{L2^2 - L1^2 - (X^2 + Y^2)}{2 \cdot L1 \cdot \sqrt{X^2 + Y^2}} \qquad \text{(law of cosines)}$$

$$\theta_1 = \operatorname{acos}\left(\frac{L2^2 - L1^2 - (X^2 + Y^2)}{2 \cdot L1 \cdot \sqrt{X^2 + Y^2}}\right) + \theta_T$$

$$\cos(180 - \theta_2) = \frac{(X^2 + Y^2) - L1^2 - L2^2}{2 \cdot L1 \cdot L2} \qquad \text{(law of cosines)}$$

$$\theta_2 = -\operatorname{acos}\left(\frac{(X^2 + Y^2) - L1^2 - L2^2}{2 \cdot L1 \cdot L2}\right)$$

FIGURE 144. Equations used in solving simple inverse kinematic problem.

$$\begin{aligned} y_1 &= f_1(x_1, x_2, x_3, x_4, x_5, x_6) \\ y_2 &= f_2(x_1, x_2, x_3, x_4, x_5, x_6) \\ y_3 &= f_3(x_1, x_2, x_3, x_4, x_5, x_6) \\ y_4 &= f_4(x_1, x_2, x_3, x_4, x_5, x_6) \\ y_5 &= f_5(x_1, x_2, x_3, x_4, x_5, x_6) \\ y_6 &= f_6(x_1, x_2, x_3, x_4, x_5, x_6) \end{aligned} \quad \text{(EQ 81)}$$

$$\delta y_i = \frac{\delta f_i}{\partial x_1} \cdot \delta x_1 + \frac{\delta f_i}{\partial x_2} \cdot \delta x_2 + \frac{\delta f_i}{\partial x_3} \cdot \delta x_3 + \frac{\delta f_i}{\partial x_4} \cdot \delta x_4 + \frac{\delta f_i}{\partial x_5} \cdot \delta x_5 + \frac{\delta f_i}{\partial x_6} \cdot \delta x_6 \quad \text{(EQ 82)}$$

$$Y = F(X) \quad \text{(EQ 83)}$$

$$\delta Y = \frac{\partial F}{\partial X} \cdot \partial X \quad \text{(EQ 84)}$$

$$\dot{Y} = J(X) \cdot \dot{X} \quad \text{(EQ 85)}$$

$$V = J(\theta)\dot{\theta} \quad \text{(EQ 86)}$$

$$V = [v_x, v_y, v_z, \omega_x, \omega_y, \omega_z]^T \quad \text{(EQ 87)}$$

$$\dot{\theta} = [\dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3, \ldots, \dot{\theta}_n]^T \quad \text{(EQ 88)}$$

$$J = \begin{bmatrix} \frac{\partial v_x}{\partial \theta 1} & \frac{\partial v_x}{\partial \theta 2} & \cdots & \frac{\partial v_x}{\partial \theta n} \\ \frac{\partial v_y}{\partial \theta 1} & \frac{\partial v_y}{\partial \theta 2} & \cdots & \frac{\partial v_y}{\partial \theta n} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial \omega_z}{\partial \theta 1} & \frac{\partial \omega_z}{\partial \theta 2} & \cdots & \frac{\partial \omega_z}{\partial \theta n} \end{bmatrix} \quad \text{(EQ 89)}$$

.

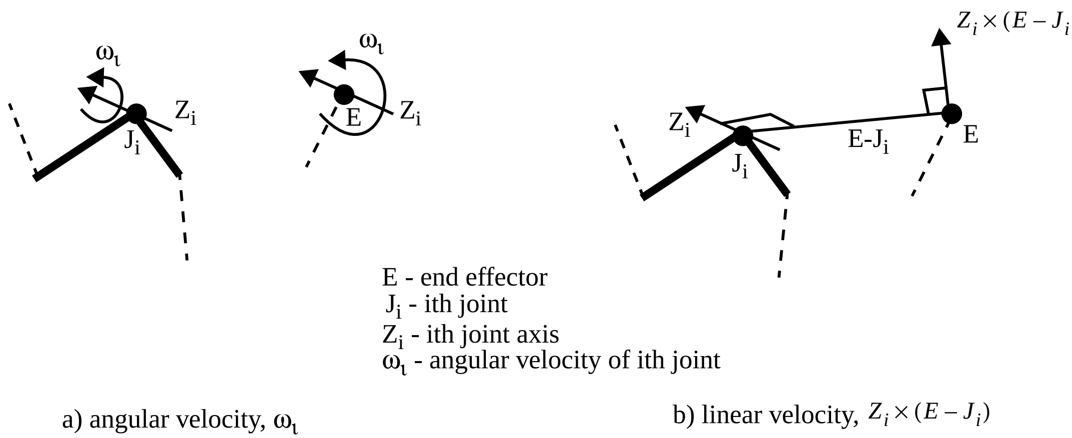


FIGURE 145. Angular and linear velocities induced by joint axis rotation.

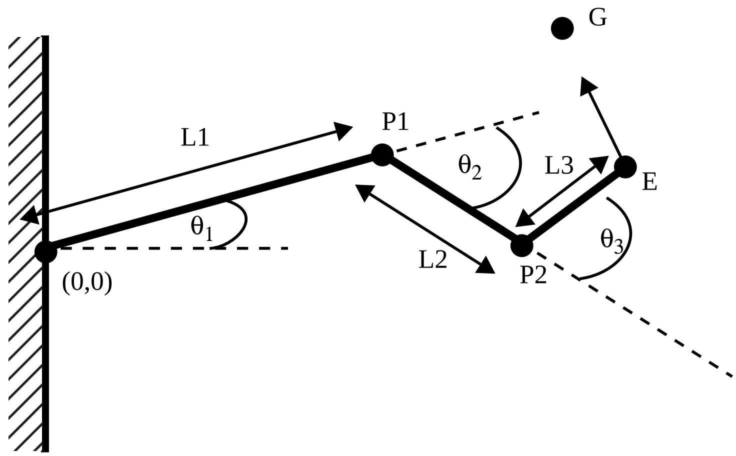


FIGURE 146. Planar, Three Joint Manipulator.

.

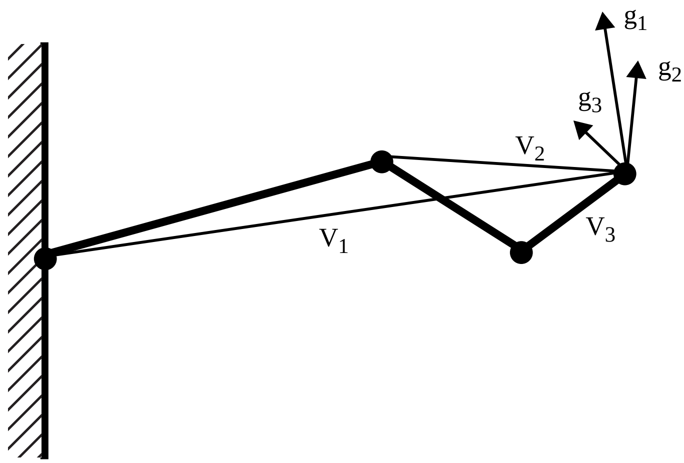
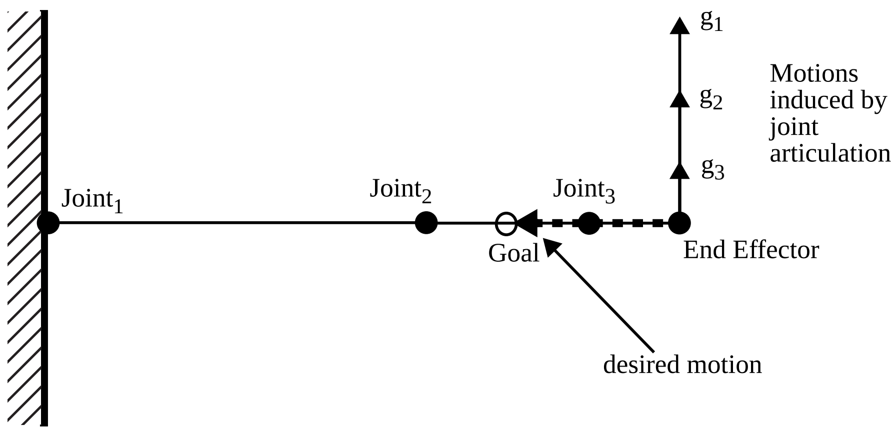


FIGURE 147. **Instantaneous changes in position induced by joint angle rotations.**

$$\begin{bmatrix} (G-E)_x \\ (G-E)_y \\ (G-E)_z \end{bmatrix} = \begin{bmatrix} ((0,0,1)\times E)_x & (0,0,1)\times(E-P1)_x & (0,0,1)\times(E-P2)_x \\ ((0,0,1)\times E)_y & (0,0,1)\times(E-P1)_y & (0,0,1)\times(E-P2)_y \\ ((0,0,1)\times E)_z & (0,0,1)\times(E-P1)_z & (0,0,1)\times(E-P2)_z \end{bmatrix} \cdot \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} \qquad \textbf{(EQ 90)}$$

$$V = J\dot{\theta} \qquad \textbf{(EQ 91)}$$

$$J^{-1}V = \dot{\theta} \qquad \textbf{(EQ 92)}$$


g1
g2
g3
Motions induced by joint articulation
Joint1
Joint2
Joint3
Goal
End Effector
desired motion

$$
\begin{aligned}
V &= J\dot{\theta} \\
J^T V &= J^T J\dot{\theta} \\
(J^T J)^{-1} J^T V &= (J^T J)^{-1} J^T J\dot{\theta} \\
J^+ V &= \dot{\theta}
\end{aligned}
\tag{EQ 93}
$$

$$
\begin{aligned}
J^+ V &= \dot{\theta} \\
J^T (JJ^T)^{-1} V &= \dot{\theta} \\
\beta &= (JJ^T)^{-1} V
\end{aligned}
\tag{EQ 94}
$$

$$
(JJ^T)\beta = V \tag{EQ 95}
$$

$$
J^T\beta = \dot{\theta} \tag{EQ 96}
$$

$$
\dot{\theta} = (J^+ J - I)z \tag{EQ 97}
$$

$$
\begin{aligned}
V &= J\dot{\theta} \\
V &= J(J^+ J - I)z \\
V &= (JJ^+ J - J)z \\
V &= (J - J)z \\
V &= 0 \cdot z \\
V &= 0
\end{aligned}
\tag{EQ 98}
$$

$$
H = \sum_{i=1}^{n} \alpha_i \cdot (\theta_i - \theta_{ci})^{\Psi} \tag{EQ 99}
$$

$$z = \nabla_\theta H = \frac{dH}{d\theta} = \psi \sum_{i=1}^{n} \alpha_i \cdot (\theta_i - \theta_{ci})^{\psi - 1}$$ **(EQ 100)**

$$\dot{\theta} = J^+ V + (J^+ J - I) \nabla_\theta H$$ **(EQ 101)**

$$\dot{\theta} = J^+ V + (J^+ J - I) \nabla_\theta H$$
$$\dot{\theta} = J^+ V + J^+ J \nabla_\theta H - I \nabla_\theta H$$
$$\dot{\theta} = J^+ (V + J \nabla_\theta H) - \nabla_\theta H$$ **(EQ 102)**
$$\dot{\theta} = J^T (J J^T)^{-1} (V + J \nabla_\theta H) - \nabla_\theta H$$
$$\dot{\theta} = J^T [(J J^T)^{-1} (V + J \nabla_\theta H)] - \nabla_\theta H$$

$$\beta = (J J^T)^{-1} (V + J \nabla_\theta H)$$

$$\dot{\theta} = J^T \beta - \nabla_\theta H$$ **(EQ 103)**

$$V + J \nabla_\theta H = (J J^T) \beta$$ **(EQ 104)**

.

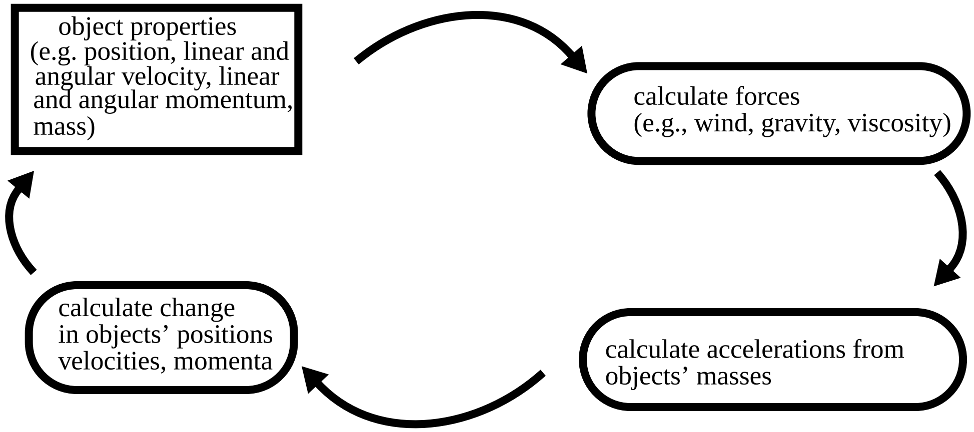


FIGURE 148. **Rigid body simulation update cycle.**

$$x(t + \Delta t) = x(t) + v(t) \cdot \Delta t \qquad \text{(EQ 105)}$$

$$v(t + \Delta t) = v(t) + a(t) \cdot \Delta t \qquad \text{(EQ 106)}$$

$$x(t + \Delta t) = x(t) + ((v(t) + v(t + \delta t)) / 2) \cdot \Delta t \qquad \text{(EQ 107)}$$

$$x(t + \Delta t) = x(t) + v(t) \cdot \Delta t + \frac{1}{2} \cdot a(t) \cdot \Delta t^2 \qquad \text{(EQ 108)}$$

(EQ 109)

etc.

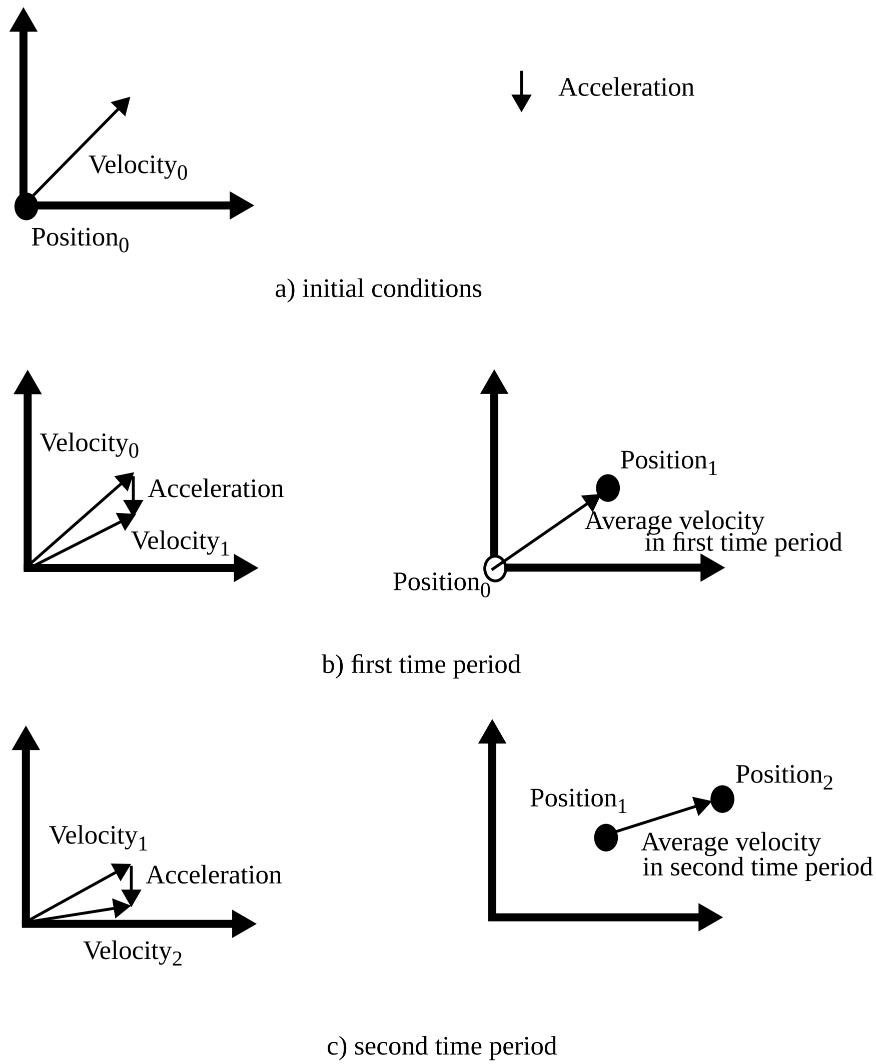


FIGURE 149. Modeling of a point's position at discrete time intervals (vector lengths are for illustrative purposes and are not accurate)

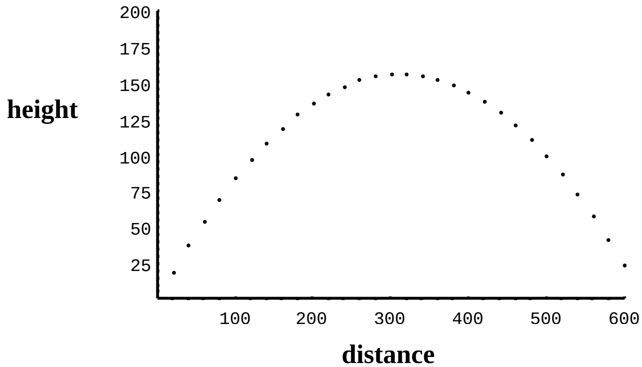


FIGURE 150. **Path of a particle in the simple example from the text.**

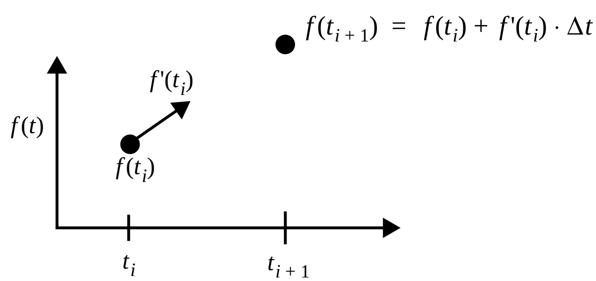
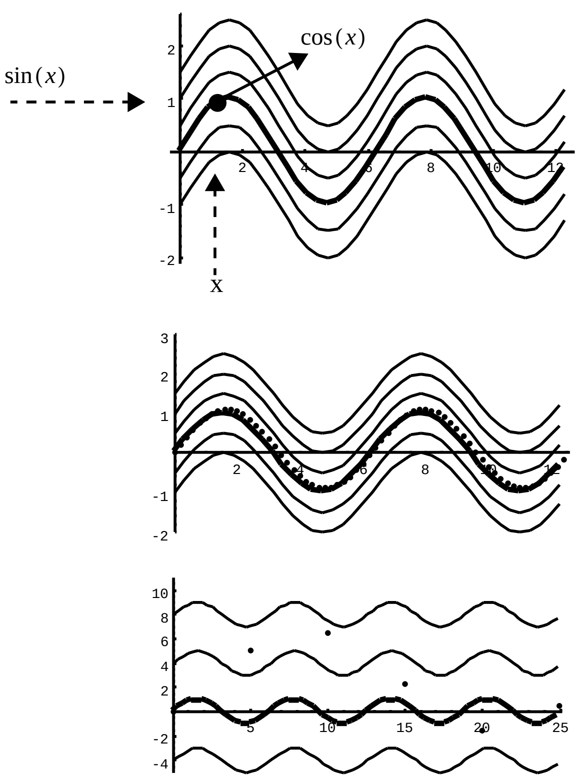


FIGURE 151. **Euler integration**

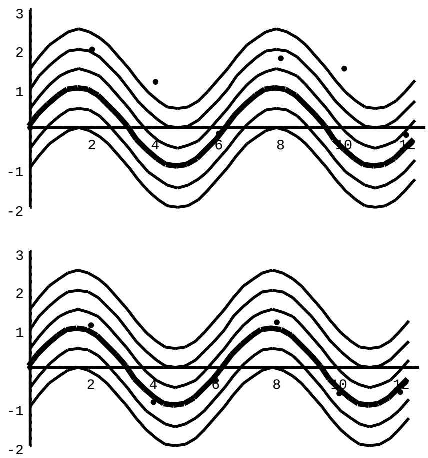


a) In this example, the sine function is the underlying (unknown) function. The objective is to reconstruct it based on an initial point and knowledge about the derivative function (cosine). Start out at any (x,y) location and, if the reconstruction is completely accurate, the sinusoidal curve which passes through that point should be followed. In the examples below, the initial point is (0,0) so the thicker sine curve should be followed.

b) Updating the function values by taking small enough steps along the direction indicated by the derivative generates a good approximation to the function. In this example, $\Delta x=0.2$.

c) However, if the step size becomes too large then the function reconstructed from the sample points can deviate widely from the underlying function. In this example, $\Delta x=5$.

FIGURE 152. **Approximating the sine curve by stepping in the direction of its derivative.**



a) Using step sizes of 2 with the Euler method

b) Using step sizes of 2 with the midpoint method

FIGURE 153. Euler method and the midpoint method.

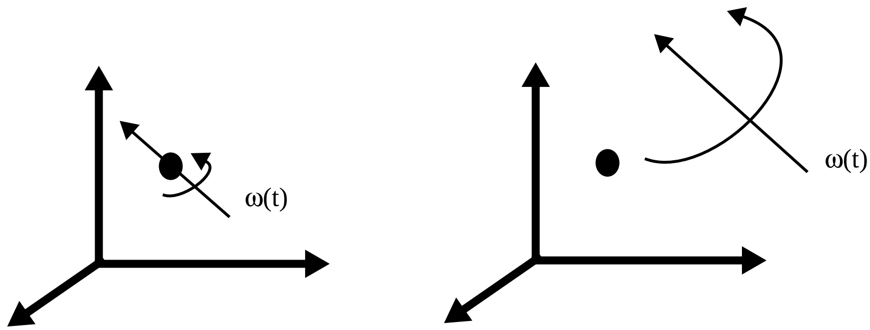


FIGURE 154. **For a given number of rotations per unit of time, the angular velocity is the same whether the axis of rotation is near or far away.**

).

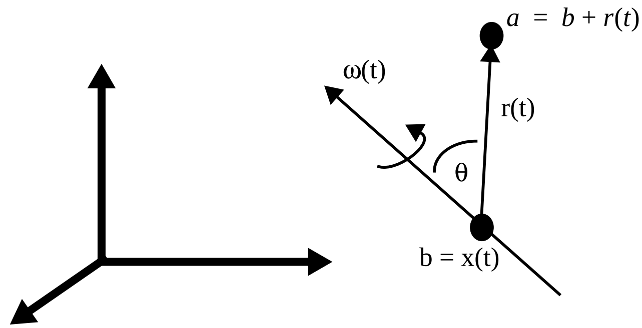


FIGURE 155. A point rotating about an axis.

$$\dot{r}(t) = \omega(t) \times r(t)$$
$$|\dot{r}(t)| = |\omega(t)||r(t)| \sin\theta \qquad \text{(EQ 110)}$$

$$R(t) = \begin{bmatrix} R1(t) & R2(t) & R3(t) \end{bmatrix}$$
$$\dot{R}(t) = \begin{bmatrix} \omega(t) \times R1(t) & \omega(t) \times R2(t) & \omega(t) \times R3(t) \end{bmatrix} \qquad \text{(EQ 111)}$$

$$A \times B = \begin{bmatrix} A_y \cdot B_z - A_z \cdot B_y \\ -(A_z \cdot B_x) + A_x \cdot B_z \\ A_x \cdot B_y - A_y \cdot B_x \end{bmatrix} = \begin{bmatrix} 0 & -A_x & A_y \\ A_z & 0 & -A_x \\ -A_y & A_x & 0 \end{bmatrix} \begin{bmatrix} B_x \\ B_y \\ B_z \end{bmatrix} = A^* B \qquad \text{(EQ 112)}$$

$$\dot{R}(t) = \omega(t)^* R(t) \qquad \text{(EQ 113)}$$

$$q(t) = R(t)q + x(t) \qquad \text{(EQ 114)}$$

$$\dot{q}(t) = \omega(t)^* R(t)q + v(t) \qquad \text{(EQ 115)}$$

$$\dot{q}(t) = \omega(t) \times (q(t) - x(t)) + v(t) \qquad \text{(EQ 116)}$$

$$M = \Sigma m_i \tag{EQ 117}$$

$$x(t) = \frac{\Sigma m_i q_i(t)}{M} \tag{EQ 118}$$

$$F = m \cdot a \tag{EQ 119}$$

$$a = F / m \tag{EQ 120}$$

$$F(t) = \Sigma f_i(t) \tag{EQ 121}$$

$$\begin{aligned} \tau_t(t) &= (q(t) - x(t)) \times f_i(t) \\ \tau(t) &= \Sigma \tau_t(t) \end{aligned} \tag{EQ 122}$$

$$p = m \cdot v \tag{EQ 123}$$

$$P(t) = \Sigma m_i \dot{q}_i(t) \tag{EQ 124}$$

$$P(t) = M \cdot v(t) \tag{EQ 125}$$

$$\dot{P}(t) = M \cdot \dot{v}(t) = F(t) \tag{EQ 126}$$

$$\begin{aligned} L(t) &= \Sigma((q(t) - x(t)) \times m_i \cdot (\dot{q}(t) - v(t))) \\ &= \Sigma(R(t)q \times m_i \cdot (\omega(t) \times (q(t) - x(t)))) \\ &= \Sigma(m_i \cdot (R(t)q \times (\omega(t) \times R(t)q))) \end{aligned} \tag{EQ 127}$$

$$\dot{L}(t) = \tau(t) \tag{EQ 128}$$

$$L(t) = I(t) \cdot \omega(t) \tag{EQ 129}$$

$$I_{\text{object}} = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{xy} & I_{yy} & I_{yz} \\ I_{xz} & I_{yz} & I_{zz} \end{bmatrix} \quad \text{(EQ 130)}$$

$$I_{xx} = \iiint \rho(q) \cdot (q_y^2 + q_z^2)\,\mathrm{dx\,dy\,dz} \quad \text{(EQ 131)}$$

$$\begin{array}{ll} I_{xx} = \sum m_i \cdot (y_i^2 + z_i^2) & I_{xy} = \sum m_i \cdot x_i \cdot y_i \\ I_{yy} = \sum m_i \cdot (x_i^2 + z_i^2) & I_{xz} = \sum m_i \cdot x_i \cdot z_i \\ I_{zz} = \sum m_i \cdot (x_i^2 + y_i^2) & I_{yz} = \sum m_i \cdot y_i \cdot z_i \end{array} \quad \text{(EQ 132)}$$

$$S(t) = \begin{bmatrix} x(t) \\ R(t) \\ P(t) \\ L(t) \end{bmatrix} \quad \text{(EQ 133)}$$

$$I(t) = R(t) I_{\text{object}} R(t)^T \quad \text{(EQ 134)}$$

$$\omega(t) = I(t)^{-1} L(t) \quad \text{(EQ 135)}$$

$$v(t) = \frac{P(t)}{M} \quad \text{(EQ 136)}$$

$$\frac{d}{dt} S(t) = \frac{d}{dt} \begin{bmatrix} x(t) \\ R(t) \\ P(t) \\ L(t) \end{bmatrix} = \begin{bmatrix} v(t) \\ \omega(t)^* R(t) \\ F(t) \\ \tau(t) \end{bmatrix} \quad \text{(EQ 137)}$$

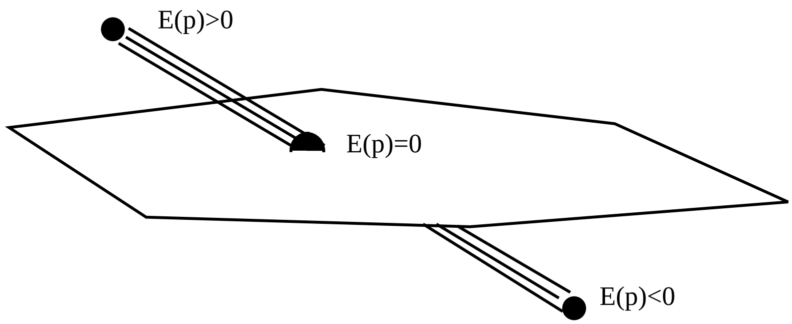


FIGURE 156. Point-plane collision.

$$E(p) = a \cdot x + b \cdot y + c \cdot z + d \qquad \text{(EQ 138)}$$

$$p(t_i) = p(t_{i-1}) + \partial t \cdot v_{ave}(t) \qquad \text{(EQ 139)}$$

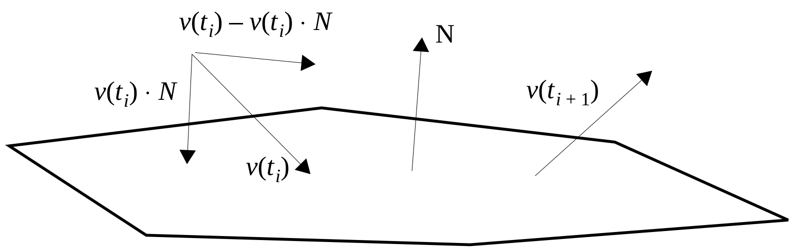


FIGURE 157. Kinematic solution for collision reaction

$$v(t_{i+1}) = v(t_i) - v(t_i) \cdot N - k \cdot v(t_i) \cdot N = v(t_i) - (1+k) \cdot v(t_i) \cdot N \quad \text{(EQ 140)}$$

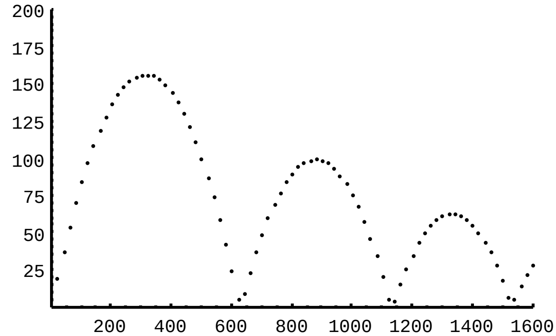


FIGURE 158. Kinematic response to collisions with ground using 0.8 as the damping factor for example from Section 4.3.1.1

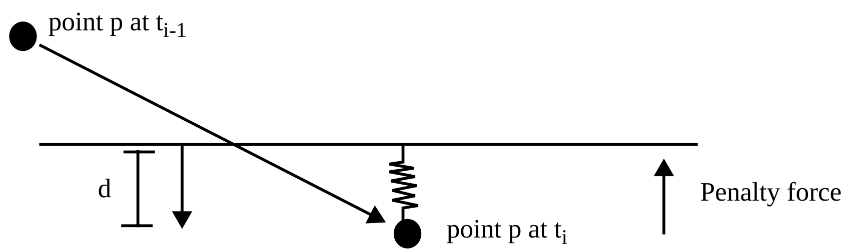


FIGURE 159. Penalty spring.

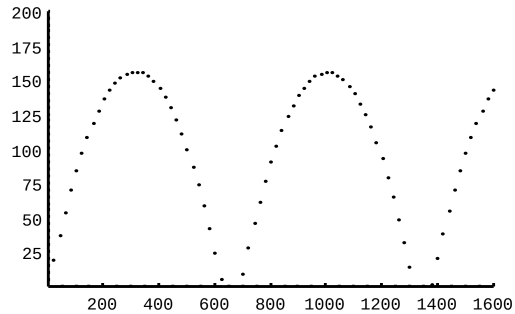
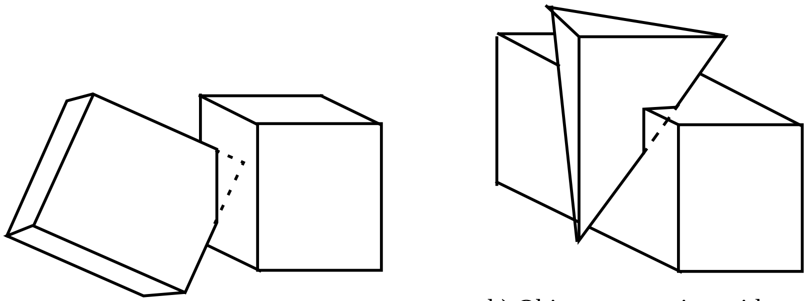


FIGURE 160. **Penalty method with a spring constant of 250 and a point mass of 10 for example from Section 4.3.1.1.**



a) Vertex inside a polyhedron

b) Object penetration without a vertex of one object contained in the other

FIGURE 161. Detecting polyhedra intersections.

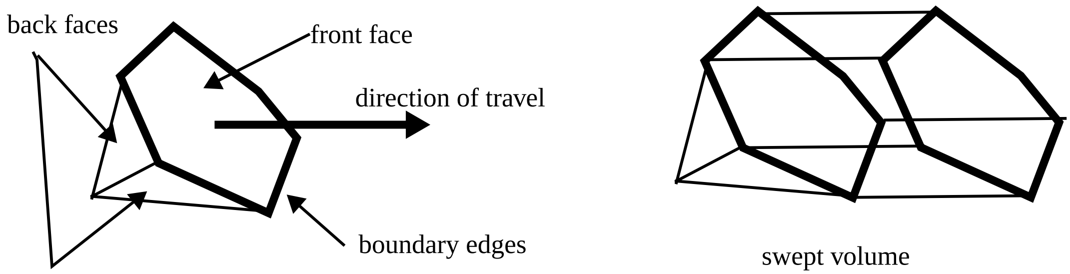


FIGURE 162. Swept Volume

.

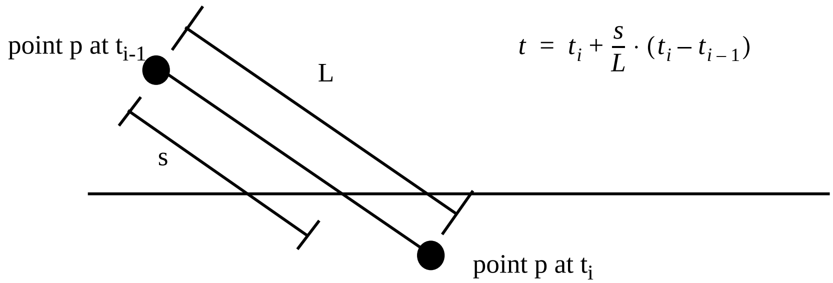
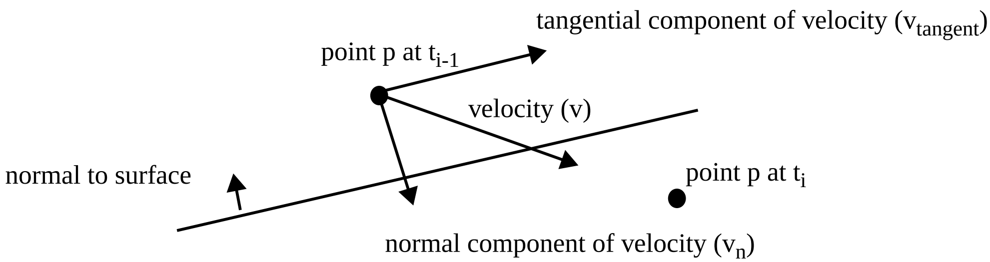
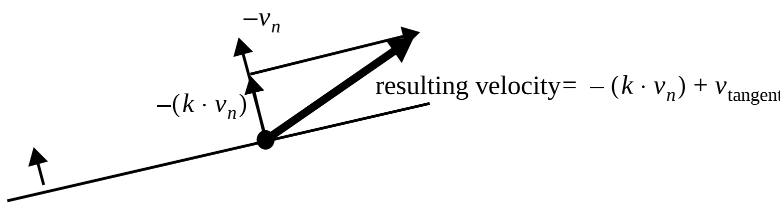


FIGURE 163. Linearly estimating time of impact, t.



a) components of a particle's velocity colliding with a plane



b) computing velocity resulting from a collision (k is the coefficient of restitution)

FIGURE 164. Impact response of a point with a plane

$$J = F \cdot \Delta t \qquad \text{(EQ 141)}$$

$$J = F \cdot \Delta t = M \cdot a \cdot \Delta t = M \cdot \Delta v = \Delta(M \cdot v) = \Delta P \qquad \text{(EQ 142)}$$

$$v_{rel}^{+} = -\varepsilon \cdot v_{rel}^{-} \qquad \text{(EQ 143)}$$

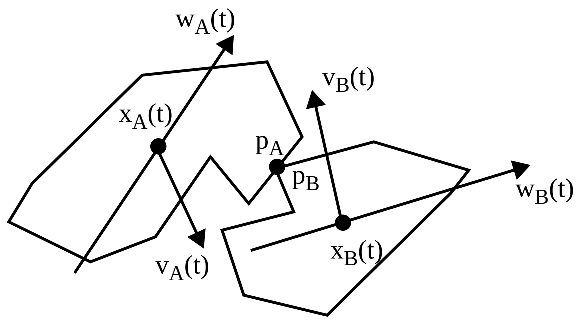


FIGURE 165. Configuration of colliding objects.

$$r_A = p_A - x_A(t)$$
$$r_B = p_B - x_B(t) \quad \text{(EQ 144)}$$

$$v_{rel} = (\dot{p}_A(t) - \dot{p}_A(t)) \cdot n \quad \text{(EQ 145)}$$

$$\dot{p}_A(t) = v_A(t) + \omega_A(t) \times r_A$$
$$\dot{p}_B(t) = v_B(t) + \omega_B(t) \times r_B \quad \text{(EQ 146)}$$

$$v_A^+ = v_A^- + \frac{j \cdot n}{M_A}$$
$$v_B^+ = v_B^- + \frac{j \cdot n}{M_B} \quad \text{(EQ 147)}$$

$$\omega^+{}_A = \omega^-{}_A + I^{-1}{}_A(t) \cdot (r_A \times j \cdot n)$$
$$\omega^+{}_B = \omega^-{}_B + I^{-1}{}_B(t) \cdot (r_B \times j \cdot n) \quad \text{(EQ 148)}$$

$$v_{rel}^+ = n \cdot (\dot{p}_A^+(t) - \dot{p}_B^+(t)) \quad \text{(EQ 149)}$$

$$v_{rel}^+ = n \cdot (v^+{}_A(t) + \omega^+{}_A(t) \times r_A - (v^+{}_B(t) + \omega^+{}_B(t) \times r_B)) \quad \text{(EQ 150)}$$

$$j = \frac{-((1+\varepsilon) \cdot v_{rel}^-)}{\frac{1}{M_A} + \frac{1}{M_B} + n \cdot (I_A^{-1}(t)(r_A \times n)) \times r_A + (I_B^{-1}(t)(r_B \times n)) \times r_B} \quad \text{(EQ 151)}$$

```
Compute VA, VB                              ; EQ 121
Compute Vrelative = dot(N,(VA-VB)           ; relative velocity of two contact points
if V-relative > threshold
        compute j                           ; EQ 126
        J = j*n
        PA += J                             ; update linear momentum
        PB -= J                             ; update linear momentum
        LA += rA x J                        ; update angular momentum
        LB -= rB x J                        ;update angular momentum
else if Vrelative <  -threshold
        resting contact
else
        objects are moving away from each other
```

.

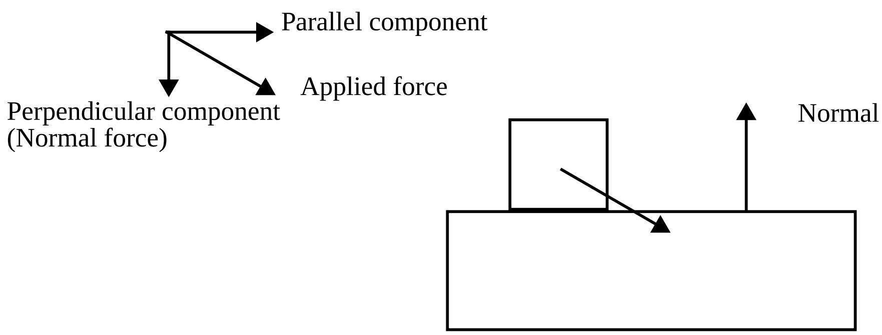


FIGURE 166. **Horizontal and vertical components of force.**

$$F_s = \mu_s \cdot F_N \quad \text{(EQ 152)}$$

$$F_k = \mu_k \cdot F_N \quad \text{(EQ 153)}$$

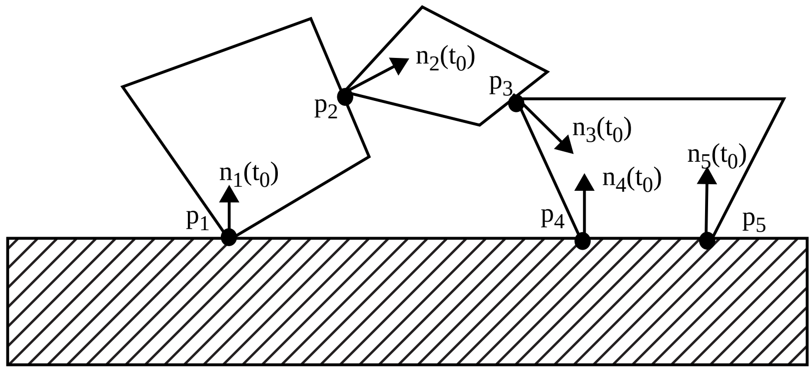


FIGURE 167. Multiple object resting contacts

$$d_i(t) = n_i(t) \cdot (p_A(t) - p_B(t)) \quad \text{(EQ 154)}$$

$$\dot{d}_i(t) = \dot{n}_i(t) \cdot (p_A(t) - p_B(t)) + n_i(t) \cdot (\dot{p}_A(t) - \dot{p}_B(t)) \quad \text{(EQ 155)}$$

$$\ddot{d}_i(t) = \ddot{n}_i(t) \cdot (p_A(t) - p_B(t)) + \dot{n}_i(t) \cdot (\dot{p}_A(t) - \dot{p}_B(t)) + \dot{n}_i(t) \cdot (\dot{p}_A(t) - \dot{p}_B(t)) + n_i(t) \cdot (\ddot{p}_A(t) - \ddot{p}_B(t)) \quad \text{(EQ 156)}$$

$$\ddot{d}_i(t_0) = n_i(t_0) \cdot (\ddot{p}_A(t_0) - \ddot{p}_B(t_0)) + 2 \cdot \dot{n}_i(t_0) \cdot (\dot{p}_A(t_0) - \dot{p}_B(t_0)) \quad \text{(EQ 157)}$$

$$\ddot{d}_i(t) \geq 0 \quad \text{(EQ 158)}$$

$$f_i \geq 0 \quad \text{(EQ 159)}$$

$$\ddot{d}_i(t) \cdot f_i = 0 \quad \text{(EQ 160)}$$

$$\ddot{d}_i(t_0) = b_i + \sum_{j=1}^{n} (a_{ij} \cdot f_j) \quad \text{(EQ 161)}$$

$$\ddot{p}_A(t) = \dot{v}_A(t) + \dot{\omega}_A(t) \times r_A + \omega_A(t) \times (\omega_A(t) \times r_A) \quad \text{(EQ 162)}$$

$$\dot{\omega}_A(t) = I_A^{-1}(t) \cdot \tau_A(t) + I_A^{-1}(t) \cdot (L_A(t) \times \omega_A(t)) \tag{EQ 163}$$

$$f_j \cdot \left(\frac{n_j(t_0)}{m_A} + (I_A^{-1}(t_0) \cdot (p_j - x_A(t_0)) \times n_j(t_0)) \times r_A \right) \tag{EQ 164}$$

$$\frac{F_A(t_0)}{m_A} + I_A^{-1}(t) \cdot \tau_A(t) + \omega_A(t) \times (\omega_A(t) \times r_A) + (I_A^{-1}(t) \cdot (L_A(t) \times \omega_A(t))) \times r_A \tag{EQ 165}$$

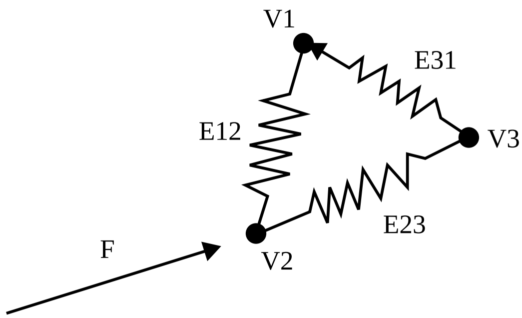


FIGURE 168. A simple spring-mass model of a flexible object.

$$F_{i,j}^{spring} = -F_{j,i}^{spring} = k_s \cdot (dist_{i,j}(t) - \mathrm{len}_{i,j}) \cdot v_{i,j}$$ **(EQ 166)**

$$F_i^{damper} = -k_d \cdot v_i(t)$$ **(EQ 167)**

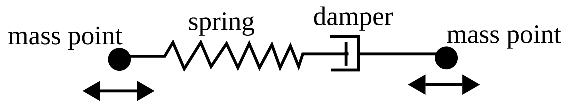


FIGURE 169. Spring-damper configuration.

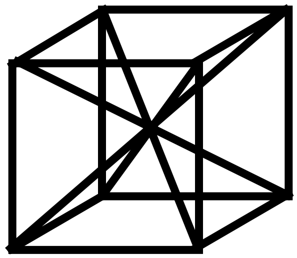


FIGURE 170. **Interior springs to help stabilize the object's configuration.**

.

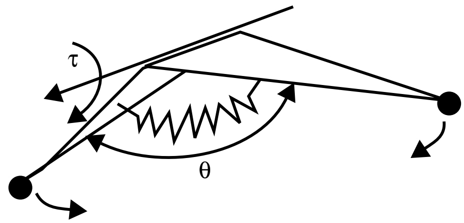


FIGURE 171. **An angular spring imparting restoring torques.**

$$\tau = k_s \cdot (\theta(t) - \theta_{rest}) - k_d \cdot \dot{\theta}(t) \qquad \text{(EQ 168)}$$

$$\tau = k_s \cdot (\theta(t) - \theta_{desired}) - k_d \cdot (\dot{\theta}(t) - \dot{\theta}_{desired}) \qquad \text{(EQ 169)}$$

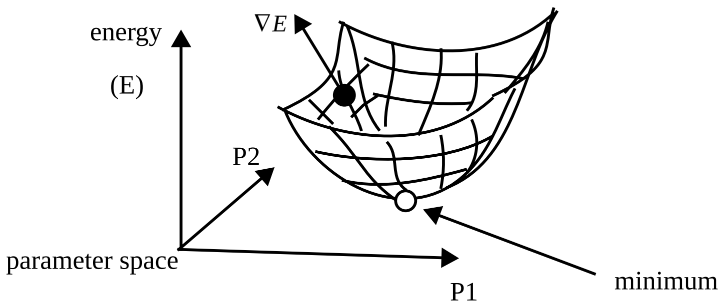


FIGURE 172. Sample simple energy function.

$$F(t_{i+1}) = F(t_i) - h \cdot \nabla E \qquad \text{(EQ 170)}$$

Point-to-point: $E = \left|P^q(u_a, v_a) - P^b(u_b, v_b)\right|^2$

Point-to-point locally abutting: $E = \left|P^q(u_a, v_a) - P^b(u_b, v_b)\right|^2 + N^a(u_a, v_a) \cdot N^b(u_b, v_b) + 1.0$

Floating attachment: $E = (I^b(P^q(u_a, v_a)))^2$

Floating attachment locally abutting: $E = (I^b(P^q(u_a, v_a)))^2 + N^a(u_a, v_a) \cdot \frac{\nabla I^b(P^q(u_a, v_a))}{\left|\nabla I^b(P^q(u_a, v_a))\right|} + 1.0$

$$m \cdot \ddot{x}(t) - f(t) - m \cdot g = 0 \quad \text{(EQ 171)}$$

$$\begin{aligned} x(t_0) &= a \\ x(t_1) &= b \end{aligned} \quad \text{(EQ 172)}$$

$$R = \int_{t_0}^{t_1} |f|^2 \mathrm{dt} \quad \text{(EQ 173)}$$

$$\dot{x}_i = \frac{x_i - x_{i-1}}{h} \quad \text{(EQ 174)}$$

$$\ddot{x}_i = \frac{x_{i+1} - 2 \cdot x_i + x_{i-1}}{h^2} \quad \text{(EQ 175)}$$

$$p_i = m \cdot \frac{x_{i+1} - 2 \cdot x_i + x_{i-1}}{h^2} - f - m \cdot g = 0 \quad \text{(EQ 176)}$$

$$\begin{aligned} c_a &= x_1 - a = 0 \\ c_b &= x_n - b = 0 \end{aligned} \quad \text{(EQ 177)}$$

$$J_{ij} = \frac{\partial C_i}{\partial S_j} \qquad \text{(EQ 178)}$$

$$H_{ij} = \frac{\partial^2 R}{\partial S_i \partial S_j} \qquad \text{(EQ 179)}$$

$$-\frac{\partial R}{\partial S_i} = \sum_j H_{ij} \hat{S}_j \qquad \text{(EQ 180)}$$

$$-C_i = \sum_j J_{ij} (\hat{S}_j + \tilde{S}_j) \qquad \text{(EQ 181)}$$

.

Type of Group	Number of elements	Incorporated physics	Intelligence
Particles	many	much - with environment	none
Flocks	some	some - with environment and other elements	limited
Autonomous Behavior	few	little	much

TABLE 5. **Characteristics of Group Types**

.

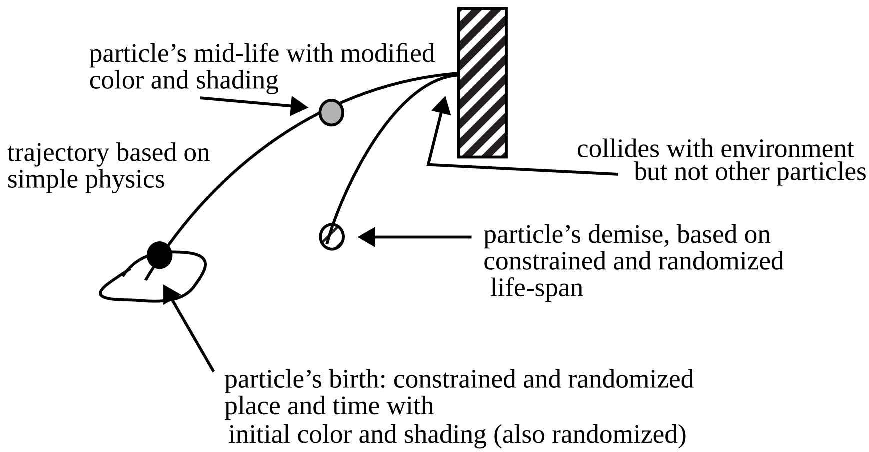


FIGURE 173. The life of a particle

$$\# \text{ of particles } = average + Rand(\_) \cdot range \tag{EQ 182}$$

$$\# \text{ of particles } = average + Rand(\_) \cdot range \cdot screenArea \tag{EQ 183}$$

```
typedef particle_struct struct {
  vector3D          p;
  vector3D          v;
  vector3D          f;
  float             mass;
} particle;
```

```
typedef particleSystem_struct struct {
  particle   *p;
  int                        n;
  float                      t;
} particleSystem;
```

```
update(pSystem)
{
  clearForces(pSystem)
  computeForces(pSystem)
  getState(pSystem,array1)
  computeDerivative(pSystem,arrayw)
  addVector(array1,array2,array2,n)
  saveState(array2,pSystem)
  t += deltaT
}
```

be aware of itself and two or three of its neighbors.
be aware of what's in front of it and have a limited field of view (fov).
have distance-limited field of view (fov).
be influenced by objects within the line of sight.
be influenced by objects based on distance and size (angle subtended in the fov).
be affected by things using a distance-squared or distance-cubed weighting function.
have a general migratory urge, but no global objective.
not follow a designated leader.
not have knowledge about a global flock center.

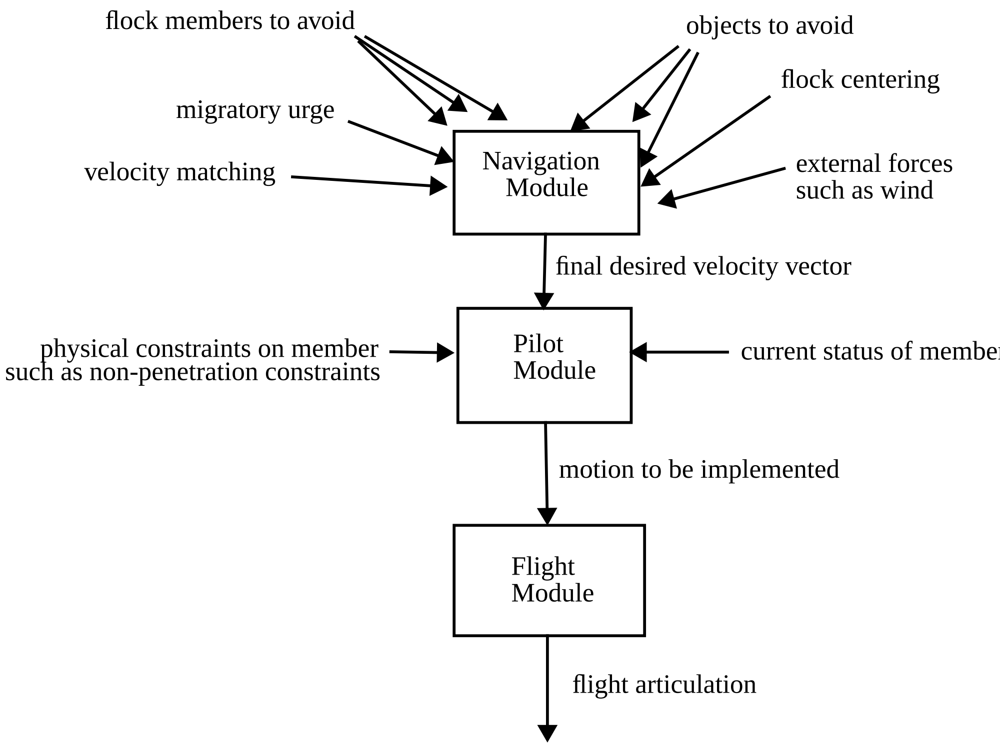


FIGURE 174. Negotiating the Motion

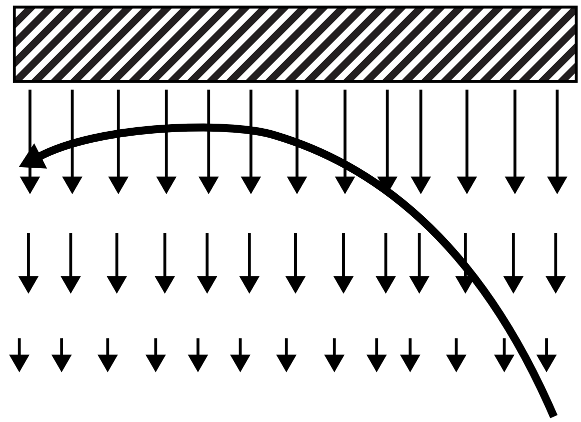


FIGURE 175. **Force field collision avoidance.**

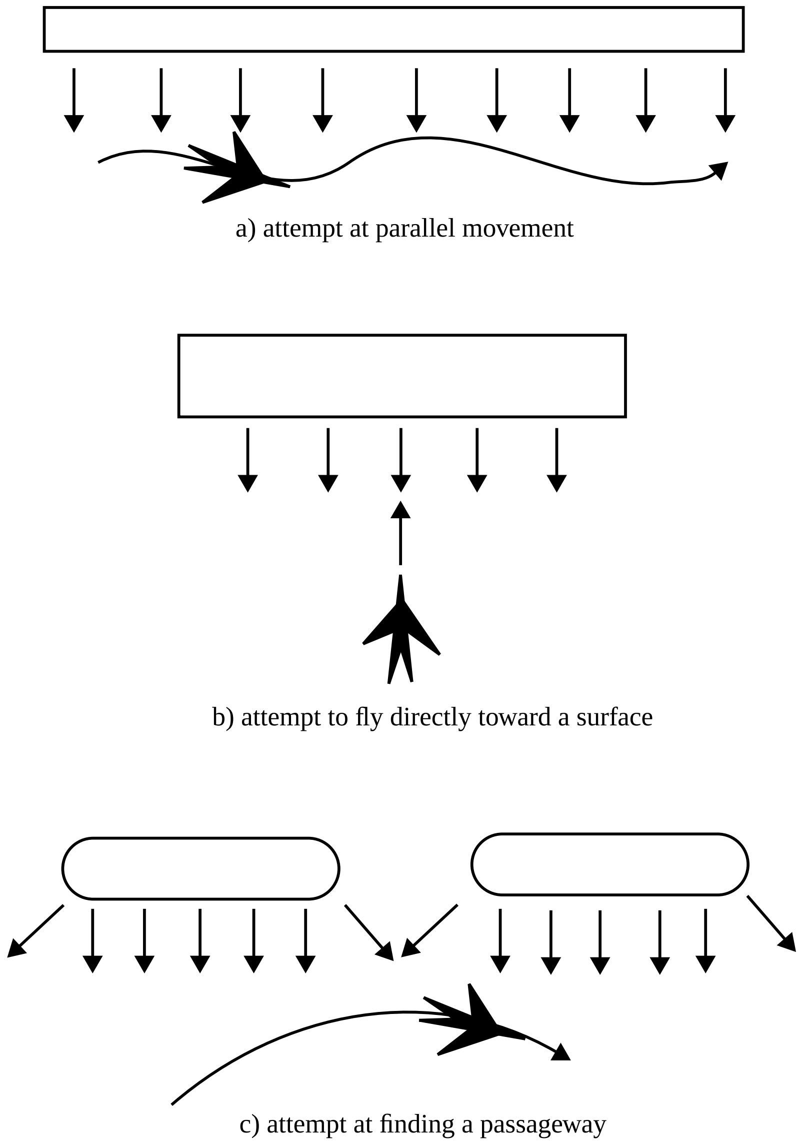


FIGURE 176. Problems with force field collision avoidance

T

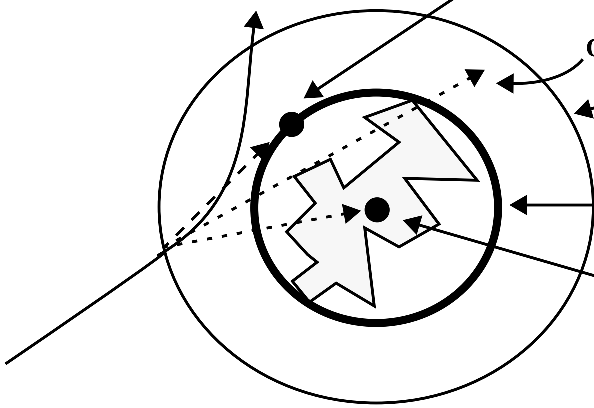
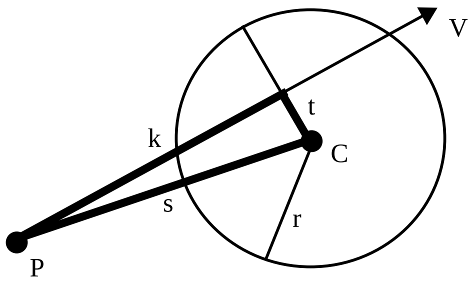


FIGURE 177. **Steer to avoid bounding sphere.**



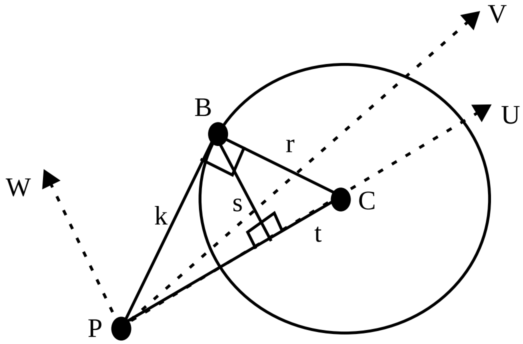
$$s = |C - P|$$

$$k = (C - P) \bullet \frac{V}{|V|}$$

$$t = \sqrt{s^2 - k^2}$$

$t < r$ indicates penetration with bounding sphere

FIGURE 178. Testing for potential collision with a bounding sphere.



$$k = \sqrt{|C-P|^2 - r^w}$$

$$r^2 = s^2 + t^2$$

$$k^2 = s^2 + (|C-P| - t)^2$$

$$k^2 = r^2 - t^2 + (|C-P| - t)^2$$

$$k^2 = r^2 - t^2 + |C-P|^2 - 2 \cdot |C-P| \cdot t + t^2$$

$$t = \frac{k^2 - r^2 - |C-P|^2}{-2 \cdot |C-P|}$$

$$s = \sqrt{r^2 - t^2}$$

$$U = \frac{C-P}{|C-P|}$$

$$W = (U \times V) \times U$$

$$B = P + (|C-P| - t) \cdot U + s \cdot W$$

FIGURE 179. Calculation of point B on boundary of sphere

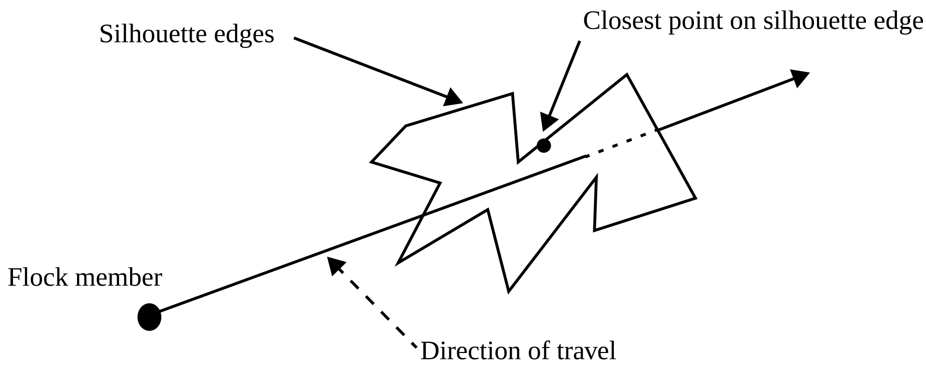


FIGURE 180. Determining steer-to point from silhouette edges

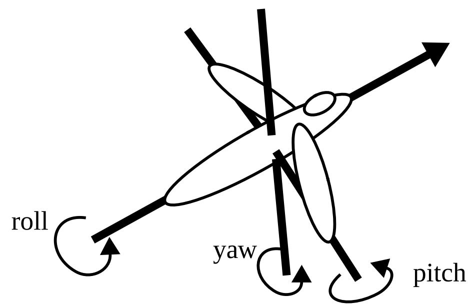


FIGURE 181. **Roll, pitch, and yaw of local coordinate system.**

.

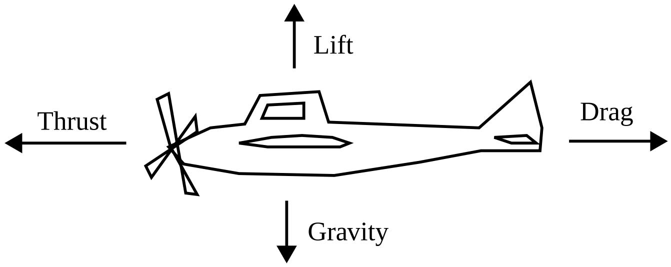


FIGURE 182. **The forces of flight.**

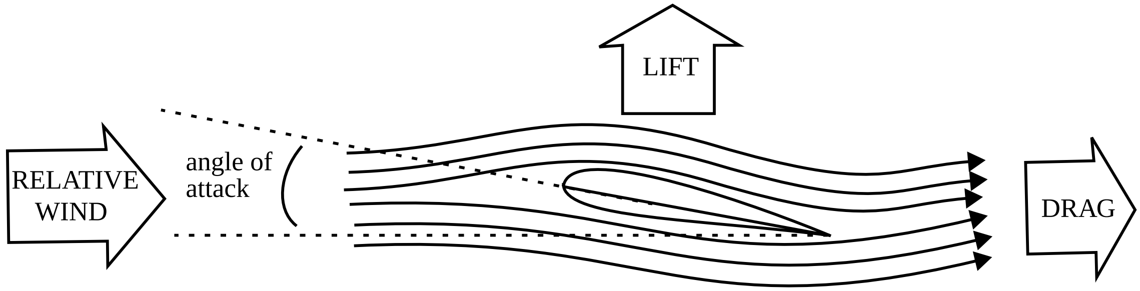
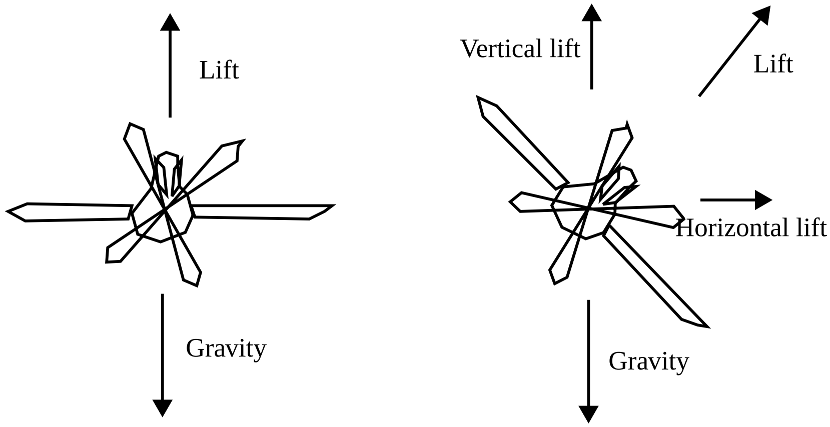


FIGURE 183. Lift produced by an airfoil.


Lift
Gravity
Vertical lift
Lift
Horizontal lift
Gravity

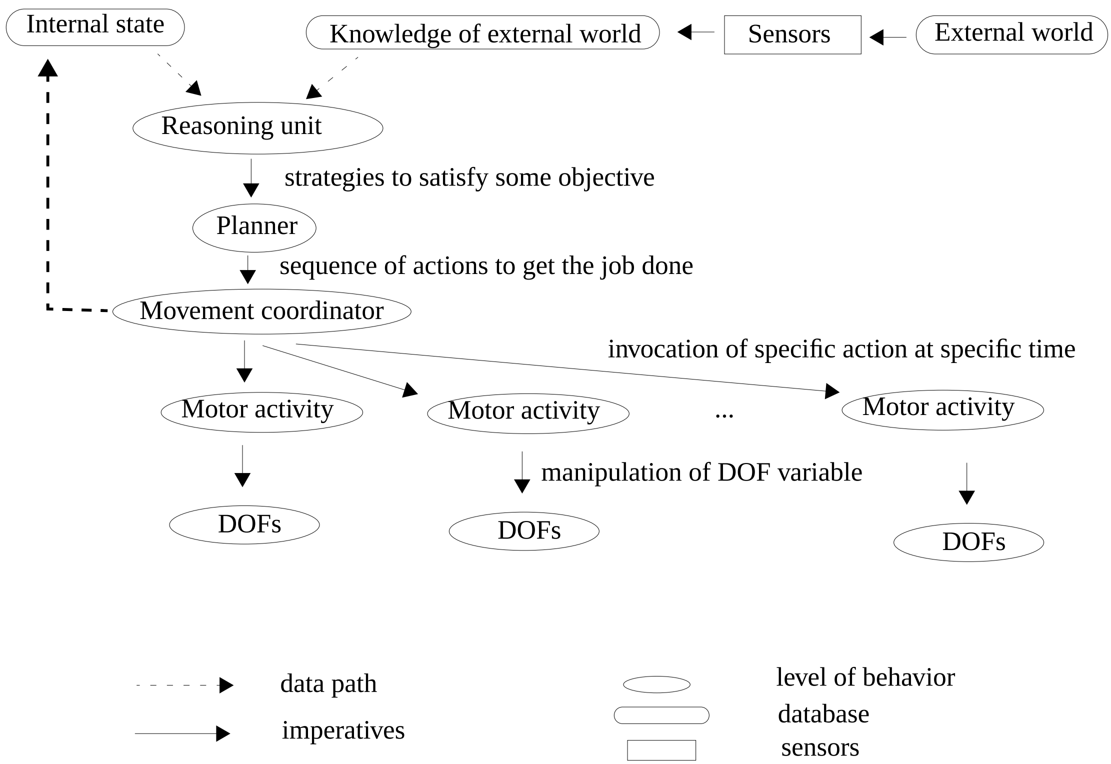


FIGURE 184. Levels of behavior

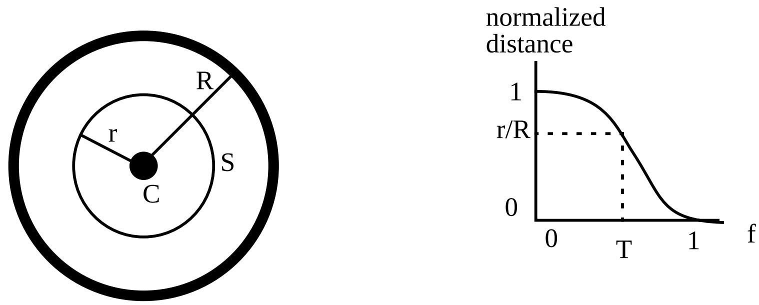


FIGURE 185. The metaball and a sample density function.

$$F(P) = \Sigma w_i f_i(P) - T \qquad \text{(EQ 184)}$$

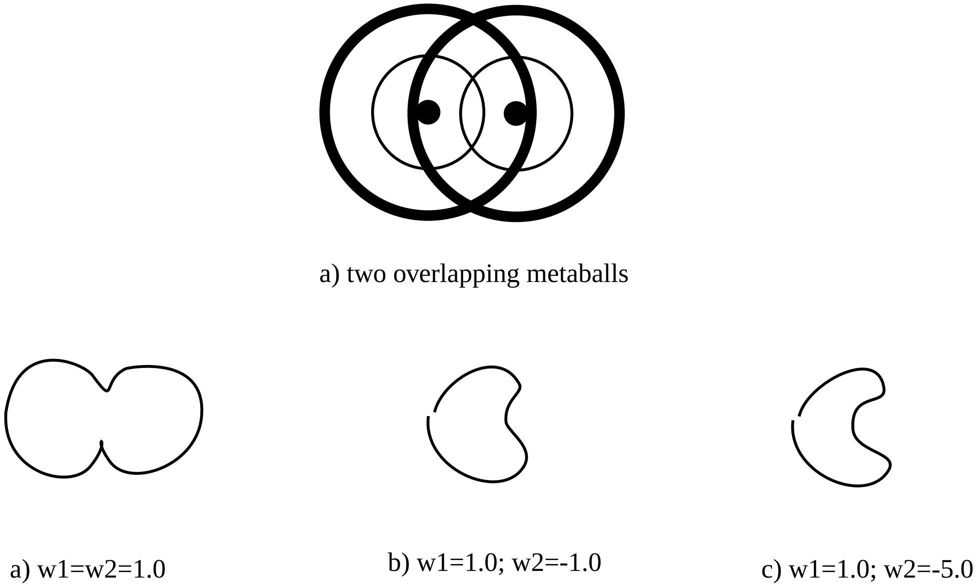


FIGURE 186. Compound implicit surface.

T.

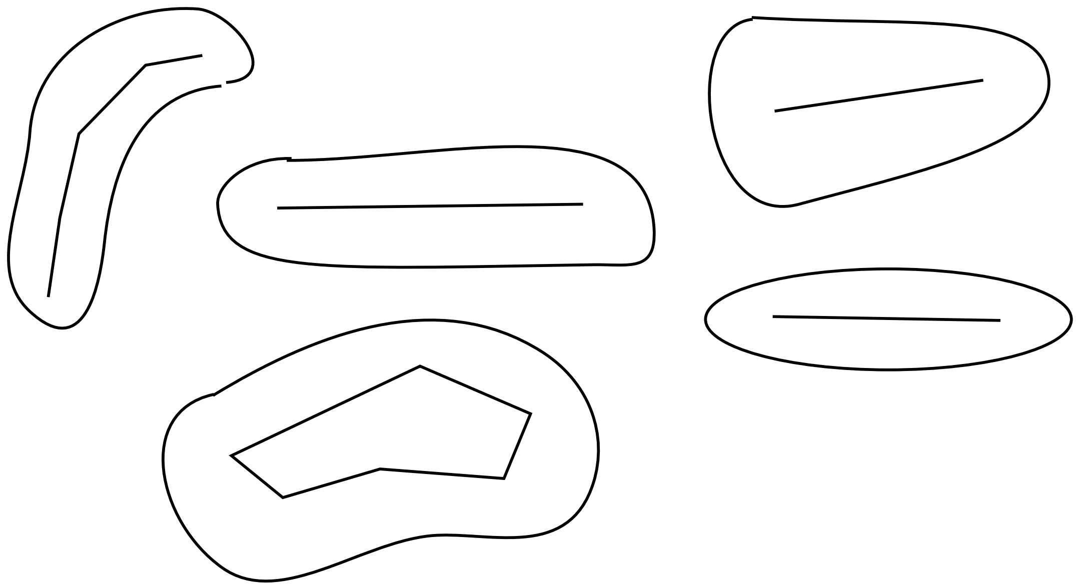


FIGURE 187. Distance-based implicit surfaces.

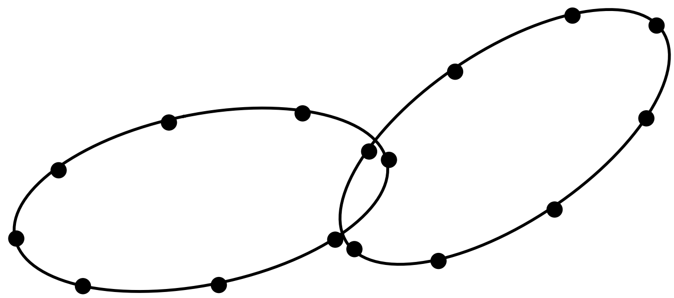


FIGURE 188. Point samples used to test for collisions.

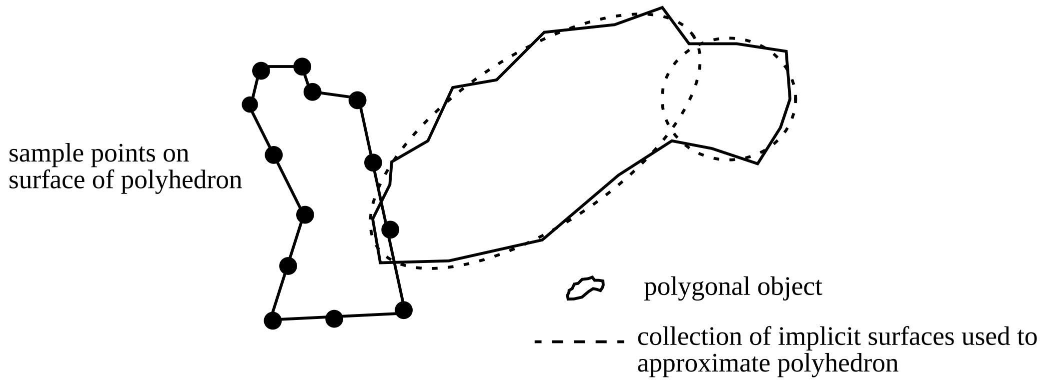


FIGURE 189. Using implicit surfaces for detecting collisions between polyhedral objects.

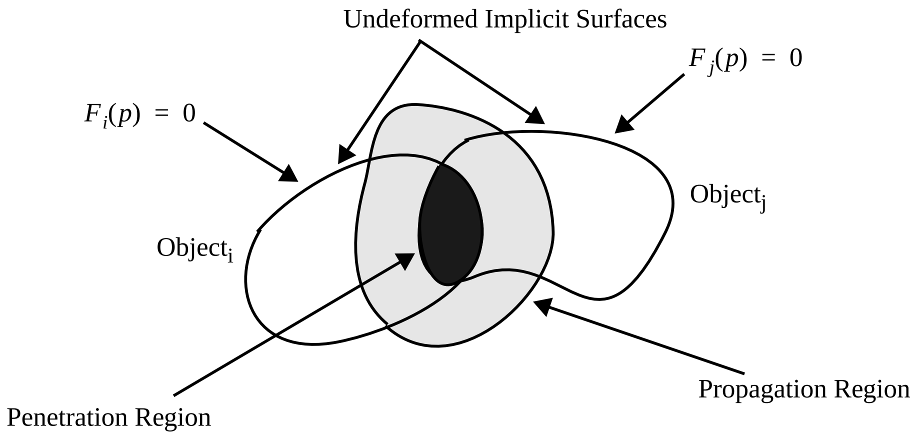


FIGURE 190. Penetrating Implicit Surfaces.

.

$$G_{ij}(p) = -F_j(p)$$
$$G_{ji}(p) = -F_i(p) \quad \text{(EQ 185)}$$

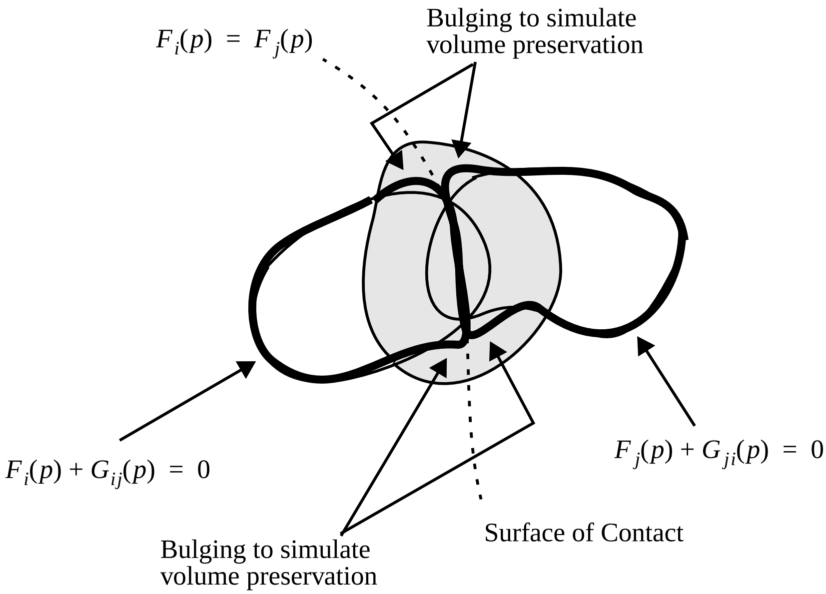


FIGURE 191. Implicit surfaces after deformation due to collision.

.

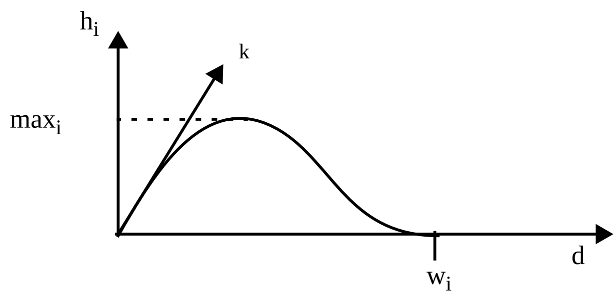


FIGURE 192. **Deformation function in the propagation region.**