Multi-joint kinematics and dynamics

Emo Todorov

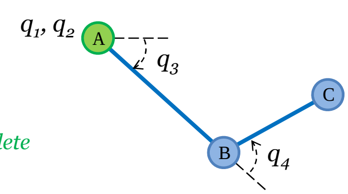
Applied Mathematics
Computer Science and Engineering

University of Washington

Kinematics in generalized vs. Cartesian coordinates

generalized $q = (q_1, q_2, q_3, q_4)^T$

Cartesian $x = (A_x, A_y, B_x, B_y, C_x, C_y)^T$



dim(q) equals the number of degrees of freedom (DOFs)
dim(x)>dim(q), i.e. Cartesian coordinates are over-complete

forward kinematics:

always well-defined mapping from q to $x = h(q)$

$$A_x = q_1$$
$$A_y = q_2$$
$$B_x = q_1 + |AB| \cos(q_3)$$
$$B_y = q_2 + |AB| \sin(q_3)$$
$$C_x = q_1 + |AB| \cos(q_3) + |BC| \cos(q_3 + q_4)$$
$$C_y = q_2 + |AB| \sin(q_3) + |BC| \sin(q_3 + q_4)$$

inverse kinematics:

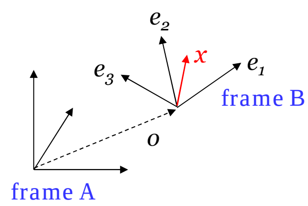
usually not well-defined, but one can resolve redundancy via optimization:
of all q satisfying $x = h(q)$ for given x, pick the one closest to a preferred q^*

$$q = \arg\min_{\hat{q} : x = h(\hat{q})} \|\hat{q} - q^*\|$$

Kinematics in 3D

Attach a spatial frame to each body - usually at the center of mass, or at the center of the joint connecting it to its parent body. Rotations can be expressed as

$$ {}^{A}_{B}R = \begin{pmatrix} | & | & | \\ e_1 & e_2 & e_3 \\ | & | & | \end{pmatrix} \qquad R^T = R^{-1},\ \det(R) = 1 $$



Transformation between frames: ${}^{A}x = {}^{A}_{B}o + {}^{A}_{B}R\ {}^{B}x$

Addition and multiplication can be combined using homogeneous coordinates:

$$ {}^{A}_{B}\hat{R} = \begin{pmatrix} {}^{A}_{B}R & {}^{A}_{B}o \\ 0 & 1 \end{pmatrix} \qquad \hat{x} = \begin{pmatrix} x \\ 1 \end{pmatrix} \quad : \quad {}^{A}\hat{x} = {}^{A}_{B}\hat{R}\ {}^{B}\hat{x} \qquad {}^{A}_{C}\hat{R} = {}^{A}_{B}\hat{R}\ {}^{B}_{C}\hat{R} $$

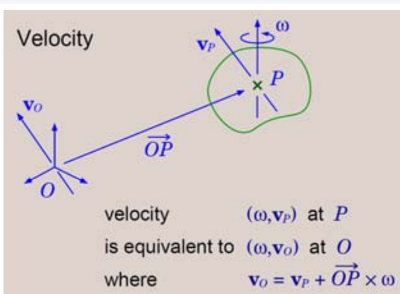
The spatial relations between body frames depend on the joint parameters q.
Let $q_{(n)}$ be the parameters specifying the joint between body n and its parent.
Then the corresponding transformation is parameterized as ${}^{\text{parent}(n)}_{n}\hat{R}\left(q_{(n)}\right)$

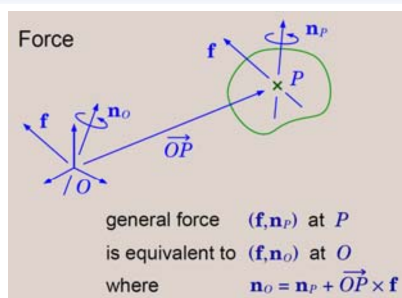
Computing the forward kinematics involves a forward recursion:

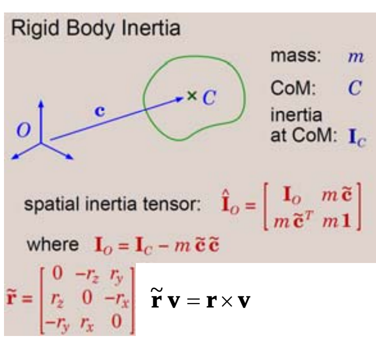
$$ {}^{\text{world}}_{n}\hat{R} = {}^{\text{world}}_{\text{parent}(n)}\hat{R}\ {}^{\text{parent}(n)}_{n}\hat{R}\left(q_{(n)}\right) $$

Numerically, forward kinematics is more accurate using ***quaternions*** *instead of 3x3 matrices.*

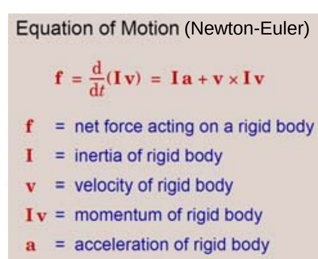
Rigid-body dynamics







$\tilde{\mathbf{r}}\,\mathbf{v} = \mathbf{r} \times \mathbf{v}$



see Featherstone's slides on *Spatial Vector Algebra*

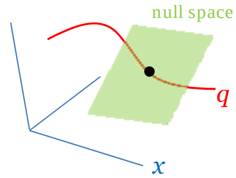
Newtonian mechanics with implicit constraints

Newton's second law for a scalar point mass is $m\ddot{x} = f$

For a set of n point masses in 3D we have
$$\begin{pmatrix} m_1 I_3 & & \\ & \ddots & \\ & & m_n I_3 \end{pmatrix} \begin{pmatrix} \ddot{x}_{1,1} \\ \vdots \\ \ddot{x}_{n,3} \end{pmatrix} = \begin{pmatrix} f_{1,1} \\ \vdots \\ f_{n,3} \end{pmatrix}$$

which in vector notation is $D\ddot{x} = f$

Now consider a set of m positional equality constraints defined *implicitly* as $\phi(x) = 0$
They could specify that some masses belong to the same rigid body, or that some rigid bodies are constrained by joints, etc. The constraints eliminate m DOFs and create a $3n$-m dimensional configuration manifold parameterized by q.



The constraint forces can only act within the null space, which is spanned by the rows of the Jacobian matrix $J = \frac{\partial \phi}{\partial x}$.
Thus $f_{tot} = f + J^T \lambda$ for some m-dimensional vector λ, found by taking into account the differentiated constraints:

$$\dot{\phi} = J\dot{x}, \quad \ddot{\phi} = J\ddot{x} + \dot{J}\dot{x} = 0, \quad \text{where } \dot{J} = \sum_i \frac{\partial J}{\partial x_i}\dot{x}_i$$

The constrained dynamics $D\ddot{x} = f_{tot}$ are the solution to the linear in $\ddot{x}, \lambda$ equation

$$\begin{pmatrix} D & -J^T \\ -J & 0 \end{pmatrix} \begin{pmatrix} \ddot{x} \\ \lambda \end{pmatrix} = \begin{pmatrix} f \\ \dot{J}\dot{x} \end{pmatrix}$$

The constrained dynamics are

$$D\ddot{x} = f - J^T \left(JD^{-1}J^T\right)^{-1} \left(\dot{J}\dot{x} + JD^{-1} f\right)$$

Constrained inertia and the Gauss principle

When the system is stationary, the constrained dynamics simplify to $\ddot{x} = Af$
where A is the inverse of the constrained inertia matrix:

$$A = D^{-1} - D^{-1}J^T \left(JD^{-1}J^T\right)^{-1} JD^{-1}$$

There is no acceleration in the null space: $J\ddot{x} = JAf = 0$, which follows from

$$JA = JD^{-1} - JD^{-1}J^T \left(JD^{-1}J^T\right)^{-1} JD^{-1} = JD^{-1} - JD^{-1} = 0$$

A is singular, with *rank(A) = dim(q)*.

Using the matrix inversion lemma, we can represent A as

$$A = \lim_{\varepsilon \to \infty} \left(D + \varepsilon J^T J\right)^{-1}$$

Thus the constrained inertia is "$D + \infty J^T J$", and is infinite in the null space.

The same results can be obtained from the more general **Gauss principle**:
the constrained acceleration $\ddot{x}$ is the solution to the minimization problem

$$\ddot{x} = \arg\min_a \left(a - \ddot{x}_0\right)^T D \left(a - \ddot{x}_0\right) \quad \text{s.t. } Ja = b$$

$\ddot{x}_0$ is the unconstrained acceleration; J, b can encode general constraints.

Explicit constraints

The implicitly-constrained dynamics $D\ddot{x} = f - J_\phi^T\left(J_\phi D^{-1} J_\phi^T\right)^{-1}\left(\dot{J}_\phi \dot{x} + J_\phi D^{-1} f\right)$
are expressed in over-complete Cartesian coordinates (x), which is often undesirable.
Instead it is better to express the dynamics in generalized (q) coordinates. This is done through explicit constraints given by the forward kinematics function $x = h(q)$

Differentiating the constraints twice yields $\ddot{x} = J(q)\ddot{q} + \dot{J}(q)\dot{q}$

The dynamics are $D\ddot{x} = f + f_c$ where f_c are the constraint forces.
Since the columns of J span the tangent space to the manifold, $J(q)^T f_c = 0$

Assembling these equations, we obtain a system which is linear in $\ddot{x}, \ddot{q}, f_c$

$$\begin{pmatrix} D & -I & 0 \\ I & 0 & -J \\ 0 & J^T & 0 \end{pmatrix}\begin{pmatrix} \ddot{x} \\ f_c \\ \ddot{q} \end{pmatrix} = \begin{pmatrix} f \\ \dot{J}\dot{q} \\ 0 \end{pmatrix}$$

The constrained dynamics are

$$\boxed{M(q)\ddot{q} + c(q,\dot{q}) = \tau}$$

where
$$M = J^T D J$$
$$c = J^T D \dot{J}\dot{q}$$
$$\tau = J^T f$$

Coordinate transformations

Consider any set of coordinates x, related to q as $x = h(q)$

Velocities in the two coordinate systems relate as $\boxed{\dot{x} = J(q)\dot{q}}$

Let f and τ denote the same force expressed in x and q coordinates respectively.
Power is coordinate-independent:

$$\dot{q}^T \tau = \dot{x}^T f = \dot{q}^T J^T f$$

Since this holds for any velocity, forces in the two coordinate systems relate as

$$\boxed{\tau = J(q)^T f}$$

Let D and M denote the same inertia expressed in x and q coordinates respectively.
Kinetic energy is coordinate-independent:

$$\dot{q}^T M \dot{q} = \dot{x}^T D \dot{x} = \dot{q}^T J^T D J \dot{q}$$

Since this holds for any velocity, inertias in the two coordinate systems relate as

$$\boxed{M(q) = J(q)^T D(x) J(q)}$$

Equality constraints in generalized coordinates

Equality constraints are handled as in the case of point-mass dynamics:
we solve the linear in $\ddot{q}, \lambda$ equation

$$M(q)\ddot{q} + c(q,\dot{q}) = \tau + J(q)^T \lambda$$
$$J(q)\ddot{q} + \dot{J}(q)\dot{q} = 0$$

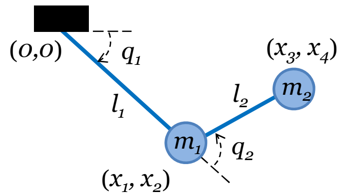
Here the constraints are $\phi(q) = 0$ and the Jacobian is $J(q) = \frac{\partial \phi}{\partial q}$

Equality constraints are often used to create kinematic loops (e.g. holding hands)

In simulations, the constraints can be violated numerically due to integration errors.
Thus it is necessary to introduce constraint stabilization (resembling PD control).

Example: 2-link arm

$$D = \begin{pmatrix} m_1 & & & \\ & m_1 & & \\ & & m_2 & \\ & & & m_2 \end{pmatrix}$$



Implicit constraints:

$$0 = \phi(x) = \begin{pmatrix} x_1^2 + x_2^2 - l_1^2 \\ (x_3 - x_1)^2 + (x_4 - x_2)^2 - l_2^2 \end{pmatrix} \qquad J_\phi(x) = 2\begin{pmatrix} x_1 & x_2 & 0 & 0 \\ x_1 - x_3 & x_2 - x_4 & x_3 - x_1 & x_4 - x_2 \end{pmatrix}$$

Explicit constraints:

$$x = h(q) = \begin{pmatrix} l_1 \cos(q_1) \\ l_1 \sin(q_1) \\ l_1 \cos(q_1) + l_2 \cos(q_1 + q_2) \\ l_1 \sin(q_1) + l_2 \sin(q_1 + q_2) \end{pmatrix} \qquad J_h(x) = \begin{pmatrix} -l_1 \sin(q_1) & 0 \\ l_1 \cos(q_1) & 0 \\ -l_1 \sin(q_1) - l_2 \sin(q_1 + q_2) & -l_2 \sin(q_1 + q_2) \\ l_1 \cos(q_1) + l_2 \cos(q_1 + q_2) & l_2 \cos(q_1 + q_2) \end{pmatrix}$$

Fast recursive computation of M and c

Computing $M = J^T D J$ and $c = J^T D \dot{J} \dot{q}$ directly is inefficient.
Instead one can use faster algorithms exploiting the structure of kinematic trees.
Let s_i be the 6D motion vector of the (1-dof) joint connecting body i to its parent.

Composite Rigid Body algorithm for computing the inertia matrix $M(q)$

(1) backward recursion:

$$D_i^{\text{comp}} = D_i + \sum_{j \in \text{children}(i)} D_j^{\text{comp}}$$

(2) set:

$$M_{ij} = \begin{cases} s_j^T D_i^{\text{comp}} s_i & \text{if } i \in \text{descendants}(j) \\ s_j^T D_j^{\text{comp}} s_i & \text{if } j \in \text{descendants}(i) \\ 0 & \text{otherwise} \end{cases}$$

Recursive Newton-Euler algorithm for computing the inverse dynamics $(q, \dot{q}, \ddot{q}) \to \tau$

(1) forward recursion:

$$\dot{x}_i = \dot{x}_{\text{parent}(i)} + s_i \dot{q}_i$$
$$\ddot{x}_i = \ddot{x}_{\text{parent}(i)} + \dot{s}_i \dot{q}_i + s_i \ddot{q}_i$$

(2) backward recursion:

$$f_i = D_i \ddot{x}_i + \dot{x}_i \times D_i \dot{x}_i + \sum_{j \in \text{children}(i)} f_j$$

(3) set:

$$\tau_i = s_i^T f_i$$

running this algorithm with $\ddot{q} = 0$ yields $-c(q, \dot{q})$

Once M and c are computed, we can compute $\ddot{q} = M^{-1}(\tau - c)$ and integrate.

Dynamics in generalized coordinates

$$M(q)\ddot{q} + c(q,\dot{q}) = \tau + g(q)$$

where $c_k(q,\dot{q}) = \sum_{ij} \Gamma_{ij,k}(q) \dot{q}_i \dot{q}_j$

$$\Gamma_{ij,k}(q) = \frac{1}{2}\left(\frac{\partial M_{ik}(q)}{\partial q_j} + \frac{\partial M_{jk}(q)}{\partial q_i} - \frac{\partial M_{ij}(q)}{\partial q_k} \right)$$

M inertia matrix
c Coriolis and centrifugal forces
g gravitational forces
τ applied/control forces
Γ Christoffel symbols

This can be derived from the **Euler-Lagrange equation:**

$$\frac{d}{dt}\frac{\partial L(q,\dot{q})}{\partial \dot{q}} - \frac{\partial L(q,\dot{q})}{\partial q} = \tau$$

where the *Lagrangian* is the kinetic energy minus the potential energy:

$$L(q,\dot{q}) = K(q,\dot{q}) - P(q)$$
$$K(q,\dot{q}) = \frac{1}{2}\dot{q}^T M(q) \dot{q}$$
$$P(q) = \sum_n 9.81 m_n h_n(q), \quad g(q) = -\frac{\partial P(q)}{\partial q}$$

If M does not depend on q, then c=0 and we have Newton's second law: $M\ddot{q} = \tau + g$

Hamiltonian formulation

The same dynamics can be obtained from the equivalent Hamiltonian formulation, based on the *Hamiltonian* $H = K + P$ instead of the Lagrangian $L = K - P$.
Now the state is represented in terms of q and the generalized momentum $p = M(q)\dot{q}$

H and L are related by the *Legendre* transformation $H = \dot{q}^T p - L$

Kinetic energy in the new coordinates is $K(q,p) = \frac{1}{2} p^T M(q)^{-1} p = \frac{1}{2}\dot{q}^T M(q)\dot{q} = K(q,\dot{q})$

Hamilton's equations are:

$$\dot{p} = -\frac{\partial H(q,p)}{\partial q} + \tau$$

$$\dot{q} = \frac{\partial H(q,p)}{\partial p}$$

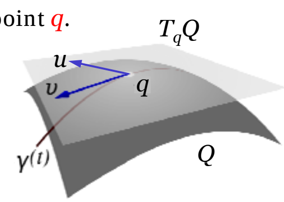
The rate of change of the Hamiltonian (i.e. the total energy) equals power:

$$\frac{d}{dt}H(q,p) = \frac{\partial H}{\partial q^T}\dot{q} + \frac{\partial H}{\partial p^T}\dot{p} = \frac{\partial H}{\partial q^T}\frac{\partial H}{\partial p} - \frac{\partial H}{\partial p^T}\frac{\partial H}{\partial q} + \frac{\partial H}{\partial p^T}\tau = \dot{q}^T\tau$$

In the absence of external forces, the Hamiltonian is conserved.

Manifolds and metrics

Q is a differentiable manifold and $T_q Q$ the tangent space at point q.
$T^*_q Q$ denotes the co-tangent (or dual) space.



A *metric* defines a dot-product on the tangent space:

$$\langle u,v\rangle_q = u^T M(q) v = \sum_{ij} M_{ij} u^i v^j \equiv M_{ij} u^i v^j \text{ (Einstein)}$$

The manifold is *Riemannian* if $M(q)$ is s.p.d. for all q.

The dot-product on the co-tangent space is defined by the inverse of M:

$$\langle u^*,v^*\rangle_q = u^{*T} M(q)^{-1} v^* = M^{ij} u_i v_j \quad \text{where} \quad (M^{ij}) \equiv (M_{ij})^{-1},\, u = (u^i),\, u^* = (u_i)$$

The metric provides the mapping between the two spaces:

$$u^* = Mu,\, u = M^{-1}u^*; \quad \text{in coordinates, } u_i = M_{ij}u^j,\, u^i = M^{ij}u_j$$

Tangent and co-tangent vectors are multiplied directly: $u^T v^* = u^i v_i = u^T M v$

Application to multi-joint dynamics:
The configuration space of a multi-joint system is a Riemannian manifold with metric given by the joint-space inertia matrix $M(q)$. The tangent vectors are velocities $\dot{q}$.
The co-tangent vectors are forces f and momenta $p = M(q)\dot{q}$.
$p^T\dot{q}$ is kinetic energy; $f^T\dot{q}$ is power.

Covariant derivatives and geodesics

The tangent basis vectors are associated with partial derivatives: $e_i = \partial/\partial q_i$
The co-tangent basis vectors are associated with differential forms: $\varepsilon_i = dq_i$
If *f(q)* is scalar and $v = v^i e_i$ is a tangent vector, then vf is the directional derivative:

$$vf = v^i e_i f = v^i \frac{\partial f}{\partial q_i} = v^T \text{ grad}(f)$$

A *connection* specifies how nearby coordinate frames "connect", i.e. how the basis vectors change over the manifold. The usual vector directional derivative is replaced with the *covariant derivative,* defined in coordinates by the *Christoffel* symbols $\Gamma_{ij}^k(q)$

$$\nabla_{e_i} e_j = \Gamma_{ij}^k\, e_k$$

For general vectors $u = u^i e_i, v = v^i e_i$ the covariant derivative is $\nabla_v u = \left(v^i \frac{\partial u^k}{\partial q_i} + \Gamma_{ij}^k u^i v^j \right) e_k$

A connection is *flat* when $\Gamma_{ij}^k = 0$. In that case we recover the regular derivative.

For a Riemannian manifold with metric $M(q)$, there exists a unique metric-preserving torsion-free connection (the *Levi-Civita* connection) with Christoffel symbols:

$$\Gamma_{ij}^k = M^{ks} \Gamma_{ij,s} \qquad \Gamma_{ij,s} = \frac{1}{2}\left(\frac{\partial M_{is}}{\partial q_j} + \frac{\partial M_{js}}{\partial q_i} - \frac{\partial M_{ij}}{\partial q_s} \right)$$

A *geodesic* is a curve *γ(t)* such that $\nabla_{\dot\gamma} \dot\gamma = 0$, i.e. $\ddot\gamma^k + \Gamma_{ij}^k \dot\gamma^i \dot\gamma^j = 0$ for all *k*.
This is called the *geodesic equation*.

Unforced motions as geodesics

The unforced motions of a multi-joint system satisfy $M(q)\ddot q + c(q,\dot q) = 0$

which we can rewrite (using the fact that *M* is s.p.d.) as $\ddot q + M(q)^{-1} c(q,\dot q) = 0$

Recalling the expression for *c*, this can be written in component form as

$$\ddot q_k + \left(M^{-1}c\right)_k = \ddot q_k + \sum_{ij} \Gamma_{ij}^k \dot q_i \dot q_j = 0$$

where $\Gamma_{ij}^k = \sum_s M_{ks}^{-1} \Gamma_{ij,s}$ and $\Gamma_{ij,s} = \frac{1}{2}\left(\frac{\partial M_{is}}{\partial q_j} + \frac{\partial M_{js}}{\partial q_i} - \frac{\partial M_{ij}}{\partial q_s} \right)$

Thus we have recovered the geodesic equation $\nabla_{\dot q} \dot q = 0$

The Levi-Civita connection for the Riemannian metric defined by the inertia matrix is called the ***mechanical connection***. Its geodesics are the unforced motions.

With external forces and gravity, the dynamics become $M(q) \nabla_{\dot q} \dot q = \tau + g$

This is equivalent to Newton's s second law,
with the covariant derivative in place of the regular derivative: $\frac{d}{dt}\dot q \to \nabla_{\dot q} \dot q$