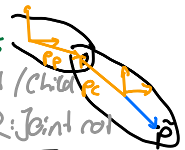
KINEMATICS

Coordinate Frames



p_p, p_c: Joint pos in Parent / Child
$\tilde{p}$: point in Rigid Body R: Joint rot

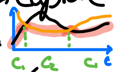
$p_{inParent} = p_p + R(\tilde{p} - p_c)$ For world coordinates, assume Parent = world

Joint Types
- Pivot / Hinge: 1DoF
- Condyloid / Metacarpal / Plane: 2DoF
- Ball: 3DoF

General Coordinates
\# Base DoF (usually 6) + \# Actuated DoF

Forward Kinematics
Given pose q where is $\tilde{p}$ in world coords $p(q, \tilde{p})$
Application from leaf to root of: $p_{inParent} = p_p + R(\tilde{p} - p_c)$ Then final rot/trans
$p(q) = x_{root} + {}_wR_{root}\,\bar{p}^{bo}$

Motion Graphs
Given Motion Capture clips we can sequence them



① Build similarity matrix from clips
② Build Graph of possible transitions

Inverse Kinematics
Given target $\tilde{x}$, what pose θ reaches it?
Solved by finding $\min_\theta E(\theta)$

$$E(\theta) = \frac{1}{2}(\tilde{x} - x(\theta))^T(\tilde{x} - x(\theta))$$

$$\nabla_\theta E = \frac{dE}{d\theta} = \frac{dx^T}{d\theta}\cdot\frac{dE}{dx} = \frac{dx^T}{d\theta}(\tilde{x} - x(\theta)) = J^T(\tilde{x} - x(\theta))$$

Gradient Descent (first-order)
$\theta_{k+1} = \theta_k - \alpha\nabla f(x_k)$
⊖ Solution can oscillate $|[\nabla f(x_k)]_i| \gg |[\nabla f(x_k)]_j|$
⊖ Takes many small steps (slow convergence)

Jacobian Transpose
$\theta_{k+1} \leftarrow \theta_k + \alpha J^T(\tilde{x} - x(\theta))$ used for badly conditioned Jacobians. Guaranteed convergence for small α

Newtons Method (second-order)
$\theta_{k+1} = \theta_k - \alpha(\nabla^2 f(x_k))^{-1}\nabla f(x_k) = \theta_k - \tau H^{-1}g$
Fits a quadratic function at θ_k and steps to it's minimum

Regularizers
We want $H \succ 0$ (pos. def) since then every direction is a decent. To "fix" negative Eigenvalues add regularizer
$H + Ir = V^T\Sigma V + Ir = V^T(\Sigma + Ir)V \xrightarrow{r\to\infty} Ir$
So Newton's Method becomes GD for $r \to \infty$
Regularizer minimizes $E(\theta) = \|\tilde{x} - x(\theta)\|^2 + r^2\|\theta\|^2$

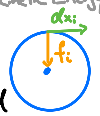
Jacobian Pseudo-Inverse
$H^{-1}g \approx (J^TJ)^{-1}J^T =: J^+$ (Gauss-Newton)
The Hessian is approximated by J^TJ
So: $\theta_{k+1} \leftarrow \theta_k + \alpha(J^TJ)^{-1}J^T(\tilde{x} - x(\theta))$

Taylor Expansion
$O(x + \Delta x) = O(x) + \Delta x^T\nabla_x O + \frac{1}{2}\Delta x^T\nabla_x^2 O\,\Delta x + O(\|\Delta x\|^3)$

DYNAMICS

Particle Dynamics
$v(t) = \dot{x}(t)$ $a(t) = \dot{v}(t) = \ddot{x}(t)$
Linear momentum: $p(t) = mv(t)$
Newton's 2nd Law: $f(t) = ma(t) = \dot{p}(t)$

Particle Systems
Conservation of Momentum
$F = \sum_i f_i = 0 \Leftrightarrow \sum_i \dot{p}_i = \dot{P} = 0 \Leftrightarrow P$ const
Linear momentum is conserved if $F = 0$
Work: $\int_{t_1}^{t_2} f_i \cdot dx_i = \int_{t_1}^{t_2} \frac{d}{dt}\left(\frac{1}{2}mv^2\right)dt = T_{t_2} - T_{t_1}$ ↖ Kinetic Energy T
- Force does no work $\int f_i\,dx_i = 0 \Rightarrow T_2 = T_1$ example:



- Force derives from potential U
$f_i = -\frac{\delta U}{\delta x_i}$ $\int f_i\,dx_i = \int -\frac{\delta U}{\delta x_i}dx_i = U_1 - U_2 \Rightarrow -\Delta U = \Delta T$

Generalized Coordinates
Describe state of each individual DoF of a system
position: $x_i(t) = x_i(q(t))$
velocity: $\dot{x}_i(t) = \frac{dx}{dq}\frac{dq}{dt} = J\dot{q}(t)$
Newton's 2nd Law: $f_q(q, \dot{q}, t) = M_q(q)\ddot{q}$ ← generalized accelerations
Kinetic Energy: $\frac{1}{2}\dot{x}^TM\dot{x} = \frac{1}{2}\dot{q}^TJ^TMJ\dot{q} = \frac{1}{2}\dot{q}^TM_q\dot{q}$

ODEs
Equation of the form: $y^{(n)} = f(t, y, y', \dots, y^{(n-1)})$
Derivatives only w.r.t a <u>single</u> variable.

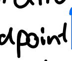
Initial Value problems
Define $y(0) = C$
Picard-Lindeloef: For $\dot{y} = ay \rightarrow y(t) = Ce^{at}$
If f is Lipschitz cont. then the IVP sol. is unique

Lagrangian Mechanics
Kinetic Energy: K (due to $\dot{q}$)
Potential Energy: U (due to $m\,g\,g$) gen. mc = gen. F
Lagrangian: $L = K - U$ Euler-Lagrange: $\frac{d}{dt}\frac{\delta L}{\delta \dot{q}} = \frac{\delta L}{\delta q}$

Newtonian Mechanics
Free Cut method in regular coordinates
① $F = ma$, $0 \cdot \ddot{q} = K$ ② Add constraint forces

Numerical ODE Solvers
$\dot{q}(t) = f(t, q) \rightarrow q(t+h) = q(t) + \int_t^{t+h}\dot{q}(t)\,dt$

Numerical Integration

Rectangle  Midpoint  Trapezoid 

$\Delta q_i \approx \dot{q}(t)\cdot h$ $\Delta q_i \approx \dot{q}(t + h/2)\cdot h$ $\Delta q_i \approx \frac{\dot{q}(t)h + \dot{q}(t+h)h}{2}$

Explicit Euler
$q_{i+1} = q_i + h\cdot\dot{q}_i$ (Rectangle rule)
Explicit: No future timesteps needed
⊖ Simulation explodes for h too large

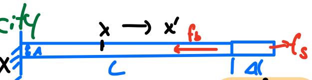
Implicit Euler
$q_{i+1} = q_i + h\dot{q}_{i+1}$
Need to solve for future time step $\dot{q}_{i+1}$
⊖ Gives small damping Effect

Evaluation Criteria
Convergence: $q_i \rightarrow q(t_i)$ as $h \rightarrow 0$
Accuracy: How fast does $|q_i - q(t_i)| \xrightarrow{h\to 0} 0$
Stability: Solution bounded $|q_i| < \infty$
Efficiency: # Function Evaluations
Stiffness: The ODE requires very small time steps in smooth parts of the solution

CONTINUUM MECHANICS

Elasticity with Mass-Spring Systems
Discretize Mesh & add spring at each Edge
$f_{12} = -k\left(\frac{|x_1 - x_0|}{L_0} - 1\right)\frac{x_1 - x_0}{|x_1 - x_0|}$
⊕ Easy to understand / implement
⊖ Behavior depends on mesh / No area volume preservation

Continuous Elasticity



Displacement: $u(x) = x' - x$
Strain (rel. stretch): $\varepsilon = \frac{\Delta L}{L}$, at arbitrary point: $\varepsilon = \frac{\delta u}{\delta x}$
Stress (internal force per area): $\sigma = f_{int}/A$
→ Hooke's Law: $\sigma = E\varepsilon$ (E: elasticity modulus)
↖ c.f. in a spring $F = k\Delta L$

Net internal force: $f_{int} = A\frac{\delta\sigma}{\delta x} = AE\frac{\delta^2 u}{\delta x^2}$

Discretization Strategies

Continuous
Deformation + Material Model
⇓
Strain Energy

FEM Discretization (Governing Equations (PDE) ⇓ Weak Form ⇓ Discretization) → Algebraic Equations

Discrete Energy (conservative potentials only) (Discretization ⇓ Per-element energy ⇓ Minimum Energy Principle) → Algebraic Equations

Equilibrium Conditions
$x \in]0, 1[$

$EA\,\delta_{xx}u(x) = -f^b(x)$ (ext. int. F balance)
$EA\,\delta_x u(1) = -f^s$ (force constraint)
$u(0) = 0$ (disp constraint)

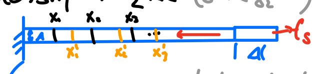
Strong Form PDE
- Requires u 2x diffable
- Displacement + Force constraints

→ Solve by finite Differences

Weak Form PDE
Reformulation of a Strong Form PDE by using test functions sat. boundary conditions
① Choose satisfying test function $\bar{u}(x)$, $\bar{u}(x) = 0$
② Assume PDE holds under $\bar{u}$
$\int_0^1(EA\,\delta_{xx}u(x) + f^b(x))\,\bar{u}(x)\,dx = 0$
$\Rightarrow$ $\int_0^1 EA\,\delta_x u\,\delta_x\bar{u}\,dx = \int_0^1 f^b\bar{u}\,dx - f^s\bar{u}(1)$
$u(0) = 0$
$\forall\bar{u}$ s.t. $\bar{u}(0) = 0$
- Requires u 1x differentiable
- Only displacement constraints

Discrete Energy
↙ Simplest postulate
Strain energy density: $\Psi = \frac{1}{2}k\varepsilon^2$ (Since $\frac{\delta\Psi}{\delta\varepsilon} = \sigma$)

Element Strain



$\varepsilon_{ij} = \frac{x'_{i+1} - x'_i - L_i}{L_i}$ $L_i = x_{i+1} - x_i$
constant per element
Element strain Energy: $W_i = \int\Psi_i(x)\,dx = \frac{1}{2}k\varepsilon_i^2 AL_i$
Total strain Energy: $W = \sum_i W_i$
Forces: $f_i = -\frac{\delta W}{\delta x'_i} = -\frac{\delta W_{i-1}}{\delta x'_i} - \frac{\delta W_i}{\delta x'_i} = -kA(\varepsilon_{i-1} - \varepsilon_i)$ $f_1 = -k\varepsilon_1$, $f_n = -k\varepsilon_{n-1}$

3D Deformations

Displacement Field: $u(x) = x' - x$; $d = x_1 - x_2$; $d' = x_1' - x_2'$

Deformation Gradient: $d' \approx \underbrace{(I + \nabla u)}_{=:F} d$

3D Strain Deformation at a point

$|d'|^2 - |d|^2 = d'^T d' - d^T d \approx d^T (F^T F - I) d$

Green Strain: $E = \frac{1}{2}(F^T F - I)$

$= \frac{1}{2}(\nabla u + \nabla u^T + \nabla u^T \nabla u)$ (quadratic in ∇u)

⊕ Rotation Invariant (for $F' = RF$: $E' = E$)

Cauchy Strain: (drops quadratic term)

$\varepsilon = \frac{1}{2}(\nabla u + \nabla u^T) = \frac{1}{2}(F + F^T) - I$ ⊖ varies with rotations

3D Stress

Traction force f_n on dA

Traction force density $t_n = \frac{df_n}{dA}$ for $dA \to 0$

Cauchy's stress: $t(x, n) = \sigma(x) \cdot n$ (Cauchy stress tensor)

Material Model: Links strain and Energy (stress)

• Linear Isotropic Material

Energy Density: $\Psi = \frac{1}{2}\lambda tr(\varepsilon)^2 + \mu tr(\varepsilon^2)$

Cauchy Stress: $\sigma = \frac{\delta\Psi}{\delta\varepsilon} = \lambda tr(\varepsilon)^2 + \mu tr(\varepsilon)$

→ λ, μ are Lamé parameters (Material constants)

• $tr(\varepsilon^2) = tr(\varepsilon^T\varepsilon) = \|\varepsilon\|_F^2$ penalizes all strain components

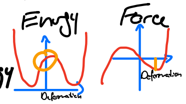
• $\lambda tr(\varepsilon)^2$ penalizes dilatations (Volume changes)

⊖ Relies on Cauchy strain → varies with rot

St. Venant Kirchhoff Material (StVK)

Replace Cauchy strain with Green strain

$\Psi_{StVK} = \frac{1}{2}\lambda tr(E)^2 + \mu tr(E^2)$



⊖ Compression to 0 with finite Energy

⊖ Force decreases with more compression

Workaround: $\Psi_{mod} = 2(\det(F) - 1)^2 + \mu tr(E^2)$

Neo-Hookean Material — adds Volume term

$\Psi_{NH} = \frac{\mu}{2}(tr(C) - 3) - \mu \ln J + \frac{\lambda}{2}\ln(J)^2$

• $tr(C) = \|F\|_F^2$ penalizes all deformations equally

• $J = \det(F) \to \infty$ for increasing compression

• stress-strain initially linear then plateau ⊕ good for ≤20% comp.

Finite Elements

Every Element consists of:

n nodal basis functions $N_i: \Omega_e \to \mathbb{R}$

n vectors of nodal variables $\bar{x}_i \in \mathbb{R}^d$

For all basis functions: $N_i(\bar{x}_i) = \delta_{ij} = \begin{cases} 1 & i=j \\ 0 & \text{else} \end{cases}$

2D: $N_i(x, y) = a_i x + b_i y + c_i$ 3D: $N_i(x, y, z) = a_i x + b_i y + c_i z + d$

↳ Obtain basis functions through linear sys. solve

⇒ Interpolation: $x(\bar{x}, \bar{y}, \bar{z}) = \sum_i N_i(\bar{x}, \bar{y}, \bar{z}) x_i$

Deformation Gradient

$x = F\bar{x}$

$F = I + \frac{\delta u(\bar{x})}{\delta\bar{x}} = \frac{\delta x(\bar{x})}{\delta\bar{x}} = \sum_i x_i \left(\frac{\delta N_i}{\delta\bar{x}}\right)^T$

Since N_i are linear on the element → F const.

So $W^e = \int_{\Omega_e} \Psi = \Psi(F) \cdot \bar{V}^e$

Nodal Forces and Force Jacobian

Force at each node: $f_i = -\frac{\delta W}{\delta x_i}$ (neg Energy Gradient)

Total Energy is sum of Elements: $W = \sum_e W_e$

$W_e \xleftarrow[\text{depends}]{\text{non-linear}} F_e \xleftarrow[\text{depends}]{\text{linear}} x_i$ Force Jacobian = - Energy Hessian

Time Integration

① For each node: $m a_i = f_i = f_i^{int} + m g$

② Convert to first order ODE system: $\dot{x} = v$, $\dot{v} = M^{-1} f(x)$

③ Discretize (Implicit Euler)

$x_{n+1} = x_n + h v_{n+1}$ $\quad v_{n+1} = v_n + h a_{n+1}$

④ Solve using Newton's Method

$x_{n+1} = x_n + h v_{n+1} = x_n + h v_n + h^2 M^{-1} f(x_{n+1})$

⇒ $g(x_{n+1}) = M(x_{n+1} - x_n) - h^2 f(x_{n+1}) - hM v_n = 0$

CONSTRAINED OPTIMIZATION

Optimization $\min_x f(x)$ s.t. $C(x) = 0$

Equilibrium Constraint i.e. $f^{ext} + f^{int} = 0$

Result from equations of motion

Equality Constraints

Assume $C(x) = 0$ for some x. Then $\exists\, dx$ such that:

$C(x + dx) = 0 \;\wedge\; f(x + dx) < f(x)$

For linear constraints: $C(x + dx) = C(x) + \nabla C^T dx$

So $\nabla f(x)^T dx < 0 \;\wedge\; \nabla C^T dx = 0$ $\quad \lambda \in \mathbb{R}$

⇒ At optimum $\nabla f = \lambda \nabla C$ ⟺ $\begin{bmatrix} \nabla f(x)^T \\ \nabla C(x)^T \end{bmatrix} dx = \begin{bmatrix} -1 \\ 0 \end{bmatrix}$ has no solution

First Order Optimality

$\nabla f(x) + \nabla C(x)^T \lambda = 0$ and $C(x) = 0$

(KKT Conditions) necessary, but not sufficient

Strict Minimizer $\forall dx \neq 0$: $f(x + \alpha dx) > f(x)$

Minimizer: $\forall dx \neq 0$: $f(x + \alpha dx) \geq f(x)$

Quadratic program (QP)

Assume $f(x) = \frac{1}{2}x^T H x + g^T x + a$ (quadratic)

Has m equality constraints $C(x) = Ax - b$

And nonredundant constraints & feasible

First-Order feasible dirs: $F(x) = \{d \neq 0 \mid \nabla C(x) d = 0\}$

All directions orthogonal to all constraint gradients

Second Order Optimality

Let x^* be a strict local minimizer satisfying KKT. For all feasible Dirs $w \in F(x^*)$: $\nabla f^T w = 0 \to f(x^* + \varepsilon w) = f(x^*) + \varepsilon\nabla f^T w + O(\varepsilon^2) = f(x^*)$

Cannot decide if $f(x^* + w) > f(x^*)$ or $f(x^* + w) \leq f(x^*)$

using first order information only

Taylor Expansion: $f(x^* + w) = f(x^*) + \nabla f(x^*)^T w + \frac{1}{2} w^T H w$

For a strict opt. we have: $\quad = f(x^*) + \frac{1}{2} w^T H w$

$f(x^* + w) > f(x^*)$ ⇒ $w^T H w > 0 \quad \forall w \in F$

Let $Z \in \mathbb{R}^{n \times (n-m)}$ be a basis of F (nullspace of ∇C)

Then: $w^T H w > 0 \;\forall x \in F \Leftrightarrow u^T Z^T H Z u > 0 \;\forall u \in \mathbb{R}^{n \times (n-m)}$ (pos. def)

Second Order sufficient Conditions:

① x^* satisfies KKT and ② $Z^T H Z$ is pos. def.

Lagrangian $L(x, y) = f(x) + \lambda^T C(x)$

Gradients: $\nabla_x L = \nabla f + \nabla C^T \lambda \quad \nabla_\lambda L = C(x)$

① First order optimality is $\nabla L = \begin{bmatrix} \nabla_x L \\ \nabla_\lambda L \end{bmatrix} = 0$

② Since L is unbounded, $\nabla L = 0$ is a saddle point

KKT System / Matrix

For a QP: $L(x, \lambda) = \frac{1}{2} w^T H w + g^T w + a + \lambda^T (Ax - b)$

⇒ $\nabla_x L = Hx + g + A^T\lambda \quad \nabla_\lambda L = Ax - b$

For First-Order Optimality we need $\nabla L = 0$

⇒ KKT-Matrix: $\begin{bmatrix} H & A^T \\ A & 0 \end{bmatrix}\begin{bmatrix} x \\ \lambda \end{bmatrix} = \begin{bmatrix} -g \\ b \end{bmatrix}$ • symmetric • indefinite

QP is convex and has unique solution

⟺ A is full rank and H is pos. def. on subspace orthogonal to A

Solving KKT System

• Directly: LU Decomp, Pardiso

• Iterative solvers • QP Solver

NonLinear Programming

f non-quadratic, c non-linear

• KKT conditions still necessary

• Second Order conditions still sufficient

NonLinear KKT Conditions

$L(x, \lambda) = f(x) + \lambda^T C(x)$ For $s = \begin{bmatrix} x \\ \lambda \end{bmatrix}$

Find Δs s.t. $L(s + \Delta s) = 0$ $\quad \nabla_{ss} L = \begin{bmatrix} \frac{\delta^2 L}{\delta x^2} & \frac{\delta}{\delta\lambda}\frac{\delta L}{\delta x} \\ \frac{\delta}{\delta x}\frac{\delta L}{\delta\lambda} & \frac{\delta^2 L}{\delta\lambda^2} \end{bmatrix}$

$\nabla_s L(s + \Delta s) = \nabla_s L(s) + \nabla_{ss} L(s)\Delta s + O(\Delta s^2)$

First Order Approximation: $\nabla_{ss} L \cdot \Delta s = -\nabla_s L$

$\begin{bmatrix} \nabla_{xx} L(x) & \nabla C(x)^T \\ \nabla C(x) & 0 \end{bmatrix}\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = -\begin{bmatrix} \nabla_x L \\ \nabla_\lambda L \end{bmatrix}$ $\quad \nabla_{xx} L = \nabla_{xx} f + \nabla_{xx} C^T \lambda$

However, $\nabla_s L(s + \Delta s) \neq 0$ in general. So iterate

```
Sequential Quadratic Programming
Until converged
  solve ∇ss L · Δs = -∇s L
  line search(s, Δs)
  update s = s + αΔs
end
```

⊖ $\nabla_{xx} C(x)$ can introduce indefiniteness

↳ Use approximation of Hessian (Quasi-Newton)

⊖ Line search must balance $f(x)$ and $C(x) \neq 0$

⊕ For Inequality constraints use interior point methods

Sensitivity Analysis for Equilibrium constraints

By simulating we get $x = \text{simulate}(p)$ DoF

To optimize we need $\frac{\delta T}{\delta p} = \left(\frac{\delta x}{\delta p}\right)^T \frac{\delta T}{\delta p}$ ← cannot compute this directly

$x(p)$ is not given explicitly

but through Equilibrium constraints $f(x, p) = 0$

For every Δp $\exists \Delta x$: $f(x + \Delta x, p + \Delta p) = 0$

⇒ $\frac{\delta f}{\delta p} = \frac{\delta f}{\delta x}\frac{\delta x}{\delta p} + \frac{\delta f}{\delta p} = 0$ ⇒

Sensitivity Matrix: $S = \frac{\delta x}{\delta p} = \left(\frac{\delta f}{\delta x}\right)^{-1}\frac{\delta f}{\delta p}$

maps infinitesimal Δp (parameters) to changes in equilibrium Δs

$\Delta x = S\Delta p$ $\quad$ computed row-wise by solving $\frac{\delta f}{\delta x} S_i = -\frac{\delta f}{\delta p_i}$

Sensitivity Analysis Steepest Descent (SASD)

```
Until convergance
    S = -∇x f^-1 ∇p f          objective T
    Δp = -S^T ∇x T
    α = line search(Δp)
    p = p + αΔp
    x = simulate(x, p)
end
```

RIGID BODY DYNAMICS

Rigid Body Kinematics

Center of Mass $p = \sum_i m_i x_i / M = \frac{\int_\Omega x \rho \, dV}{\int_\Omega \rho \, dV}$

Rotation: $R(t)$ $\quad \dot p(t) = v(t)$

Linear velocity $\dot x(t) = v(t) + \dot R(t)\bar x$
$= v(t) + \omega(t) \times (x(t) - p(t))$

Linear Acceleration $\ddot x(t) = a + ([\dot\omega] + [\omega]^2)(x-p)$

Rotation Derivative $\dot R(t) = [\omega(t)]_\times R$

Angular Velocity $\omega(t)$

Point $\bar x$ in local frame moves with $\dot{\bar x} = \omega \times c$

Linear Momentum $\rho = Mv(t)$
$F = Ma = \dot\rho$

Angular Momentum $L = I\omega$

Conservation of Angular Mom. $\dot L = I\dot\omega + \dot I\omega = \tau$
$\Rightarrow I\dot\omega = \tau - \omega \times I\omega$

Rigid Body Dynamics

Kinetic Energy: $K = \frac12 \sum \dot x_i^T m_i \dot x_i = \frac12 v^T M v + \frac12 \omega^T I \omega$

Inertia: $\sum_i m_i (r_i^T r_i \mathbb{1} - r_i r_i^T)$ (world coords),
$= R \underbrace{\sum_i m_i (\bar x_i^T \bar x_i \mathbb{1} - \bar x_i \bar x_i^T)}_{I_0} R^T$ (local coords)

We can always choose local coordinates s.t. $I_0 = diag(I_1, I_2, I_3)$

Force: $F(t) = \sum_i f_i(t)$

Torque: $\tau(t) = \sum_i r_i \times f_i(t)$

Newton Euler EOM

$$\begin{bmatrix} F \\ \tau - \omega \times I\omega \end{bmatrix} = \begin{bmatrix} M\mathbb{1}_{3\times3} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \dot v \\ \dot\omega \end{bmatrix}$$

Discrete Time (Symplectic Euler)

1. Compute F, τ
2. $v_{i+1} = v_i + h\frac{F}{M}$ $\quad \omega_{i+1} = \omega_i + h I^{-1}(\tau - \omega_i \times I\omega_i)$
3. Update CoM $p_{i+1} = p_i + h v_{i+1}$
4. Update Orientation: $R_{i+1} = R_i + h \dot R_{i+1}$

→ R_{i+1} will not be orthonormal, fix using Gram-Schmidt or use Quaternions

Quaternions

$q = [\omega, v]$, $v = [x, y, z]$

axis angle $(\theta_a, v_a) \to q = \begin{bmatrix} \omega \\ v \end{bmatrix} = \begin{bmatrix} \cos\frac{\theta_a}{2} \\ \sin\frac{\theta_a}{2} v_a \end{bmatrix}$

Rotating vectors: $Rot(v, q) = qv$
$= q(0, v) q^{-1}$

Discrete Time Quaternions

1. Compute F, τ
2. $v_{i+1} = v_i + h\frac{F}{M}$ $\quad \omega_{i+1} = \omega_i + h I^{-1}(\tau - \omega_i \times I\omega_i)$
3. Update CoM $p_{i+1} = p_i + h v_{i+1}$
4. Update Orientation: $q_{i+1} = q\left(h|\omega_{i+1}|, \frac{\omega_{i+1}}{|\omega_{i+1}|}\right) q_i$
or $\dot q_{i+1} = \frac12 (0, \omega_i) q_i$ $\quad q_{i+1} = q_i + h \dot q_{i+1}$

→ Still have to normalize q_{i+1} but it's much easier

ARTICULATED RIGID BODIES

Spring Joints

Model Joints as Hookean springs:
$f_s = -k_p(x_a - x_b) - k_d(\dot x_a - \dot x_b)$

⊖ Always gives non-zero error for const F

⊖ Oscillations when undamped

⊕ Good approximation for some real Systems

Velocity Level Constraints

1. $C(p) = 0$ when the system is in a valid config.
p: position and orientation of all rigid bodies
$C = x_a - x_b$
2. $\dot C = \frac{dC}{dt} = \frac{\delta C}{\delta p} \frac{dp}{dt} = Av$ $\quad$ A: Constraint Jacobian, v: Un-tang velocities
$A = [\mathbb{1} \; -[R_a \bar r_a] \; -\mathbb{1} \; [R_b \bar r_b]]$ $\quad v = [v_a^T \, \omega_a^T \, v_b^T \, \omega_b^T]^T$
3. $v_{i+1} = v_i + hM^{-1}F$ with $F = F_{ext} + F_c$ $\quad F_c = A^T\lambda$
λ: Lagrange mult. $A^T\lambda$ Projects system towards constraint manifold as fast as possible

Compute λ: 1. directly: better numerically

$\lambda = -k_p C_i - k_d \dot C_i$ (spring forces) $\rightarrow \lambda = -k_p C_{i+1} - k_d \dot C_{i+1}$ (implicit)

$C_{i+1} = C_i + h\dot C_{i+1} \Rightarrow \lambda = -k_p C_i - (hk_p + k_d)\dot C_{i+1}$

$\Leftrightarrow \dot C_{i+1} = -\underbrace{\frac{1}{hk_p + k_d}}_{r} \lambda - \underbrace{\frac{k_p}{hk_p + k_d}}_{c} C_i$

Set r, c by choosing k_p, k_d (stiffness, damping)

2. as a target for $\dot C_{i+1}$: $C_{i+1} \approx C_i + h\dot C_{i+1}$
Want $C_{i+1} = 0$ and can only influence $\dot C_{i+1}$
Choose: $\dot C_{i+1} = -\frac{\varepsilon}{h} C_i$ usually $\varepsilon \approx 0.3$

$\dot C_{i+1} = Av_{i+1} = Av_i + hAM^{-1}(F_{ext} + hAM^{-1}A^T\lambda)$

$\underbrace{(hAM^{-1}A^T + r\mathbb{1})}_{D}\lambda = \underbrace{-Av_i - hAM^{-1}F_{ext} - \frac{\varepsilon}{h}C_i}_{d}$

$\Rightarrow D\lambda = d \Rightarrow \lambda = D^{-1}d$ ← not always invertible

Lagrangian Mechanics

1. Choose generalized coordinates q. $x(q)$ to cartesian coords must be explicitly given
$\dot x = \frac{\delta x}{\delta q}\dot q = J\dot q$
2. Setup Lagrangian $L = K - U$ ← kinetic Energy, potential energy
3. Solve Euler-Lagrange: $\frac{d}{dt}\frac{\delta L}{\delta \dot q} = \frac{\delta L}{\delta q}$

Usually $K = \frac12 \dot x^T M \dot x$ $\quad U = -F^T x$ (pendulum) mgh

$M(q, \dot q) + C(q, \dot q) = Q = J^T M J \ddot q + J^T M \dot J \dot q - J^T F$
gen. mass, coriolis & centrifugal terms, gen. forces

CONTACT MECHANICS

Contact Forces

Normal: Preventing intersections

Tangential: Friction

Complementarity Problem

Find $z \in \mathbb{R}^n$: $z \ge 0, f(z) \ge 0$ and $z_i f_i(z) = 0 \;\forall i$ $\quad f: \mathbb{R}^n \to \mathbb{R}^n$

Shorthand: $0 \le z \perp f(z) \ge 0$

Allows modelling of contact forces without discontinuities

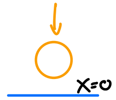
Linear Complementarity Problem (LCP)

Find $z \in \mathbb{R}^n$: $0 \le z \perp f(z) \ge 0$ where $f(z) = Az + q$

Alternative Formulation: $\quad f: \mathbb{R}^n \to \mathbb{R}^n$ (affine)
$Az + q = w$ ← w is a slack variable
$0 \le w \perp z \ge 0$ product between opt. variables

• LCP is non-linear • Can be transformed to a QP

Contact Constraints 1D



EoM: $m\ddot x = -9.81m + \lambda$ $\quad x \ge 0$

Contact does not pull: $\lambda \ge 0$

Contact force only when in contact: $x\lambda = 0$

$\Rightarrow 0 \le x \perp \lambda \ge 0$

Impact Law

Describes behavior at impact

velocity reflection: $x = 0 \wedge \dot x^- < 0$ pre impact
post impact $\Rightarrow \dot x^+ = -\dot x^-$

Free flight: $x > 0$
$m\ddot x = -9.81m$

In contact: $x = 0 \wedge \dot x^- < 0$
$m\ddot x = -9.81m + \lambda$
$0 \le \lambda \perp (\dot x^+ + \dot x^-) \ge 0$

• Models Restitution with impact law

Time Discretization

1. To FO ODE: $m\ddot x = f \Rightarrow m\dot v = f \wedge \dot x = v$
2. Time Discret.: $m(v_{i+1} - v_i) = \Delta_t f(x_{i+1})$
$x_{i+1} - x_i = \Delta_t v_{i+1}$
3. Force Balance: $\underline{m v_{i+1} - m v_i - \Delta_t f(x_i + \Delta_t v_{i+1})} = 0$
4. Adding constraints: $r(v_{i+1}) = 0$
$r(v_{i+1}) = \lambda$ $\quad 0 \le \lambda \perp (\underbrace{x_i + \Delta_t v_{i+1}}_{x_{i+1}}) \ge 0$
Contact Constraints are integrated as a complementarity condition w/ slack variable
5. Factor out velocities unknown Δv:
$r(v + \Delta v) \approx r(v) + \frac{dr}{dv}\Delta v$ $\quad$ LCP: $\frac{dr}{dv}\Delta v = \lambda - r(v)$
$0 \le \lambda \perp (x_i + \Delta t(v + \Delta v)) \ge 0$
6. Formulate as QP:
$\min \frac12 \Delta v^T \frac{dr}{dv} \Delta v + \Delta v \, r(v)$ $\quad$ λ, comp. cond.
s.t. $\Delta v \ge -(v + \frac{1}{\Delta t} x_i)$ $\quad$ now implicit

Hard vs Soft Constraints

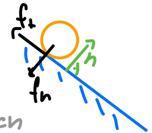
Hard Constraints: $x \ge 0$ $\quad f_{floor} = \lambda$ →

Soft Constraints: $x < 0$: $f_{floor} = kx$ (stiffness); $x \ge 0$: $f_{floor} = 0$

⊕ Unconstrained Problem ⊖ Requires constraint violation for penalty to act

⊖ Stiffness important

Coulomb Friction



Coulomb model: $f_t \le \mu f_n$

when sliding under friction

μ: Coefficient of friction (≈ 0.3 - 0.6)

Contact force: $f_{floor} = f_n + f_t$

Static: $f_t < \mu f_n \wedge v_t = 0$ $\quad$ sliding: $f_t = \mu f_n$

Maximum Dissipation Principle

Set of all admissible tangential forces:
$F = \{f_t \mid \|f_t\| \le \mu \|f_n\|\}$ $\quad f_t$ maximizes energy dissipation: $f_t = \arg\max_{f_t \in F} -f_t^T \dot x$
(opposite to velocity at contact point)

Admissible Friction Forces

max dissip. sliding: $\max(-f_t^T T \dot x)$ s.t. $\|f_t\| = \mu \|f_n\|$ $\quad T = (I - nn^T)$

⇒ Admissible friction force depends on normal force

Friction Cone $\|f_t\| \leq \mu \|f_n\|$

Admissible cone of friction forces

⊕ radius depends linearly on $\|f_n\|$ ⊖ Non-linear friction constraint difficult to model

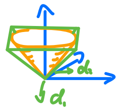
Pyramidal Friction cone

Polygonal friction cone approximation.

Non-linear cone $\|f_t\| \leq \mu \|f_n\|$

Pyramid $f_t = \beta_1 d_1 + \beta_2 d_2$

$-\tilde{\mu}\lambda \leq \beta_i \leq \tilde{\mu}\lambda$



Inner Approximation $\tilde{\mu} = \mu / \sqrt{2}$

⊖ Introduces Directional Bias

m-sided approx.: $f_t = D\beta \quad \mathbb{1}^T\beta \leq \mu\lambda \quad \beta \geq 0$

Simulation

① Collision Detection (normal & point)

② Apply Impact Law

③ Include approx. friction constraints

Equations: $r(v_{i+1}) = n\lambda + D\beta$

$0 \leq \lambda \perp n^T(v_{i+1} - v_i) \geq 0$

$\mathbb{1}^T\beta \leq \mu\lambda \perp \gamma \geq 0 \quad -D^T v_{i+1} \leq \gamma \mathbb{1} \perp \beta \geq 0$

④ Eliminate Δv

$r(v_{i+1}) = n\lambda + D\beta \rightarrow r(v + \Delta v) \approx r(v) + S\Delta v$

with $S = \frac{dr}{dv} \quad \Delta v = S^{-1}(n\lambda + D\beta - r(v))$

⑤ LCP Form:

$$z = \begin{pmatrix} \lambda \\ \beta \\ \gamma \end{pmatrix} \quad A = \begin{pmatrix} n^T S^{-1} n & n^T S^{-1} D & 0 \\ D^T S^{-1} n & D^T S^{-1} D & \mathbb{1} \\ \mu & -\mathbb{1}^T & 0 \end{pmatrix}$$

Soft Constraints Formulation

Normal Forces: $n^T x \geq 0$, $f_n = n\lambda$ $\rightarrow$ $n^T x < 0: f_n = -k_n n n^T x$; $n^T x \geq 0: f_n = 0$

Static: $T\dot{x} = 0$ $\longrightarrow$ $f_t = -k_t T\dot{x}$ if $\|f_t\| < \mu\|f_n\|$

$T := (I - nn^T)$

Sliding: $\max(-f_t^T \dot{x})$ s.t. $\|f_t\| = \mu\|f_n\|$ $\longrightarrow$ $f_t = -\mu\|f_n\| \frac{T\dot{x}}{\|T\dot{x}\|}$

FEM with Contacts

Hard Constraints ⊖ Difficult for friction

Soft Constraints ⊖ Drift for static contacts

TRAJECTORY OPTIMIZATION

Transcription Method The way in which we discretize a continuous control problem

Direct Transcription:

$$\min_{x_{1:n} u_{1:n}} L_f(x_n) + \sum_{i=0}^{n-1} L(x_i, u_i) \quad \text{s.t. } \dot{x}_i = f(x_i, u_i)$$

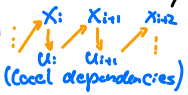
- Many ways of estimating $\dot{x}_i$: $\left(\frac{x_i - x_{i-1}}{h}, \frac{x_{i+1} - x_{i-1}}{2h}\right)$
- Dynamics couple $x_{i-1}, x_i, x_{i+1}, \dots$

Defect: $\dot{x}_i - f(x_i, u_i)$ (residual)

⊕ Better Conditioning esp. for small timesteps

⊕ States as params makes some constraints easy

⊕ Sparsity makes parallelization easy (specific solvers)



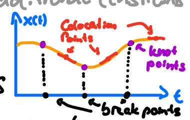
Direct Collocation Represent state / action trajectories as piece-wise polynomials

e.g. $u(t)$: piecewise linear $x(t)$: piecewise cubic

⊕ Defect can be evaluated everywhere

$$\min_{x_{1:n} u_{1:n}} L_f(x_n) + \sum_{i=0}^{n-1} L(x_i, u_i) \quad \text{s.t. } \dot{x}_{i,c} = f(x_{i,c}, u_{i,c})$$ + additional constraints

Defect now needs to be 0 at collocation (approximation) points



⊕ Defect = 0 at break points by construction

⊕ Better accuracy for same # decision vars

For linear / cubic choose collocation points in the middle of $[t_i, t_{i+1}]$

Can compute $x_{i,c}$ as a function of $x_i, f(x_i, u_i), x_{i+1}, f(x_{i+1}, u_{i+1})$

Direct single shooting

Given $x_0, u_0, \dots, u_{n-1}$ $x_1, \dots, x_n$ can be uniquely determined through simulation i.e. $x(u)$

$\min_u L(x, x(u))$ + additional constraints

s.t. $x(u) = \text{forwardSim}(x_0, u)$

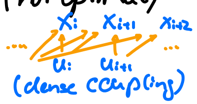
→ Dynamics constraints already in simulation

⊕ fewer decision vars ⊖ Need to compute $\frac{dx}{du}$

⊕ Simpler formulation, no hard constraints

⊕ Results always correct (even if not optimal)

⊕ Scales better to high-dim state spaces



Multiple shooting

Compute short trajectories with direct single shooting. Enforce consistency at trajectory bounds using defect constraints

Derivatives from Sensitivity Analysis

For direct shooting we need to compute:

$\frac{dL}{du} = \frac{\delta L}{\delta x}\frac{\delta x}{\delta u} + \frac{\delta L}{\delta u}$ Define $G(x, u) = 0$ iff. x is consistent with u. $(G(x(u), u) = 0 \ \forall u)$

FE: $G = \begin{bmatrix} x_1 - F(x_0, u_1) \\ x_2 - F(x_1, u_2) \\ \vdots \\ x_n - F(x_{n-1}, u_n) \end{bmatrix} = 0$ BE: $G = \begin{bmatrix} x_1 - F(x_1, u_1) \\ x_2 - F(x_2, u_2) \\ \vdots \\ x_n - F(x_n, u_n) \end{bmatrix} = 0$ F=ma: $G = \begin{bmatrix} M\ddot{x}_1 - F(x_1, u_1) \\ M\ddot{x}_2 - F(x_2, u_2) \\ \vdots \\ M\ddot{x}_n - F(x_n, u_n) \end{bmatrix} = 0$

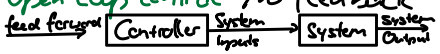
$\Rightarrow \frac{\delta G}{\delta u} = \frac{\delta G}{\delta x}\frac{dx}{du} + \frac{\delta G}{\delta u} = 0 \Rightarrow \frac{dx}{du} = -\left(\frac{\delta G}{\delta x}\right)^{-1}\frac{\delta G}{\delta u}$

sparse 2x diagonal   diagonal

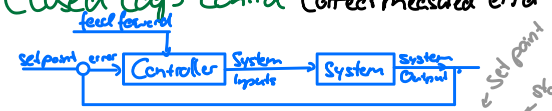
Neural Network Dynamics Use NN to learn the dynamics. This gives easy to compute derivatives

FEEDBACK CONTROL

Open Loop control No feedback



Closed Loop Control Correct measured error



set point ↙ state ↙

Proportional gain: $F_f = F_{max}\, e/\|e\|$ $\quad e := \bar{x} - x$

Bang-Bang Control For systems with bounds on $|u| < \lambda$ the optimal policy is $u(t) = \{-u, u\}$

⇒ Always apply maximal input to reach target as fast as possible

PD - Control $F_f = k_p e + k_d \dot{e}$

k_p: Affects only frequency of an undamped oscillation around the set point (ideal spring)

k_d: Affects damping of the system as $\dot{e} = \dot{\bar{x}} - \dot{x} = \dot{x} = v$ (assuming const $\bar{x}$)

→ penalizes oscillations

PID - Control $F_f = k_p e + k_i \int_0^t e(\tau)d\tau + k_d \dot{e}$

k_i: Integrator gives zero steady state error $\left(\lim_{t \to \infty} e(t) = 0\right)$

⊖ Integral term can give unstable / unsafe responses (windup)

⊕ Simple systems: $k_d = 2\sqrt{k_p}$ (critically damped)

Implicit PD - Control PD controllers act like virtual springs: $\bar{a} = k_p e + k_d \dot{e}$ $= k_p(\bar{x} - x_t) + k_d(\dot{\bar{x}} - \dot{x}_t)$ Implicit Formulation:

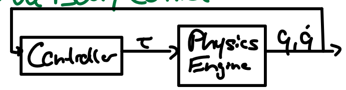
$\bar{a} = k_p(\bar{x} - x_{t+1}) + k_d(\dot{\bar{x}} - \dot{x}_{t+1})$

We know: $\dot{x}_{t+1} = \dot{x}_t + h\bar{a} \quad x_{t+1} = x_t + h\dot{x}_{t+1}$

$$\Rightarrow \bar{a} = \frac{-k_p(x_t + h\dot{x}_t - \bar{x}) - k_d(\dot{x}_t - \dot{\bar{x}})}{1 + h^2 k_p + h k_d}$$

Gaits Each gait has a particular speed for which it is most energy efficient

Whole Body Control



$\bar{a}_{com} = k_p e + k_d \dot{e}$ ← target acceleration for CoM

$\min_{\ddot{q}, u, f} \|\bar{a}_{com} - a_{com}(\ddot{q})\|_2^2$ ← track motion targets

s.t. $M\ddot{q} + C(q, \dot{q}) + J^T f = \begin{bmatrix} 0 \\ u \end{bmatrix}$ ← no control on root DoF

$f \in K \wedge u \in L \wedge J\ddot{q} + \dot{J}\dot{q} = 0$ ← acceleration of ground contact pts = 0

friction cone   possible control inputs

Walking Motions

Hildebrand Gait Diagram

Duty Cycle: Percentage of time a limb is in stance

Walking Gaits > 50% Running Gaits < 50%

Footfall pattern manages role of limbs

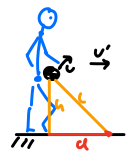
⇒ Adds constraints / objectives

Raibert Heuristic Simple heuristic to choose foot placement positions

Total Energy: $E = \frac{1}{2}mv^2 + mgh$

$= \frac{1}{2}mv'^2 + mgl$

$d = d_f(v_d) + (v - v_d)\sqrt{\frac{h}{g}}$



BASICS

Integration by Parts: $\int_a^b f g' = [fg]_a^b - \int_a^b f' g$

$[x]^T[x] = r^T r I - r r^T$

$\dot{R} = [\omega]R \qquad a \times b = -b \times a$