Last Time:

- Interpreting Least Action
- Manipulator Equation
- Non-Conservative
- Constraints

Today:

- Simulation with Constraints
- Differential Algebraic Equations (DAEs)
- Variational Integrators

Simulations With Constraints:

- Last time we saw how to write dynamics with a constraint $c(q)=0$:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) - \frac{\partial L}{\partial q} - \lambda^T \frac{\partial c}{\partial q} = 0$$

$$M(q)\ddot{q} + C(q,\dot{q}) + G(q) = J(q)^T \lambda$$

- To calculate λ, we diff $c(q)$ twice:

$$c(q)=0 \Rightarrow \dot{c}(q) = \frac{\partial c}{\partial q}\dot{q} = J(q)\dot{q} = 0$$

$$\Rightarrow \ddot{c}(q) = J(q)\ddot{q} + \underbrace{\frac{\partial}{\partial q}\left(J(q)\dot{q}\right)\dot{q}}_{d(q,\dot{q})} = 0$$

- Now we can solve a KKT system ~~for~~ $\ddot{q}$ and λ:

$$\begin{bmatrix} M(q) & J^T(q) \\ J(q) & 0 \end{bmatrix} \begin{bmatrix} \ddot{q} \\ -\lambda \end{bmatrix} = \begin{bmatrix} -C(q,\dot{q}) - G(q) \\ -d(q,\dot{q}) \end{bmatrix}$$

- Solving this linear system directly is generally the way to go numerically.

- Another common strategy: We can solve the first line for $\ddot{q}$ and plug it into the 2nd line:

$$M\ddot{q} - J^T\lambda = -C - G \Rightarrow \ddot{q} = M^{-1}J^T\lambda - M^{-1}(C+G)$$

$$J\ddot{q} = -d \Rightarrow [JM^{-1}J^T]\lambda = JM^{-1}(C+G) - d$$

- Now we have λ and we can plug it back into the dynamics to solve for $\ddot{q}$

- Note that second method is smaller/faster but numerically can have worse conditioning (worse accuracy). Generally solving the KKT system is better.

* Pendulum Simulation

- Naive numerical implementation has constraint drift.

- Note that $c(q)$ does not appear in the dynamics at all (only $\ddot{c}(q)$)

Differential Algebraic Equations (DAEs)

- We really want to solve the system of equations:

$$\dot{x} = f(x)$$
$$0 = c(x)$$

for $x(t)$

- This is called a DAE
- There are several techniques for solving these.

Baumgarte Stabilization:

- Simplest way to solve DAEs
- Transform the DAE into an equivalent ODE
- Modify ODE by adding a PD controller to track the constraint:

$$M(q)\ddot{q} + C(q,\dot{q}) + G(q) = \overbrace{J(q)^T \lambda}^{\text{Constraint force}}$$

constraint Normal vector

constraint force magnitude

- Modify λ with a PD feedback term:

$$M(q)\ddot{q} + C(q,\dot{q}) + G(q) = J(q)^T(\lambda - \alpha c(q) - \beta \dot{c}(q))$$

- Remember $\dot{C}(q) = J(q)\,\dot{q}$

- This works with any standard ODE solver

- PD gains α and β must be tuned based on system, integrator, time steps, etc.

- We can also fit this into the KKT system:

$$\begin{bmatrix} M & J^T \\ J & 0 \end{bmatrix} \begin{bmatrix} \ddot{q} \\ \lambda \end{bmatrix} = \begin{bmatrix} -C - G \\ -d - e \end{bmatrix}$$

$$e = [J M^{-1} J^T](\alpha C(q) + \beta \dot{C}(q, \dot{q}))$$

- Can show that constraints are now Lyapunov stable.

* Pendulum example with stabilization

- We can achieve good constraint satisfaction but never exact

- ODE becomes stiff as PD gains increase $\Rightarrow$ more expensive to solve with tighter constraint tolerances.

Variational Integrators:

- So far we've looked at generic ODE solvers for $\dot{x} = f(x)$, $x \in \mathbb{R}^n$
- Also talked about specialized methods for Lie Groups (RK-MK)
- Now we'll look at specialized methods for Lagrangian Systems.