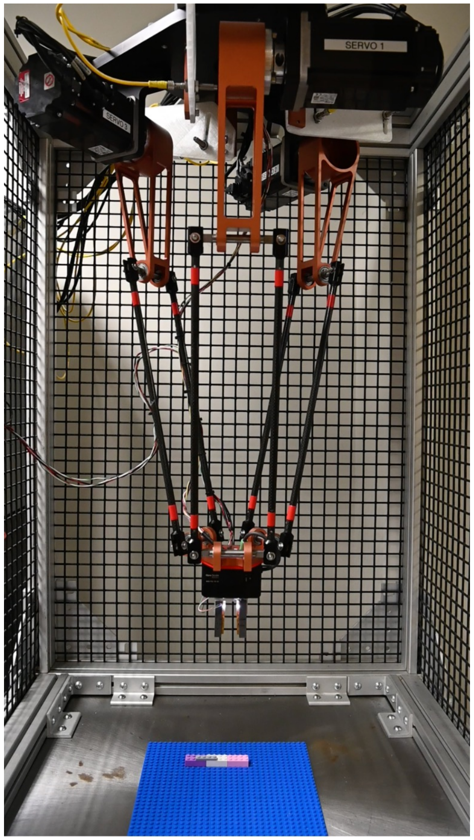


Modelica-Based Control of a Delta Robot

Scott A. Bortoff
Chief Scientist
Multi Physical Systems

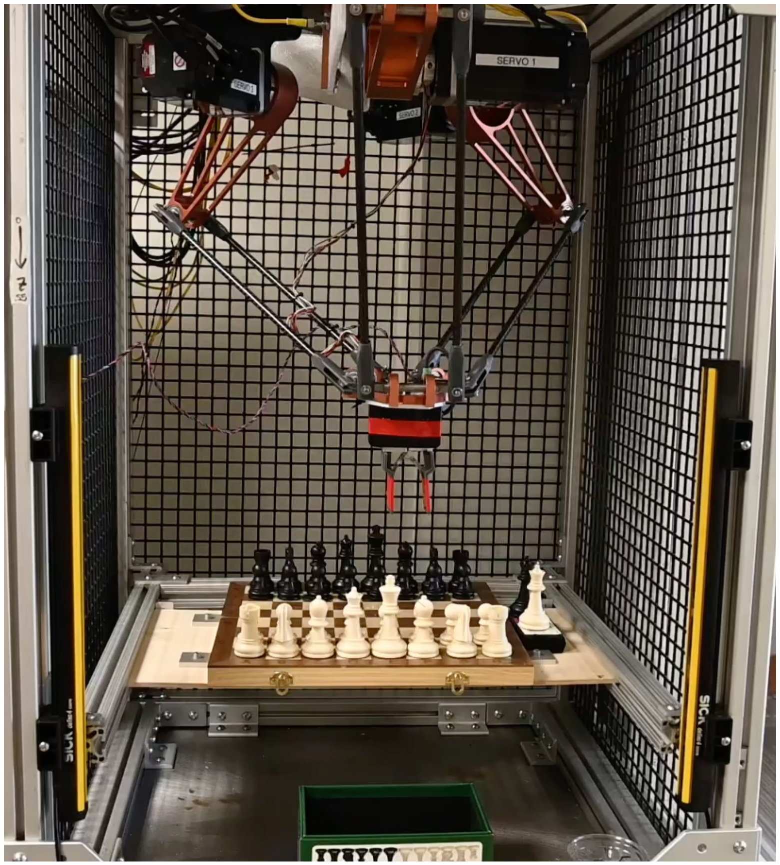
26-October-2022

MITSUBISHI ELECTRIC RESEARCH LABORATORIES (MERL)
Cambridge, Massachusetts, USA
http://www.merl.com

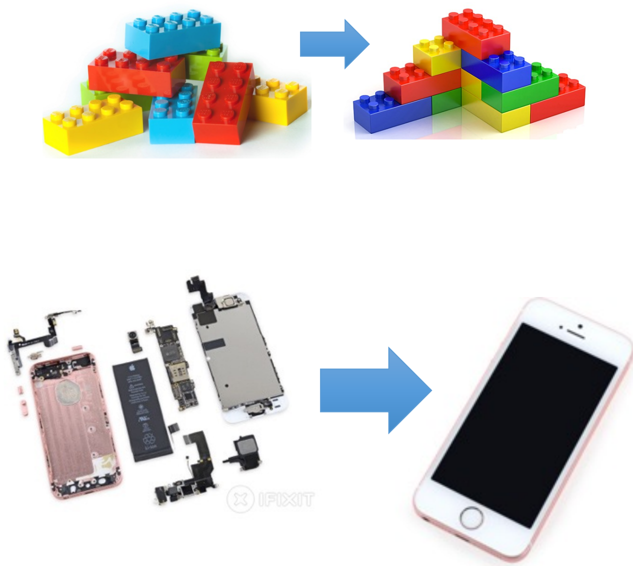
Kamaji
千と千尋の神隠し
Spirited Away
2001

Outline - 2 Related Topics

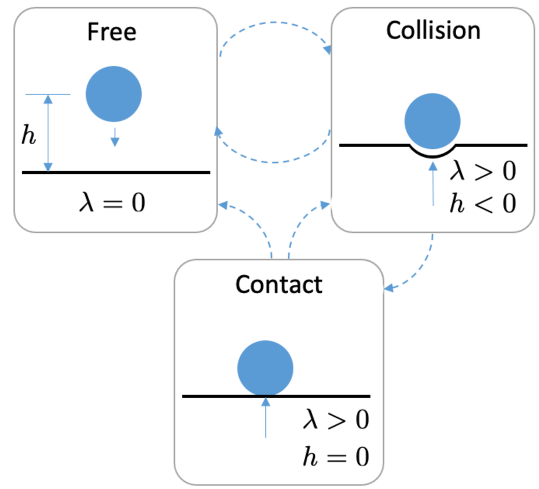
Topic 1:
Modeling & Control of Kamaji



Goal:
Object Manipulation → Assembly



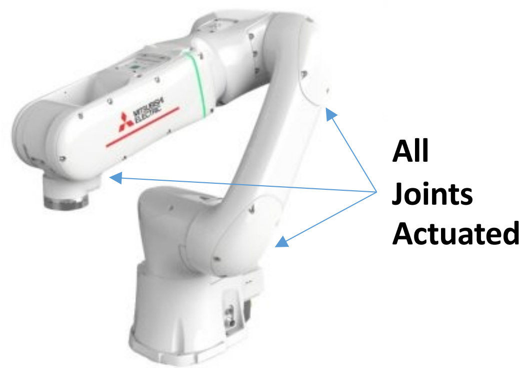
Topic 2:
Modeling & Control of Contact & Collisions



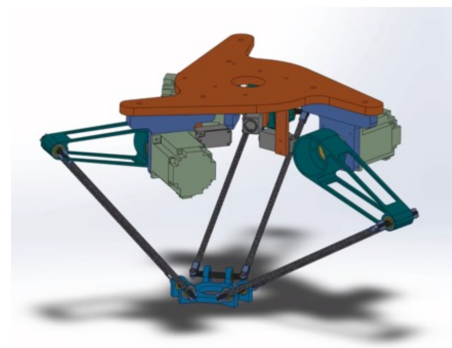
MITSUBISHI ELECTRIC
Changes for the Better

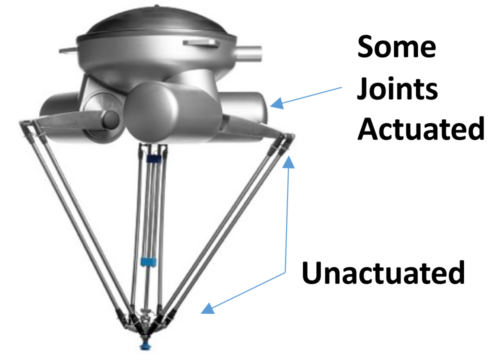
Serial-Link Vs. Delta Robots – Mechanical Design

Serial Link Robotic Manipulators



Closed-Chain Robotic Manipulators





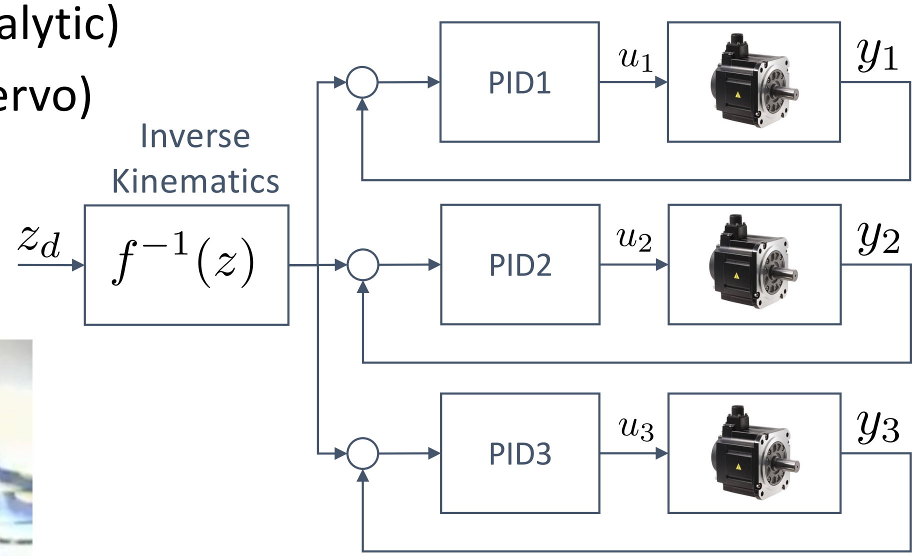
Codian - Delta Robot

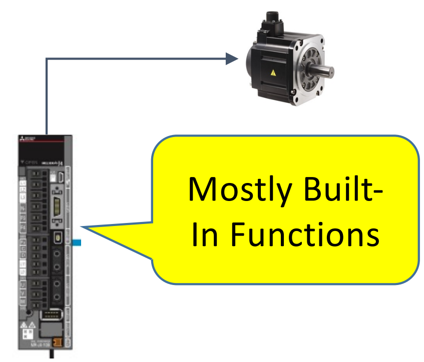
- **Open Chain**
- **High Mass, High Inertia**
- **Geared Joints → High Friction**
- **High Impedance – Stiff**
- **Coupled Force / Torque**
- High Precision Mechanical Position Control
- **High Joint & Link Bending**
- Larger work volume
- Many big parts → More expensive
- **Pick and Place, Assembly Applications**

- **Complex Closed Chain**
- **Low Mass, Low Inertia**
- **Kamaji has Direct Drive → Low Friction**
- **Low Impedance – Soft**
- **Decoupled Force / Torque**
- High Precision Mechanical Position Control
- **Low Joint & Link Bending**
- Smaller work volume
- Fewer, simple parts (symmetry) → Less expensive
- **Pick and Place Applications**

Industrial Delta Robot Used for Pick and Place Operations

- Trajectory computed by inverse kinematics (analytic)
- High-gain PID on each servomotor (built into servo)
- Simple and fast, but…
- Robot is very stiff. No good for contact.

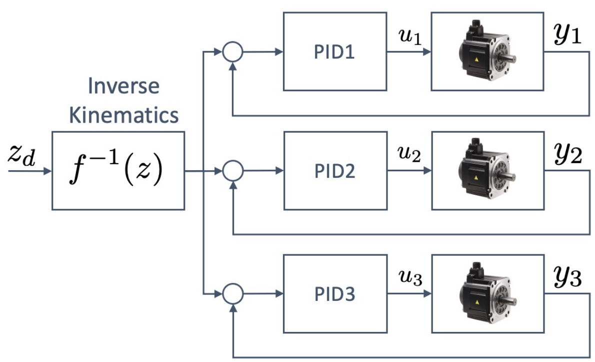




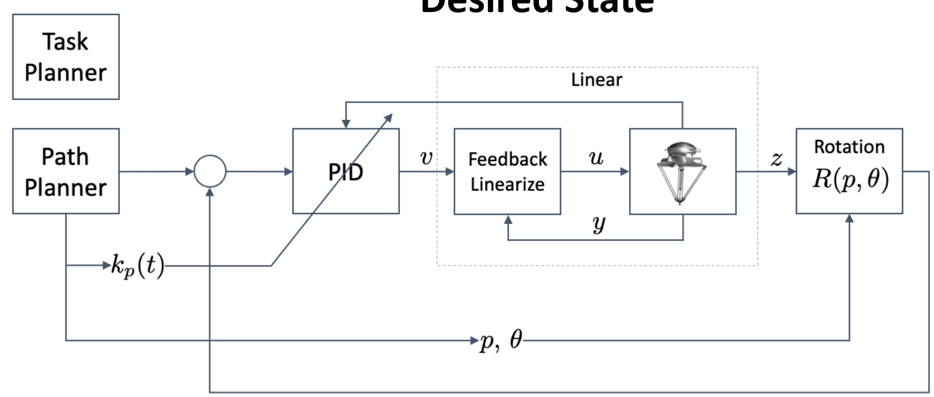
Research Objectives

- Delta Advantages – Simple, Precise, **Low Mass, Fast**, Decoupled.
- Desired Use – Assembly. Collisions and contact. Need nonlinear control & programable impedance (**soft**).
- **How do we control the robot trajectory and its impedance to assemble something, robustly?**
- **How do we avoid switching among hybrid modes of operation (position, velocity, force etc.)?**

State-of-the Art

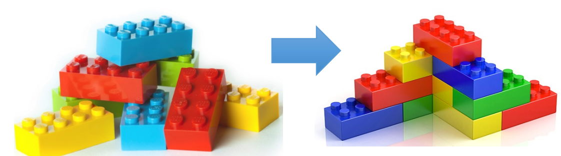


Desired State



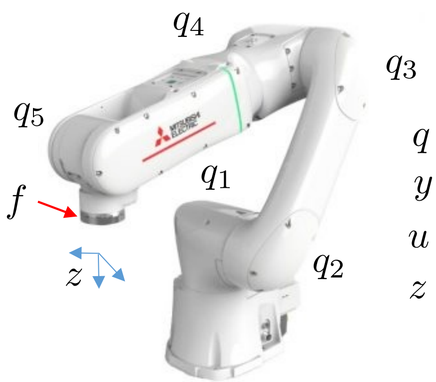
Pick & Place Applications

Assembly Problems



Serial-Link versus Delta Robot – Kinematics, Dynamics

Open Chain – Conventional Calculations



$q \in \mathcal{R}^5$ Joint Angles
$y = q$ Measurements
$u \in \mathcal{R}^5$ Inputs (Co-located)
$z \in \mathcal{R}^3$ End effector

Forward Kinematics – Explicit (analytic, closed-form)

$$z = F(q)$$

Inverse Kinematics – Implicit (no closed-form)

$$q = F^{-1}(z)$$

Jacobian – Explicit (analytic closed form)

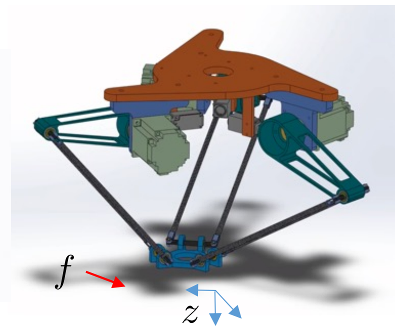
$$J(q) = \frac{\partial F(q)}{\partial q} \Rightarrow \tau = J^T(q) f$$

Dynamics – ODE (all analytic, closed-form)

$$M(q)\dot{v} + C(q, v) + D(v) + G(q) = u + \tau$$

Closed Kinematic Chain → Kinematics, Dynamics

A little more work....



$q \in \mathcal{R}^9$ Joint Angles
$y \in \mathcal{R}^3$ Measurements
$u \in \mathcal{R}^3$ Inputs

Forward Kinematics – Implicit (no closed-form)

$$F(q, z) = 0$$ (6 equations, 6 unknowns, solved numerically)

Inverse Kinematics – Explicit (closed-form)

$$q = F^{-1}(z)$$

TWO Jacobians

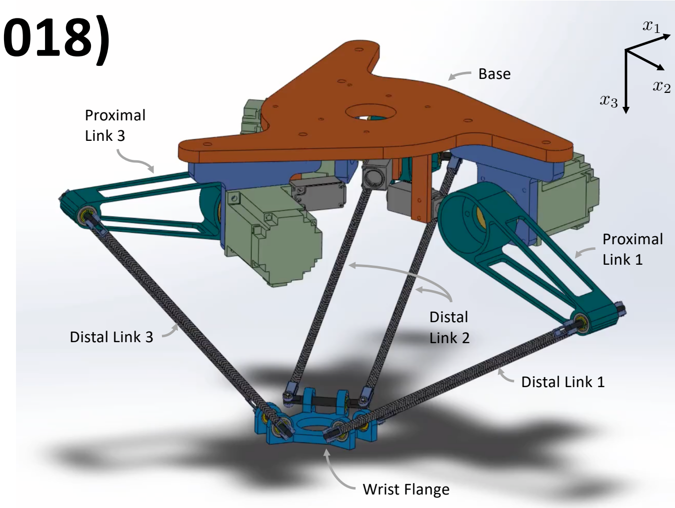
$$J_c = \frac{\partial F(y)}{\partial y}$$ (Implicit. For control)

$$J_v = \frac{\partial F(q)}{\partial q}$$ (Explicit. For simulation)

Dynamics – Explicit if DAE

$$M(q)\ddot{q} + C(q, \dot{q}) + D(\dot{q}) + G(q) = \lambda^T H(q) + Bu$$
$$h(q) = 0$$

Delta Robot Differential-Algebraic Model (2018)



- **Forward Kinematics of each arm…**

$$\psi(q_i) = \begin{bmatrix} l_2 \sin(q_{i2}) \sin(q_{i3}) \\ l_1 \cos(q_{i1}) + l_2 \cos(q_{i2}) \\ l_1 \sin(q_{i1}) + l_2 \sin(q_{i2}) \cos(q_{i3}) \end{bmatrix}$$

Arm 1: $q_1 = [q_{11},\ q_{12},\ q_{13}]^T$
Arm 2: $q_2 = [q_{21},\ q_{22},\ q_{23}]^T$
Arm 3: $q_3 = [q_{31},\ q_{32},\ q_{33}]^T$

- **Constraint at wrist flange (6 equations)…**

$$h(q) = \begin{bmatrix} \psi(q_1) - R_z(2\pi/3) \cdot \psi(q_2) \\ \psi(q_1) - R_z(-2\pi/3) \cdot \psi(q_3) \end{bmatrix} \qquad h(q) : \mathcal{R}^9 \to \mathcal{R}^6$$

- **Dynamics for each arm (conventional)…**

$$\underbrace{m(q_i)}_{\substack{\text{3x3} \\ \text{inertia}}}\dot{v}_i + \underbrace{c(q_i, v_i)}_{\substack{\text{3x3} \\ \text{centripetal} \\ \text{Corriolis}}} + \underbrace{g(q_i)}_{\substack{\text{3x1} \\ \text{gravity}}} = \underbrace{bu_i}_{\substack{\text{Motor} \\ \text{torque}}} \qquad b = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

- **Robot Dynamics…Index 3 DAE**

$$H = \frac{\partial h}{\partial q}$$

$$\begin{aligned} \dot{q} &= v \\ M(q)\dot{v} + C(q,v) + G(q) &= \lambda^T H(q) + Bu \\ \boxed{h(q) = 0} \end{aligned}$$

Replace

- **Robot Dynamics … Index 1 DAE**

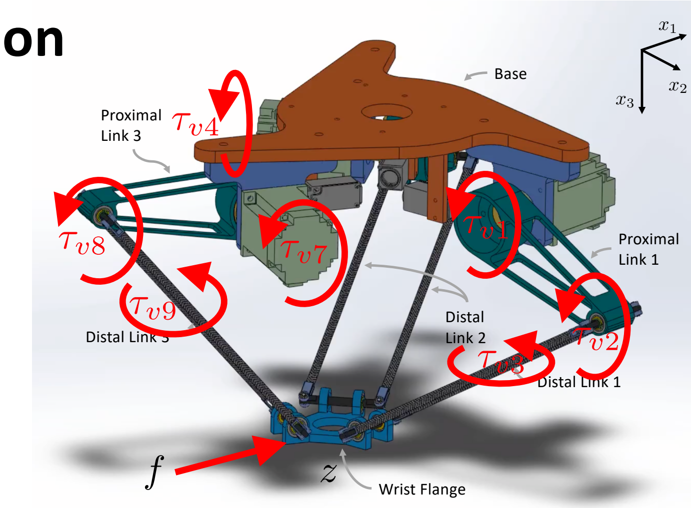
$$\begin{aligned} \dot{q} &= v \\ M(q)\dot{v} + C(q,v) + G(q) &= \lambda^T H(q) + Bu \end{aligned}$$

$$\boxed{h''(q, v, \lambda) + \alpha_1 h'(q, v) + \alpha_0 h(q) = 0}$$

- **24 equations, 24 states** $(q,\ v,\ \lambda)$

$$\begin{aligned} M(q) &= \mathrm{diag}\,(m(q_1), m(q_2), m(q_3)) \in \mathbb{R}^{9\times 9}, \\ C(q,v) &= \mathrm{diag}(c(q_1,v_1), c(q_2,v_2), c(q_3,v_3)) \in \mathbb{R}^9, \\ G(q) &= \mathrm{diag}(g(q_1), g(q_2), g(q_3)) \in \mathbb{R}^9, \\ B &= \mathrm{diag}\,(b, b, b) \in \mathbb{R}^{9\times 3}. \end{aligned}$$

Delta Robot "Virtual" Jacobian for Simulation



- **Robot Dynamics (3) ...**

$$H = \frac{\partial h}{\partial q}$$

$$\dot{q} = v$$

$$M(q)\dot{v} + C(q,v) + D(v) + G(q) = H^T(q)\lambda + B(u + \tau_u) + \tau_v$$

$$\ddot{h}(q,v,\dot{v}) + \alpha_1 \dot{h}(q,v) + \alpha_0 h(q) = 0$$

$$\boxed{\tau_v = J_v^T(q) \cdot f}$$

- **Location of wrist flange as function of q ...**

$$z = \psi(q_1) = \begin{bmatrix} l_2 \sin(q_{12}) \sin(q_{13}) \\ l_0 - l_3 + l_1 \cos(q_{11}) + l_2 \cos(q_{12}) \\ l_1 \sin(q_{11}) + l_2 \sin(q_{12}) \cos(q_{13}), \end{bmatrix}$$

Forward kinematics of Arm 1

$$z = R_2 \psi(q_2)$$

$$z = R_3 \psi(q_3)$$

... and Arm 2 and Arm 3

- **Sum and divide by 3 (not unique)...**

$$z = (\psi(q_1) + R_2 \psi(q_2) + R_3 \psi(q_3))/3$$

- **Compute Jacobian ...**

$$\boxed{J_v(q) = \frac{1}{3}\left[\frac{\partial \psi}{\partial q}(q_1) \quad R_2 \frac{\partial \psi}{\partial q}(q_2) \quad R_3 \frac{\partial \psi}{\partial q}(q_3)\right]}$$

- **Closed-form, analytic formula. Use in simulations with (3)**

Delta Robot "Control" Jacobian

- **Robot Dynamics (DAE)...**

$$H = \frac{\partial h}{\partial q}$$

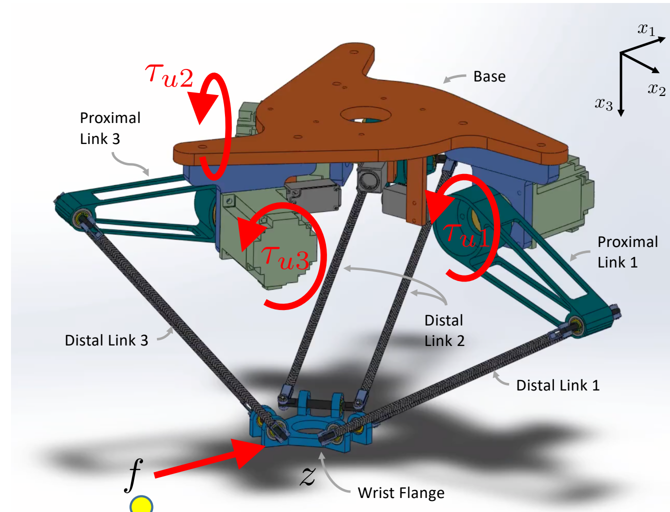
$$\dot{q} = v$$

$$M(q)\dot{v} + C(q,v) + D(v) + G(q) = H^T(q)\lambda + B(u + \tau_u) + \tau_v$$

$$\ddot{h}(q,v,\dot{v}) + \alpha_1 \dot{h}(q,v) + \alpha_0 h(q) = 0$$

Matched with u

$$\boxed{\tau_u = J_c^T(y) \cdot f}$$



- **Constraint at wrist flange (6 equations, 3 measured, 3 unmeasured variables)...**

$$h(q) = \begin{bmatrix} \psi(q_1) - R_2\psi(q_2) \\ \psi(q_1) - R_3\psi(q_3) \end{bmatrix} = 0$$

$y = [q_{11}\ q_{21}\ q_{31}]^T$ ← Measured

$x = [q_{12}\ q_{13}\ q_{22}\ q_{23}\ q_{32}\ q_{33}]^T$ ← Not Measured

- **Forward kinematics computed algorithmically by solving** $h(x,y) = 0$.

$$\frac{\partial h}{\partial x}(x_k, y) \cdot (x_{k+1} - x_k) = -h(x_k, y)$$

Note: $\left(\frac{\partial h}{\partial x}\right)^{-1}$ is computed here.

$J_c^T(y)$ is useful in control algorithms, but should ***not*** be used to *simulate* the effect of f on the robot, because it is computed algorithmically.

- **Implicit function theorem, there exists** $g : \mathbb{R}^3 \to \mathbb{R}^6$ **such that** $h(g(y), y) = 0$
- **Control Jacobian is**

$$\boxed{J_c(y) = \frac{\partial z}{\partial y} = \frac{\partial \Psi}{\partial x} \cdot \frac{\partial g}{\partial y} + \frac{\partial \Psi}{\partial y}}$$

where

$$\begin{cases} z = \Psi(x,y) = \psi(y_1, x_1, x_2) \\ \frac{\partial g}{\partial y} = -\left(\frac{\partial h}{\partial x}\right)^{-1} \cdot \frac{\partial h}{\partial y} \\ h(x,y) = \begin{bmatrix} \psi(y_1, x_1, x_2) - R_2\psi(y_2, x_3, x_4) \\ \psi(y_1, x_1, x_2) - R_3\psi(y_3, x_5, x_6) \end{bmatrix} \end{cases}$$

Computed numerically.

Modelica Realization of the DAE Model

MITSUBISHI ELECTRIC
Changes for the Better

Declare 3 arms

$$\dot{q}_i = v_i$$

$$m(q_i)\dot{v}_i + c(q_i, v_i) + g(q_i) = b\tau$$

Declare Lagrange Multiplier

Declare Constraint

Constant Rotation Matrices

$$\dot{q} = v$$

$$M(q)\dot{v} + C(q,v) + D(v) + G(q) = H^T(q)\lambda + B(u + \tau_u) + \tau_v$$

$$\ddot{h}(q,v,\dot{v}) + \alpha_1 \dot{h}(q,v) + \alpha_0 h(q) = 0$$

Servo Angle Measurements $y, \ \dot{y}$

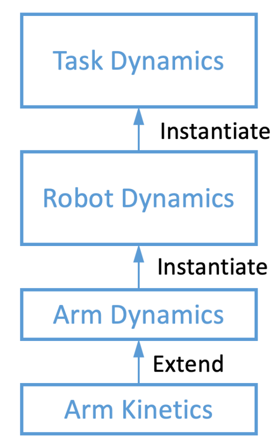
Virtual Jacobian J_v

Wrist Flange location $z, \ \dot{z}$

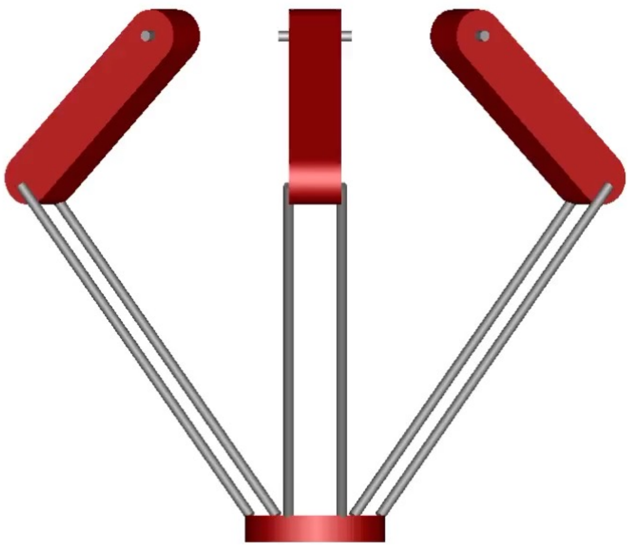
```
model deltaRobotLagrange

 Arms.deltaRobotArmLagrange arm1, arm2, arm3;
 Real lambda[6];      // Lagrange multiplier
 Real h0[6], h1[6], h2[6];
 Input Real u[3], f[3];      // torque inputs, force inputs
 parameter Real POLE = 5.0;

 constant Real Rot2[3,3] = Utilities.RotZ(2.0*PI/3.0);
 constant Real Rot3[3,3] = Utilities.RotZ(-2.0*PI/3.0);
 constant Real B[3] = {1, 0, 0};  // input torque vector

equation

 arm1.tau =  transpose(arm1.dh) * lambda[1:3] + transpose(arm1.dh) * lambda[4:6] + transpose(dz1) * f + B * u[1];
 arm2.tau = -transpose(Rot2 * arm2.dh) * lambda[1:3] + B * u[2] + transpose(dz2) * f;
 arm3.tau = -transpose(Rot3 * arm3.dh) * lambda[4:6] + B * u[3] + transpose(dz3) * f;

 h0 = cat(1,arm1.psi-Rot2*arm2.psi,arm1.ps -Rot3*arm3.psi);
 h1 = der(h0);
 h2 = der(h1);
 zeros(6) = h2 + 2.0 * POLE * h1 + POLEˆ2 * h0;
 // zeros(6) = h0;  // Or this constraint

 y = cat(1, vector(arm1.q[1]), vector(arm2.q[1]), vector(arm3.q[1]));
 yDot = cat(1, vector(arm1.v[1]), vector(arm2.v[1]), vector(arm3.v[1]));

 dz1 = Controllers.Functions.armJacobian(arm1.q) / 3.0;
 dz2 = R2*Controllers.Functions.armJacobian(arm2.q) / 3.0;
 dz3 = R3*Controllers.Functions.armJacobian(arm3.q) / 3.0;
 Jv = cat(2, dz1, dz2, dz3);

 z = arm1.h;
 zDot = Jv * cat(1, vector(arm1.v), vector(arm2.v), vector(arm3.v));

end deltaRobotLagrange;
```

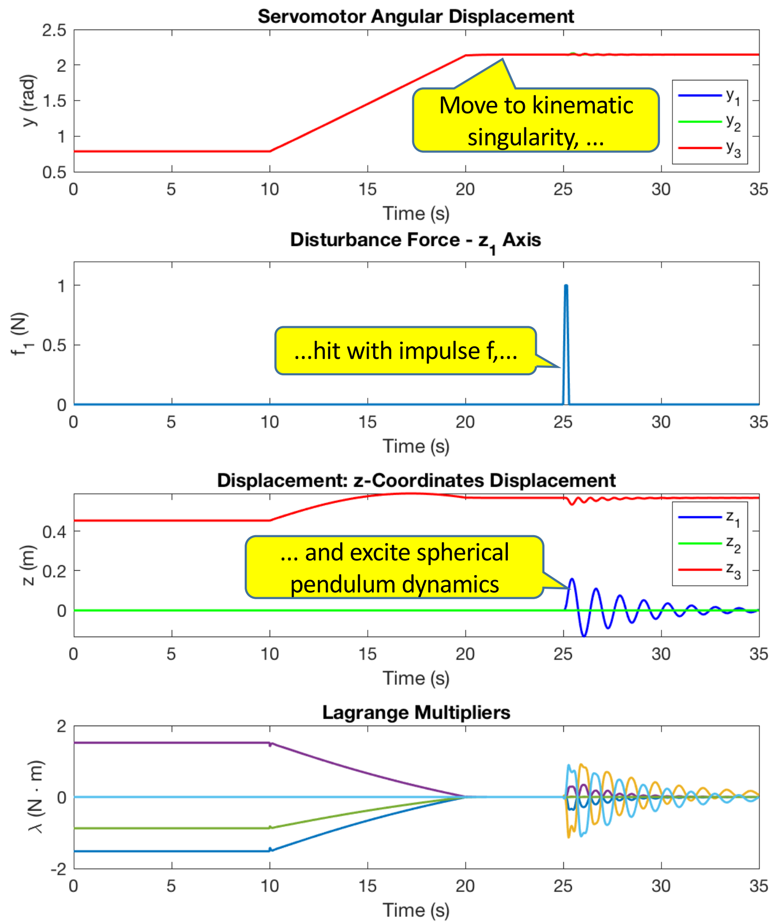
Why Modelica?

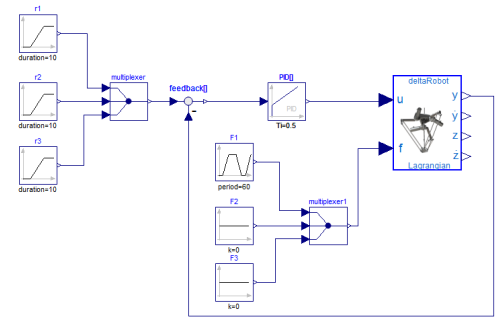
- Hybrid DAE Model of Computation
- Object-Oriented for organization
- Multi-Physical: Mechanics + Control



Simulation Thorough Kinematic Singularity







Inner Loop Feedback Linearization

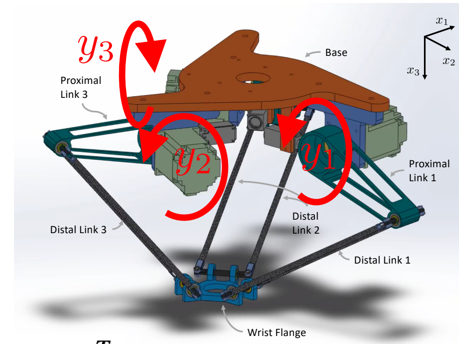
1. DAE Model...

$$\dot{q} = v$$

$$M(q)\dot{v} + C(q,v) + D(v) + \boxed{G(q) = H^T(q)\lambda + B(u+\tau_u)} + \tau_v$$

$$\ddot{h}(q,v,\dot{v}) + \alpha_1 \dot{h}(q,v) + \alpha_0 h(q) = 0$$

$$B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$



2. Apply gravity-cancelling feedback by solving 9x9 system...

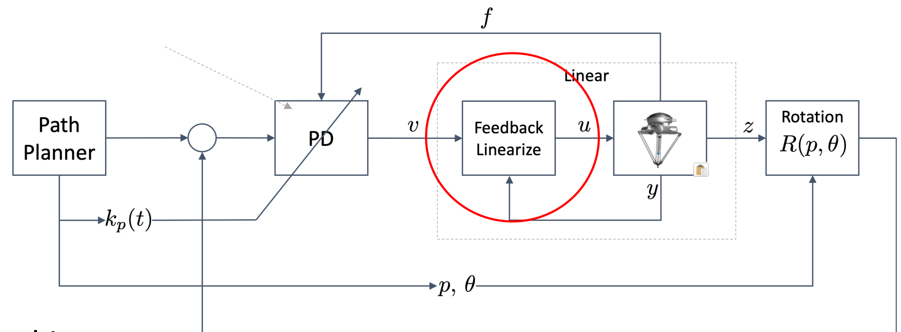
$$\begin{bmatrix} \tau_u \\ \lambda \end{bmatrix} \cdot \begin{bmatrix} B & H^T(q) \end{bmatrix} = G(q) \Rightarrow M(q)\dot{v} + C(q,v) + D(v) = Bu$$

$y = [q_{11}\ q_{21}\ q_{31}]^T$ (Servo angles - measured)

$x = [q_{12}\ q_{13}\ q_{22}\ q_{23}\ q_{32}\ q_{33}]^T$ (Not measured)

3. Reorder into measured (y) and unmeasured (x)...

$$\begin{bmatrix} \bar{M}_{11}(q) & \bar{M}_{12}(q) \\ \bar{M}_{21}(q) & \bar{M}_{22}(q) \end{bmatrix} \begin{bmatrix} \ddot{y} \\ \ddot{x} \end{bmatrix} + \begin{bmatrix} \bar{C}_1(q,v) \\ \bar{C}_2(q,v) \end{bmatrix} + \begin{bmatrix} \bar{D}_1(v) \\ \bar{D}_2(v) \end{bmatrix} = \begin{bmatrix} u \\ 0 \end{bmatrix}$$



4. Differentiate constraint twice, solve for $\ddot{x}$...

$$\frac{\partial h}{\partial y}\ddot{y} + \frac{\partial h}{\partial x}\ddot{x} + \dot{y}^T \frac{\partial^2 h}{\partial y^2}\dot{y} + \dot{x}^T \frac{\partial^2 h}{\partial x^2}\dot{x} = 0$$

5. Ignore higher-order terms $\bar{C}(q,v)$

$$\bar{M}_y \cdot \ddot{y} = u \quad \text{(3 x 3)}$$

where $\bar{M}_y = \bar{M}_{11} - \bar{M}_{12} \cdot \frac{\partial h}{\partial x}^{-1} \cdot \frac{\partial h}{\partial y}$

- Then control is...

$$\boxed{u = \bar{M}_y J_c^{-1} \left(k_i \int_0^t (r-z)d\tau + k_p(r-z) + k_d(\dot{r}-\dot{z}) + \ddot{r} \right)}$$

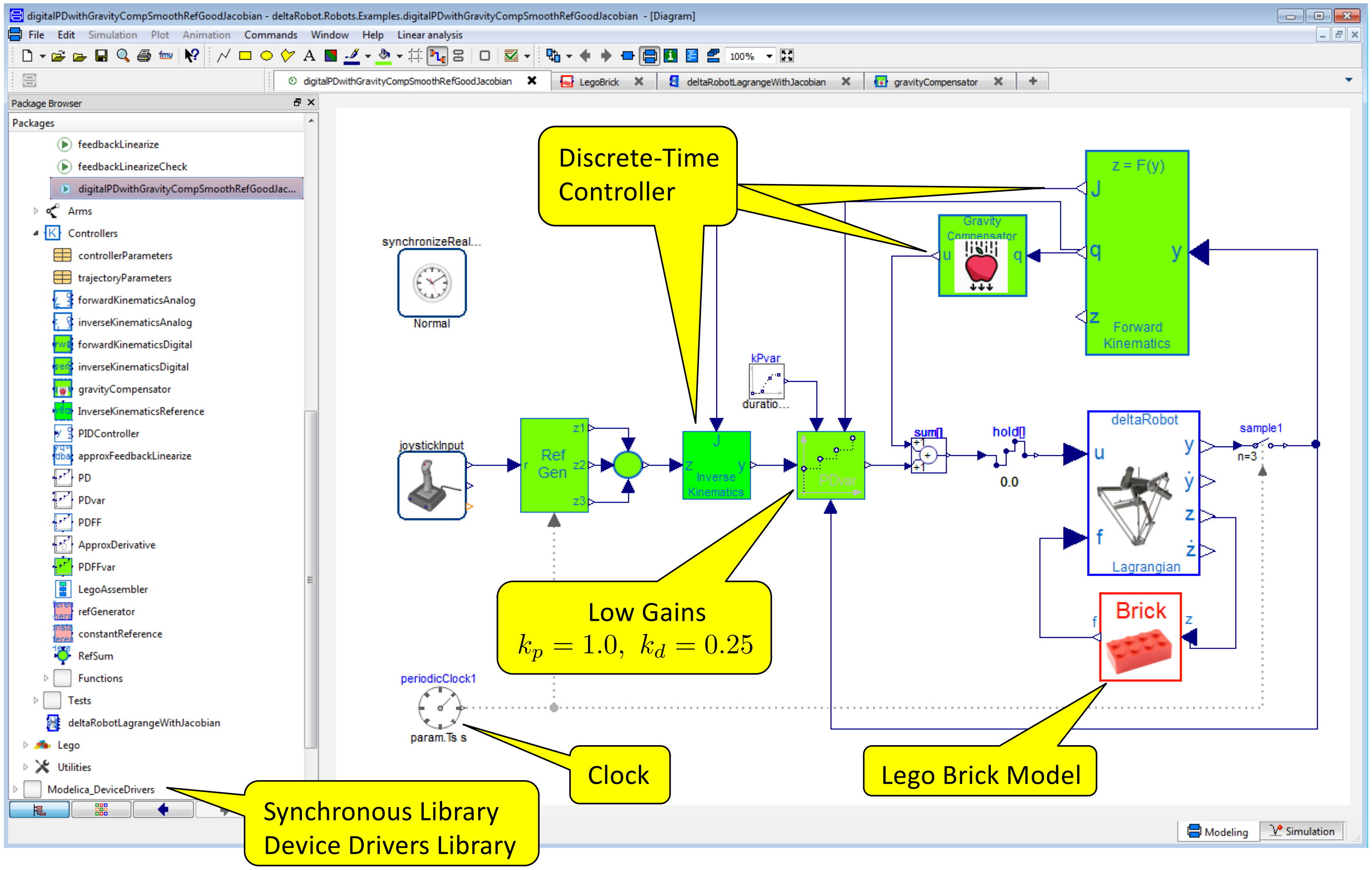
- In z coordinates

$$k_i \int_0^t (r-z)d\tau + k_p(r-z) + k_d(\dot{r}-\dot{z}) = 0$$

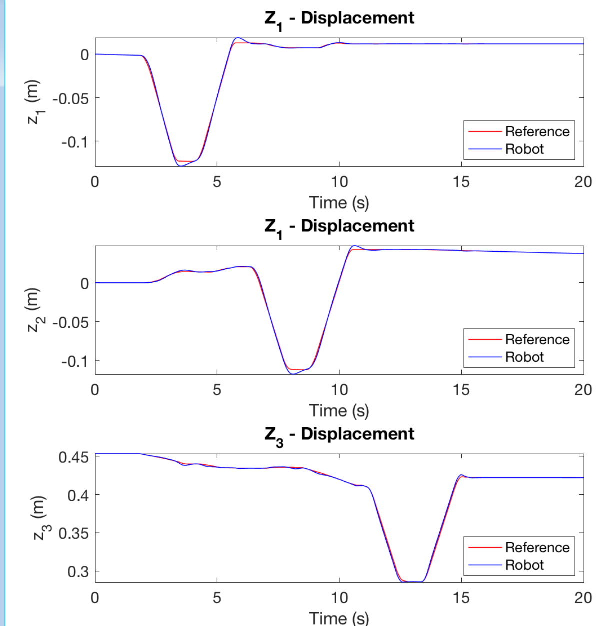
PID outer loop with reference feed forward, adjustable gains

MITSUBISHI ELECTRIC
Changes for the Better

Modelica for Real-Time Desktop Simulation

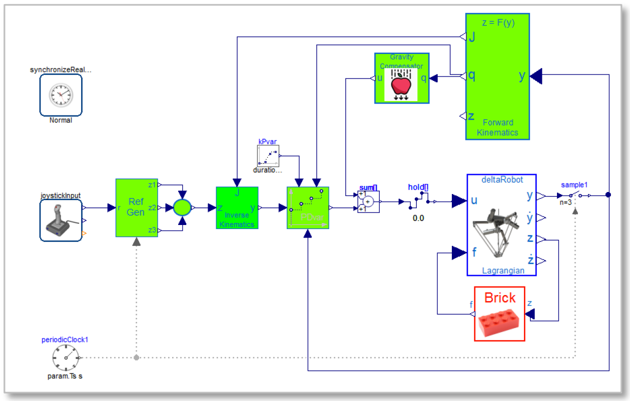


Wrist Flange Location: Simulation

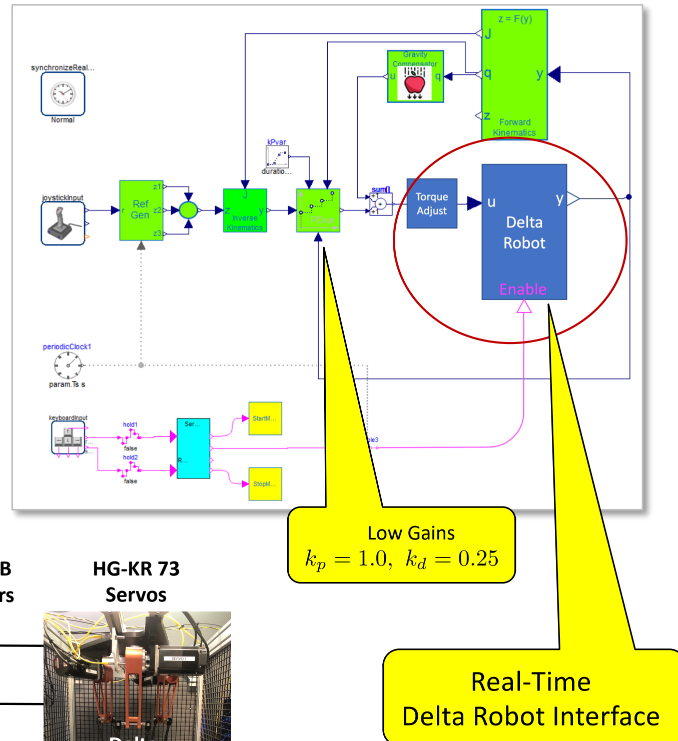


Modelica for Experimental Testing

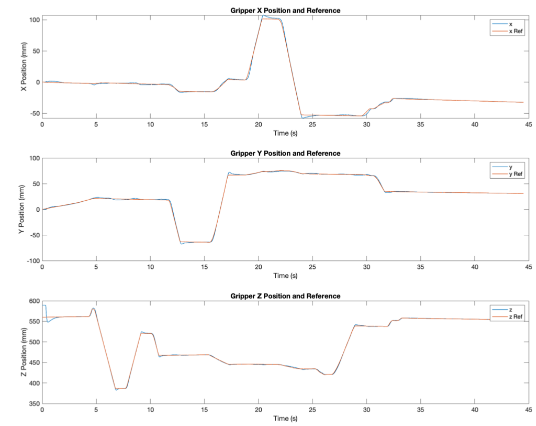
Simulation



Experiment



Wrist Flange Location: Experiment



MERL Delta Robot Modelica Robot Library

MERL Delta Robot Modelica Control Library

Modelica Synchronous Library

MERL Delta Robot Real-Time Modelica Library

Modelica Device Drivers Library

Modelica Standard Library

MR-MC 210 C Language API

Windows 10 OS

OpenModelica

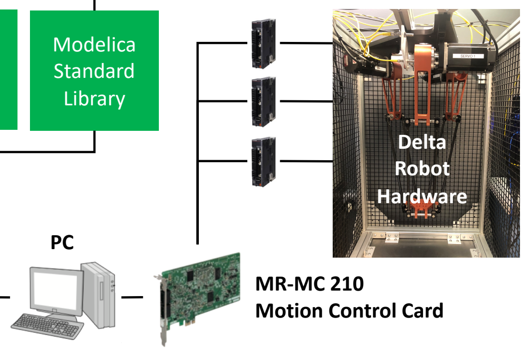
Dymola

- Developed at MERL
- Open Source
- Vendor-Supplied

MR-J4-B Inverters

HG-KR 73 Servos

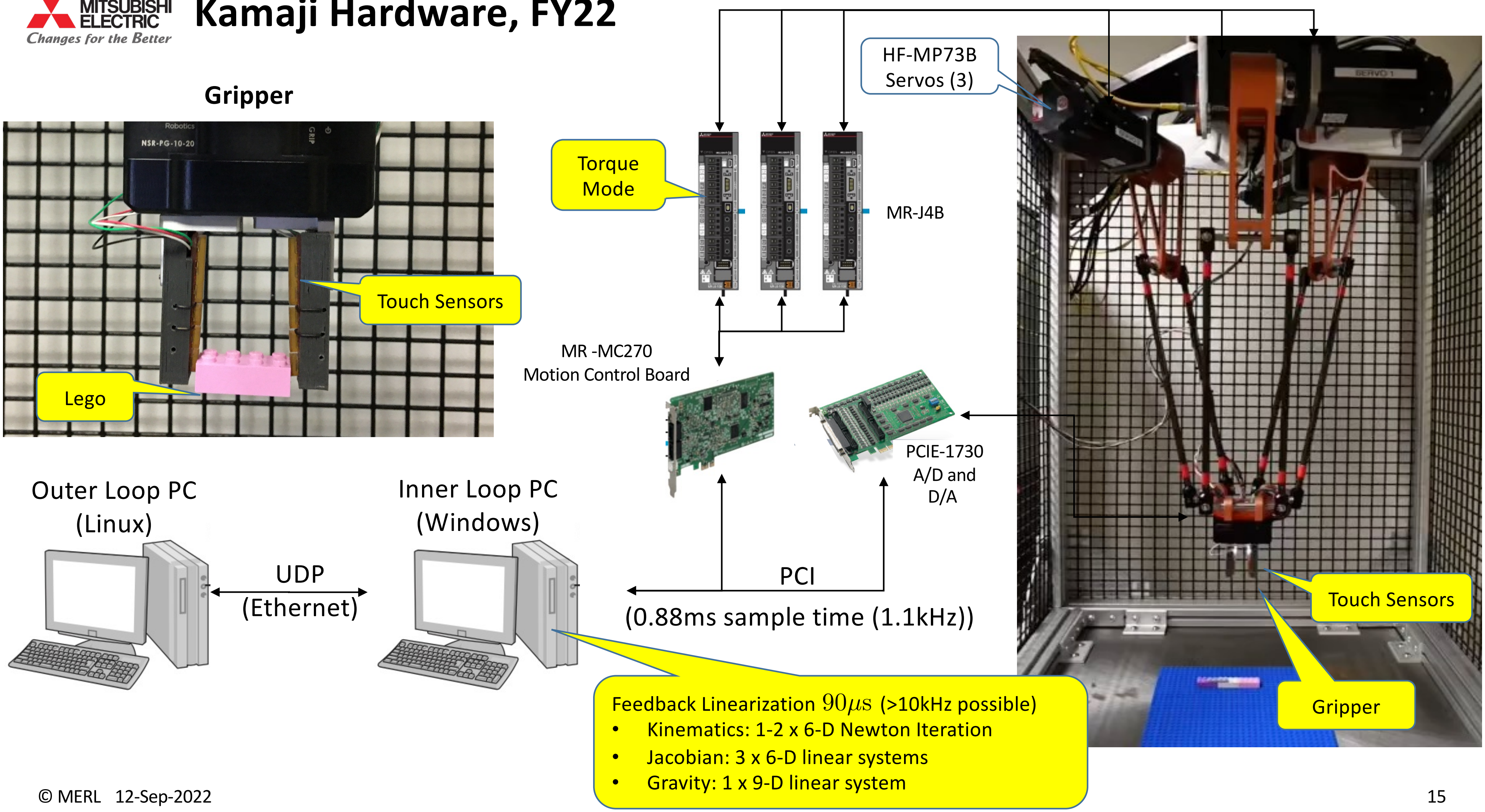
PC



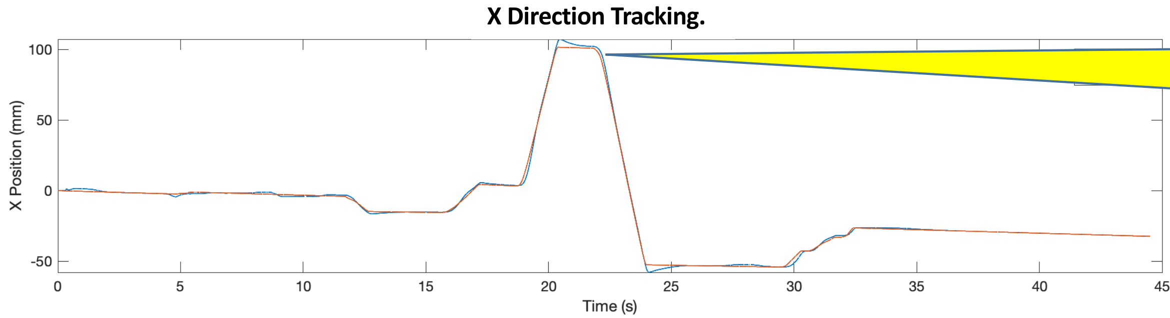
MR-MC 210 Motion Control Card

Ahmed Okasha

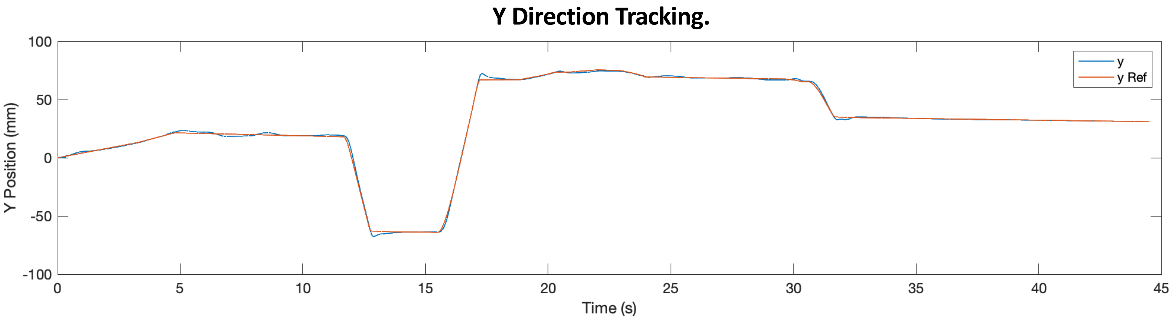
Kamaji Hardware, FY22



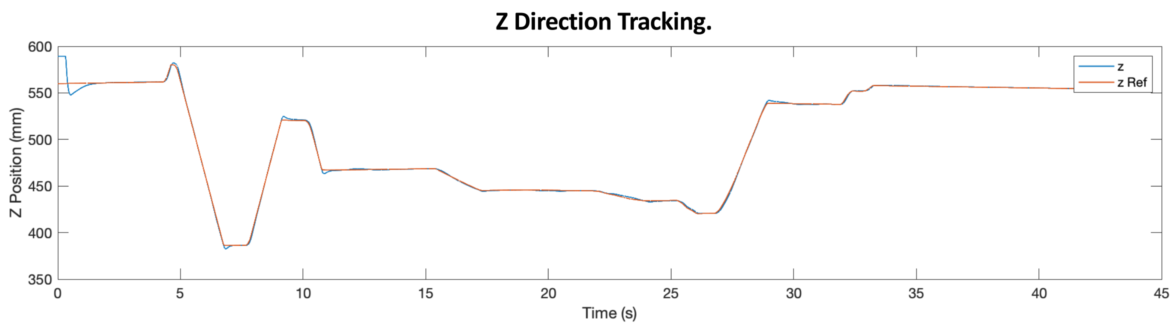
Trajectory Tracking Experiment with Joystick Reference



Large tracking error during motion transient because reference acceleration is not fed forward



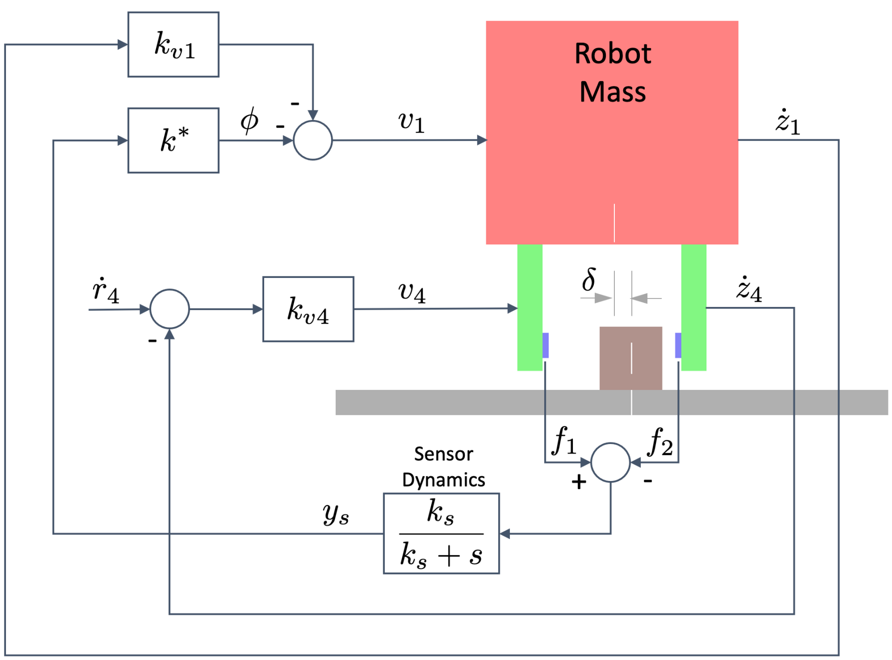
Steady-state tracking error <30μm

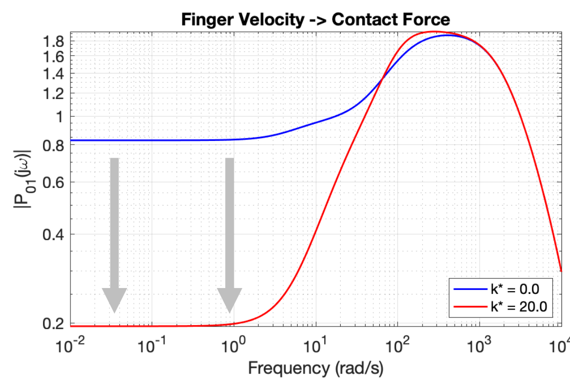


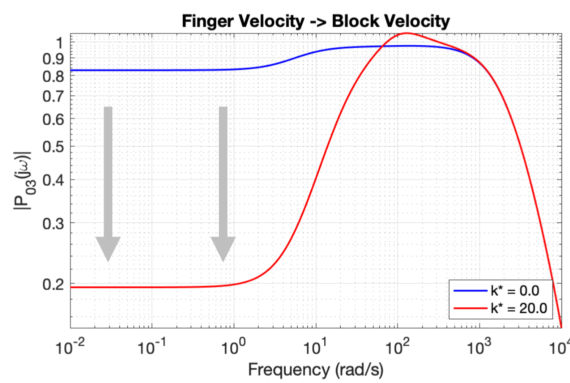
Outer Loop Gains…
Poles placed at s=-p=-7 (soft, low gain)
kI = p^3 = 343
kP = 3*p^2 = 147
kD = 3*p = 21
MAX Integral = 1.0 (see anti-windup PID)
Stability maintained when integrator saturates

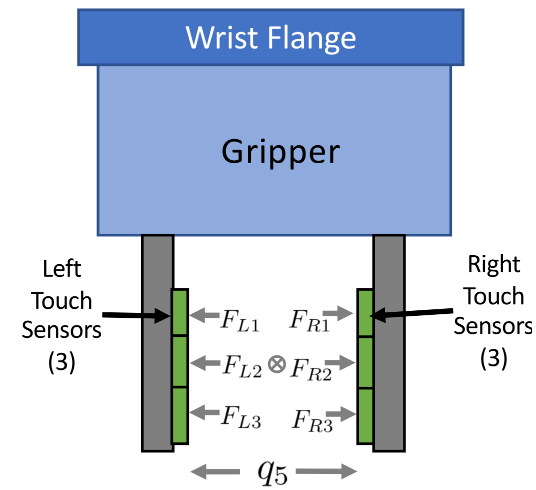
Soft-Contact Feedback Control with Tactile Sensor Feedback

- Feedback loop from difference of sensors to robot horizontal position
- Reduces robot impedance in frequency range
- No switching required in grasp

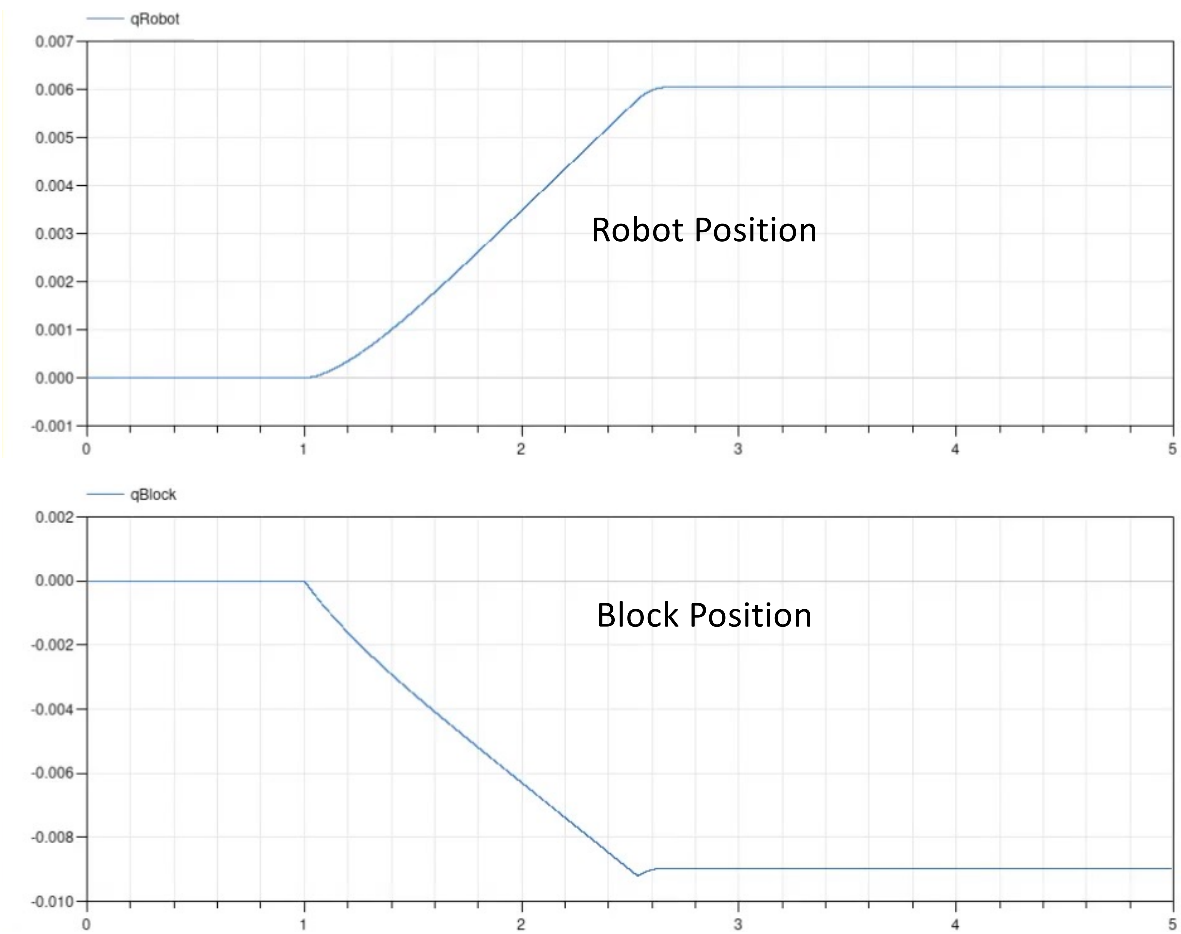


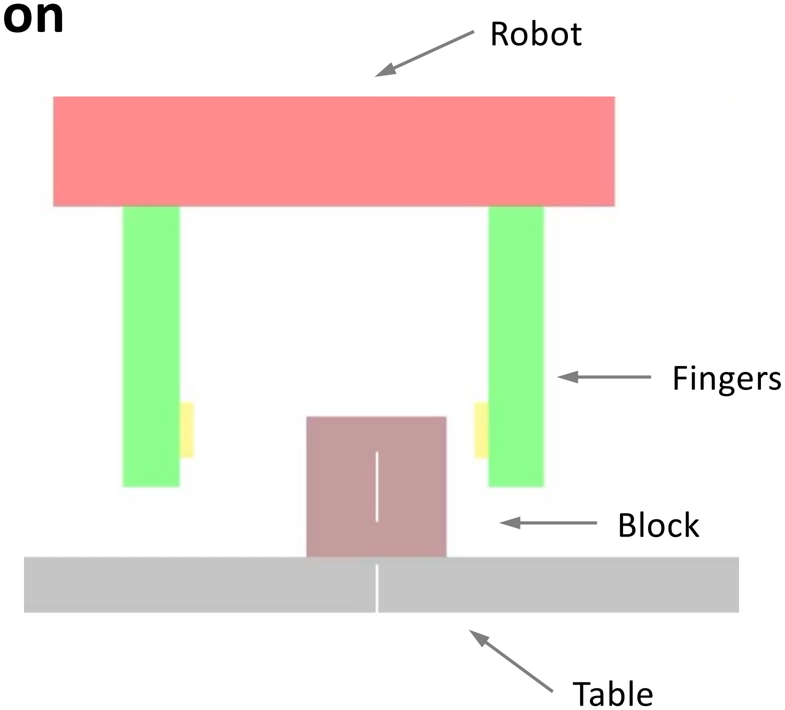




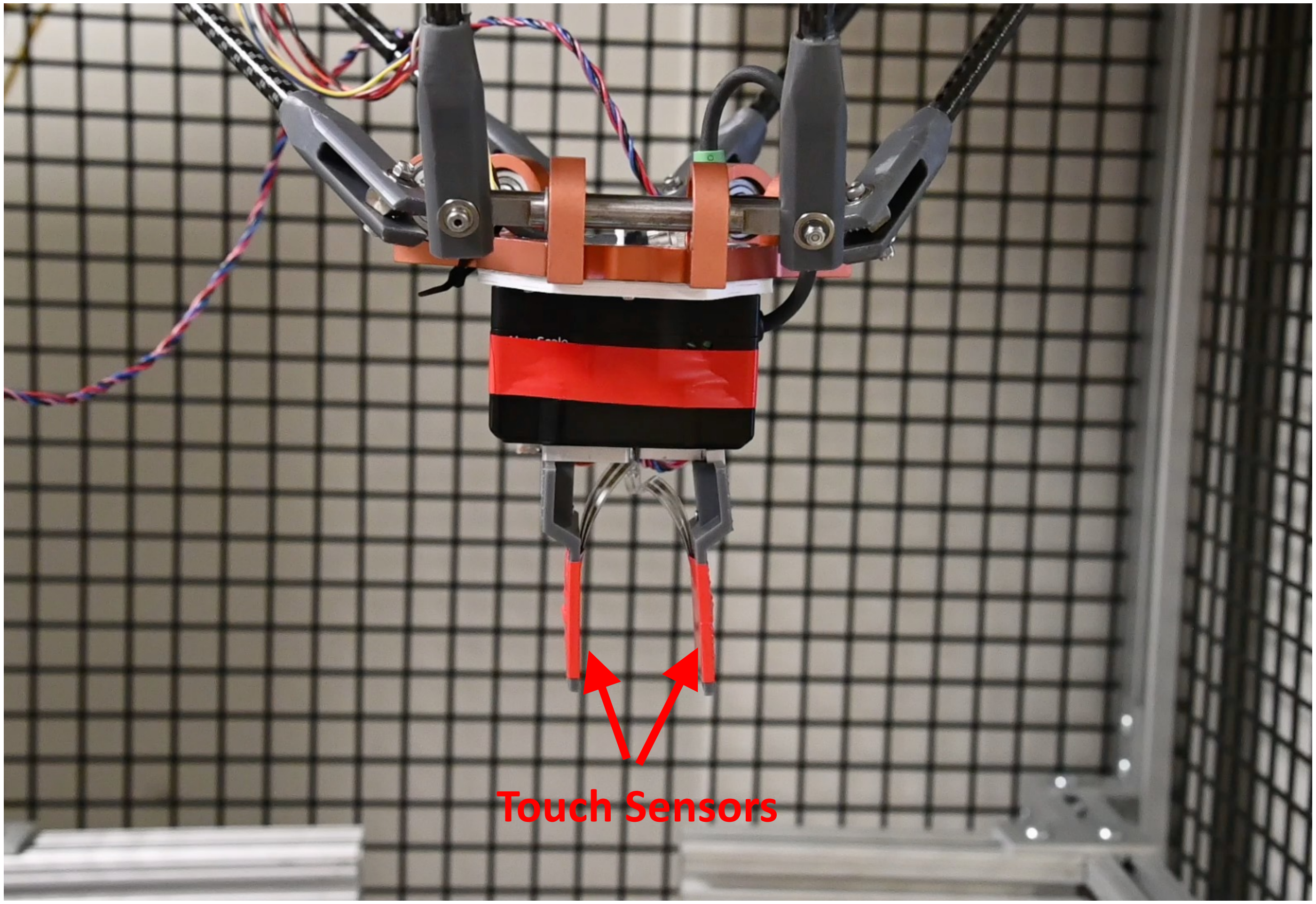


Soft-Contact Feedback Control Simulation





Soft-Contact Feedback Control Experiment

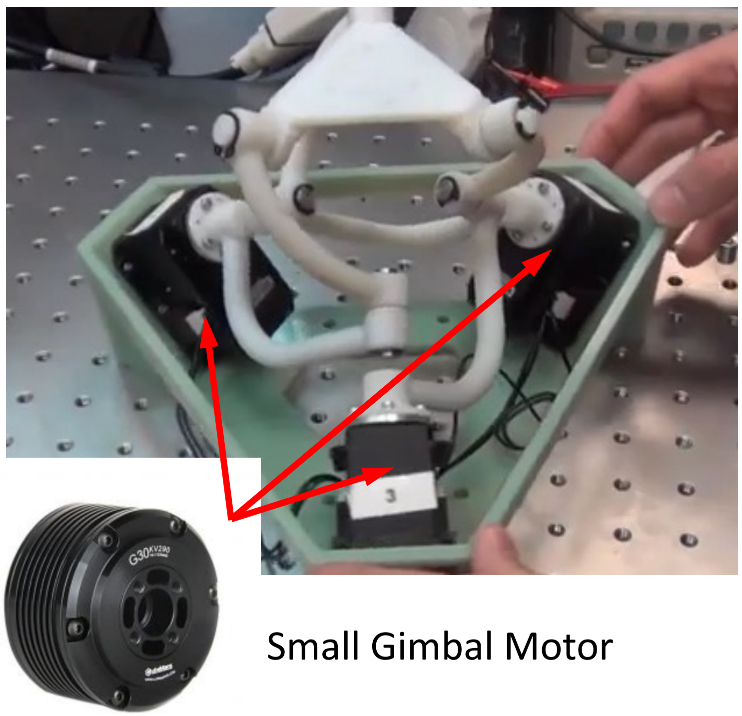


Touch Sensor

Next Steps: Add Spherical Wrist, Study Assembly

New 3 DOF spherical wrist

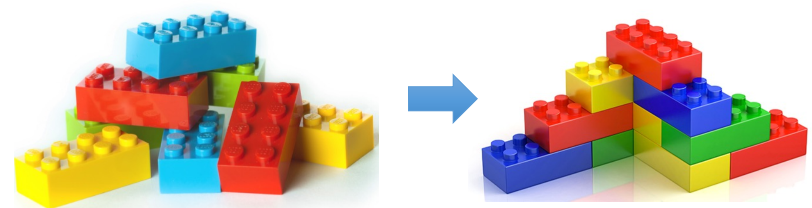
Direct Drive, low impedance, low mass/inertia



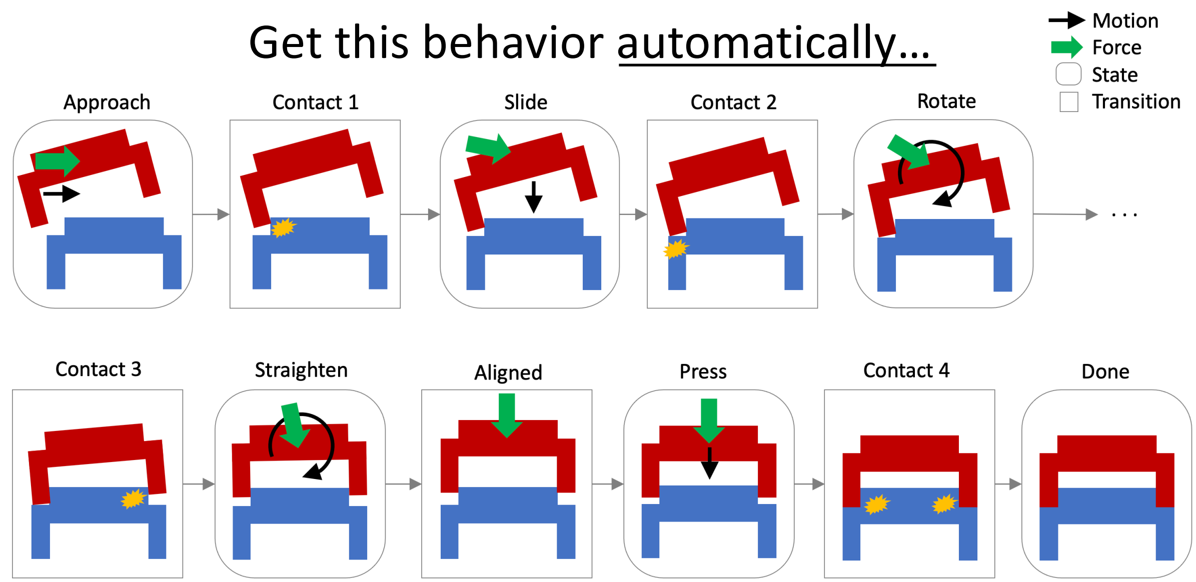
Small Gimbal Motor

How to solve Assembly Problem

Using Impedance Control and MPC



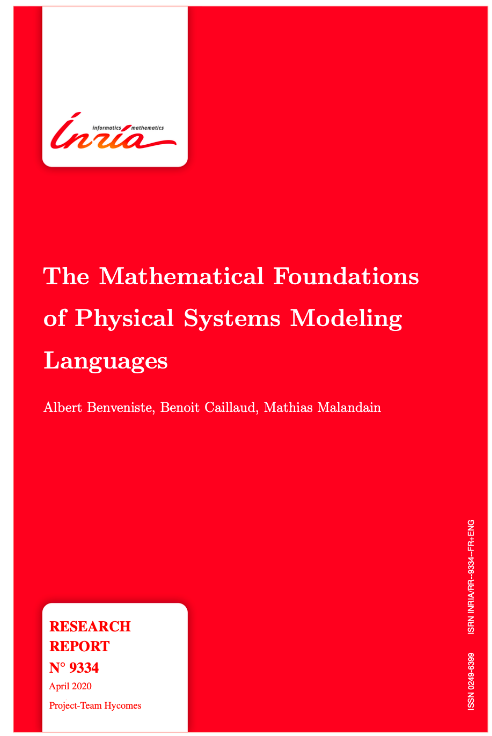
Get this behavior <u>automatically…</u>



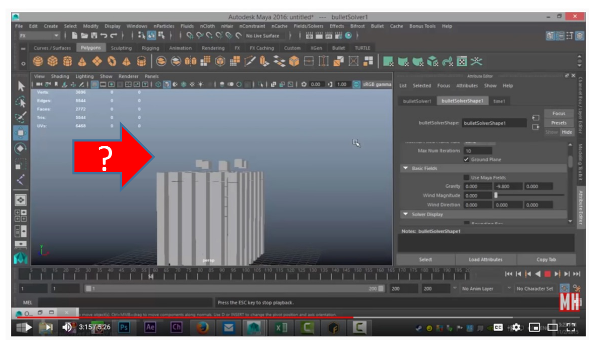
Collisions, Contact & Hybrid (MultiMode) DAEs: Lots of Previous Work

- Physics-Based Animation: Bullet, PhyX, Gazebo (with different engines such as DART), etc.
- State-of-the-art: Represent as Nonlinear or Linear Complementary Problem (LCP), solve
  - Solve a QP problem at each discrete time step
  - Fixed time step, usually first-order (Euler) symplectic (to approx. conserve energy) integrator
  - Simulation is the purpose of the model
- Limitations...
  - Discrete time – mixes modes of computation.
  - LCP formulation is static – are dynamics correctly captured during collisions?
  - Explicit integration
- Multi-Mode DAE (variable # states & structure) formulations
  - Mathematically rigorous structural analysis
  - Algorithms for index reduction
  - Emerging tools (compilers & code generation)

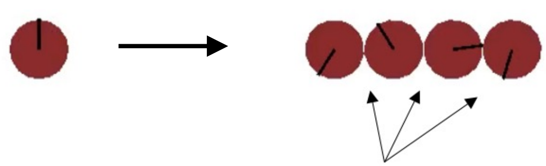
Multi-Mode DAEs



Bullet Simulation



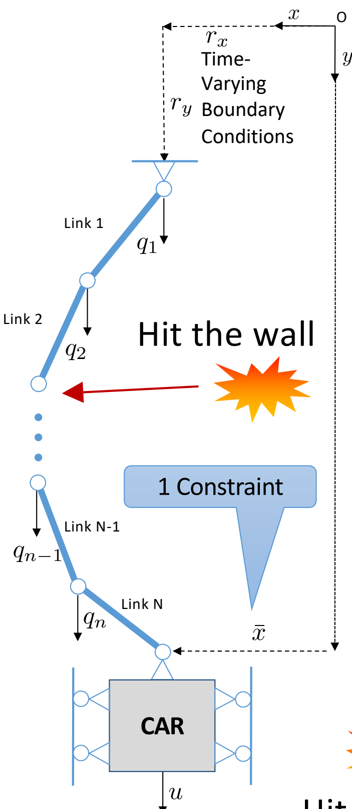
Newton's Cradle – Difficult for LCP



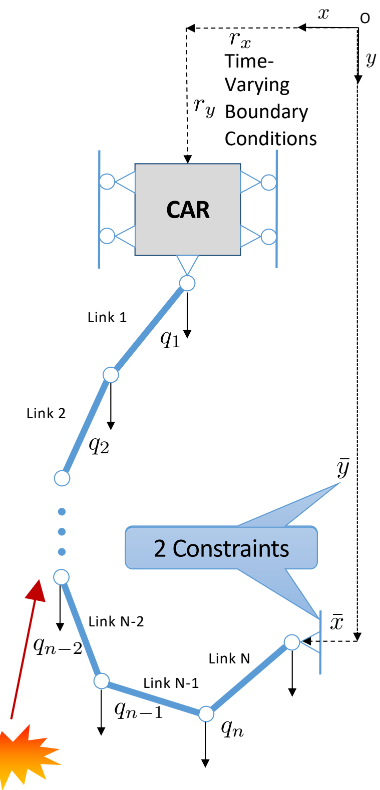
Touching, but no force

Many Examples of Contact and Collision

Rope Sway



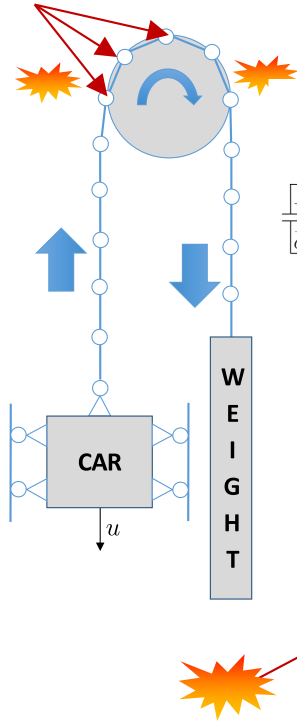
Cable Sway



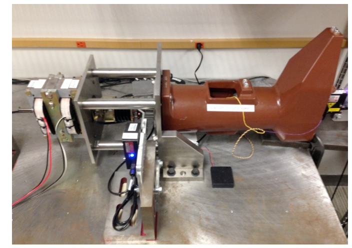
Hit the wall

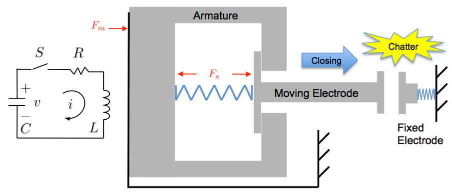
Car Motion

Contact

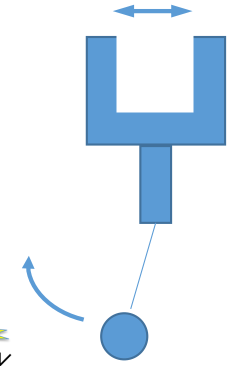


Circuit Breaker





Ball & Cup

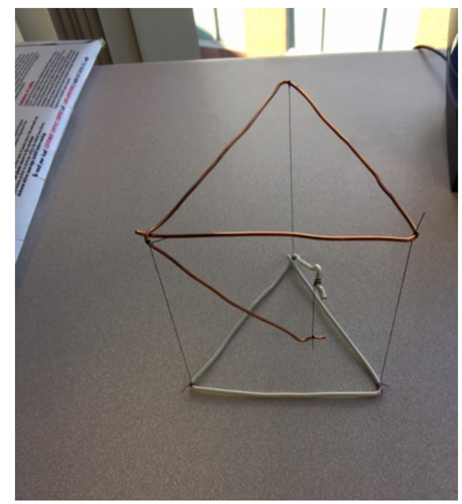


Ball Maze

Chain Fountain

Lift off pile

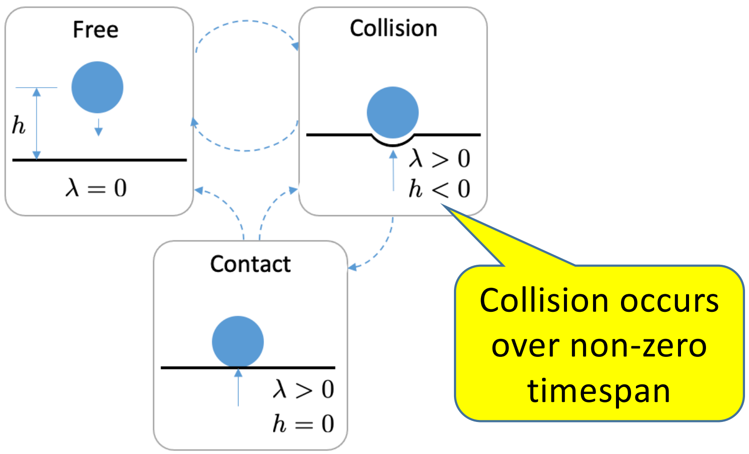
Alan's Paperclip Art Toy



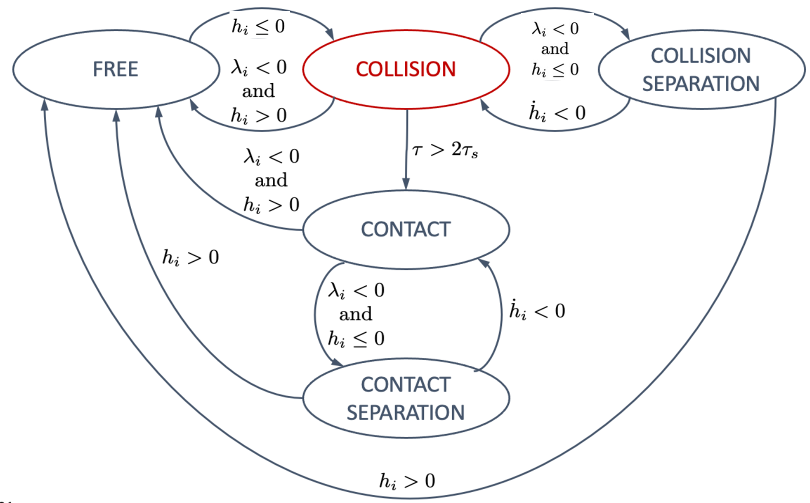
Hybrid (“Multi mode”) DAE Model of Collision & Contact

- Solves LCP* problem as a DAE – guarantees existence of solution, allows for use multi-step solver
- Mathematical model useful for simulation, synthesis of controller (dynamic optimization)

Basic Idea



Finite State Machine



Differential Algebraic Equations (DAE)

$$\dot{q} = v$$

$$M(q)\dot{v} + C(q,v)v + d(q)v + g(q) = Bu + \sum_{i=1}^{n-1} H_i^T(q)\lambda_i$$

$$\ddot{h}_i(q) + \alpha_1 \dot{h}_i(q) + \alpha_0 h_i(q) = 0 \qquad 1 \le i \le n-1$$

$$H_i^T(q) = \frac{\partial h_i(q)}{\partial q}$$

Represents elastic impact

Example – Difficult for LCP Formulation

Touching, but no force

*LCP = Linear Complementary Problem

Details of DAE Model of Contact / Collision

- Robot + Task represented as Lagrangian system + constraint functions

Lagrangian Dynamics $\dot{q} = v$

$$M(q)\dot{v} + C(q,v)v + d(q)v + g(q) = Bu$$

Constraint Functions

$$h_i(q) \qquad i = 1, 2, \ldots, N$$

- If n-1 constraints are active, and NO constraints are in the COLLISION state, use...

"SLOW" System

$$\dot{q} = v$$

$$M(q)\dot{v} + C(q,v)v + d(q)v + g(q) = Bu + \sum_{i=1}^{n-1} H_i^T(q)\lambda_i \qquad H_i^T(q) = \frac{\partial h_i(q)}{\partial q}$$

$$\ddot{h}_i(q) + \alpha_1 \dot{h}_i(q) + \alpha_0 h_i(q) = 0 \qquad 1 \leq i \leq n-1$$

Constrained Lagrangian with Stabilized Constraint

- If n-1 constraints are active, and constraint n is in the COLLISION state, use...

Fast Time Scale of Singularly Perturbed System

Constant

Constant

Constant

"FAST" System

$$\dot{q} = v$$

$$M(\bar{q})\dot{v} + C(\bar{q},v)v + d(\bar{q})v + g(\bar{q}) = Bu + \sum_{i=1}^{n-1} H_i^T(\bar{q})\lambda_i + H_n^T(\bar{q})\lambda_n \qquad H_i^T(\bar{q}) = \frac{\partial h_i(q)}{\partial q}\Big|_{q=\bar{q}} \qquad \bar{q} = q(\bar{t}^-)$$

$$\ddot{h}_i(q) + \alpha_1 \dot{h}_i(q) + \alpha_0 h_i(q) = 0 \qquad 1 \leq i \leq n-1$$

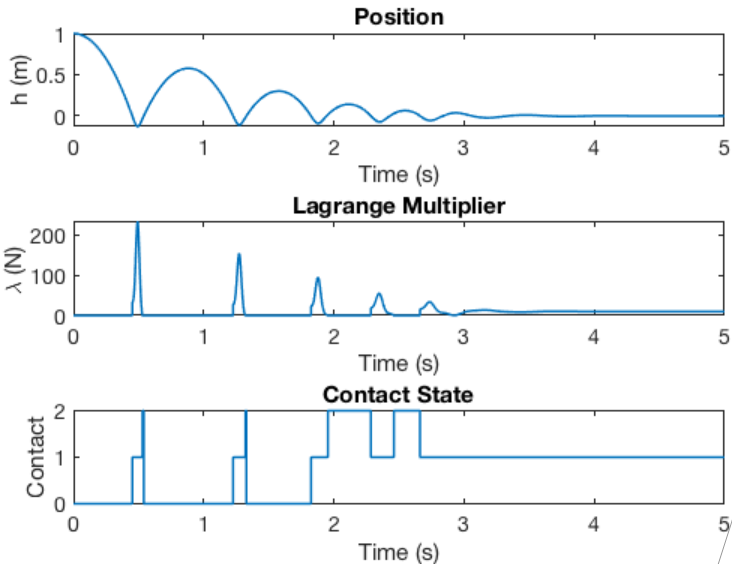
Soft spring

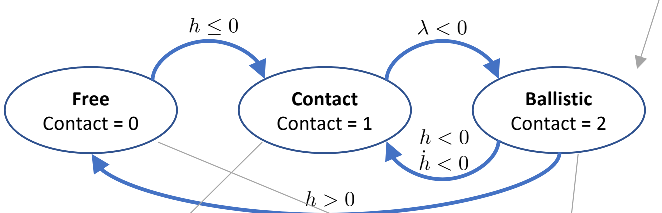
$$\ddot{h}_n(q) + \beta_1 \dot{h}_n(q) + \beta_0 h_n(q) = 0 \qquad \alpha_1 << \beta_1 \qquad \alpha_0 << \beta_0$$

Stiff spring

Time of collision

Bouncing Ball Example

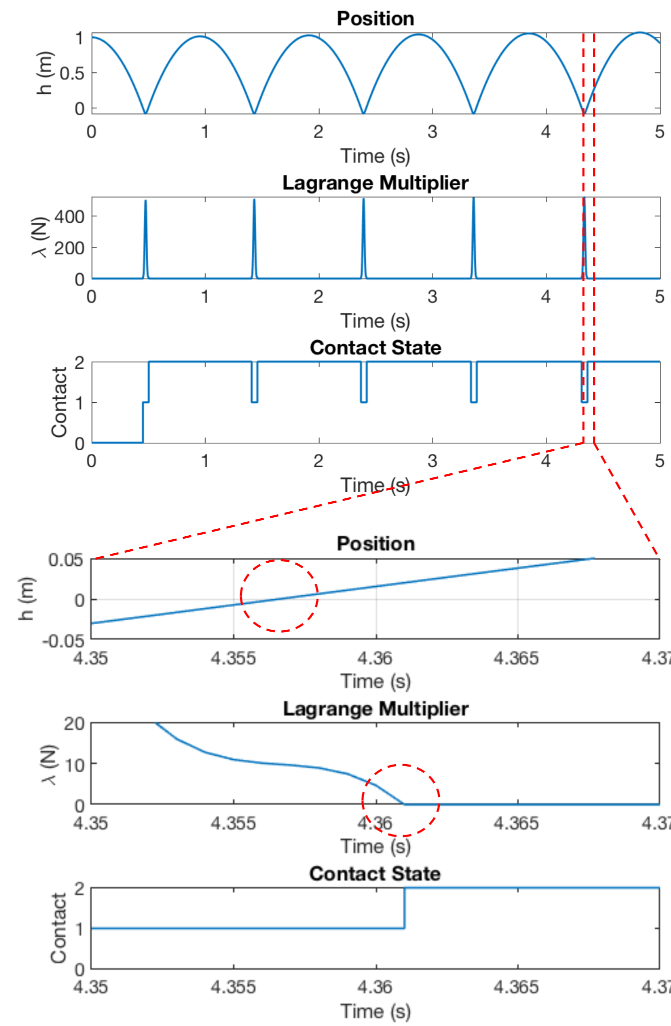




$$\dot{q} = v$$
$$m\dot{v} = -mg + \lambda$$
$$0 = \ddot{h} + \alpha_1 \dot{h} + \alpha_0 h + \alpha_3 h^3$$

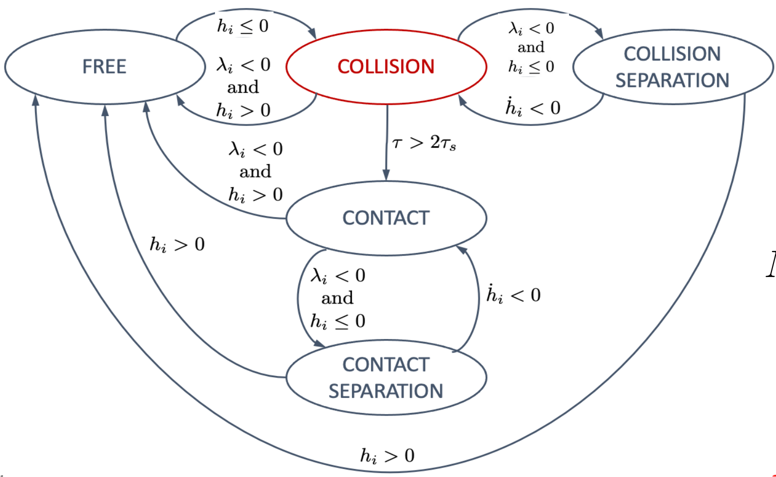
$$\dot{q} = v$$
$$m\dot{v} = -mg$$
$$\lambda = 0$$

```
model myBouncingBall

Real q(start=1.0), v(start=0.0), f;
Real h, hDot, hDotDot, lambda(start=0);
discrete Integer contact(start=0);
Boolean b1, b2, b3, b4;
parameter Real g=9.81, m=1.0;
parameter Real a0=100, a1=20, a2=1e6;
parameter Boolean linFlag = false;

algorithm
b1 := h <= 0;
b2 := lambda < 0.0;
b3 := h > 0.0;
b4 := h <= 0 and hDot < 0;
if edge(b1) and contact == 0 then
  contact := 1;
end if;
if contact == 1 and edge(b2) then
  contact := 2;
end if;
if contact == 2 and edge(b3) then
  contact := 0;
end if;
if contact == 2 and edge(b4) then
  contact := 1;
end if;

equation
if contact == 1 or linFlag then
  0 = hDotDot + a1*hDot + a0*h + a2*h^3;
else
  lambda = 0.0;
end if;

f = if linFlag then lambda
      else contact*lambda;

der(q) = v;
m * der(v) = -m * g + f;

h = q;
hDot = der(h);
hDotDot = der(hDot);

end myBouncingBall;
```

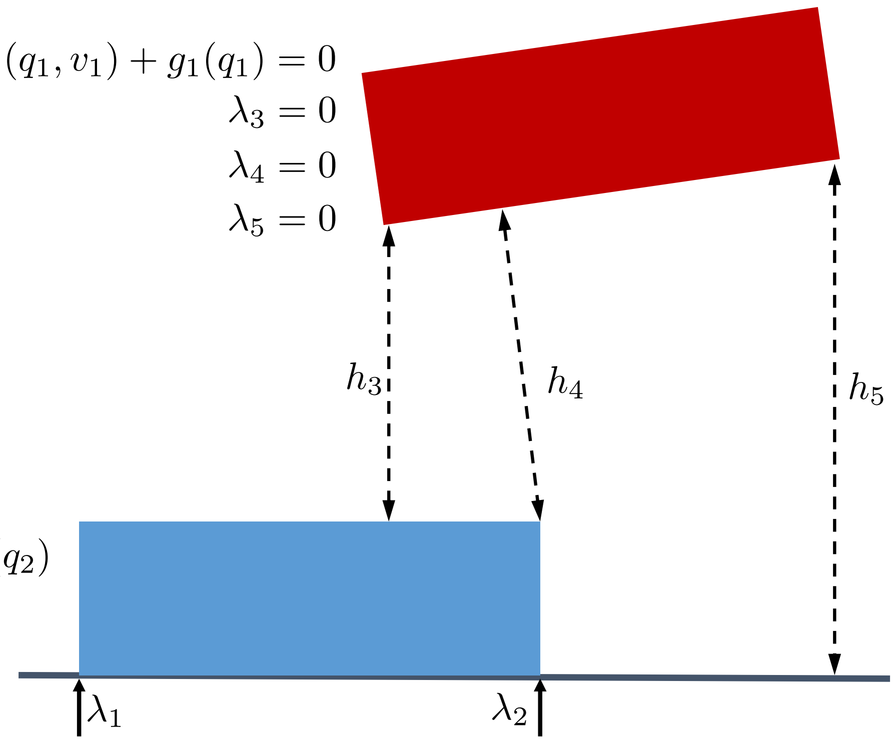


Cartoon Example – Falling Red Body is Initially Free

FSM for h_3



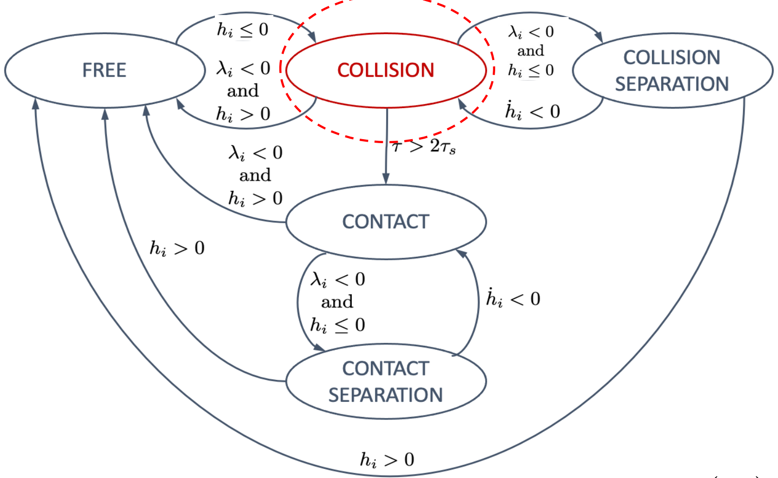
$$M_1(q_1)\dot{v}_1 + C_1(q_1, v_1) + g_1(q_1) = 0$$
$$\lambda_3 = 0$$
$$\lambda_4 = 0$$
$$\lambda_5 = 0$$



$$M_1(q_2)\dot{v}_2 + C_2(q_2, v_2) + g_2(q_2) = \lambda_1 dH_1(q_2) + \lambda_2 dH_2(q_2)$$
$$\ddot{h_1} + \dot{h_1} + h_1 = 0$$
$$\ddot{h_2} + \dot{h_2} + h_2 = 0$$

Cartoon Example – Collision

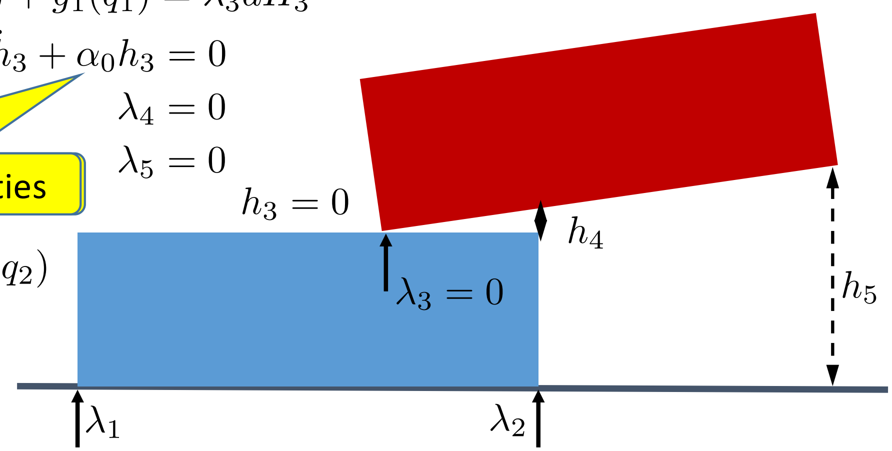
FSM for h_3



$$M_1(q_1)\dot{v}_1 + C_1(q_1, v_1) + g_1(q_1) = \lambda_3 dH_3$$
$$\ddot{h}_3 + \alpha_1 \dot{h}_3 + \alpha_0 h_3 = 0$$
$$\lambda_4 = 0$$
$$\lambda_5 = 0$$
$$h_3 = 0$$

Represents elastic impact...

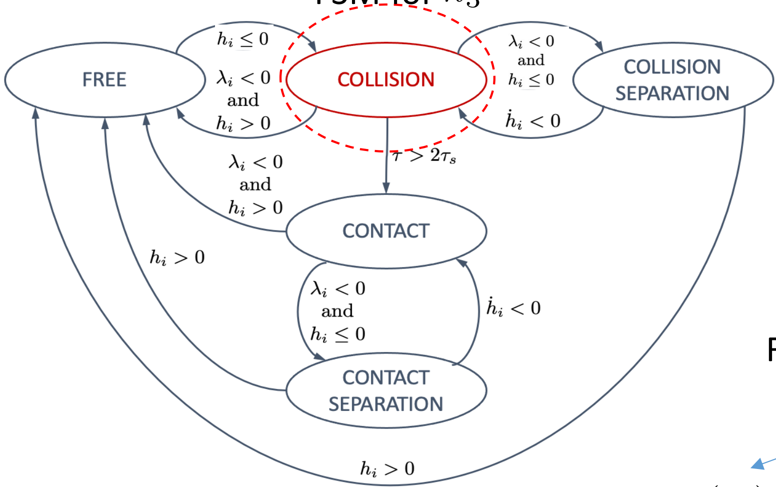
Tune parameters to material properties



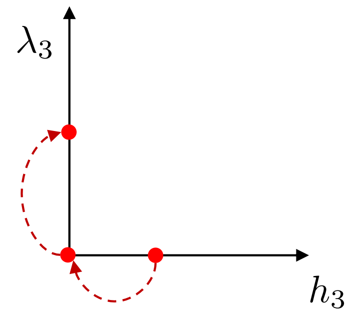
$$M_1(q_2)\dot{v}_2 + C_2(q_2, v_2) + g_2(q_2) = \lambda_1 dH_1(q_2) + \lambda_2 dH_2(q_2)$$
$$\ddot{h}_1 + \dot{h}_1 + h_1 = 0$$
$$\ddot{h}_2 + \dot{h}_2 + h_2 = 0$$

Cartoon Example – Fast Collision Time Scale

FSM for h_3



Solution for λ_3, h_3
Simulated in fast time scale



Freeze – Fast system is linear

$$M_1(q_1)\dot{v}_1 + C_1(q_1, v_1) + g_1(q_1) = \lambda_3 dH_3$$

$$\ddot{h}_3 + \alpha_1 \dot{h}_3 + \alpha_0 h_3 = 0$$

$$\lambda_4 = 0$$

$$\lambda_5 = 0$$

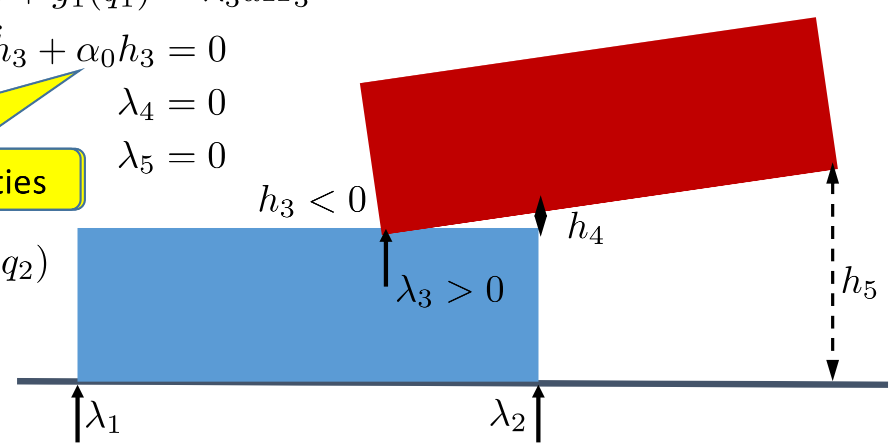
Represents elastic impact…

Tune parameters to material properties

$$M_1(q_2)\dot{v}_2 + C_2(q_2, v_2) + g_2(q_2) = \lambda_1 dH_1(q_2) + \lambda_2 dH_2(q_2)$$

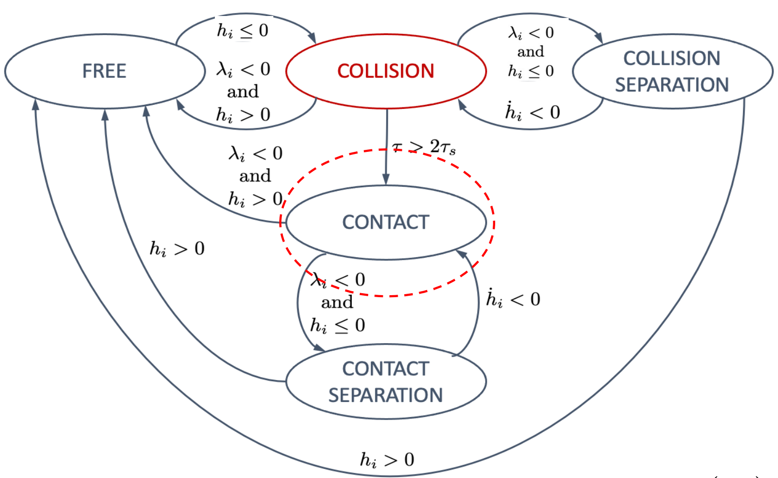
$$\ddot{h}_1 + \dot{h}_1 + h_1 = 0$$

$$\ddot{h}_2 + \dot{h}_2 + h_2 = 0$$

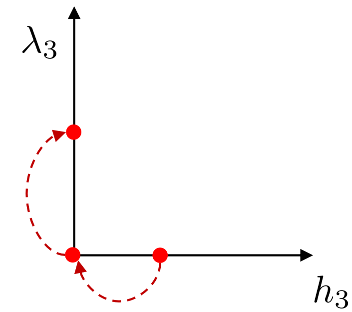


Cartoon Example – Switch to Contact State after Fast Transient

FSM for h_3



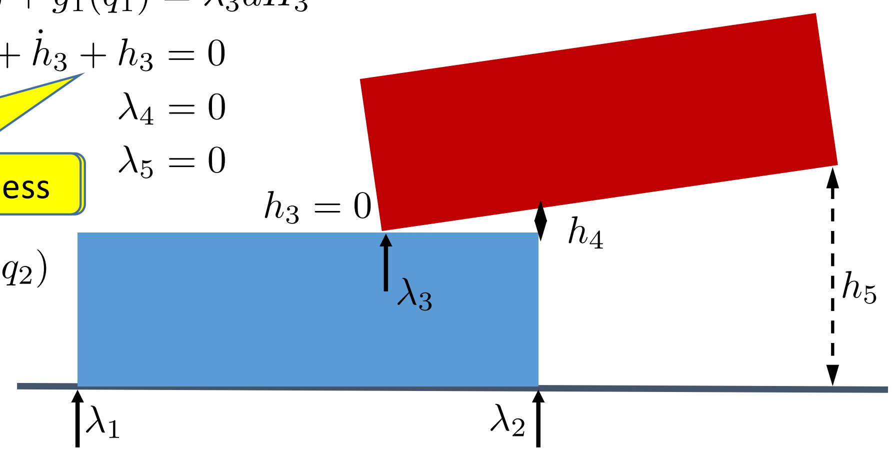
Solution for λ_3, h_3
Simulated in fast time scale



$$M_1(q_1)\dot{v}_1 + C_1(q_1, v_1) + g_1(q_1) = \lambda_3 dH_3$$
$$\ddot{h}_3 + \dot{h}_3 + h_3 = 0$$
$$\lambda_4 = 0$$
$$\lambda_5 = 0$$

Now enforces constraint

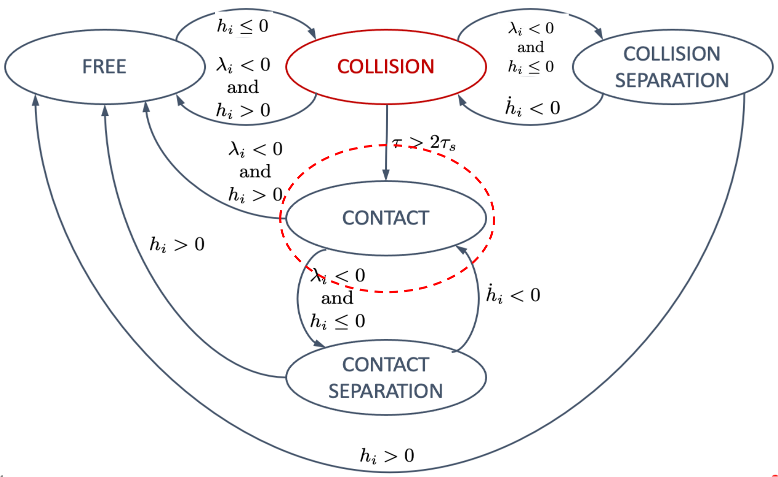
Made parameters =1 to reduce stiffness



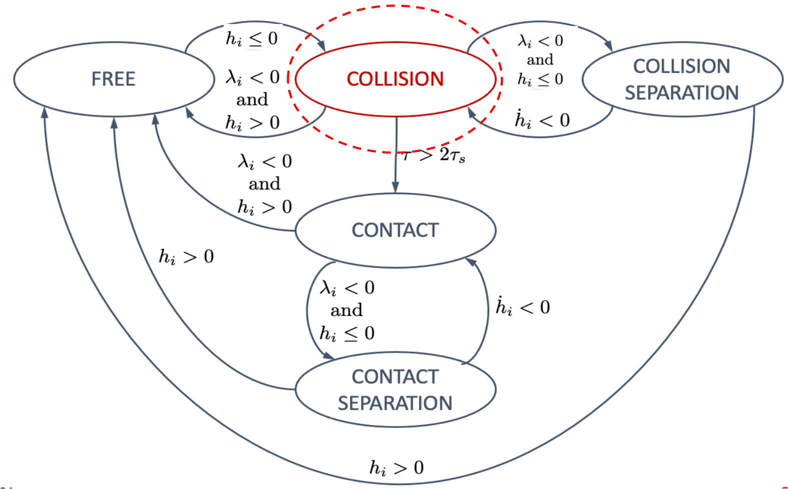
$$M_1(q_2)\dot{v}_2 + C_2(q_2, v_2) + g_2(q_2) = \lambda_1 dH_1(q_2) + \lambda_2 dH_2(q_2)$$
$$\ddot{h}_1 + \dot{h}_1 + h_1 = 0$$
$$\ddot{h}_2 + \dot{h}_2 + h_2 = 0$$

Cartoon Example – Collision

FSM for h_3

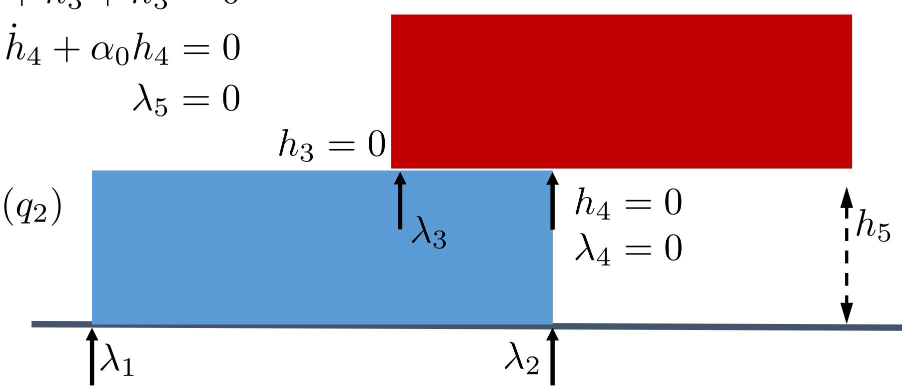


FSM for h_4



$$M_1(q_1)\dot{v}_1 + C_1(q_1, v_1) + g_1(q_1) = \lambda_3 dH3(q_1) + \lambda_4 dH_4(q_1)$$
$$\ddot{h}_3 + \dot{h}_3 + h_3 = 0$$
$$\ddot{h}_4 + \alpha_1 \dot{h}_4 + \alpha_0 h_4 = 0$$
$$\lambda_5 = 0$$

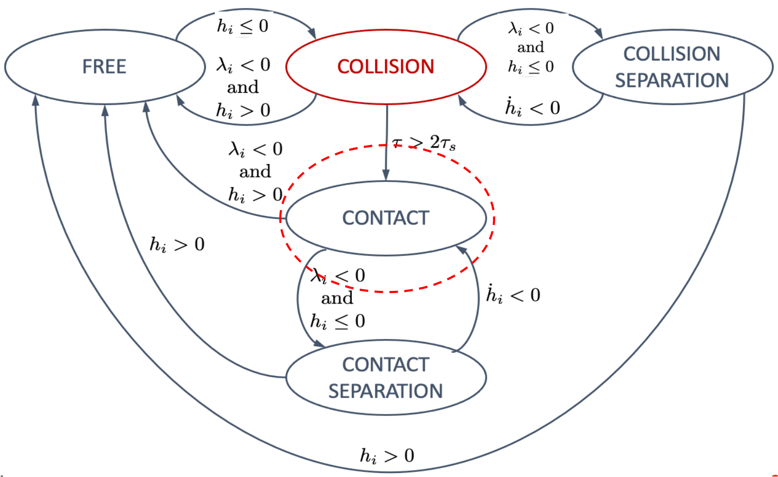
Represents second elastic impact…



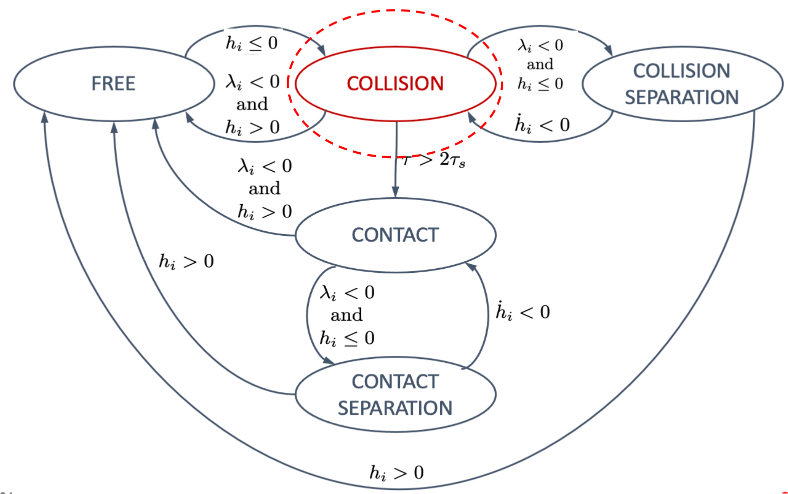
$$M_1(q_2)\dot{v}_2 + C_2(q_2, v_2) + g_2(q_2) = \lambda_1 dH_1(q_2) + \lambda_2 dH_2(q_2)$$
$$\ddot{h}_1 + \dot{h}_1 + h_1 = 0$$
$$\ddot{h}_2 + \dot{h}_2 + h_2 = 0$$

Cartoon Example – Fast Timescale during Collision

FSM for h_3



FSM for h_4



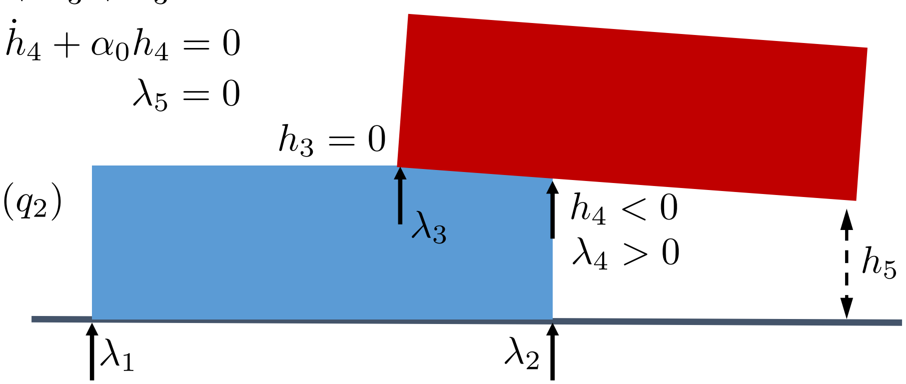
$$M_1(q_1)\dot{v}_1 + C_1(q_1, v_1) + g_1(q_1) = \lambda_3 dH3(q_1) + \lambda_4 dH_4(q_1)$$

$$\ddot{h}_3 + \dot{h}_3 + h_3 = 0$$

$$\ddot{h}_4 + \alpha_1 \dot{h}_4 + \alpha_0 h_4 = 0$$

$$\lambda_5 = 0$$

Represents second elastic impact…



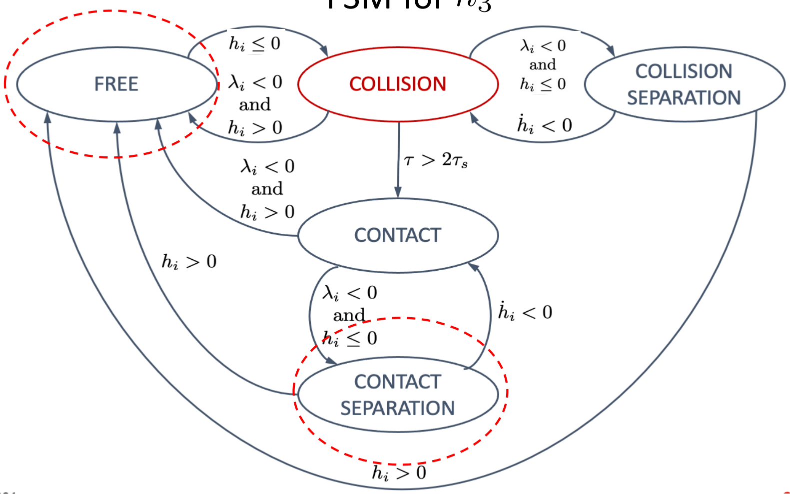
$$M_1(q_2)\dot{v}_2 + C_2(q_2, v_2) + g_2(q_2) = \lambda_1 dH_1(q_2) + \lambda_2 dH_2(q_2)$$

$$\ddot{h}_1 + \dot{h}_1 + h_1 = 0$$

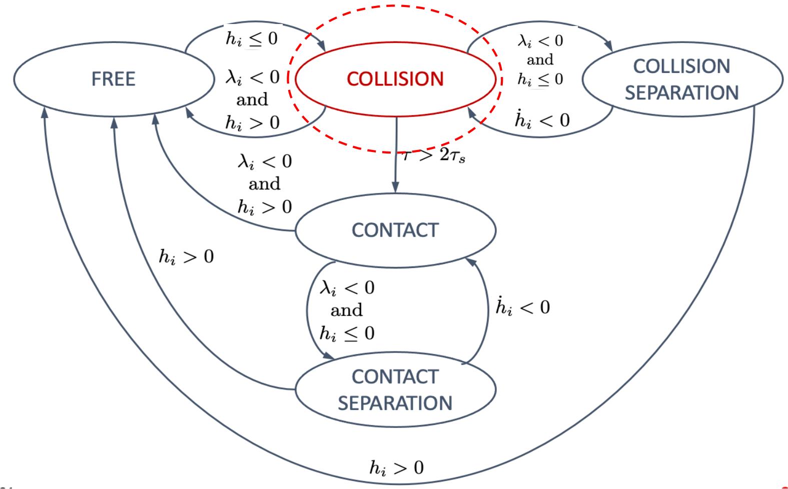
$$\ddot{h}_2 + \dot{h}_2 + h_2 = 0$$

Cartoon Example – h_3 Loses contact

FSM for h_3

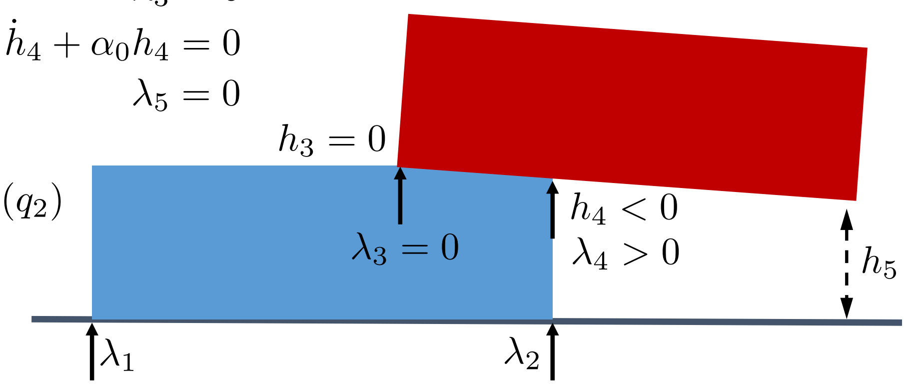


FSM for h_4



$$M_1(q_1)\dot{v}_1 + C_1(q_1, v_1) + g_1(q_1) = \lambda_4 dH_4(q_1)$$
$$\lambda_3 = 0$$
$$\ddot{h}_4 + \alpha_1 \dot{h}_4 + \alpha_0 h_4 = 0$$
$$\lambda_5 = 0$$

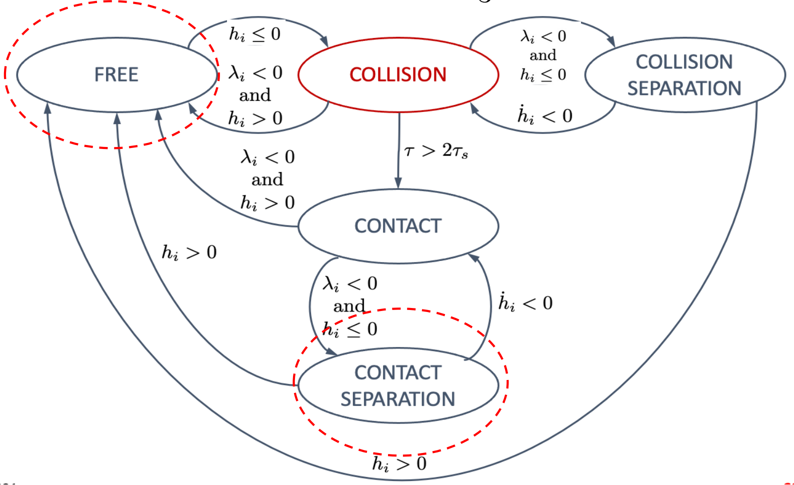
Contact constraint broken



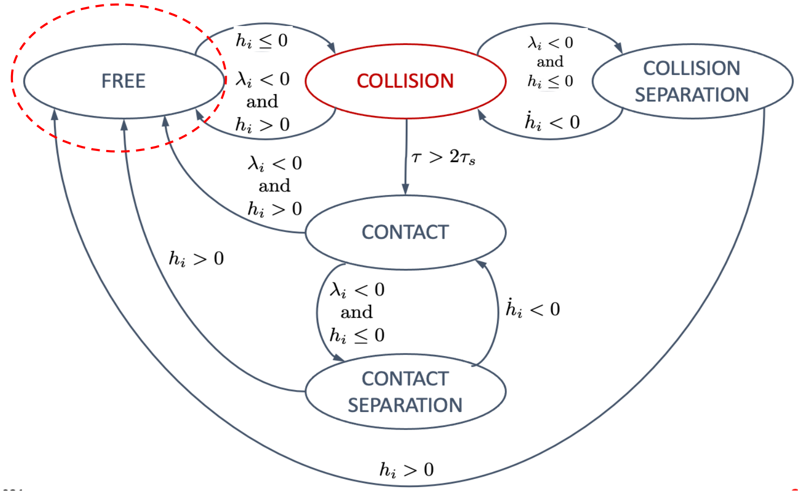
$$M_1(q_2)\dot{v}_2 + C_2(q_2, v_2) + g_2(q_2) = \lambda_1 dH_1(q_2) + \lambda_2 dH_2(q_2)$$
$$\ddot{h}_1 + \dot{h}_1 + h_1 = 0$$
$$\ddot{h}_2 + \dot{h}_2 + h_2 = 0$$

Cartoon Example – Both Constraints lose contact

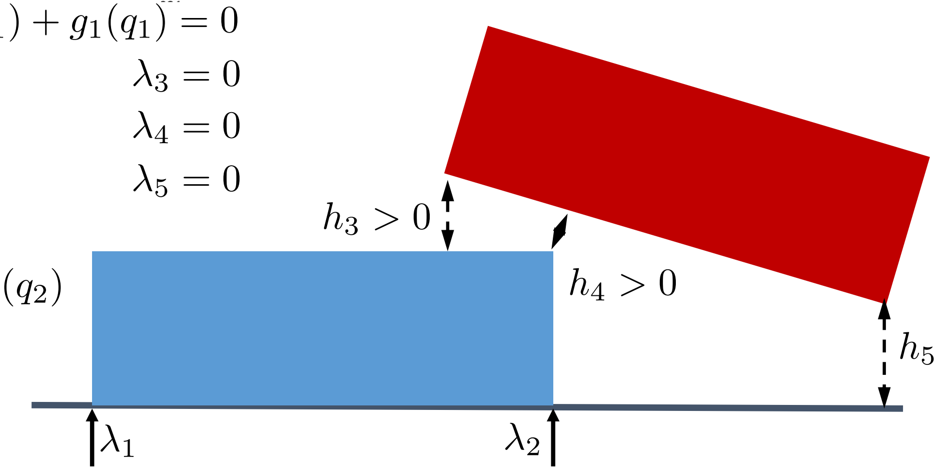
FSM for h_3



FSM for h_4



$$M_1(q_1)\dot{v}_1 + C_1(q_1, v_1) + g_1(q_1) = 0$$
$$\lambda_3 = 0$$
$$\lambda_4 = 0$$
$$\lambda_5 = 0$$



$$M_1(q_2)\dot{v}_2 + C_2(q_2, v_2) + g_2(q_2) = \lambda_1 dH_1(q_2) + \lambda_2 dH_2(q_2)$$
$$\ddot{h}_1 + \dot{h}_1 + h_1 = 0$$
$$\ddot{h}_2 + \dot{h}_2 + h_2 = 0$$

Does Not Conserve Energy

- Contact State

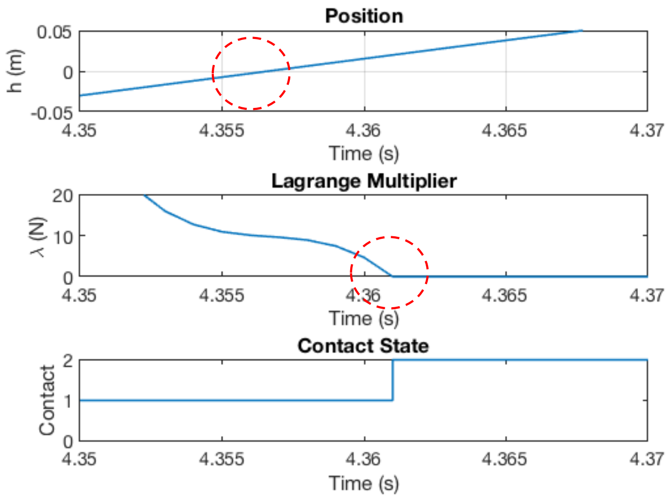
$$\dot{q} = v$$
$$m\dot{v} = -mg + \lambda$$
$$0 = \ddot{h} + \alpha_1 \dot{h} + \alpha_0 h + \alpha_3 h^3$$

- Solve for Lagrange Multiplier...

$$\lambda(t) = (\alpha_0 q(t) + \alpha_2 q^3(t) + g)m.$$

- If this were a "spring force," the g would not appear
- This force adds energy to the ball when $\lambda > 0$ and $h > 0$
- This force is responsible for
  - $h \to 0$ (no penetration in steady state) ... Good
  - Failure to conserve energy ... Bad
- But, its not *that* bad
  - Energy added only when $\lambda > 0$ and $\dot{h} \neq 0$
  - In typical applications, $\alpha_1 > 0$ (and $\alpha_1 = 0$ isn't stable)
  - LCP formulations also fail to conserve energy

Lisa, get in here! In this house we obey the laws of thermodynamics!

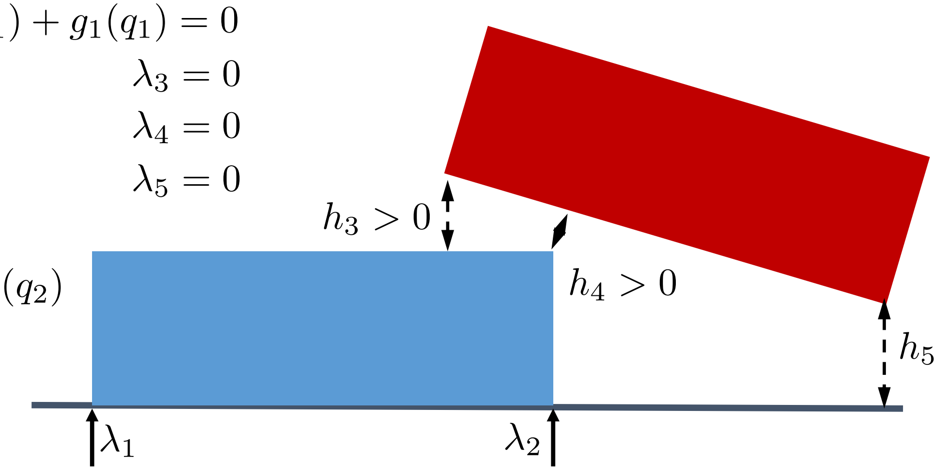


DAE Contact – Collision Model Properties

- Mathematical DAE Model
  - Does not mix modes of computation (e.g. ODE solver + QP solver)
  - Useful for design / analysis beyond simulation!
- Continuous - All states are continuous functions of time!
- Can use implicit, stiff solver e.g. DASSL.
  - Stable simulation for wide time scales
  - Adaptive, automatic step size. Faster than fastest time constant system.
- But…
  - No mature software.
  - Requires h and dH.

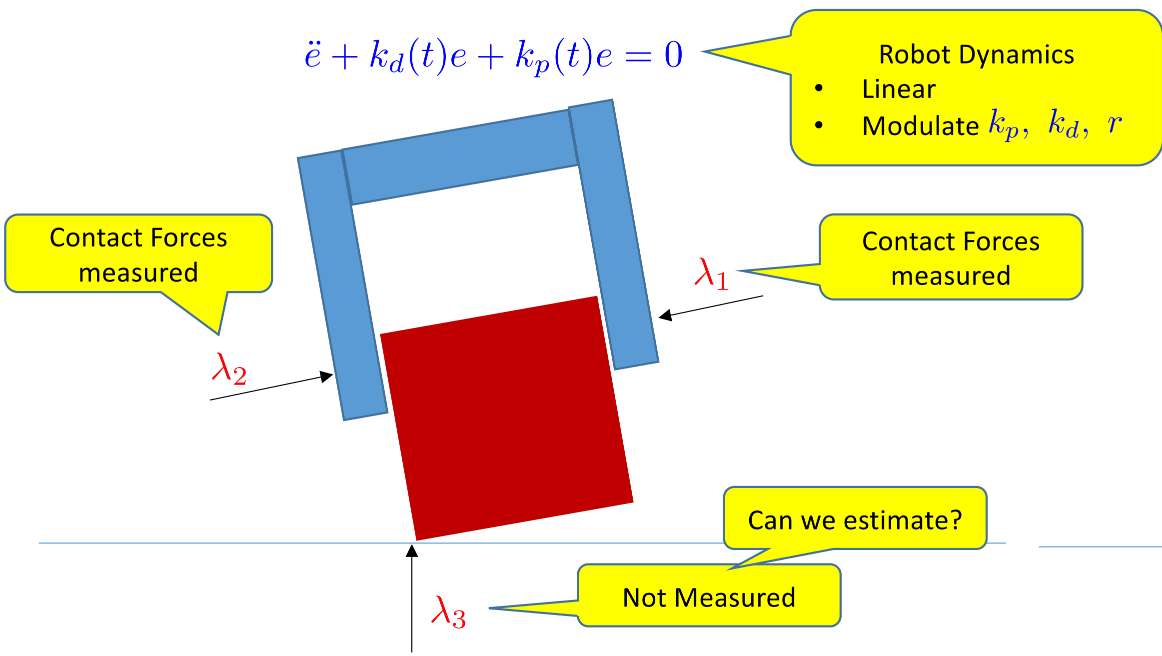
$$
\begin{aligned}
M_1(q_1)\dot{v}_1 + C_1(q_1, v_1) + g_1(q_1) &= 0 \\
\lambda_3 &= 0 \\
\lambda_4 &= 0 \\
\lambda_5 &= 0
\end{aligned}
$$

$$
\begin{aligned}
M_1(q_2)\dot{v}_2 + C_2(q_2, v_2) + g_2(q_2) &= \lambda_1 dH_1(q_2) + \lambda_2 dH_2(q_2) \\
\ddot{h}_1 + \dot{h}_1 + h_1 &= 0 \\
\ddot{h}_2 + \dot{h}_2 + h_2 &= 0
\end{aligned}
$$



Other Possible Uses of this Model

Task State or Parameter Estimation



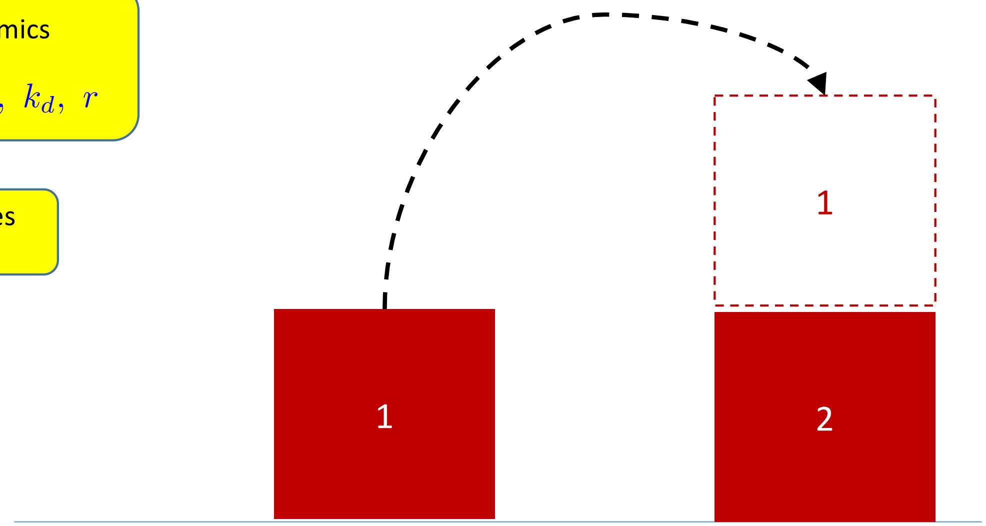
$$M(q)\dot{v}_1 + C(q,v) + g(q) = \lambda_1 H_1(q) + \lambda_2 H_2(q) + \lambda_3 H_3(q)$$

Not Measured

Can we estimate?

$$h_1(q) = 0$$
$$h_2(q) = 0$$
$$h_3(q) = 0$$

MPC for Robotic Manipulation / Assembly



Can we use MPC to compute assembly trajectory?

- Linear robot dynamics.
- Constrained dynamics during contact. Hybrid.
- DAE model is almost linear.

Can control theory define robustness metrics?

- System - oriented thinking vs. signal oriented.

MITSUBISHI ELECTRIC

Changes for the Better