Presenting JIMI:

A Hopping Monopod Robot Incorporating Nonlinear Series Elastic Actuators, Fiber-Reinforced Polymer Construction, and a Concurrent Asynchronous Dataflow-based Centroidal Momentum Balance Controller

Ivar L. Thorson

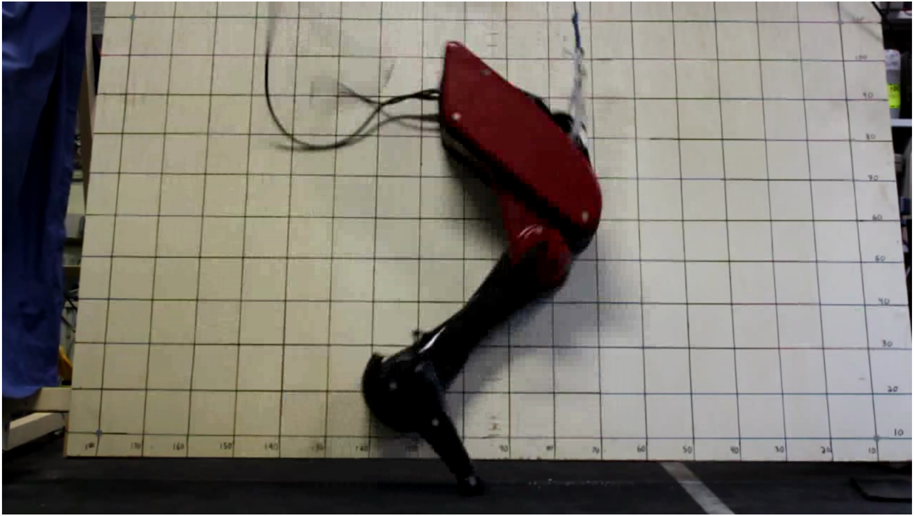
Department of Advanced Robotics
Istituto Italiano di Tecnologia
Università di Genova

April 26, 2012

The Topic of this Presentation

Name:	JIMI
Mass:	8.3kg (minimally) 10.9kg (autonomous)
Size:	~95cm tall, 27cm shank, 40cm thigh, 51cm body
Speed:	1.0 m/s, 2.1 hops/sec (design goals, not yet reached)
Height:	Jumps 36cm vertically (from squat, uncontrolled)
Power:	2x111W electric motors
Energy:	2x27J elastic energy storage

Preliminary Teaser Video



Presentation Roadmap

"JIMI" integrates many details into a state-of-the-art robot:

- **Dynamics:** Mechanical and control dynamics were designed simultaneously via simulation
- **Actuation:** Uses novel, patented nonlinear series elastic actuators
- **Control:** Balances dynamically via task-space control of centroid
- **Estimation:** Performs online, model-based system identification
- **Software:** Software is asynchronous, dataflow-based & concurrent
- **Materials:** Features lightweight, monocoque structures made of CFRP (Carbon fiber-reinforced polymer) and urethane foam

This presentation will take about 25 minutes.

Part I: The Dynamics of Running

Goal: To proceed analytically from animal-like sinusoidal vertical ground reaction forces (GRFs) to a specification of actuation and control for JIMI.

Overview:

1. Dynamics of animal running
2. A simplified model of running
3. How dynamics lead to actuation specification

Dynamics of Animal Running

What *best* characterizes running?

Dynamics of Animal Running

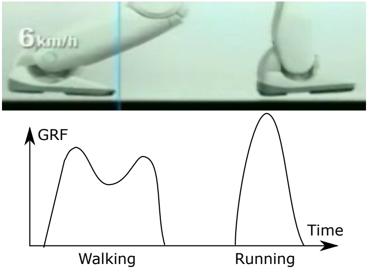
What *best* characterizes running?

- No feet touching the ground?

Dynamics of Animal Running

What *best* characterizes running?

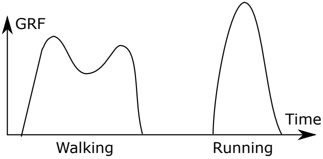
- No feet touching the ground?
- A single-hump vertical GRF?

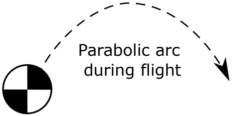


Dynamics of Animal Running

What *best* characterizes running?

- No feet touching the ground?
- A single-hump vertical GRF?
- Flight Center of Mass (CoM) motion that's ballistic?

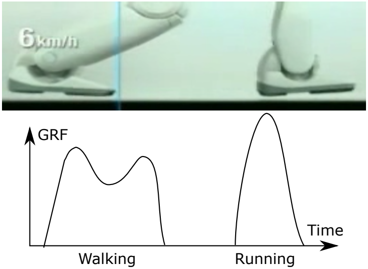


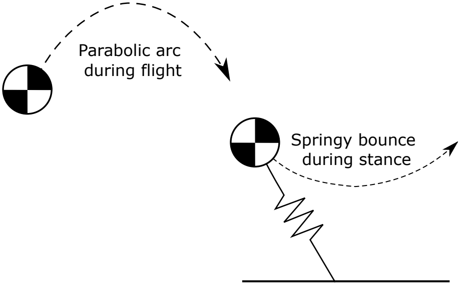


Dynamics of Animal Running

What *best* characterizes running?

- No feet touching the ground?
- A single-hump vertical GRF?
- Flight Center of Mass (CoM) motion that's ballistic?
- Stance forces that resemble an elastic collision?





Dynamics of Animal Running

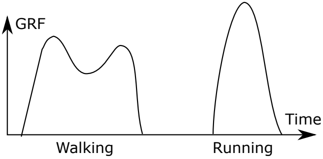
What *best* characterizes running?

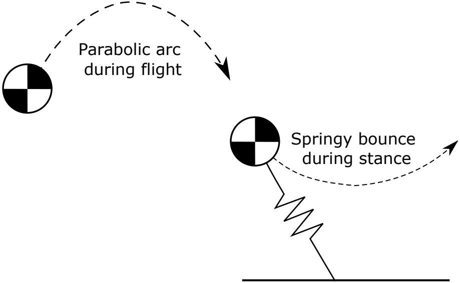
- No feet touching the ground?
- A single-hump vertical GRF?
- Flight Center of Mass (CoM) motion that's ballistic?
- Stance forces that resemble an elastic collision?

Animals exhibit all the above.

The design of JIMI assumes *a GRF resembling an elastic collision* is key.

Can we get a rough spec from this?





Peak GRF for Elastic GRFs

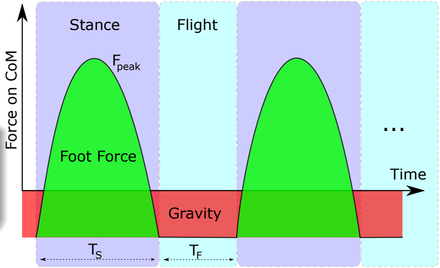
The sum of all impulses to the CoM should be zero over a stride.

Impulse due to GRF

$$I_S = \int_0^{T_s} F_{peak} \sin\left(\frac{\pi}{T_S} t\right)\, dt$$

Impulse due to gravity

$$I_g = m_c g(T_S + T_F)$$



Peak GRF for Elastic GRFs

The sum of all impulses to the CoM should be zero over a stride.

Impulse due to GRF

$$I_S = \int_0^{T_s} F_{peak} \sin\left(\frac{\pi}{T_S} t\right) \, dt$$

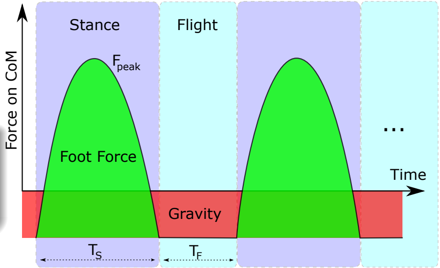
Impulse due to gravity

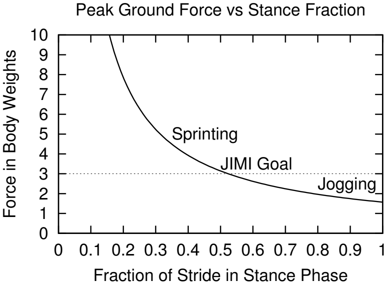
$$I_g = m_c g (T_S + T_F)$$

Peak GRF

$$F_{peak}(T_S, T_F) = \frac{\pi m_c g (T_S + T_F)}{2 T_S}$$

JIMI Goal: 50% stance, ~3B.W.





CoM Power and Energy in Stance

CoM Height

$$y_c(t) = \begin{cases} \frac{gt^2}{2} + \dot{y}_c^{LO} t & \text{(flight)} \\ \frac{(g + \frac{F_{peak}}{m_c}\sin(\frac{\pi}{T_S}t))t^2}{2} + \dot{y}_c^{TD} t \end{cases}$$

GRF Power on CoM

$$P(t) = F_f(t)\dot{y}_c(t)$$

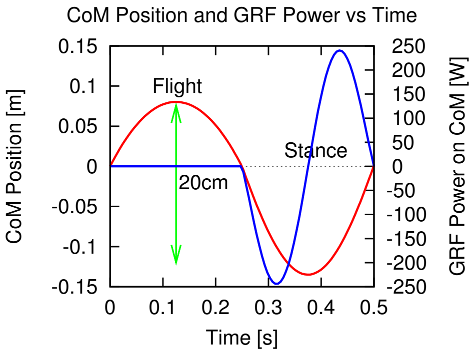
CoM Energy Change

$$\Delta V(y_c) = m_c g \Delta y_c$$

For a 10kg robot with 0.25s stance and flight times:

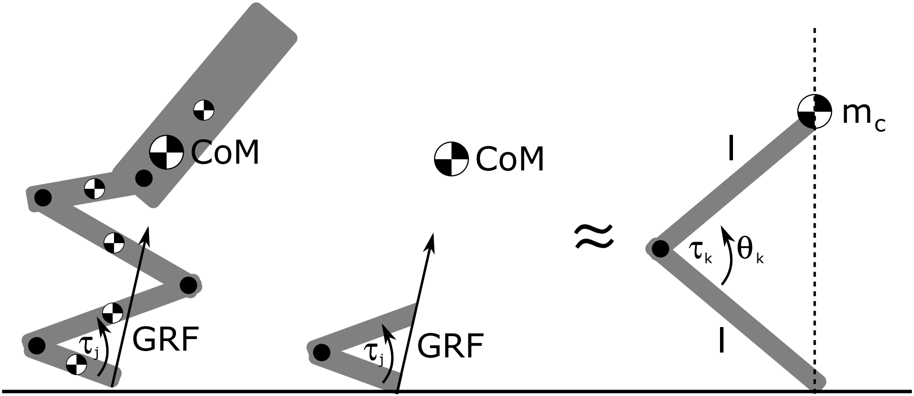
- ~20cm total vertical motion
- ~20J absorb/release per hop
- ~250W peak mech. power

How would joint torques look?



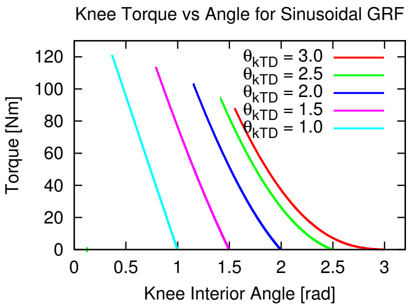
Revolute-jointed Runners

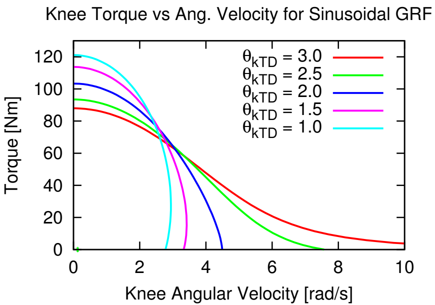
Joint torques can be estimated from GRF vector and simple kinematics.

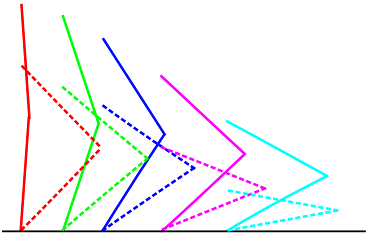


The model on the right is a reasonable general case approximation.
(We ignore moments of inertia and link length asymmetries)

Actuator Torque/Velocity Requirements







- Let $m_c = 10$kg, $l = 0.4$m, 2Hz hop, 50% stance duty cycle
- Torques <120Nm torque
- Velocity <10rads/sec
- Straighter legs more nonlinear

Summary of Running Dynamics

Assuming an elastic vertical GRF gave us:

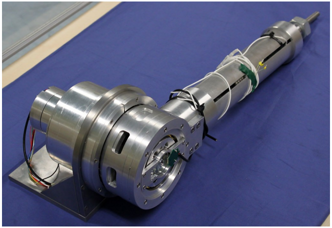
- CoM motion
- CoM power & energy
- Peak GRF levels

Assuming revolute joints gave us:

- Rough character of joint torque nonlinearity
- Velocity, torque limits

Can we now design an actuator that satisfies the above specs?

Part II: Nonlinear Series Elastic Actuation



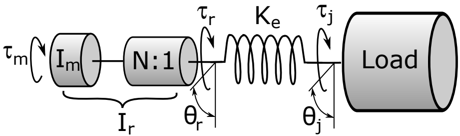
Goal: To present two novel, nonlinear series elastic actuators ideal for legged robots.

Overview:

1. Introduction to Series Elasticity
2. Optimal Series Elasticity
3. The Hypocycloid Mechanism
4. The HypoSEA-v1
5. The HypoSEA-v2

What is a Series Elastic Actuator? (SEA)

SEAs purposely introduce an elastic element between actuator and load.



Good effects:

- Improves L.F. force control
- Improves impact resistance
- Provides energy storage

Bad effects:

- Reduces force bandwidth
- Adds another DOF
- Naive controllers often waste work compressing elasticity

Transmission Effects

$$\tau_r = N\tau_m$$

$$I_r = N^2 I_m$$

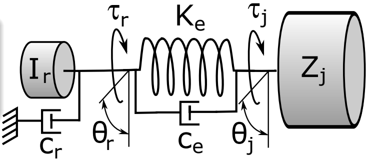
Normal Dynamic Stiffness

$$\frac{\tau_j}{\theta_j} = N^2 I_m s^2$$

SEA Dynamic Stiffness

$$\frac{\tau_j}{\theta_j} = K_e$$

Frequency Domain Analysis (*Williamson, 1995*)



Complex Rotor Torque

$$\tau_r(\tau_j, \dot{\theta}_j) = \left(\frac{I_r s^2 + c_r s}{c_e s + K_e} + 1 \right) \tau_j + (I_r s + c_r)\, \dot{\theta}_j$$

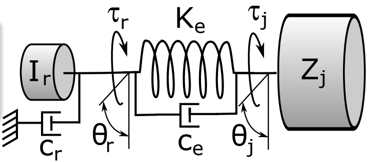
Conclusions:

- Only the red terms are unique to SEAs.

Frequency Domain Analysis (*Williamson, 1995*)

Complex Rotor Torque

$$\tau_r(\tau_j, \dot{\theta}_j) = \left(\frac{I_r s^2 + c_r s}{c_e s + K_e} + 1 \right) \tau_j + (I_r s + c_r)\, \dot{\theta}_j$$



Let $s = j\omega$, $c_r = c_e = 0$ to see spring effect:

$$\tau_r(\tau_j, \dot{\theta}_j) = \left(1 - \frac{I_r \omega^2}{K_e}\right) \tau_j + j I_r \omega \dot{\theta}_j$$

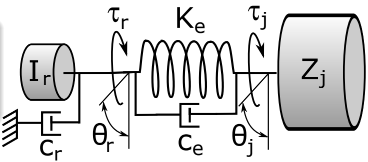
Conclusions:

- Only the red terms are unique to SEAs.
- Rotor-elastic resonance at $\sqrt{\frac{K_e}{I_r}}$

Frequency Domain Analysis (*Williamson, 1995*)

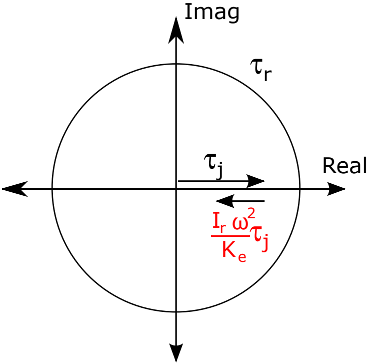
Complex Rotor Torque

$$\tau_r(\tau_j, \dot{\theta}_j) = \left(\frac{I_r s^2 + c_r s}{c_e s + K_e} + 1 \right) \tau_j + (I_r s + c_r)\, \dot{\theta}_j$$



Let $s = j\omega$, $c_r = c_e = 0$ to see spring effect:

$$\tau_r(\tau_j, \dot{\theta}_j) = \left(1 - \frac{I_r \omega^2}{K_e} \right) \tau_j + j I_r \omega \dot{\theta}_j$$

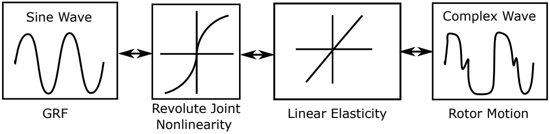


Conclusions:

- Only the red terms are unique to SEAs.
- Rotor-elastic resonance at $\sqrt{\frac{K_e}{I_r}}$
- Spring reduces τ_r for $\omega < \sqrt{2\frac{K_e}{I_r}}$
- More rotor torque "left over" to track load motion $\implies$ better force control

What about nonlinearities?

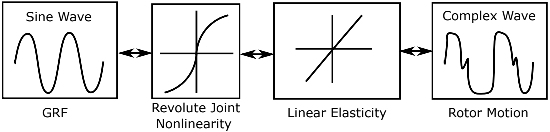
Problem: Running joint torques are nonlinear $\Longrightarrow$ excess rotor motion.



To maximize control torque available for counteracting disturbances, we want the rotor motion simple and harmonic.

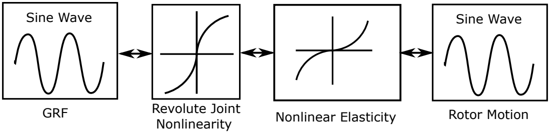
What about nonlinearities?

Problem: Running joint torques are nonlinear $\Longrightarrow$ excess rotor motion.



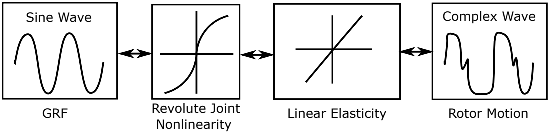
To maximize control torque available for counteracting disturbances, we want the rotor motion simple and harmonic.

Solution: For a given nonlinear stereotypical desired motion, an elastic nonlinearity exists that linearizes the desired-torque/rotor-motion relation.



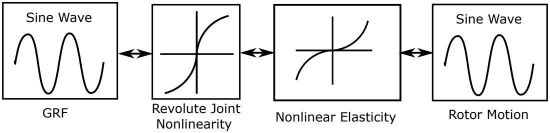
What about nonlinearities?

Problem: Running joint torques are nonlinear $\implies$ excess rotor motion.



To maximize control torque available for counteracting disturbances, we want the rotor motion simple and harmonic.

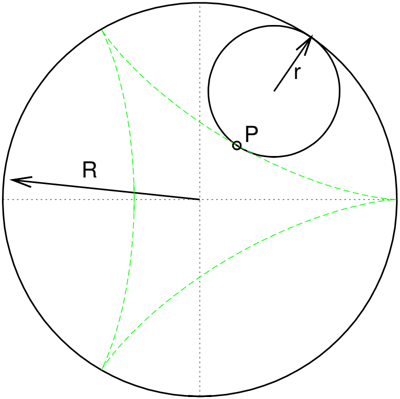
Solution: For a given nonlinear stereotypical desired motion, an elastic nonlinearity exists that linearizes the desired-torque/rotor-motion relation.

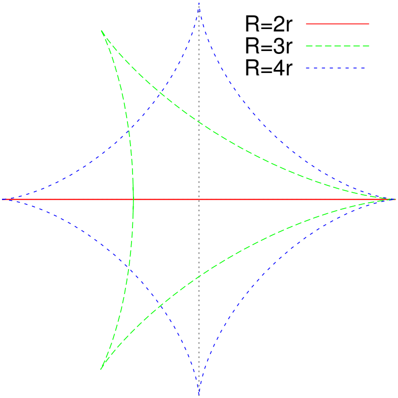


What mechanisms produce the proper nonlinear elasticity for running?

What's a Hypocycloid?

The curve traced by a point on a small circle rolling inside a larger circle.



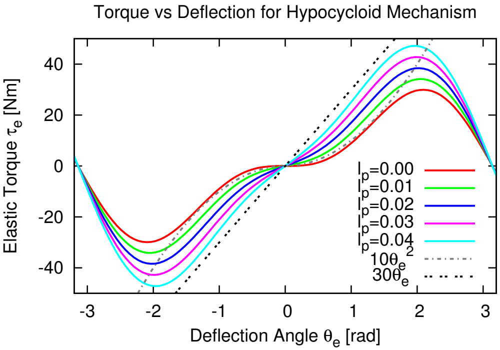


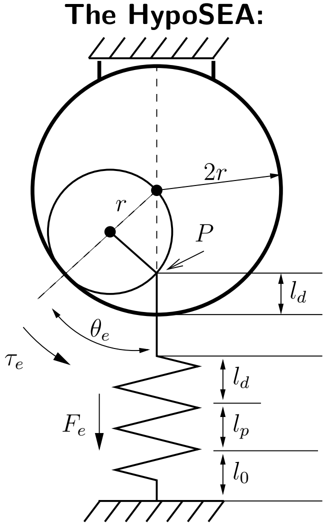
If $R = 2r$, a straight line is drawn from a revolute motion.

Hypocycloid-based Series Elastic Actuator

Torque-Angle Relation

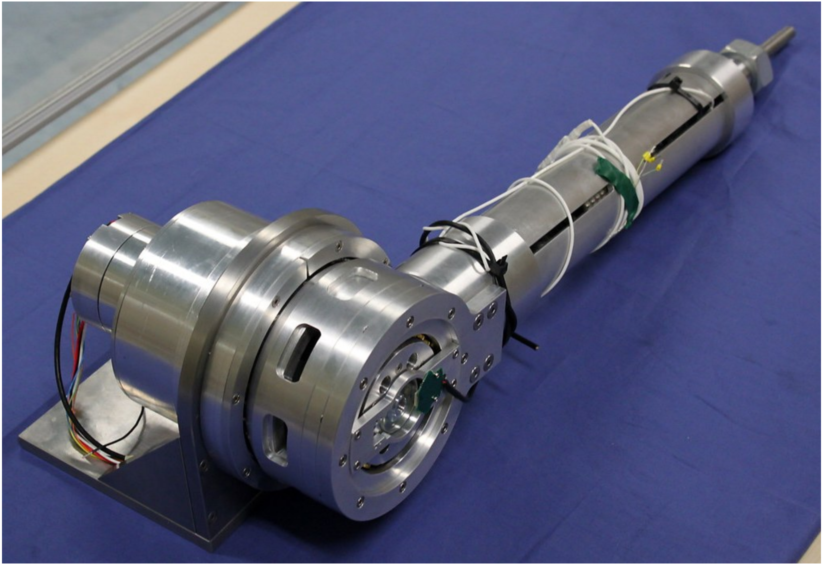
$$\tau_e(\theta_e) = 2rK_e(2r(1-\cos\theta_e)+l_p)\sin\theta_e$$



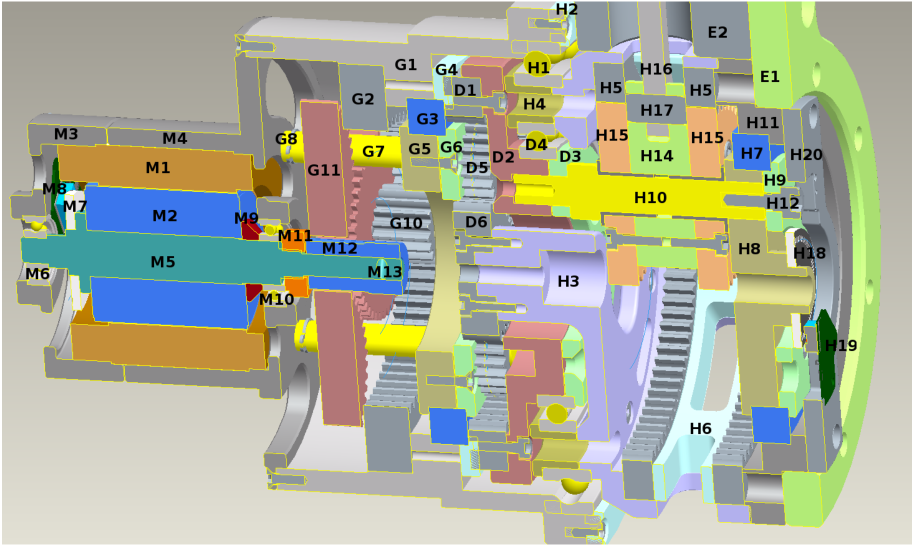


Varying the spring pretension l_p produces a useful family of curves for running robots!

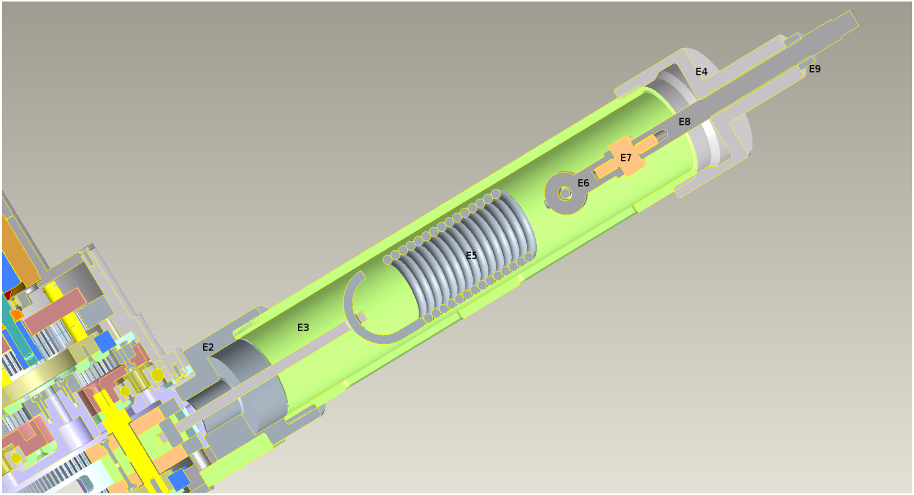
HypoSEA-v1 Photo



HypoSEA-v1 Cross Section



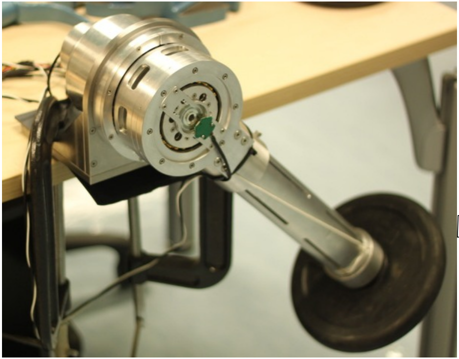
HypoSEA-v1 Cross Section

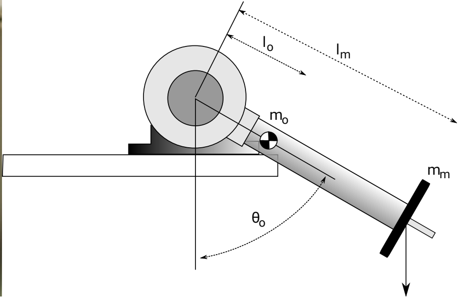


HypoSEA-v1 Video

Backdrivability $\implies$ Energy Recovery

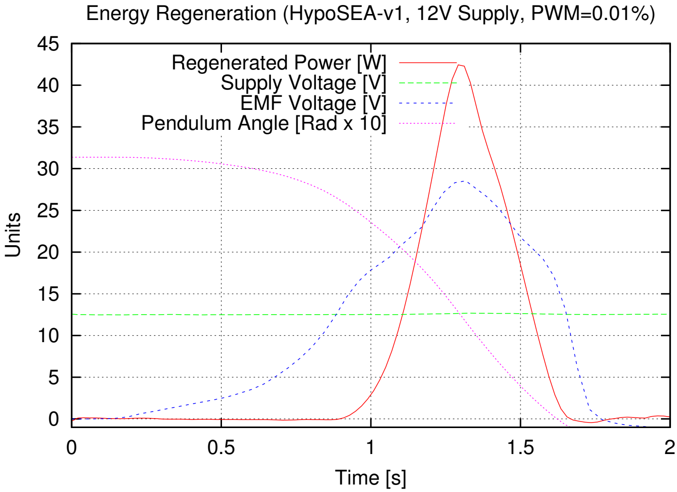
Experiment: How much energy can be recovered from a pendulum swing?





If a lead-acid battery is used as a power supply, when the BLDC motor spins fast enough, current flows into the battery even with a naive motor control board. Let's attach a 2kg mass and measure the energy absorbed.

HypoSEA-v1 Energy Recovery (65% Eff.)



Absorbed 13.7J of a possible 21.0J (excluding K.E. lost below EMF=13V).

HypoSEA-v1 Performance Results

The Good:

- Low passive mechanical impedance
- Impact resistance
- Backdrivability
- Energy regeneration efficiency (65%)
- Energy storage if rotor locked (>40J)

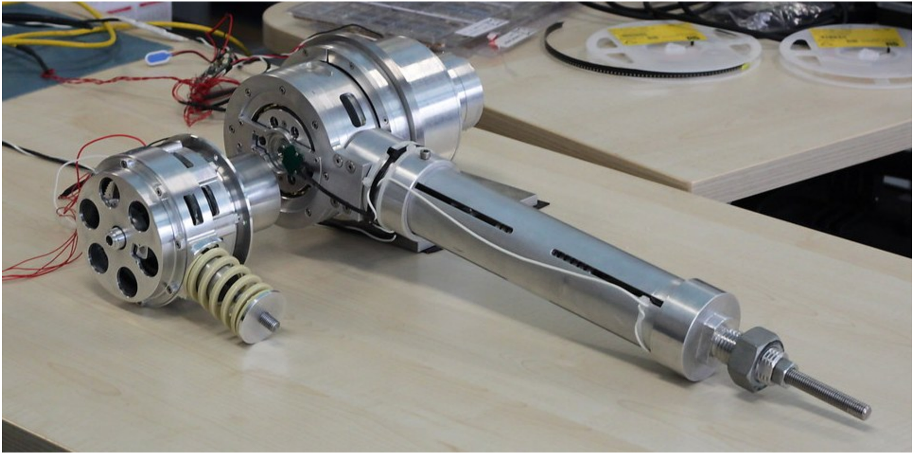
The Bad:

- Heavy (8.5kg)
- Big (0.5m)
- Too much friction (1-2Nm)
- Too little momentary torque (71Nm...goal was 120Nm)

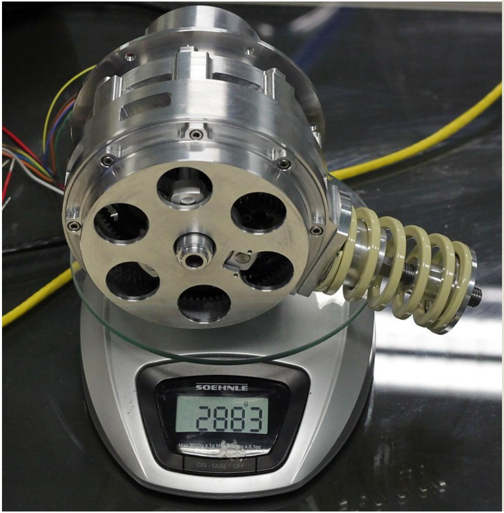
A second revision was clearly needed!

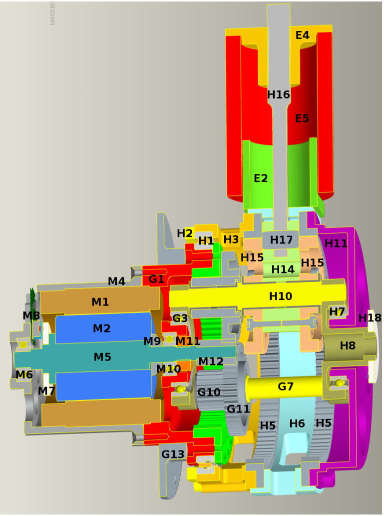
HypoSEA-v2 Photo Comparison

The HypoSEA-v2 (left) is the improved version of the HypoSEA-v1 (right).



HypoSEA-v2 Inside and Out





HypoSEA Performance Comparison

Description	v1	v2	Unit
Actuator mass	8.3	2.883	kg
Actuator diameter	14.0	12.4	cm
Longest exterior dimension	67	21	cm
Max tested joint torque	71	65	Nm
Max theoretical joint torque	126	70	Nm
Min resolvable torque	<0.02	<0.02	Nm
Max controlled joint vel	10.2	10.6	rad/s
Rotor-joint Gear Ratio	$18.\overline{3}$	17	
Elasticity-Rotor Gear Ratio	$12.8\overline{3}$	17	
Joint-Elasticity Gear Ratio	$\frac{10}{7}$	1	
Linear Spring Constant	10.09	30.82	N/mm
Max spring pretension	40	20	mm
Max spring deflection	72	48	mm
Hypocycloid gear radius	24	24	mm
Max spring energy*	42.3	27.3	J

HypoSEA Performance Comparison

Description	v1	v2	Unit	
Actuator mass	8.3	2.883	kg	Lighter!
Actuator diameter	14.0	12.4	cm	
Longest exterior dimension	67	21	cm	
Max tested joint torque	71	65	Nm	
Max theoretical joint torque	126	70	Nm	
Min resolvable torque	<0.02	<0.02	Nm	
Max controlled joint vel	10.2	10.6	rad/s	
Rotor-joint Gear Ratio	$18.\overline{3}$	17		
Elasticity-Rotor Gear Ratio	$12.8\overline{3}$	17		
Joint-Elasticity Gear Ratio	$\frac{10}{7}$	1		
Linear Spring Constant	10.09	30.82	N/mm	
Max spring pretension	40	20	mm	
Max spring deflection	72	48	mm	
Hypocycloid gear radius	24	24	mm	
Max spring energy*	42.3	27.3	J	

HypoSEA Performance Comparison

Description	v1	v2	Unit	
Actuator mass	8.3	2.883	kg	Lighter!
Actuator diameter	14.0	12.4	cm	
Longest exterior dimension	67	21	cm	Smaller!
Max tested joint torque	71	65	Nm	
Max theoretical joint torque	126	70	Nm	
Min resolvable torque	<0.02	<0.02	Nm	
Max controlled joint vel	10.2	10.6	rad/s	
Rotor-joint Gear Ratio	$18.\overline{3}$	17		
Elasticity-Rotor Gear Ratio	$12.8\overline{3}$	17		
Joint-Elasticity Gear Ratio	$\frac{10}{7}$	1		
Linear Spring Constant	10.09	30.82	N/mm	
Max spring pretension	40	20	mm	
Max spring deflection	72	48	mm	
Hypocycloid gear radius	24	24	mm	
Max spring energy*	42.3	27.3	J	

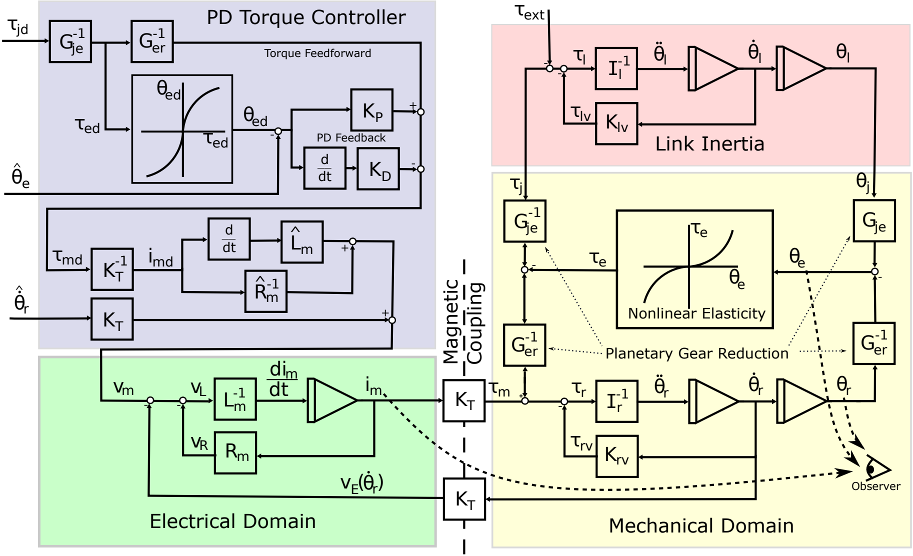
HypoSEA Performance Comparison

Description	v1	v2	Unit	
Actuator mass	8.3	2.883	kg	Lighter!
Actuator diameter	14.0	12.4	cm	
Longest exterior dimension	67	21	cm	Smaller!
Max tested joint torque	71	65	Nm	Same Trq!
Max theoretical joint torque	126	70	Nm	
Min resolvable torque	<0.02	<0.02	Nm	
Max controlled joint vel	10.2	10.6	rad/s	
Rotor-joint Gear Ratio	$18.\overline{3}$	17		
Elasticity-Rotor Gear Ratio	$12.8\overline{3}$	17		
Joint-Elasticity Gear Ratio	$\frac{10}{7}$	1		
Linear Spring Constant	10.09	30.82	N/mm	
Max spring pretension	40	20	mm	
Max spring deflection	72	48	mm	
Hypocycloid gear radius	24	24	mm	
Max spring energy*	42.3	27.3	J	

HypoSEA Performance Comparison

Description	v1	v2	Unit	
Actuator mass	8.3	2.883	kg	Lighter!
Actuator diameter	14.0	12.4	cm	
Longest exterior dimension	67	21	cm	Smaller!
Max tested joint torque	71	65	Nm	Same Trq!
Max theoretical joint torque	126	70	Nm	
Min resolvable torque	<0.02	<0.02	Nm	
Max controlled joint vel	10.2	10.6	rad/s	
Rotor-joint Gear Ratio	$18.\overline{3}$	17		
Elasticity-Rotor Gear Ratio	$12.8\overline{3}$	17		
Joint-Elasticity Gear Ratio	$\frac{10}{7}$	1		
Linear Spring Constant	10.09	30.82	N/mm	
Max spring pretension	40	20	mm	
Max spring deflection	72	48	mm	
Hypocycloid gear radius	24	24	mm	
Max spring energy*	42.3	27.3	J	Worse...

HypoSEA Torque Controller



Summary of Actuation

- Hypocycloid mechanism makes the best use of limited rotor torque by closely matching the expected joint torques of running.
- HypoSEA-v2 is light enough to use in a robot.
- Bigger motor drivers would improve peak torques.

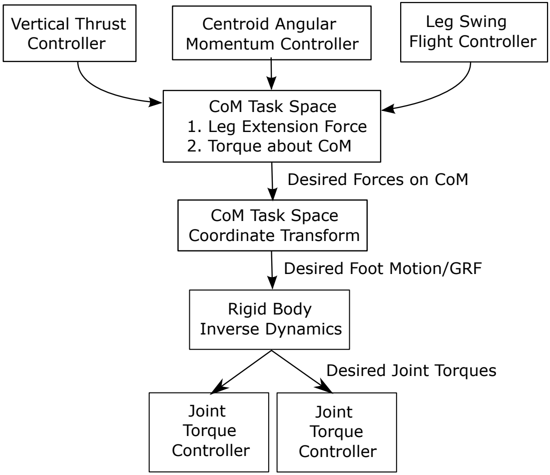
Let's now turn to JIMI's balancing controller, which sends signals to the joint torque controllers.

Part III: Dynamic Balancing

Goal: To describe a centroidal task-space controller that creates a sinusoidal vertical GRF and stabilizes the centroidal angular momentum.

Overview:

1. Model of JIMI
2. Inverse Dynamics
3. Centroid Task Space
4. Dynamic Balance Controllers
5. Simulation Results



JIMI: A Monopod Runner

Equations of Motion

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\dot{\mathbf{q}}, \mathbf{q})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \mathbf{D}_j \tau_j + \mathbf{J}_f \lambda_f$$

Definitions

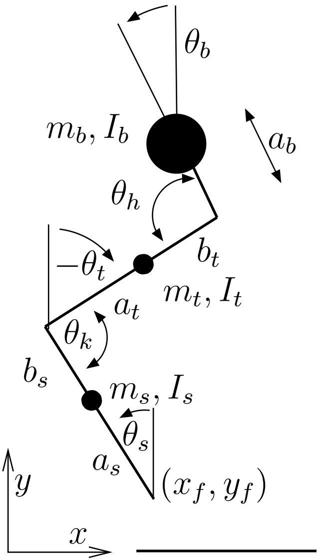
$$\mathbf{q} = \begin{bmatrix} \theta_s & \theta_t & \theta_b & x_f & y_f \end{bmatrix}^T$$

$$\tau_j = \begin{bmatrix} \tau_k & \tau_h \end{bmatrix}^T$$

$$\lambda_f = \begin{bmatrix} F_{fx} & F_{fy} \end{bmatrix}^T$$

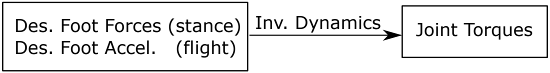
$$\mathbf{J}_f = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}^T$$

$$\mathbf{D}_j = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \end{bmatrix}^T$$



Inverse Dynamics GRF Control

Goal: To solve for joint torques that give the desired GRF during stance, and the desired foot acceleration during flight.



Stance Inverse Dynamics (Des. Foot GRFs: $\lambda_f = \lambda_{fd}$)

$$\begin{bmatrix} \mathbf{M} & \mathbf{J}_f^T & -\mathbf{D}_j^T \\ \mathbf{J}_f & 0 & 0 \\ 0 & \mathbf{I} & 0 \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ -\lambda_f \\ \tau_{jd} \end{bmatrix} = \begin{bmatrix} \tau_v - \mathbf{C}\dot{\mathbf{q}} - \mathbf{g} \\ -\dot{\mathbf{J}}_f \dot{\mathbf{q}} \\ -\lambda_{fd} \end{bmatrix}$$

Flight Inverse Dynamics (Des. Foot Motion: $\mathbf{J}_f \ddot{\mathbf{q}} = \ddot{\mathbf{q}}_{fd}$)

$$\begin{bmatrix} \mathbf{M} & -\mathbf{D}^T \\ \mathbf{J}_f & 0 \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ \tau_{jd} \end{bmatrix} = \begin{bmatrix} \tau_v - \mathbf{C}\dot{\mathbf{q}} - \mathbf{g} \\ \ddot{\mathbf{q}}_{fd} \end{bmatrix}$$

Centroid Task Space

Use horizontal GRF to control centroidal torque:

Centroidal Torque to Horiz. GRF

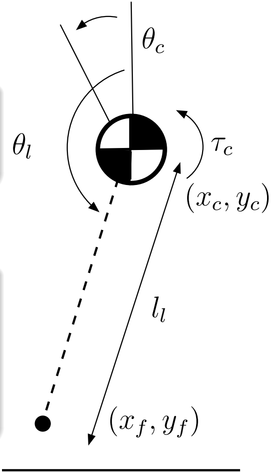
$$F_{fx}(F_{fy}, \tau_c) = \frac{(x_c - x_f)F_{fy} - \tau_c}{(y_c - y_f)}$$

Express controllers in CoM-Foot polar coords:

Polar Coordinates

$$\theta_l = \tan^{-1} \frac{x_c - x_f}{y_c - y_f}$$

$$l_l = \sqrt{(x_c - x_f)^2 + (y_c - y_f)^2}$$



3-Part Dynamic Balancing Controller

Sinusoidal Vertical GRF

$$y_R(t) = \hat{y}_{cTD} + \frac{h_R t}{T_S}$$

$$F_{fy}(t) = K_{cR}(y_R - \hat{y}_c)$$

Centroidal Angular Momentum

$$\theta_c = K_{lb}\theta_b - \theta_l + \theta_{b0}$$

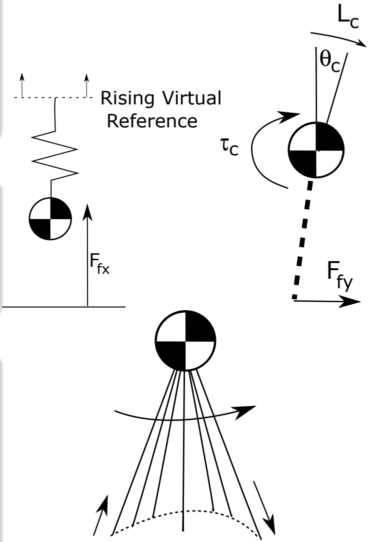
$$\tau_c = K_{cP}\theta_c + K_{cD}L_c$$

Leg Swing and CoP Control

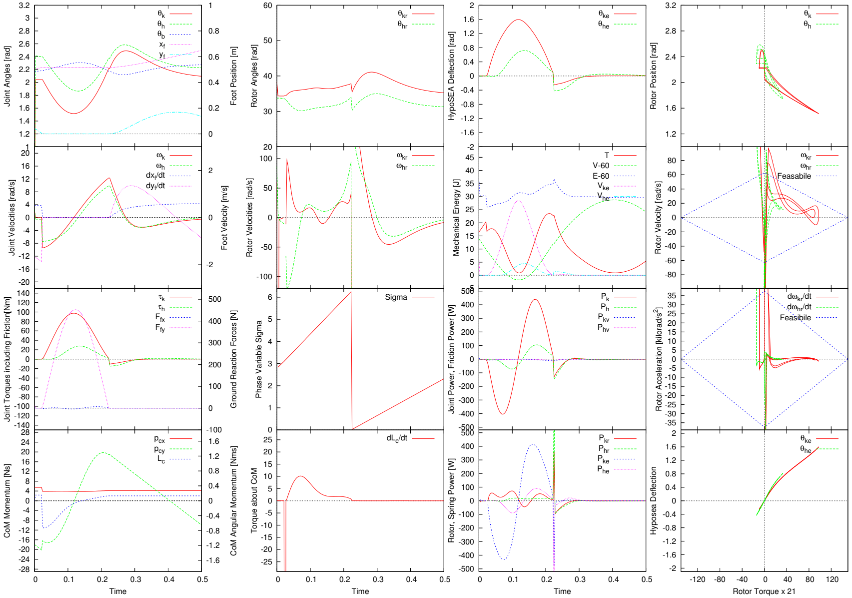
$$\sigma = \pi(t - t^{LO})/T^F + \frac{\pi}{2}$$

$$\theta_{ld}(\sigma) = K_1 \sin(\sigma + \frac{\pi}{2}) + K_2$$

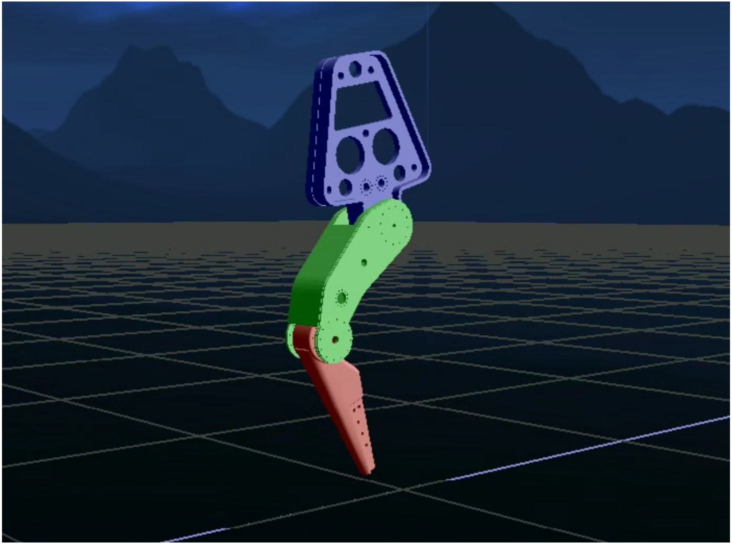
$$l_{ld}(\sigma) = K_3(\sin(\sigma) + \sin(2\sigma)) + K_4$$



Simulation Results



Old Simulation Video



Summary of Dynamic Balancing Controller

- JIMI, HypoSEA, and controller dynamics were studied in simulation *during the design process*.
- Three rules expressed in centroid task space stabilize the robot.
- Rotor work was minimized by matching passive mechanical dynamics and controller torques.

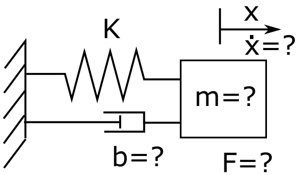
Next: How can we estimate state and model parameters for the above model-based control?

Part IV: State and Model Estimation

Goal: To describe how the state and model parameters of the JIMI were estimated using model-based least squares regression with power constraints.

1. Example: Numerical Differentiation
2. Model-based Estimation
3. HypoSEA State Observer
4. JIMI State Observer

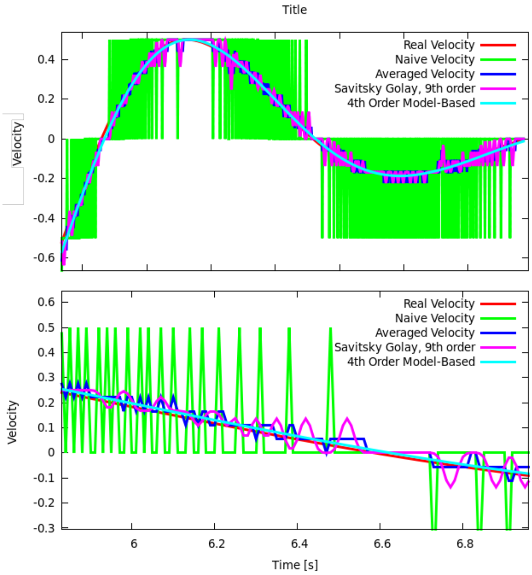
Numerical Derivatives of Noisy Data



Ways to differentiate:

- Real value
- Finite differences
- Averaging/LF pass
- Polynomial Regression
- Model-based

If models can improve *control*, models can improve *estimation*!



Model-Based Estimation

Linear ODE Model

$$\dot{x} = Ax + Bu + \delta$$

$$y = Cx + \epsilon$$

States x, observations y, input u, noise δ and ϵ.

- For realtime control, estimators must be causal – Kalman Filter and its variants work great.
- But for smoothed past values, central differences better.
- JIMI uses model-based fourth-order central-difference weighted least squares.

Kalman Filter, Unrolled, Variance Weights Hidden

$$\begin{bmatrix} C & & \\ -dt\,(I + A) & I & \\ & C & \\ & -dt(I + A) & I \\ & & C \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} y_0 \\ Bu_0 \\ y_1 \\ Bu_1 \\ y_2 \end{bmatrix}$$

Kalman Filters with Fatter Bands

Kalman Filter Differences Matrix

$$\begin{bmatrix} \dot{x}_0 \\ \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \vdots \end{bmatrix} = \frac{1}{h} \begin{bmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ \vdots \end{bmatrix}$$

4th Order Discrete Differences Matrix

$$\begin{bmatrix} \dot{x}_0 \\ \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \vdots \end{bmatrix} = \frac{1}{12h} \begin{bmatrix} 0 & 8 & -1 & & \\ -8 & 0 & 8 & \ddots & \\ 1 & -8 & \ddots & 8 & -1 \\ & \ddots & -8 & 0 & 8 \\ & & 1 & -8 & 0 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ \vdots \end{bmatrix}$$

JIMI's Model-Based Fixed-Lag Smoothing

4th Order Discrete Differences Matrix

$$\begin{bmatrix} \mathrm{diag}(C, C, \cdots) \\ \mathrm{diag}\begin{bmatrix} -\frac{1}{12}I & \frac{2}{3}I & -dtA & -\frac{2}{3}I & \frac{1}{12}I \end{bmatrix} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ \vdots \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ \vdots \\ Bu \end{bmatrix}$$

Model Parameter Estimation

Problem: How can we estimate model parameters $\dot{x} = Ax + Bu$?

MODEL PARAMETER ESTIMATION

Problem: How can we estimate model parameters $\dot{x} = Ax + Bu$?

Observation: If your model is good, excluding external disturbances, the power in and out of the system will be zero from state to state.

Model Parameter Estimation

Problem: How can we estimate model parameters $\dot{x} = Ax + Bu$?

Observation: If your model is good, excluding external disturbances, the power in and out of the system will be zero from state to state.

Solution: Least squares parameter estimation minimizing power error.

- Uses energy as a lingua franca between different physical parameters.
- Expand terms of A, B matrices; use to write power balance equation.

$$\begin{bmatrix} \dot{x}_0 \\ \dot{x}_1 \\ \vdots \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} \phi_{11} & \phi_{12} & \cdots & \phi_{1n} \\ \phi_{21} & \phi_{22} & & \\ \vdots & & \ddots & \\ \phi_{n1} & & & \phi_{nn} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} \gamma_{11} & \gamma_{12} & \cdots & \gamma_{1n} \\ \gamma_{21} & \gamma_{22} & & \\ \vdots & & \ddots & \\ \gamma_{n1} & & & \gamma_{nn} \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_n \end{bmatrix}$$

Let $\phi = \begin{bmatrix} \phi_1 & \phi_2 & \cdots & \gamma_1 & \gamma_2 & \cdots \end{bmatrix}^T$ and $P_{err}(x, \dot{x}, u) = U\phi$, and Q be weights.

$$\hat{\varphi} = \left(U^T Q U\right)^{-1} U^T Q P_{err}$$

Summary of State Estimation

- HypoSEA real time control uses a Kalman Filter.
- Smoothing is done with a higher order model-based filter.
- From smoothed data, we can iteratively improve $\hat{\phi}$ such that power is conserved.
- JIMI's inertial parameters not yet estimated.

Next: In what manner was this software written?

Part V: Asynchronous, Dataflow Programming

Goal: To present the programming style used to write the control software for JIMI, in the Clojure Language invented by Rich Hickey.

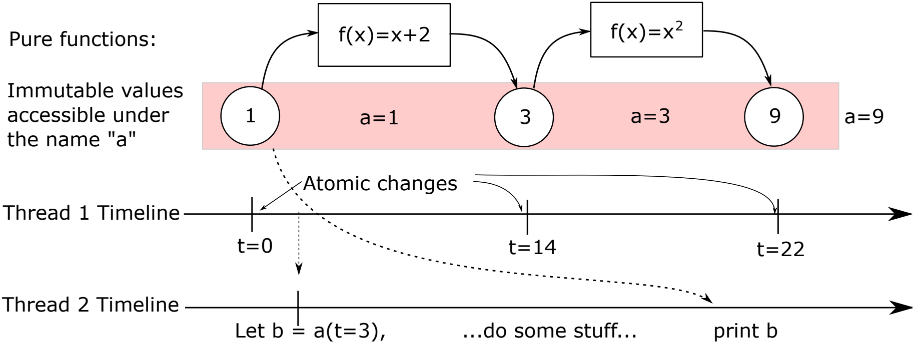
1. Why Another Robotics Software System?
2. Immutable Data and Pure Functions
3. Basics of Dataflow Programming
4. Advantages of Dataflow Programming
5. Screenshots of Developed Software
6. A Short Video

Motivation: To Be Less Irritated

Irritation	Solution
Software licenses	Liberty-based software only
Xenomai + RoboLLI crashing my PC	Ordinary Linux Kernel + JVM
Recompiling on every code change	Dynamic language, JIT compilation
Not using all my CPU cores	Concurrent dataflow model
Inter-process communication barriers	Use many threads in one process
Bugs in one thread stopping others	Contain exceptions to each dataflow
Lack of real-time visualization	DIY oscilloscope, OpenGL viewer
Integrating non-synchronous data streams	Asynchronous, event-based code

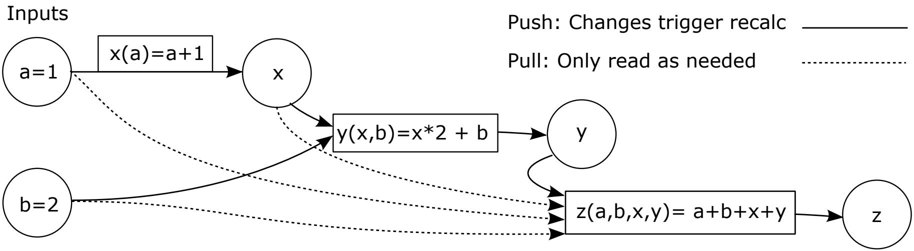
Immutable Data & Pure Functions (*Hickey*)

Use Clojure's epochal model of state, identity, transitions:



- Values are only birthed and GC'd – never modified.
- Names only point to one value at a time.
- Multiple threads can share same data.
- Tree structures can safely reuse old data to reduce copying.

Dataflow Programming



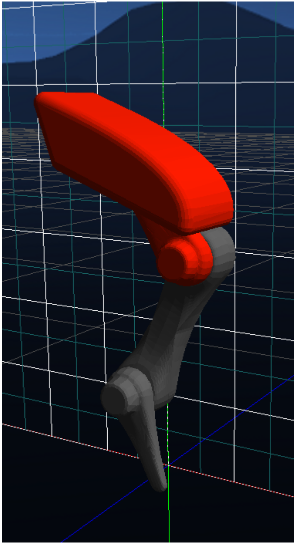
- Data keeps itself updated!
- Always safe to read!
- Bugs isolated to each flow!
- Add new flows anytime!

Dataflow Programming

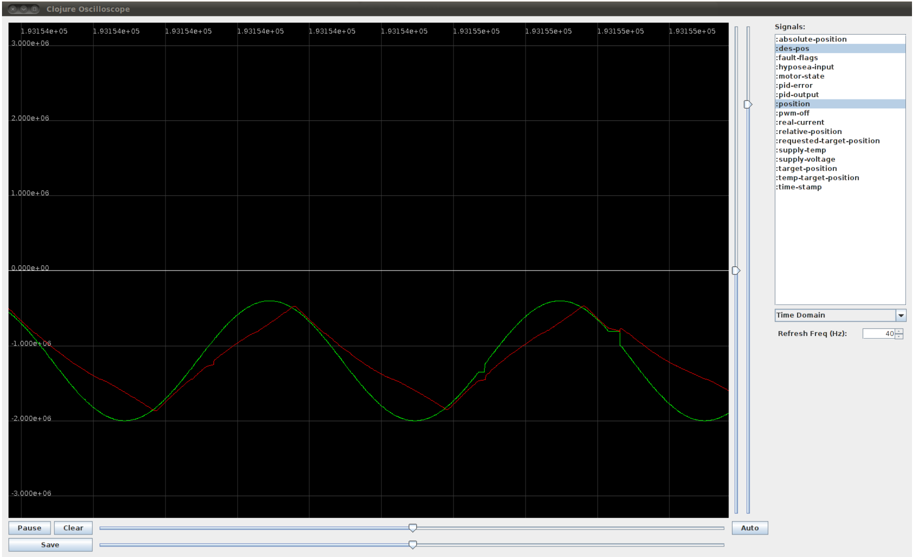
Real applications need a few more details:

- **Triggering** If events occur faster than they can be processed, you can allow skipping of intermediate values.
- **Periodicity** Achieved with scheduler that triggers functions.
- **Coordination** Coordinating several references possible with software transactional memory (but I discourage it).
- **Latency** Long chains of light computations can be forced to use same thread.
- **Need-Based** Really expensive computations can be evaluated lazily, only as needed, and with most recent values.
- **Threading** Queue arguments, execute function in a thread pool.
- **Generality** Useful to separate dependencies and recalculation trigger condition with another function.

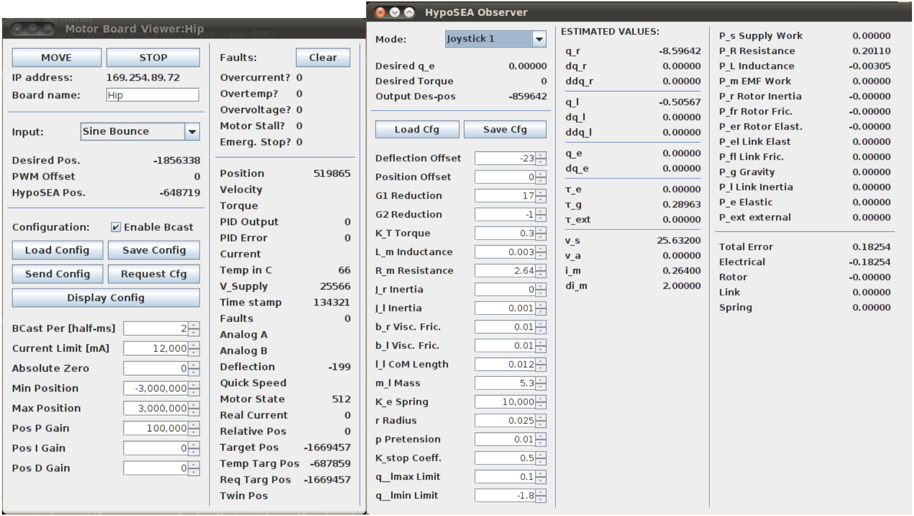
Screenshots: OpenGL Viewport



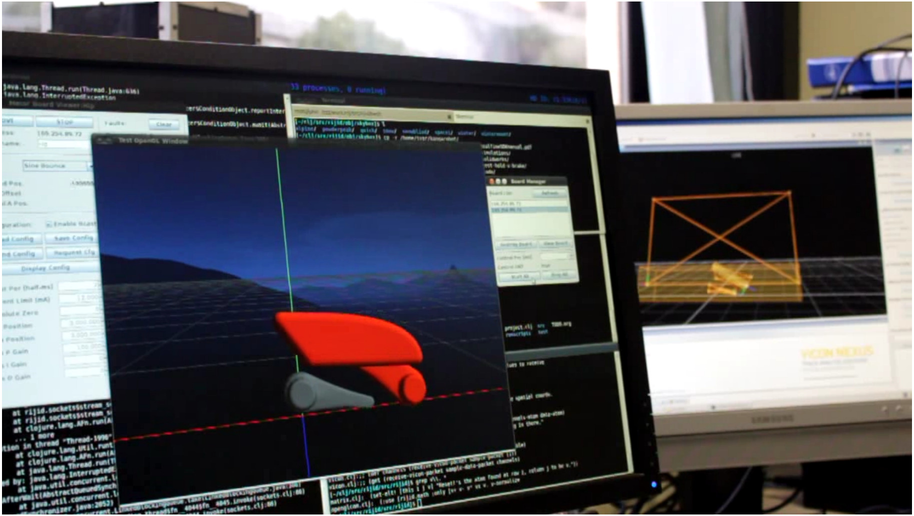
Screenshots: Oscilloscope



Screenshots: Motor Controller GUI



20 Seconds of Video



Summary of Software Architecture

- Clojure is beautiful, lispy, functional, and uniquely immutable.
- Dataflow allows great concurrency and is very simple.
- Latency is pretty good (100uS), would improve if optimized.
- Incremental, realtime GC badly needed to stop erratic 5ms pauses.
- Prioritization didn't work well (JVM thread priorities broken).

The latter two problems would probably be solved by a realtime JVM.

Part VI: Advanced Composite Materials

Goal: To present the monocoque Carbon Fiber Reinforced Polymer (CFRP) construction techniques used to make JIMI, and show they are accessible to researchers at IIT.

Overview:

1. Monocoque structures
2. Composite Sandwich Structures
3. Composite Layup Techniques
4. Construction Photos

Monocoque ("single shell") Structures

A very lightweight way to create stiff, load-carrying skins with complex shapes.

Benefits:

- Extremely light, strong, and stiff
- One molded part can replace several interconnected parts

Disadvantages:

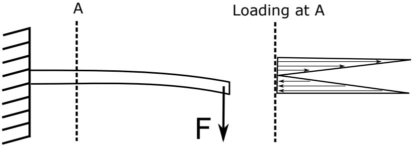
- Generally not machinable, threadable without metal embedments
- Requires time-consuming mold-making
- Very anisotropic strength properties
- Hard to mass-produce

Internal truss structure:

Monocoque CFRP structure:

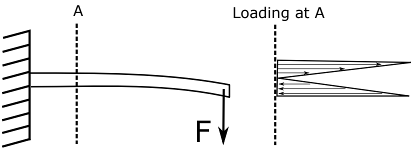
Composite Sandwich Structures

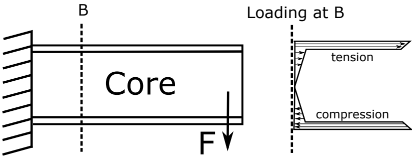
Problem: How can a thin skin carry a load without being too flexible?



Composite Sandwich Structures

Problem: How can a thin skin carry a load without being too flexible?



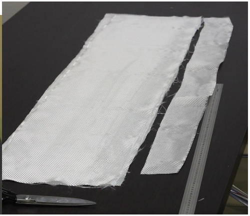


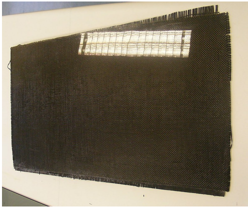
Solution: Make a sandwich structure.

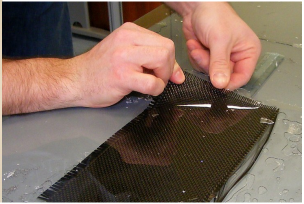
- Core materials have only low shear load
- Low density cores add almost no weight
- Effective stiffness, strength increased
- Balsa wood, plastic foams, aramid or metal honeycombs common in aircraft

Basic Wet Layup

- Essentially just "painting strong fibers with plastic glue".
- Simple, requires few tools, but makes heavier parts with bad finishes

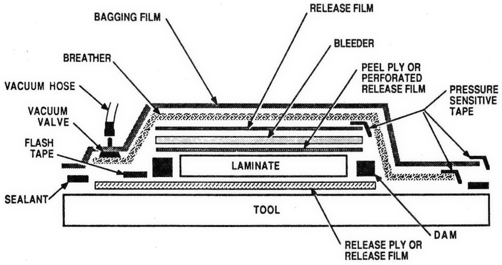




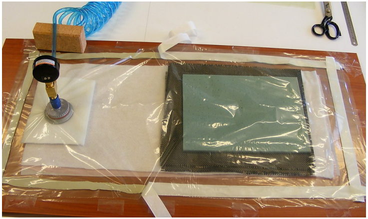


Vacuum Bag Layup

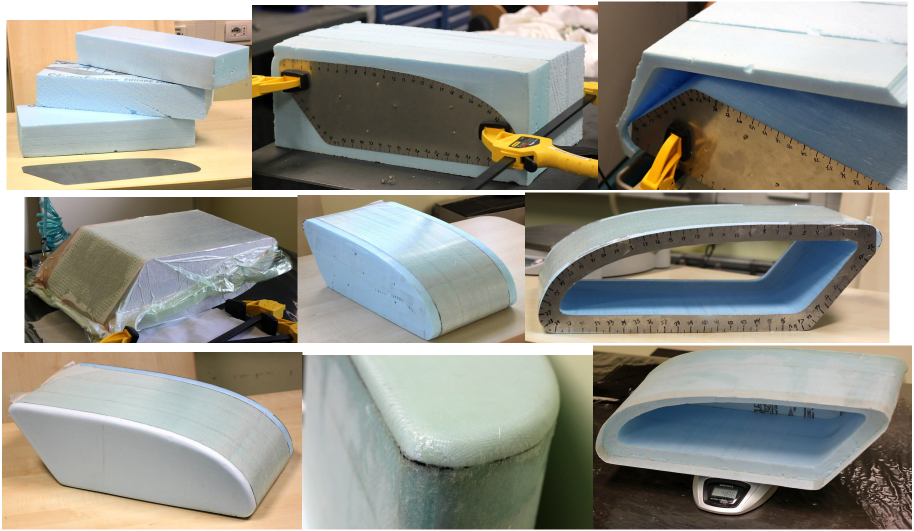
- Use atmospheric pressure to squeeze out unneeded resin, compress fibers
- Accessible technique for amateurs



LAYUP SEQUENCE FOR BAGGING OPERATION



Moldless Composite Parts



Doubly-Molded Composite Sandwich Parts

Summary

- Carbon fiber sandwich structures are uniquely lightweight and stiff.
- JIMI's CFRP structure was constructed entirely at IIT.
- JIMI's three parts took ∼160-200 hours of work.
- Shank mass: 171g
- Thigh mass: 518g
- Body mass: 976g (Moldless construction)

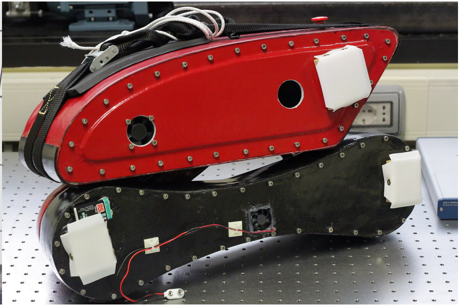
Caution: Please learn safe handling procedures before trying it yourself!

Project Conclusions

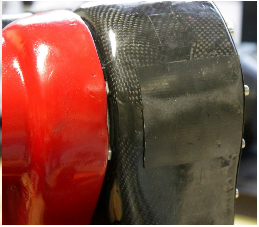
Goal: To show photos and summarize the results in 90 seconds.

1. JIMI Photo Summary
2. Successes and Failures
3. Future Work

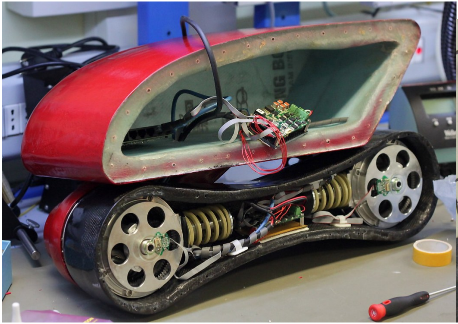
JIMI Photo Summary 1

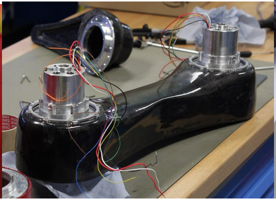


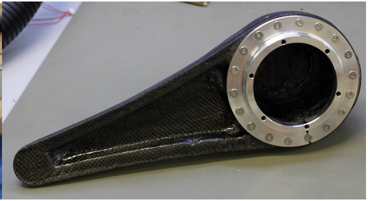




JIMI Photo Summary 2







Successes and Failures

Successes:

- Basic concept works: match actuation, control to dynamics

Failures:

- Irregular vicon latency (TCP) a problem at present
- Joint torques not yet optimal as in simulation

Successes and Failures

Successes:

- Basic concept works: match actuation, control to dynamics
- Asynchronous dataflow control is robust, easy to debug

Failures:

- Irregular vicon latency (TCP) a problem at present
- Joint torques not yet optimal as in simulation
- GRF controller has a singularity

Successes and Failures

Successes:

- Basic concept works: match actuation, control to dynamics
- Asynchronous dataflow control is robust, easy to debug
- CFRP pieces are lightweight

Failures:

- Irregular vicon latency (TCP) a problem at present
- Joint torques not yet optimal as in simulation
- GRF controller has a singularity
- Hard to disassemble JIMI

Successes and Failures

Successes:

- Basic concept works: match actuation, control to dynamics
- Asynchronous dataflow control is robust, easy to debug
- CFRP pieces are lightweight
- HypoSEA-v2 controls force well

Failures:

- Irregular vicon latency (TCP) a problem at present
- Joint torques not yet optimal as in simulation
- GRF controller has a singularity
- Hard to disassemble JIMI
- Need more powerful motor drivers badly

Successes and Failures

Successes:

- Basic concept works: match actuation, control to dynamics
- Asynchronous dataflow control is robust, easy to debug
- CFRP pieces are lightweight
- HypoSEA-v2 controls force well
- Energy regeneration a bonus

Failures:

- Irregular vicon latency (TCP) a problem at present
- Joint torques not yet optimal as in simulation
- GRF controller has a singularity
- Hard to disassemble JIMI
- Need more powerful motor drivers badly

Future Work

There are several research directions that could be pursued from here:

Performance:	How fast/high can JIMI be made to run/jump?
Software:	Clean up, document, and release the dataflow software.
Mechanical:	Can the torso be rebuilt lighter?
Actuation:	Can bigger motor drivers improve HypoSEA performance?
Energetic:	What trajectories maximize energy recovery?
Commercial:	Does the HypoSEA have any economic value?

From now until August, there is only time for me to pursue the first.

Is anybody interested in using JIMI or the HypoSEA-v1 in the future?

Special thanks to

Gianluca Pane
Phil Hudson
Dr. Nikos Tsagarakis
Dr. Darwin Caldwell
and the HyQ Group

Thank you all for your attention.

Questions welcome!