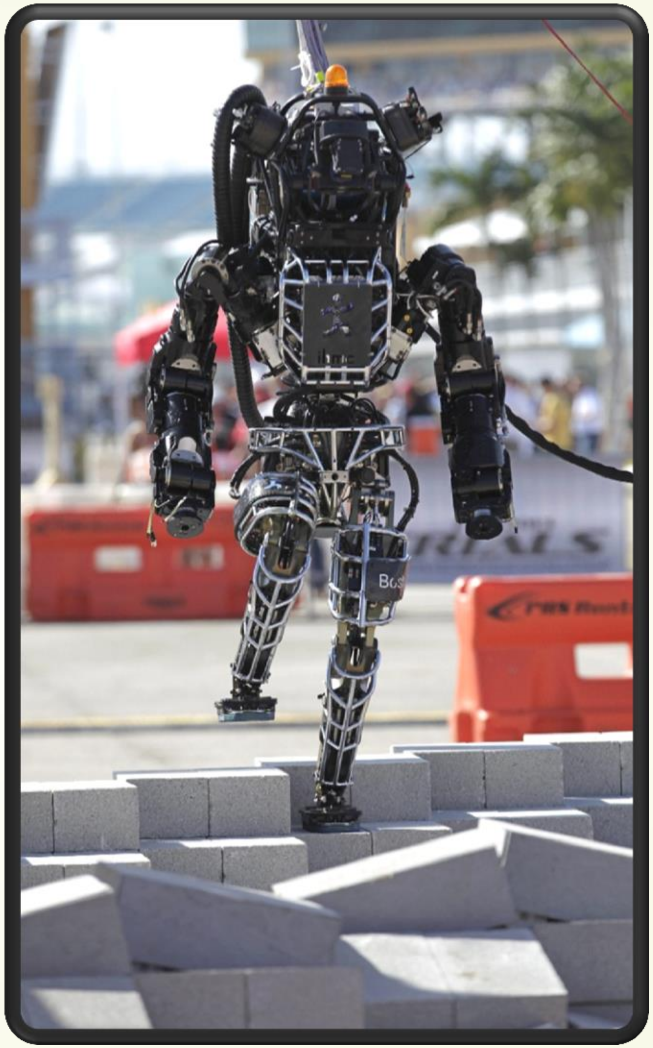
ihmc

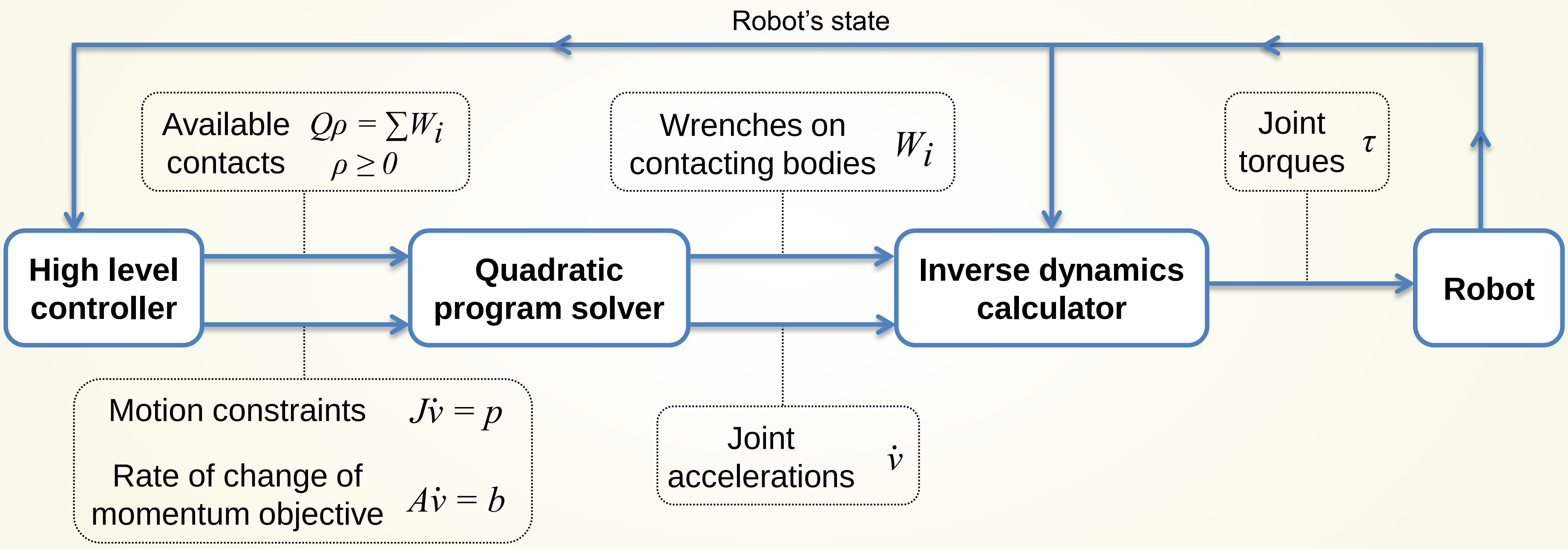
Momentum-based control framework, capturability-based walking control: application to the humanoid robot Atlas

Presented by Tingfan Wu and Sylvain Bertrand


ATLAS

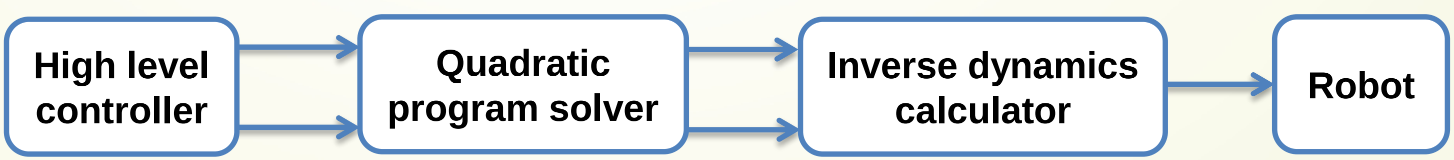


Whole-body motion control framework



Motion Constraints – Instantaneous Goals

- Examples
  - ☺Maintain balance – Keep CoM between feet
  - ☺Move my hand closer to the target
  - ☹Step on rock – instantaneous move!
- Divide long term goal to short ones
  - – Trajectory planner (eg. PID controller, swing planner)



Temporal layers	seconds	166Hz	166Hz	1kHz

Linear Constraints on Joint Acceleration

$$\min_{\dot{v}} \sum w_i \|M_i \dot{v} - p_i\|^2$$

$$s.t \quad M_j \dot{v} = p_j$$

- Connect with desires linearly, e.g.

$$\dot{v}_k = \dot{v_{d,k}}$$

- Change of reference frame via Jacobian

$$Jv = \dot{x_d}$$

$$\dot{J}v + J\dot{v} = \ddot{x}_d$$

$$J\dot{v} = \ddot{x}_d - \dot{J}v$$

Linear Constraints on Joint Acceleration

$$\min_{\dot{v}} \sum w_i \|M_i \dot{v} - p_i\|^2$$

$$s.t \quad M_j \dot{v} = p_j$$

- Connect with desires linearly, e.g.

$$\dot{v}_k = \dot{v_{d,k}}$$

- Change of reference frame via Jacobian

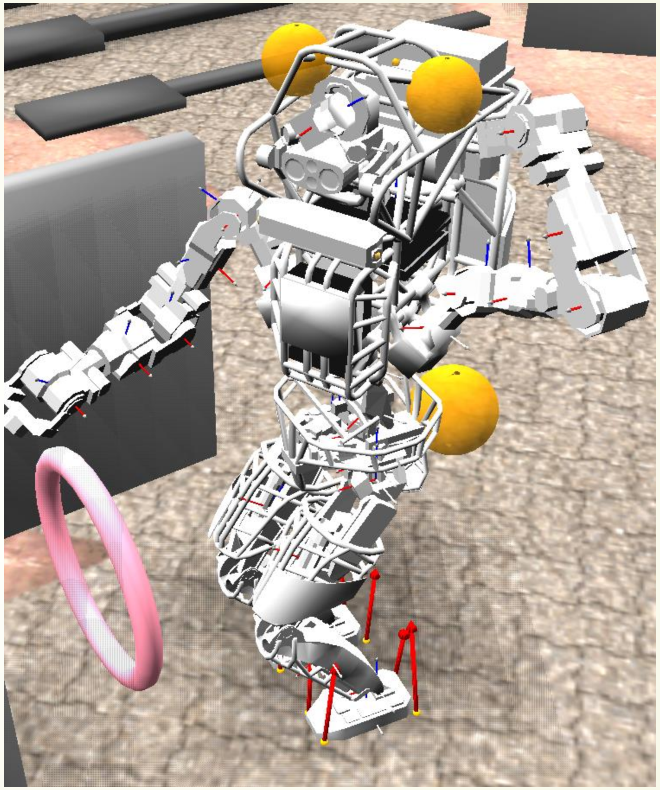
$$J\dot{v} = \ddot{x}_d - \dot{J}v$$

- Link to velocity / position goals with PD controller

$$\dot{v}_k = k_p(x_d - x) + k_d(\dot{x}_d - \dot{x})$$

Constraints / Desires of Atlas

- Chest rotation
  - Joint space goal
- Scratch head
  - Spatial goal in body
- Reach the torus
  - spatial goal in world



Centroidal Momentum

- Single Link- Angular and Linear Momentum

$$h = \begin{bmatrix} k \\ l \end{bmatrix} = \begin{bmatrix} I\omega \\ m\dot{x} \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & m\mathbf{1} \end{bmatrix} \begin{bmatrix} \omega \\ \dot{x} \end{bmatrix}$$

- Multiple Links - Centroidal Momentum
  - sum of link momenta expressed in centroidal frame
  - A **linear** function of joint velocities

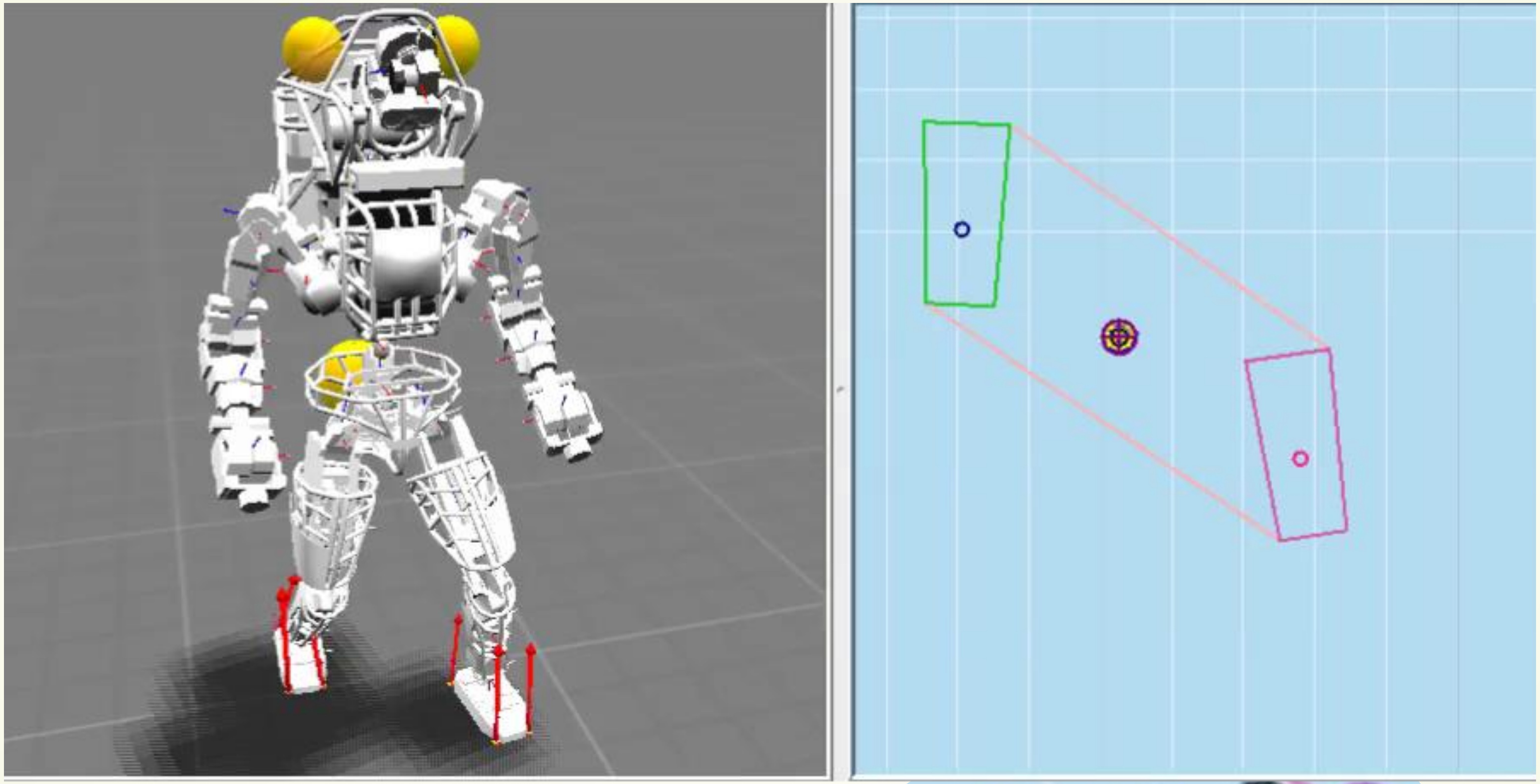
$$h = A(q)v$$

  - Enforce $\dot{h} = \dot{h}_d$ with constraint

$$\dot{A}v + A\dot{v} = \dot{h}_d$$

$$A\dot{v} = \dot{h}_d - \dot{A}v$$

Example of Controlling Centroidal Momentum



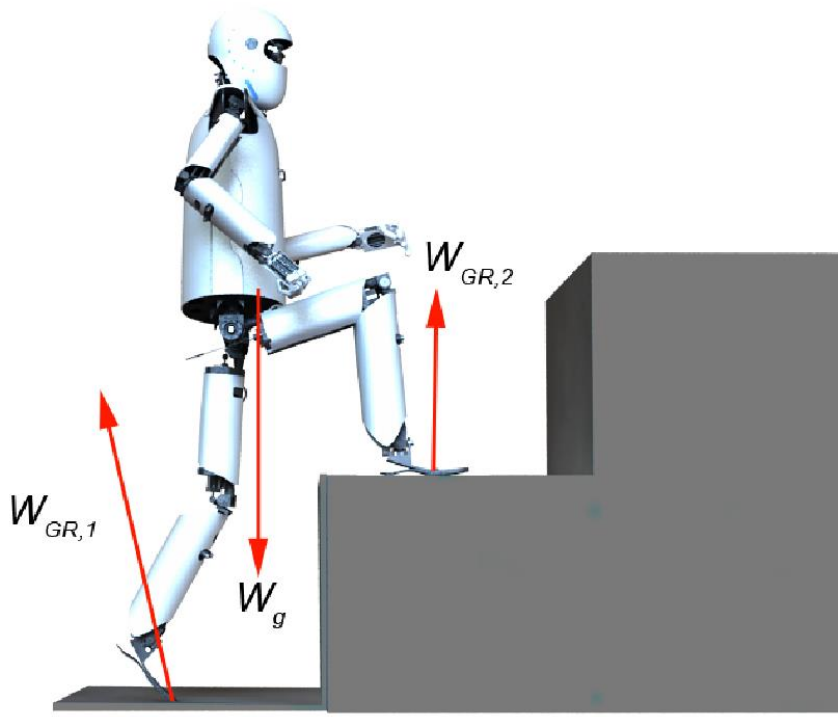
Constraint on External Wrenches

- Newtown/Euler law

$$\dot{h} = \sum external\ wrench$$

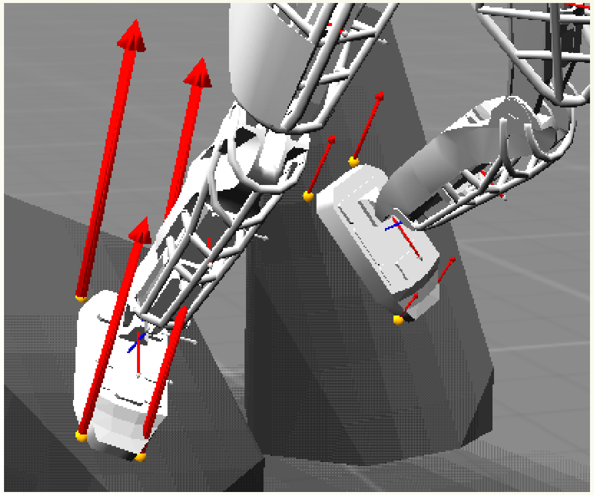
$$\dot{h} = \dot{A}v + A\dot{v} = \sum W_{ext,i}$$

$$A\dot{v} = -\dot{A}v + \sum W_{GR,i} + W_g$$



$$\sum_i W_{\text{ext},i} = \sum_i W_{\text{GR},i} + W_g$$

Constraint on Ground Reaction Wrench



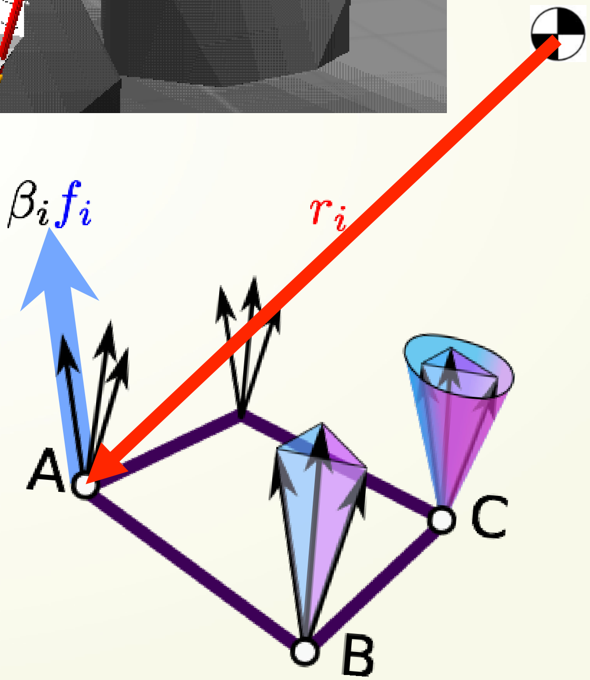
Construct force in pyramid approximation of friction cone [Pollard and Reitsma, 2001]

$$\text{force } f_i = \beta_i \rho_i, \rho_i \geq 0$$

$$\text{torque } \tau_i = r \times f_i$$

$$\text{wrench } w_i = \begin{bmatrix} \beta_i \\ r_i \times \beta_i \end{bmatrix} \rho_i = q_i \rho_i$$

$$w_{GR} = \sum w_i = \sum q_i \rho_i = Q\rho$$



Put It All Together

minimize

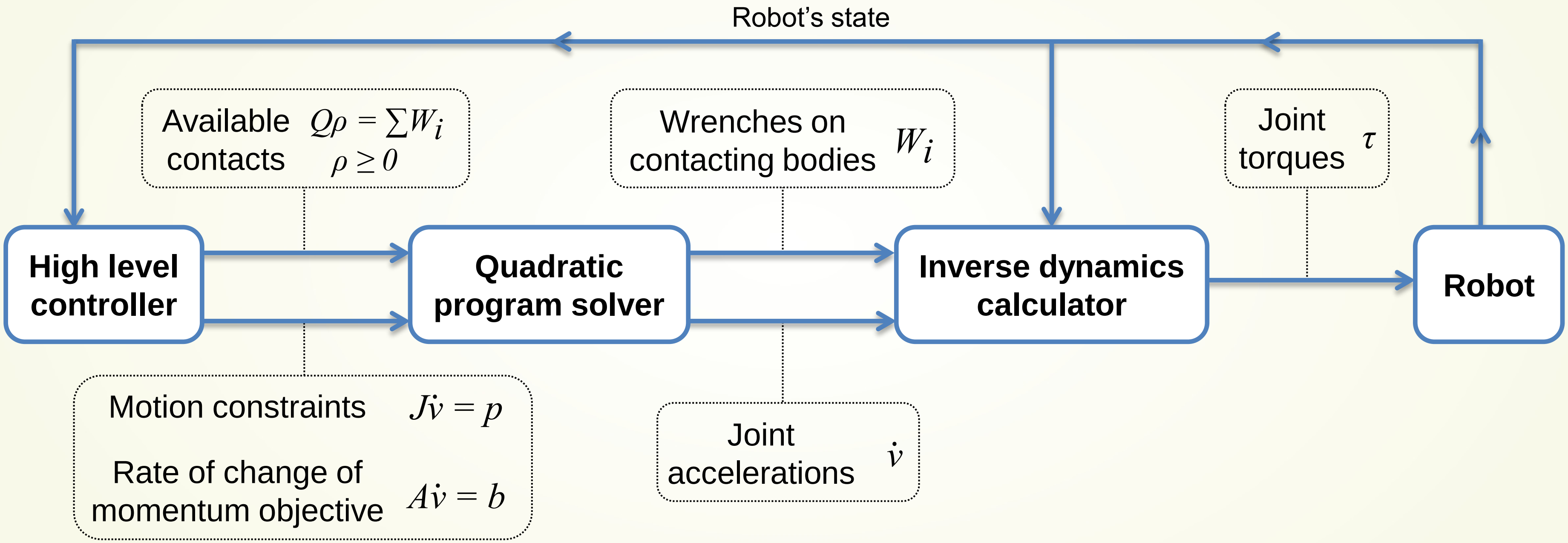
$$\min_{\dot{v},\rho} \sum w_i \|M_i \dot{v} - p_i\|^2 + w_{\dot{v}} \|\dot{v}\|^2 + w_\rho \|\rho\|$$

s.t

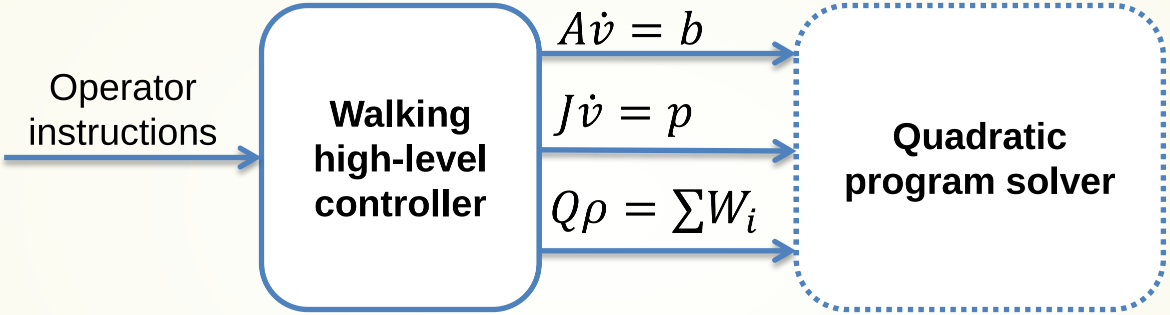
$$\rho \geq 0$$

$$A\dot{v} = -\dot{A}v + W_g + Q\rho$$

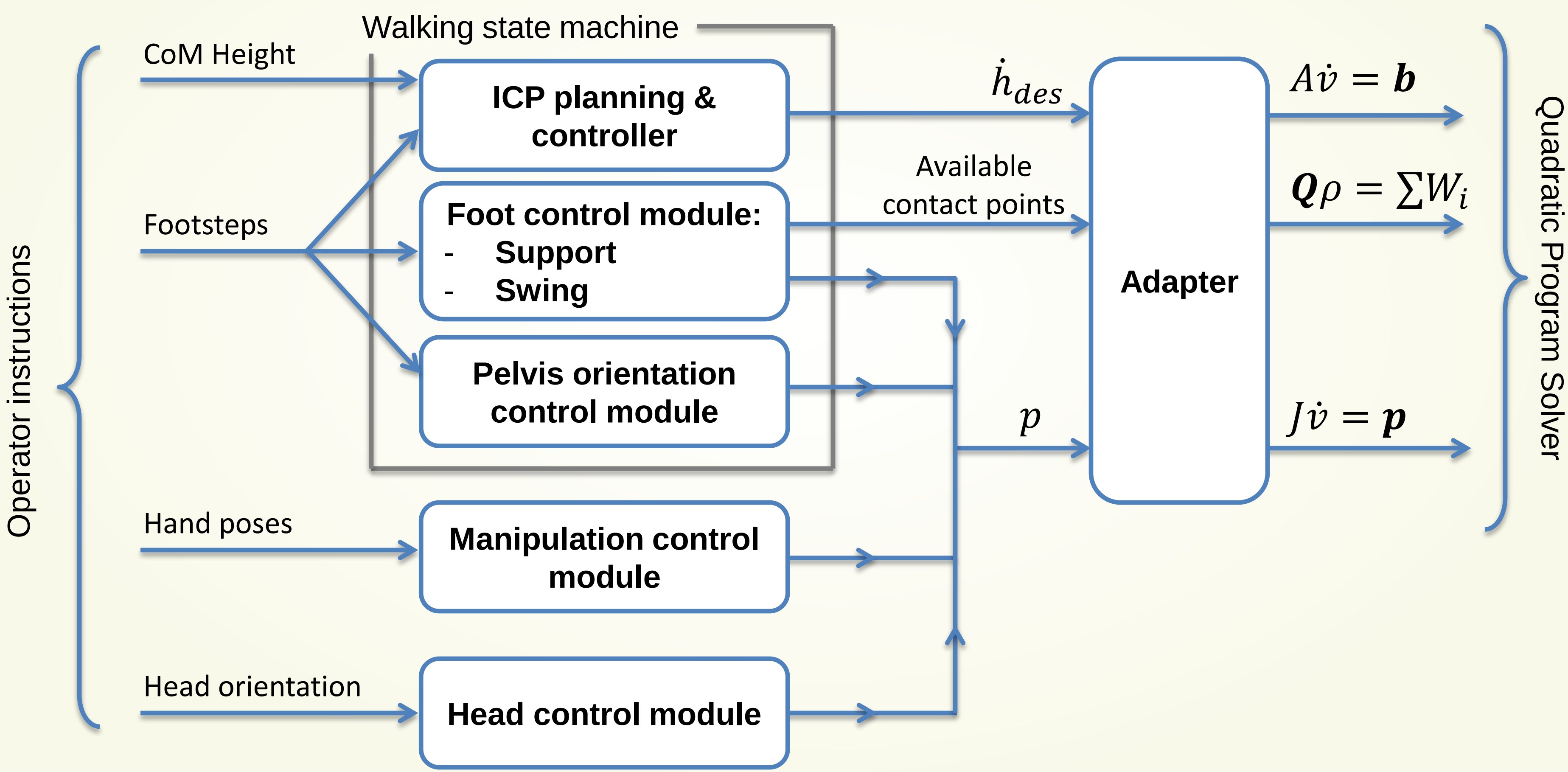
Whole-body motion control framework



Walking high-level controller



Walking high-level controller



Pelvis Orientation Control Module

- Minimum jerk trajectory generator:

$$\theta_{pelvis}^{ref}, \omega_{pelvis}^{ref}, \dot{\omega}_{pelvis}^{ref}$$

- PD controller:

$$\dot{\omega}_{pelvis}^{des} = \dot{\omega}_{pelvis}^{ref} + K_p\left(\theta_{pelvis}^{ref} - \theta_{pelvis}\right) + K_v\left(\omega_{pelvis}^{ref} - \omega_{pelvis}\right)$$

- Motion constraint:

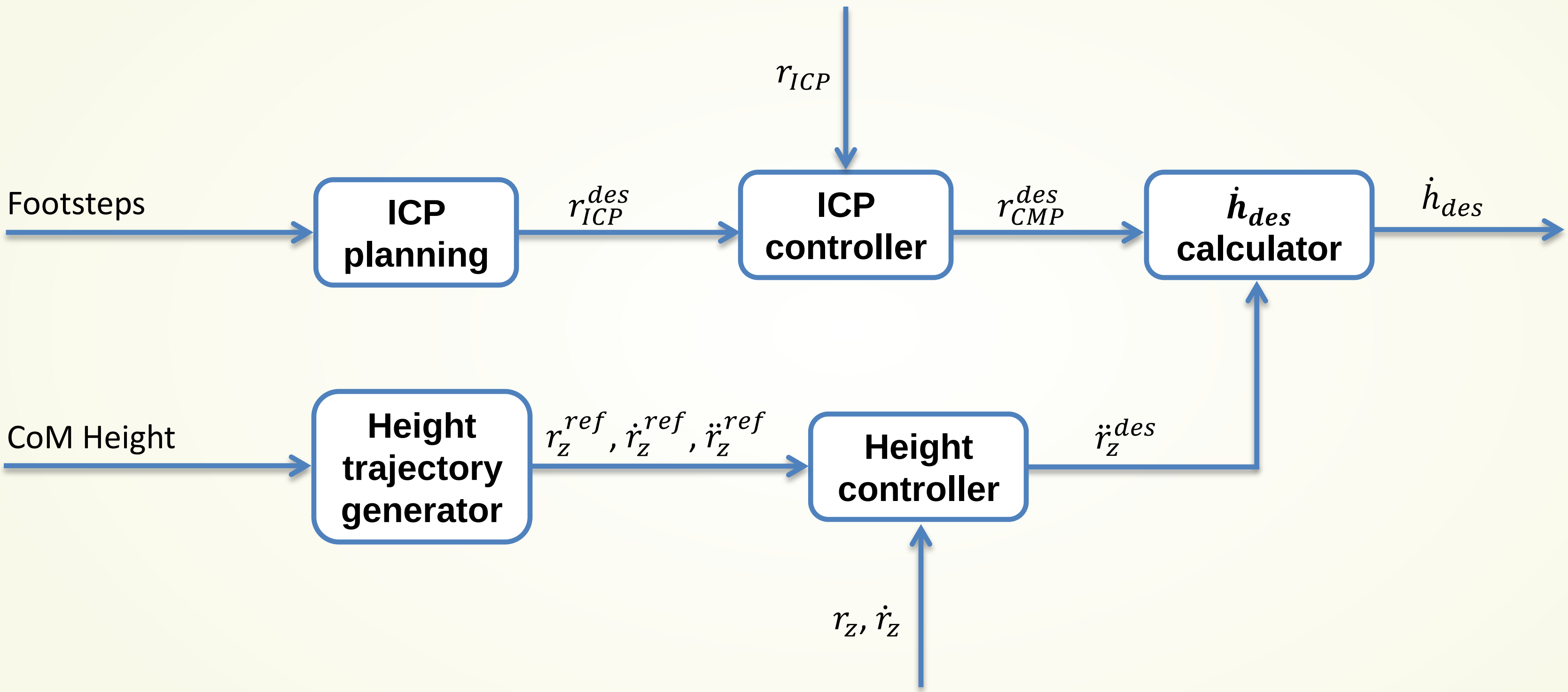
$$SJ\dot{v} = Sp$$

With:

$$p = \begin{bmatrix} \dot{\omega}_{pelvis}^{des} \\ 0_{3\times1} \end{bmatrix} - \dot{J}v$$

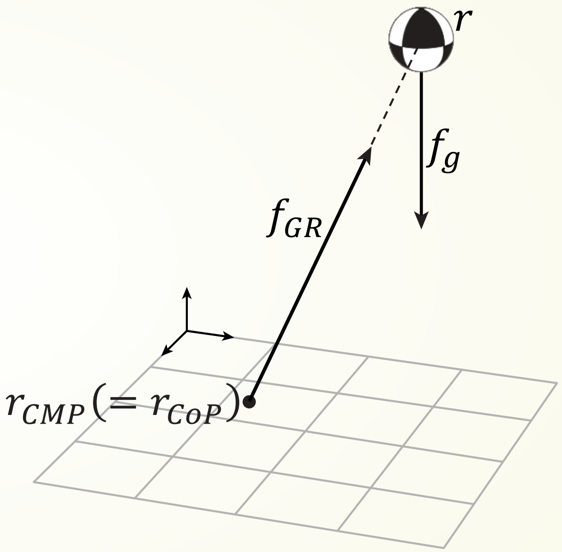
$$S = \begin{pmatrix} I_{3\times3} & 0_{3\times3} \end{pmatrix}$$

ICP Planning & Controller



ICP Overview

- Point mass:
  - $r_{CMP} = r_{CoP}$
  - CoP must be inside support polygon
  - CMP inside the support polygon too



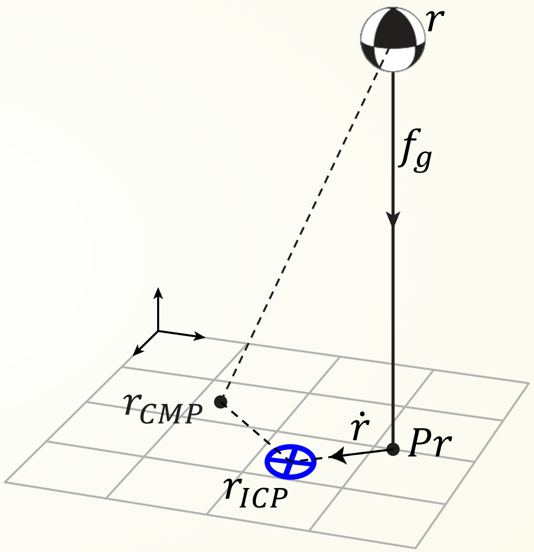
ICP Overview

- Instantaneous Capture Point:

$$r_{ICP} = Pr + \frac{1}{\omega_0}\dot{r}$$

with:

$$\omega_0 = \sqrt{\frac{g}{r_z}}$$



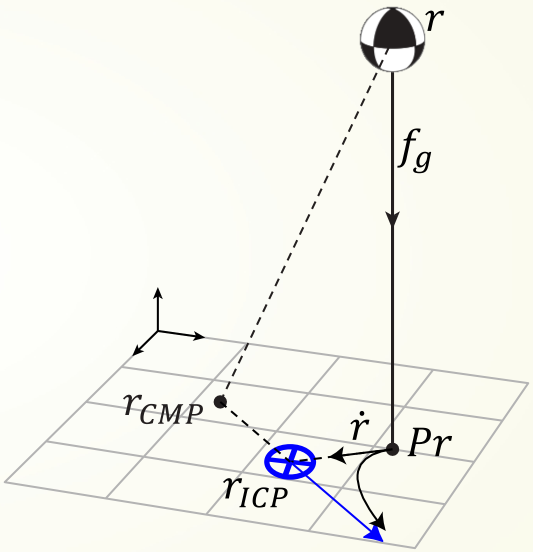
ICP Overview

- Instantaneous Capture Point:

$$r_{ICP} = Pr + \frac{1}{\omega_0}\dot{r}$$

with:

$$\omega_0 = \sqrt{\frac{g}{r_z}}$$

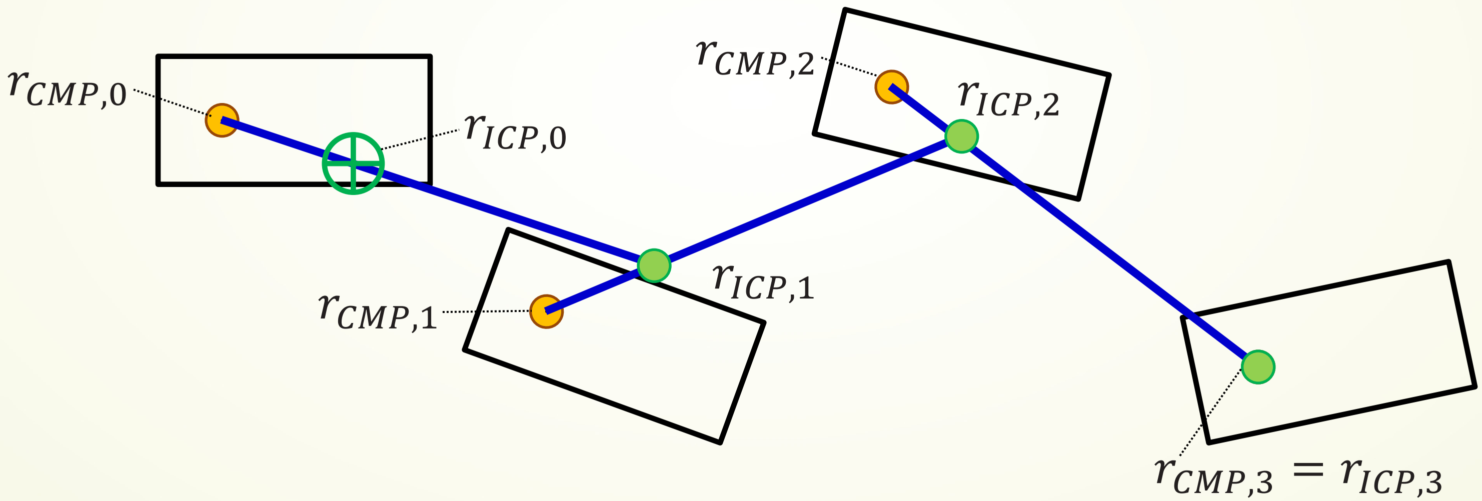


- Dynamics for constant r_z:

$$\dot{r}_{ICP} = \omega_0(r_{ICP} - r_{CMP})$$

ICP Planning (Simpler case)

- Plan r_{CMP} for single supports (shift instant. In double support)
- Find with a recursion r_{ICP} at the beginning of single support
- Generate desired trajectory for the ICP



Englsberger, J. et al. "*Three-dimensional bipedal walking control using Divergent Component of Motion*", *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2013*, 2600-2607

ICP Controller & $\dot{h}_{des}$ Calculation

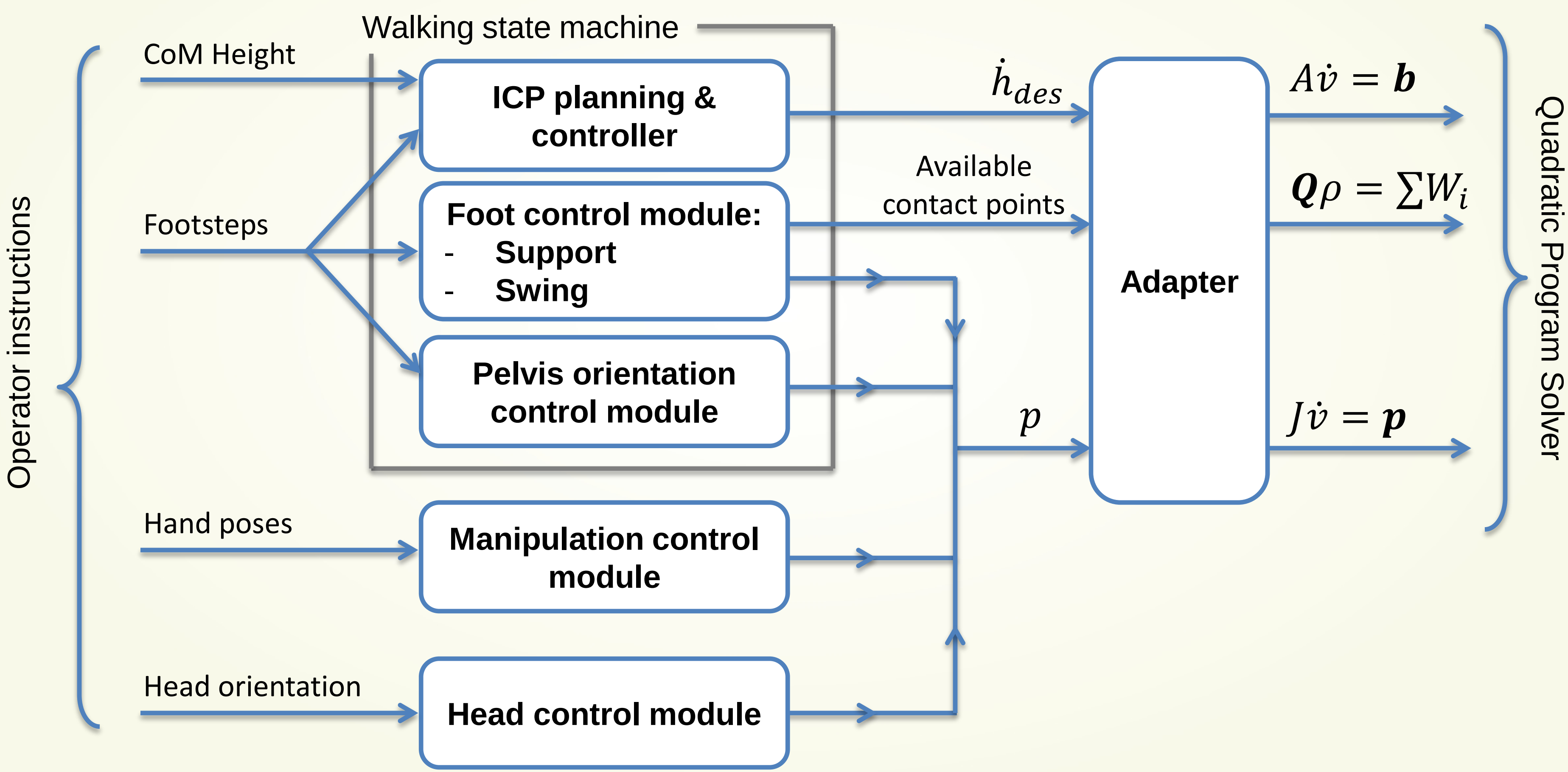
- ICP controller:

$$r_{CMP}^{des} = r_{ICP} + \frac{K_p}{\omega_0}\left(r_{ICP}^{des} - r_{ICP}\right) - \frac{1}{\omega_0}\dot{r}_{ICP}^{des}$$

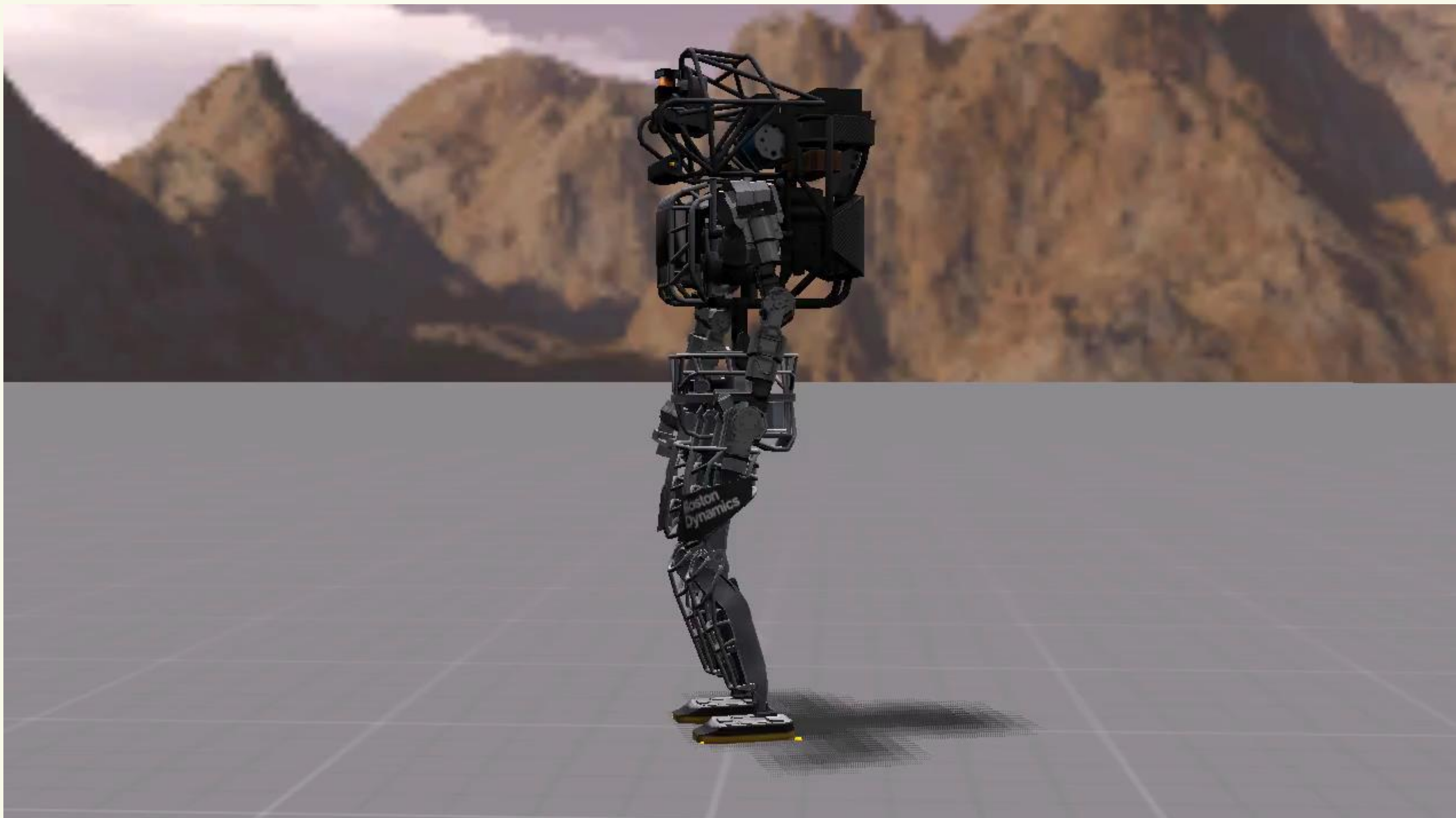
- Compute $\dot{h}_{des}$ ():

$$\dot{h}_{des} = m\begin{bmatrix} \omega_0^2\left(r_x - r_{CMP,x}\right) \\ \omega_0^2\left(r_y - r_{CMP,y}\right) \\ \ddot{r}_z^{des} \end{bmatrix}$$

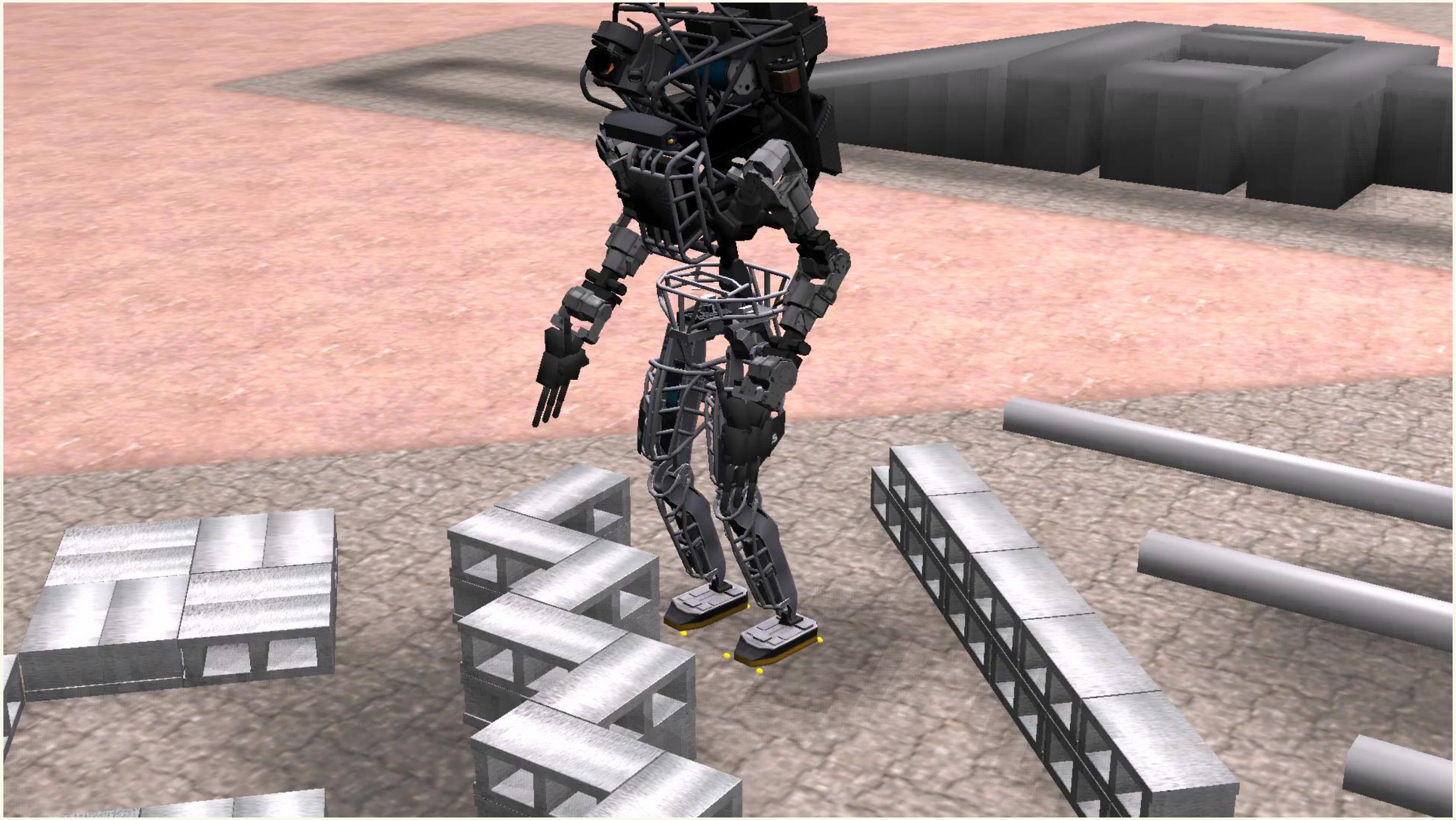
Walking high-level controller

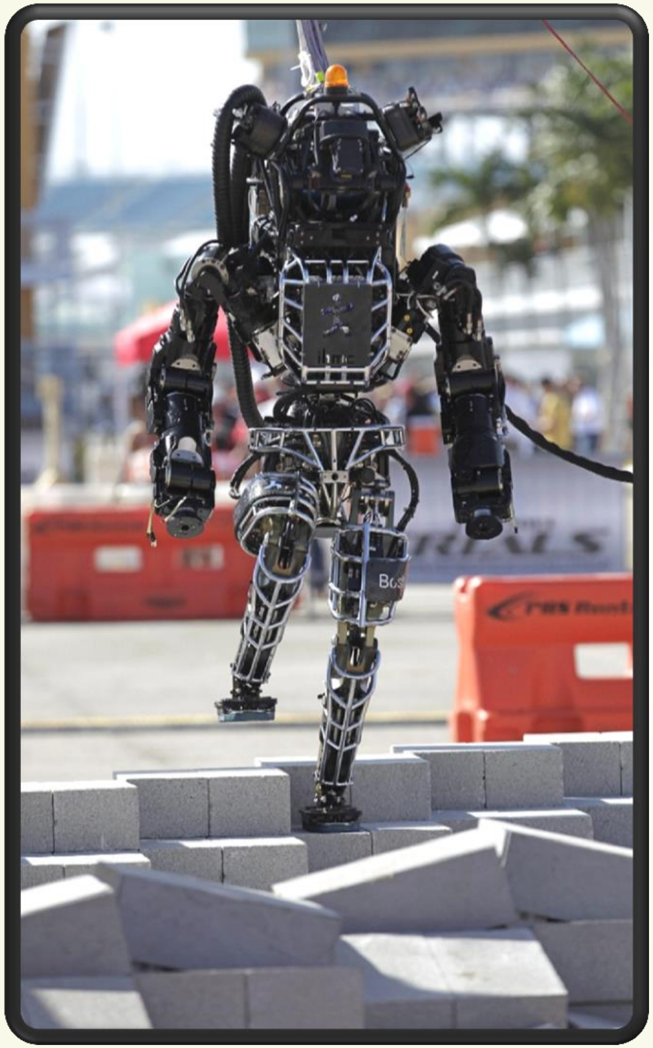


Virtual Atlas Walking



Virtual Atlas Walking

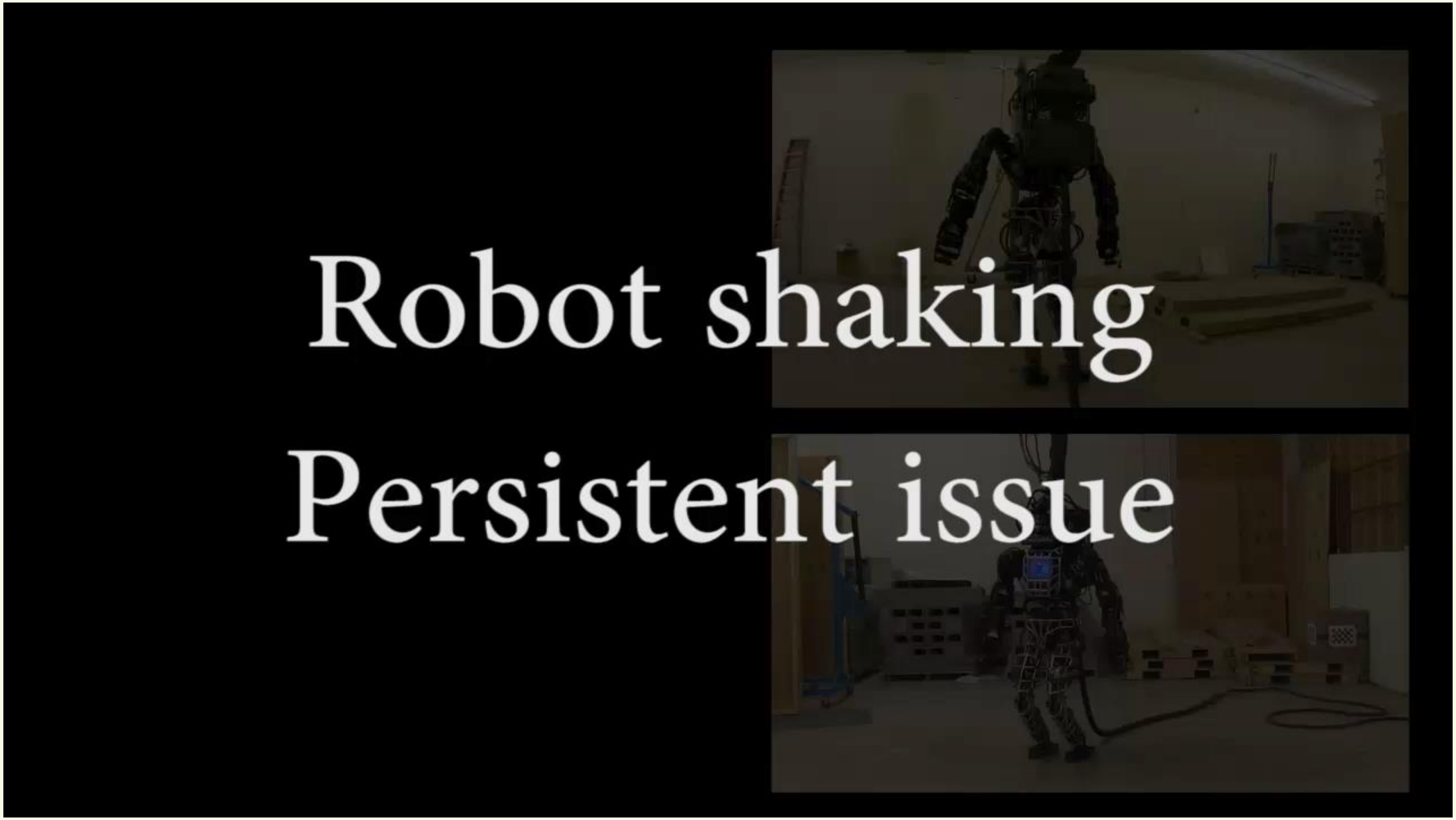




IMPLEMENTATION ON THE REAL ROBOT

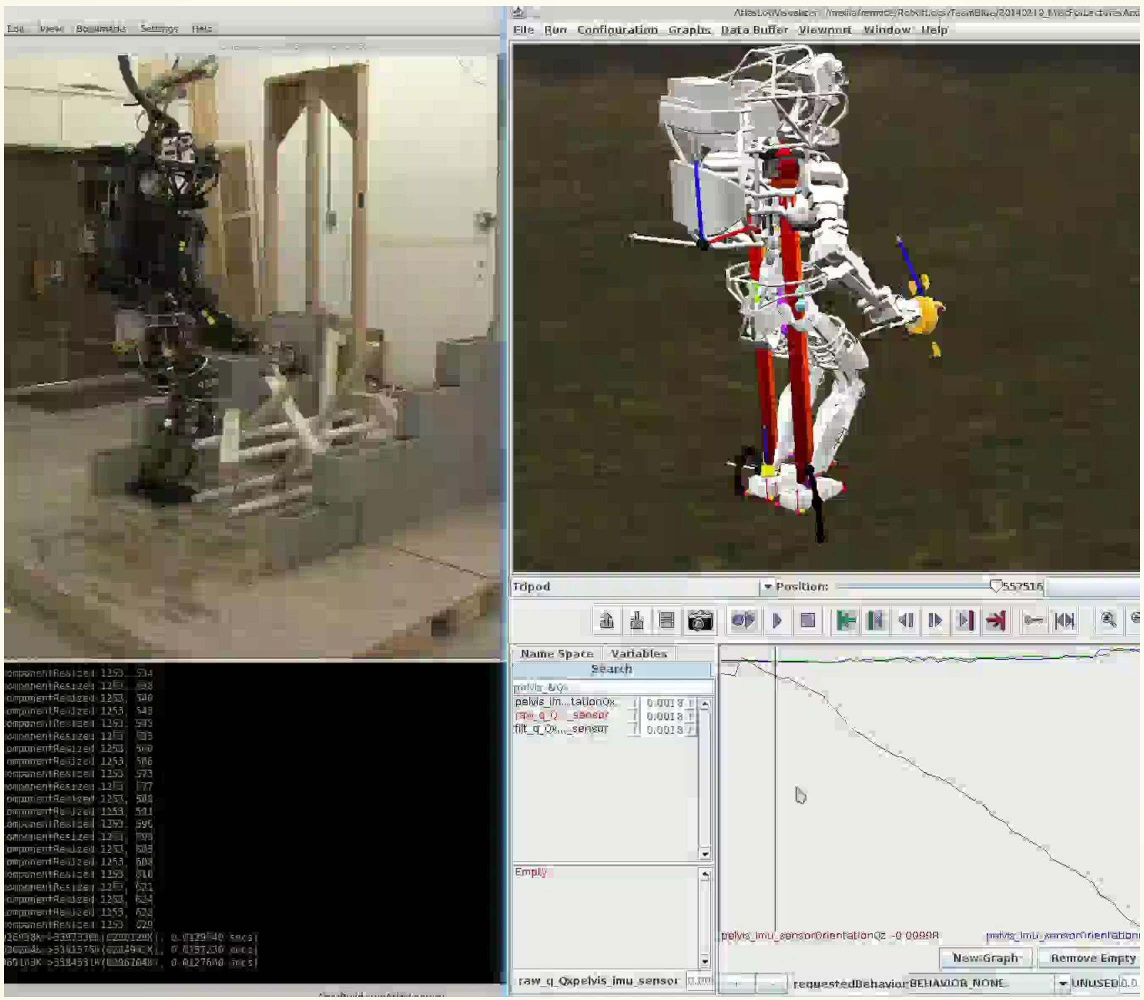
ihmc

Issues: Shakies

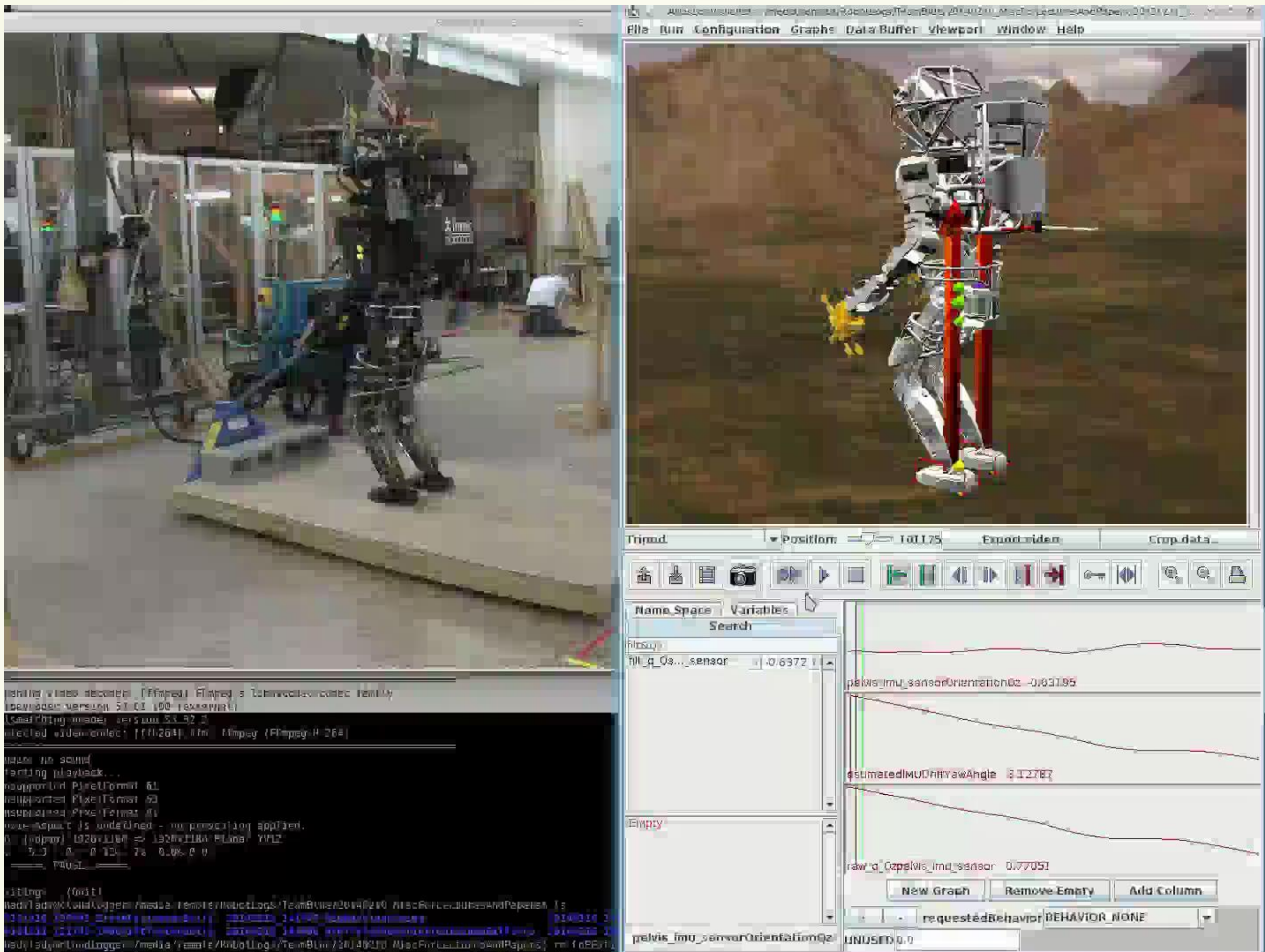


Issues: Foot slipping

Issues: IMU drift

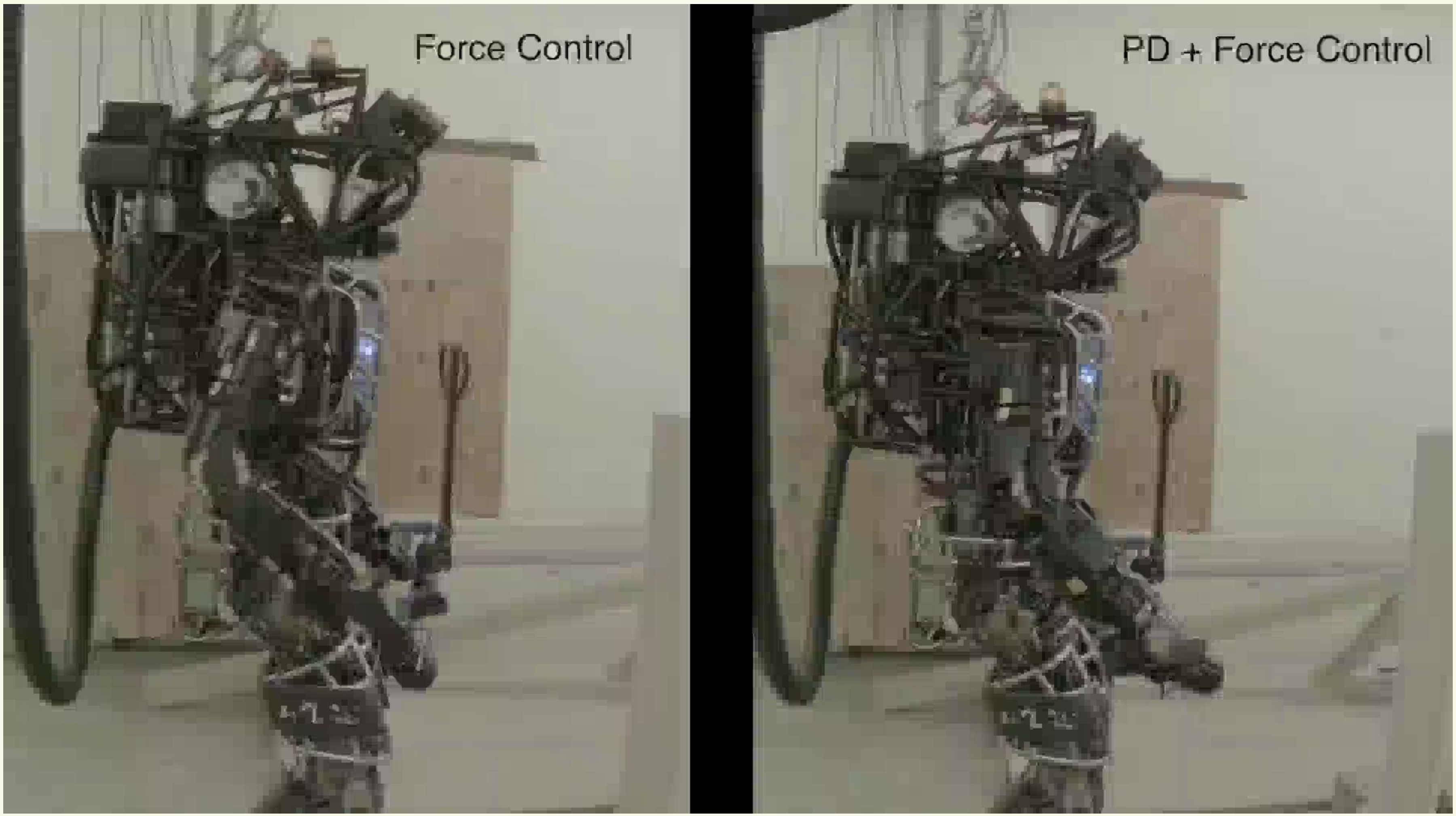


Issues: IMU drift

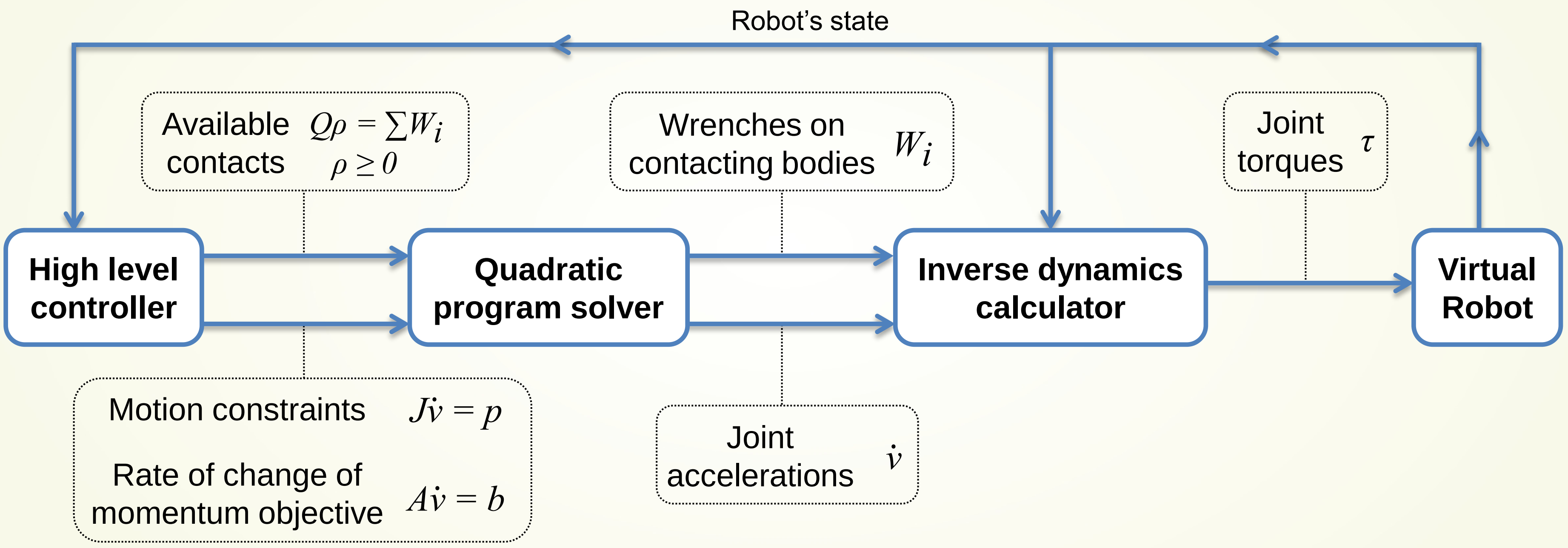


ihmc

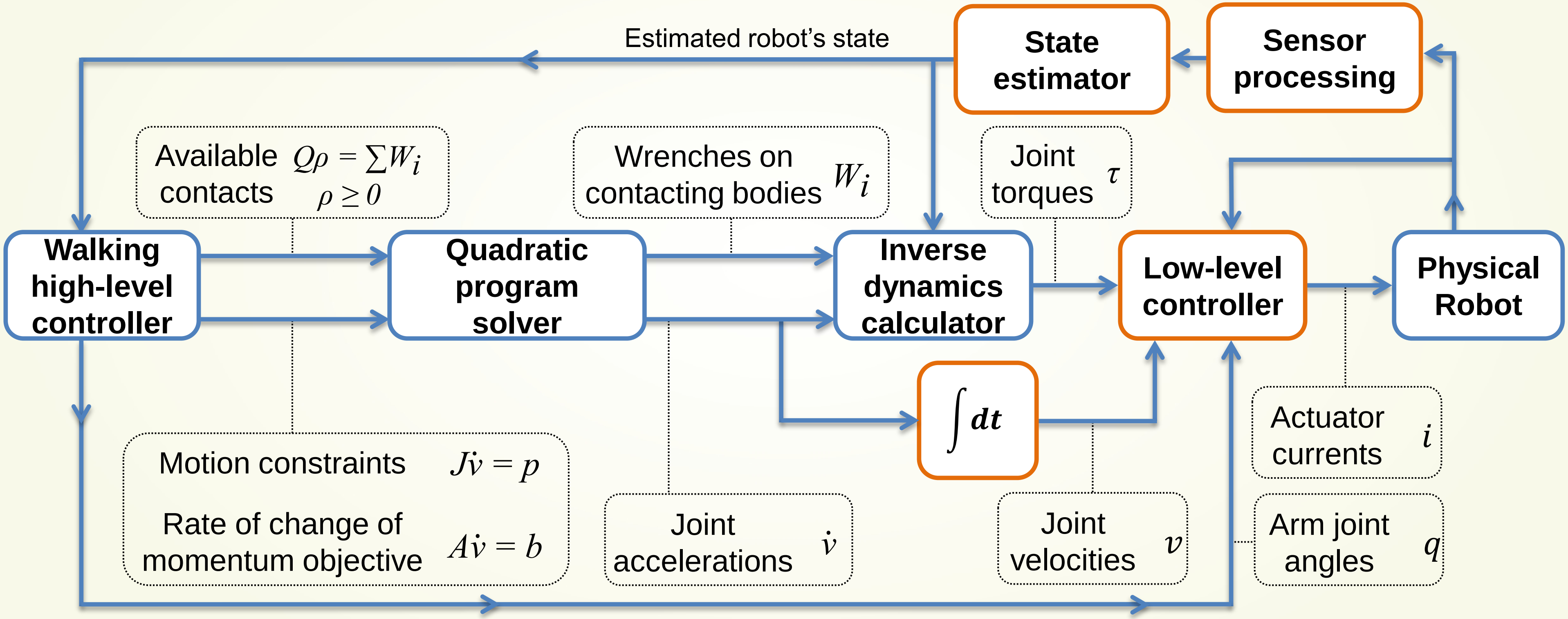
Issues: Joint stiction



Whole-body motion control framework Before



Whole-body motion control framework
After



Sensor processing

- Sensor noise: low-pass filters
- Link elasticity:

$$q_{proc} = q_{raw} + \frac{\tau}{k}$$

- Joint backlash:

$$\dot{q}_{proc}^{t} = \alpha \dot{q}_{proc}^{t-1} + (1-\alpha)\dot{q}_{raw}\eta$$

with: $\eta = \begin{cases} \frac{t-t_0}{\Delta t} & if\ (t-t_0) < \Delta t \\ 1 & else \end{cases}$

$t_0 = t$ when $\dot{q}_{proc}$ crosses zero

State Estimator (IMU drift)

- Built new state estimate for Atlas:
  - Trusts IMU for orientation and angular velocity
  - Fuses data for position and linear velocity:

$$\dot{x}^t = \alpha_{\dot{x}}(\dot{x}^{t-1} + \ddot{x}^t_{IMU}\Delta t) + (1 - \alpha_{\dot{x}})\dot{x}_{kin}$$
$$x^t = \alpha_x(x^{t-1} + \dot{x}^t\Delta t) + (1 - \alpha_x)x_{kin}$$

- Compensator for IMU drift in orientation and angular velocity
- State estimator error:
  - $\sim 1 inch/step$ on the horizontal position
  - $\sim 0.2 inch/step$ along the vertical

Walking controller (chatter/shakies)

- Chatter:
  - Reduced PD controller gains (especially damping)
  - Motion constraints: limited to a maximum acceleration and maximum jerk
- Model inaccuracies:
  - System ID (for the chest mass)
  - Added integral term to the ICP controller:

$$r_{CMP}^{des} = r_{ICP} + \frac{K_p}{\omega_0}\left(r_{ICP}^{des} - r_{ICP}\right) + \frac{K_i}{\omega_0}\int\left(r_{ICP}^{des} - r_{ICP}\right)dt - \frac{1}{\omega_0}\dot{r}_{ICP}^{des}$$

Low-level controller (joint stiction)

- Hybrid controller to fight stiction:

$$i = k_{ff_{qd}}\dot{q} + i_\tau + i_{\dot{q}} + i_q$$

$$\text{with:} \begin{cases} i_\tau = k_{\tau,p}(\tau_d - \tau) + k_{\tau,d}(\dot{\tau}_d - \dot{\tau}) + k_{\tau,i}\int(\tau_d - \tau)dt \\ i_{\dot{q}} = k_{\dot{q}}\left(\int \ddot{q}_d dt - \dot{q}\right) \\ i_q = k_{q,p}(q_d - q) + k_{q,v}(\dot{q}_d - \dot{q}) + k_{q,i}\int(q_d - q)dt \end{cases}$$

  - i_τ: PID controller on joint torque
  - $i_{\dot{q}}$: P controller on joint velocity
  - i_q: PID controller on joint position

- Arm joints: torque + position controller $\left(i = k_{ff_{qd}}\dot{q} + i_\tau + i_q\right)$
- Other joints: torque + velocity controller $\left(i = k_{ff_{qd}}\dot{q} + i_\tau + i_{\dot{q}}\right)$
  - Joint desired velocity obtained from joint desired acceleration (QP solver output)
  - Support ankle joints: only torque controller $\left(i = k_{ff_{qd}}\dot{q} + i_\tau\right)$

Good runs

At the end

- Control
  - Wanted a compliant control ended up with high impedance
  - Introduced lots of “hacks” that needs to be cleaned and improved
- Atlas
  - Gave us hard time
  - Pretty repeatable fitting our testing obsession

Future work

- General improvements
  - Modeling
  - State estimation
  - Low-level controller
  - Push-recovery
  - Fast walking

Thank you!

A big thanks to everyone who contributed:
Twan Koolen, Jesper Smith, Gray Thomas, Brandon Shrewsbury, John Carff, Nathan Mertins, Douglas Stephen, Peter Abeles, Johannes Englsberger, Stephen McCrory, Jeff van Egmond, Maarten Griffioen, Tingfan Wu, Tomas de Boer, Marshall Floyd, Samantha Kobus, Alex Lesman, Nolan Manor, Sami Alsheikh, Daniel Duran, William Rifenburgh, Larry Bunch, Eric Morphis, Luca Colasanto, Khai-Long Ho Hoang, Pushyami Kaveti, Wessel Straatman, Brooke Layton and Pepper, Peter Neuhaus, Matthew Johnson, Jerry Pratt. I hope I haven't forgotten anybody…