Drake QP Inverse Dynamics controller

$\dot{V}_d$ cost pelvis, torso basis reg cost $\|\beta\|^2$

$$\min_{\dot{v},\beta} \underbrace{(J_{com}\dot{V} + \dot{J}_{com}V}_{\ddot{p}_{com}} - \ddot{p}_{com_{des}})^2 + (\dot{V} - \dot{V}_{des})^2 + \text{body acceleration costs} + \text{regularizations}$$

(including contacts)

$\lambda' = \lambda + 10$ $-(J^T\beta)^T(\lambda+10)$ ~~centroidal momentum~~

subject to:

$A = [M_u - (J^T\beta)^T]$ $b = [h_u]$ $X = \begin{bmatrix}\dot{V}\\ \beta\end{bmatrix}\begin{matrix}36\\32\end{matrix} > 68$

- Dynamics equality 6×68 $M_u\dot{V} + h_u = (J^TB)^T_u\beta$ $M_u\dot{V} - (J^T\beta)^T\beta = -h_u$ u = only top 6 rows for floating base
- vd_eq ⊖ $\dot{V}$ equality 15×36 — constrain $\dot{V}$ values for indices that have desired values
- eq_dof_motion
- contact force basis inequality 32×32 $\beta \geq 0$ $A = [M_\ell\ (J^TB)_\ell]$ $BL = \tau_{min} - h$ $BU = \tau_{max} - h$
- torque limit inequality 30×68 $\tau_{min} - h \leq M_\ell\dot{V} - (J^TB)_\ell\beta \leq \tau_{max} - h$

in linear term

what to change for λ? β

$\left\|\begin{bmatrix}\dot{p}\\ L\end{bmatrix} - \begin{bmatrix}\dot{p}\\ L\end{bmatrix}_{des}\right\|^2$ replace with centroidal moment matrix if $\dot{p} \rightarrow \begin{bmatrix}p\\ L\end{bmatrix}$

$$\underbrace{J_{com}\dot{V} + \dot{J}_{com}V}_{\ddot{p}_{com}} - \ddot{p}_{des}$$

same as mc-rtc

- Dynamic equality
- Contact force basis inequality

Costs

- ~~$\dot{V}_d$ cost~~
- Foot contact cost 36×36 (soft no-slip constraint)

Quadratic cost!

$$\frac{1}{2}x^TQx + b^Tx + c$$

$$\ddot{p}_{contact} = J_c\ddot{q} + \dot{J}_c\dot{q} = 0 \longrightarrow \min\|J_c\ddot{q} + \dot{J}_c\dot{q} - 0\|^2$$

$A = J_c$ $b = -\dot{J}_c\dot{q}$

$\|Ax - b\|_2^2$

~~$Q = J_c^TJ_c$ $c = 2(J_c^T(\dot{J}_c\dot{q}))^T$~~

$= x^T(A^TA)x - 2(A^Tb)^Tx$

- ⊘ centroidal momentum change cost 36×36

$\ddot{p}_c + \eta\dot{p} = 0$

~~vd cost~~

$$\underbrace{J_c}_{A}\ddot{q} + \underbrace{\dot{J}_c\dot{q} + k_d J_c\dot{q}}_{-b} = 0$$

- ~~vd cost~~

$\lambda' = \lambda + 10 \geq 0$

$\lambda > -10$

$$x^T(A^TA)x + (\dot{A}^TV - \dot{C}_{des})X$$

$C \in \mathbb{R}^{6\times N}$ — $N = DOF$

$$\frac{1}{2}x^T(J_c^TJ_c)x - (A^Tb)^Tx$$

centroidal momentum matrix: maps joint rates to centroidal momentum

$$J_c^T(\dot{J}_c\dot{q} + \underbrace{k_dJ_c\dot{q}}_{\dot{V}_c})$$

$$\begin{bmatrix}\dot{p}\\ \dot{L}\end{bmatrix} = C\ddot{q} + \dot{C}\dot{q} \xleftarrow{\frac{d}{dt}} \underbrace{\begin{bmatrix}p\\ L\end{bmatrix}}_{H} = C\dot{q}$$

$$\min\left\|\begin{bmatrix}\dot{p}\\ \dot{L}\end{bmatrix}\right\|^2 \longrightarrow \min\|C\ddot{q} + \dot{C}\dot{q}\|^2$$

$\begin{bmatrix}\dot{p}\\ \dot{L}\end{bmatrix}_{des}$ $\begin{bmatrix}\dot{p}\\ \dot{L}\end{bmatrix}_{des}$

Note: no self-collision cost included here

$H = C\dot{q}V$

$A = C$ $b = -\dot{C}\dot{V}\begin{bmatrix}\dot{p}\\ \dot{L}\end{bmatrix}_{des}$

$$\frac{\partial H}{\partial t} = \dot{C}\underset{V}{\dot{q}} + C\underset{\dot{V}}{\ddot{q}}$$

Desired Centroidal Momentum

$$\begin{bmatrix} \text{angular} \\ \text{linear} \end{bmatrix} \begin{matrix} 3 \\ 3 \end{matrix}$$

$\|H - H_{des}\|^2$ constant

$\cancel{H^T H} - 2H^T H_{des} + \cancel{H_{des}^T H_{des}}$

$(C\dot{v} + \dot{C}v)^T (C\dot{v} + \dot{C}v) - 2(C\dot{v} + \dot{C}v)$

$\cancel{H_{des}^T H_{des}}$

7 + 9

16

4

min $H^T H - 2H^T H_{des}$

$(C\dot{v} + \dot{C}v)^T (C\dot{v} + \dot{C}v) - 2(C\dot{v} + \dot{C}v)^T H_{des}$

$(C\dot{v})^T (C\dot{v}) + (C\dot{v})^T (\dot{C}v) + (\dot{C}v)^T (C\dot{v}) + \underbrace{\cancel{(\dot{C}v)^T (\dot{C}v)}}_{\text{constant}} - 2(C\dot{v} + \dot{C}v)^T H_{des}$

$\dot{v}^T C \dot{v} + (\dot{C}v - \dot{v}_{des})^T \dot{v}$

→ LinearSystemSolver

Slowness is in $\boxed{\text{RigidBodyPlant::DoCalcTimeDerivatives}}$

+ closestpoints

check sparsity?

From Diagram::DoCalcTimeDerivatives

or

↓

System<T>::DoCalcTimeDerivatives → LeafSystem → RigidBodyPlant::D

From Integrator::DoStep

$\boxed{\text{Look for IK solver}}$

0.1

- New QPPlan type with separate body motions
- ~~Try different integration scheme~~

50×50'

colPivotQR 50000 ns

JacobiSVD 700000

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 6 \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 4 & 3 \end{bmatrix}$$

$$x_{exp2} \begin{bmatrix} 5/11 \\ -4/11 \\ 17/11 \end{bmatrix} \qquad \begin{bmatrix} 3 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

Drake QP inverse dynamics controller

Dynamics: $M(q)\dot{v} + h(q,v) = S\tau + J^T\lambda$ contact wrench in world frame

acceleration / selection matrix

In actual implementation: $M(q)\dot{v} + h(q,v) = S\tau + J^T B\beta$ replaced by set of point forces applied at contact points

basis

Dynamics eq
vel eq
ineq
contact force basis
torque limit ineq
LF, RF contact cost
cen mom change cost
pelvis, torso cost
vel cost
basis reg cost

Desired motions! a_{des} in Cartesian

$a_{des} = J\dot{v} + \dot{J}v$ linear wrt $\dot{v}$

becomes $\tau = (\text{TorqueLinear})\,X + (\text{TorqueConstant})$

$[M_l, -J^T B]\begin{bmatrix}\dot{v}\\ \beta\end{bmatrix} + [h_l]$

$S = \begin{bmatrix} 0 \\ I \end{bmatrix}$ first 6 zeros

$\tau = M_l\dot{v} + h_l - (J^T B)_l\beta$ Just solve for $\dot{v}$ and β

where l means bottom n_τ rows

$X = \begin{bmatrix}\dot{v}\\ \beta\end{bmatrix}$

Constraints

equality: can also be high weight cost terms

- Dynamics
- contact constraints

Inequality: $\tau_{min} - h \le M\dot{v} - J^T B\beta \le \tau_{max} - h$

- Joint torque limit
- Limits on β (forces)

$$\min \left(J_{com}\dot{v} + \dot{J}_{com}v - \ddot{P}_{com}\right)^2 + (\dot{v} - \dot{v}_{des})^2 + \text{body acceleration cost terms}$$

36 DOF $\dot{v}$
32 β basis weights
68 total

? — Dynamics eq
Aeq 6×68 start 0

? — ~~vel~~ eq
15×36 0

~~Aineq~~ contact force basis $\lambda \ge 0$
Aineq 32×32 start 36

? — torque limit
30×68 start 0

Costs
leftfoot cartes
36×36 start 0
right F cartesi
36×36 0

? — centroidal momentum change
36 36 0
pelvis torso, vel
36×36 0
basis reg
32×32 36

$A\,x = b$

$\begin{bmatrix} n \\ \times \\ 68 \end{bmatrix}\begin{bmatrix} 68 \\ \times \\ 1 \end{bmatrix} = \begin{bmatrix} n \\ 1 \end{bmatrix}$

Q had an infinity

- make general methods for getting Q, c, Aeq, Aineq, beq, bineq from QP specification in Math Program

see mosek_solver.cc

extract to somewhere else

~~Try on Docker~~

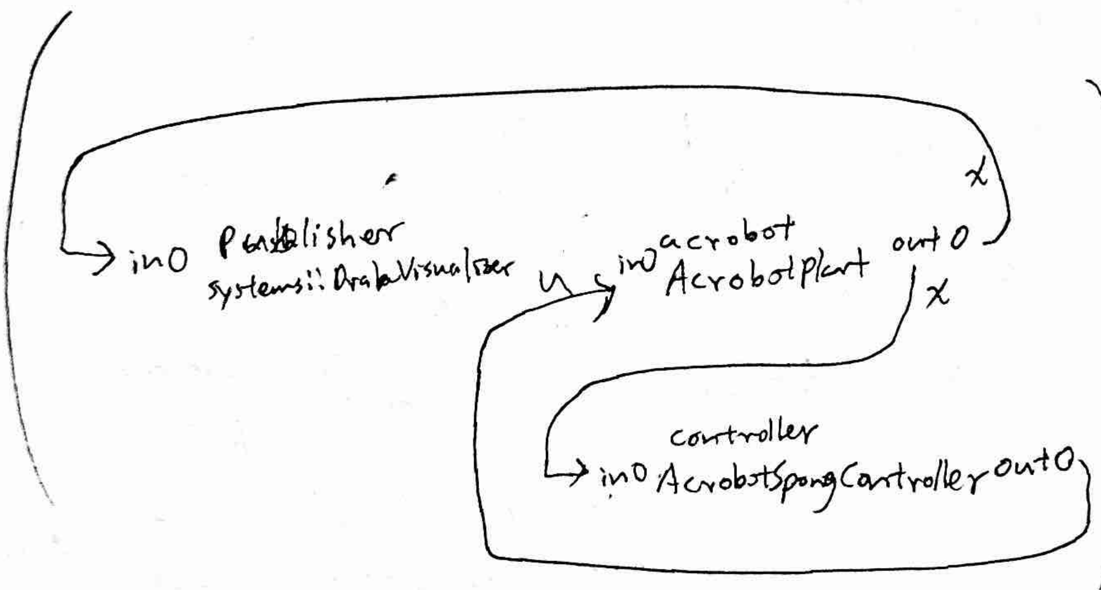
Message Naveen
Switch out Gurobi for mosek

Look at PlanEval System

$X^T(J_c^T J_c)X + J_c^T(\dot{J}_c V)$ J.

Drake

Acrobot swingup example



simulator

Basic procedure:

1. Set up diagram w/ nodes — publisher, robot, controller
2. Build diagram
3. Set up simulator w/ diagram
4. Initialize simulator → start (stepTo (time[s]))

Diagram ↓ goes into System ↓ goes into Simulator

just to set initial state

acrobot_context via GetMutableSubSystemContext

Where are system dynamics (eg. for pendulum) calculated?

Controllers class with:

inputport (state)

outputport (plantInput,

control law { function to calculate controls)

Try putting iLQR in?

shooting Method

use Linearization tool, etc

Look at:

- systems directory
- LCM basics
- Humanoid controller — Euroc Valkyrie example
- URDF parsing ↔ simple robot? threelinkarm
- way to visualize Drake diagram?
- System::ToAutoDiffXd support in System — how?

Trajectory Optimization — Mathematical Program

test gflags

- see acrobot_spong_controller for how to linearize plant and use LQR
- also note that acrobot spong example has state receiver and command sender instead of just a plant, since it's meant to deal w/ a real robot

plant inside controller just tracks real robot

3×10^{-4}

$\boxed{0.0003}$

0.05

PlanEval ~~take, state~~

inputs:

· state estimate

· Manipplan

Outputs:

Try larger timesteps

RobotStateMsg → HumanoidStatus

Input ManipulationPlan

type robotlocomotion::robot_plant

PlanEval → QPInverseDynamics

QPInput — Just make our own QPInput?

Do ExtendedCalcUnrestrictedUpdate — look here

clean RAM?

change HumanoidManipulationPlan to something else

com computed with RobotKinematicState::get_com

what is semi explicit?

~~Try hardcoding desired centroidal momentum~~

~~PD controller~~

Is control loop running at same rate as simulation?

No, see dt in PlanEvalBaseSystem, QPInverseDynamicsSystem

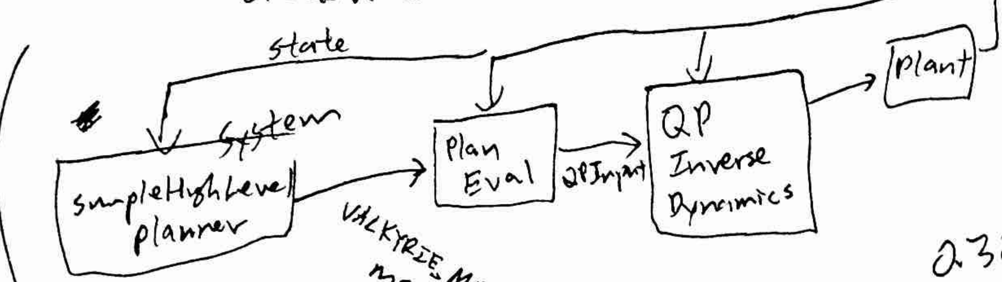
Check where this is used

Student Teacher

Person

Person* p = new Student

&



0 1 2 3

VALKYRIE_MANIP_PLAN msg

This message type (class) contains the details on how to process desired behavior from high-level planner into QPInput. (desired accelerations and positions)

LCM_DrivenLoop

Blocked by robotstatesub

(see ValkyrieBalancing demo, humanoid_controller.h)

0.308

-0.39

0.93

make this

LCMDrivenLoop(Diagram, subscriber, null, lcm, timeconverter)

SimpleHighLevelPlanner(System)

Diagram w/LCM subs, pubs