ROCR Trajectory - current thoughts

- Optimization routines can help in finding feasible, locally optimal trajectories.
- Fast, stable trajectories are the result of finely tuned/matched torque levels and transition states.
- More rotational inertia on the main body is better than less (mass being equal).
- A higher tail velocity at transition can help stabilize the gait, and require less torque.
- (A little?) more work required to get continuous torque optimization working – seeding with bang-bang torque optimization will hopefully help.

Lessons learned so far...

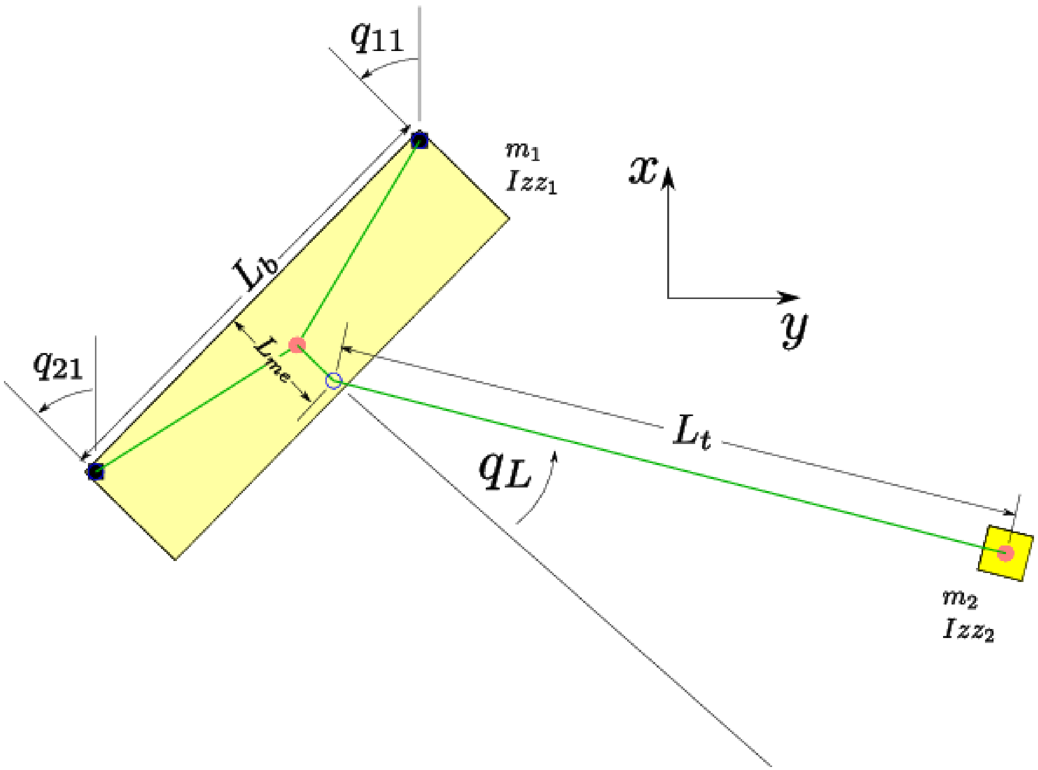
- For (erroneously) higher Izz1, optimization led to a doubling in climb rate and an improvement in return map stability.
- Results not (yet?) replicated for more realistic Izz1.

- Current method:
  - Generate steady state trajectory using tail position command
  - Use steady state switching state as a starting point for the optimization
  - Search by hand for a solvable optimization adjusting:
    - # of torque switches
    - Maximum torque
    - Sign of initial applied torque

Optimization is performed using DIRCOL and SNOPT

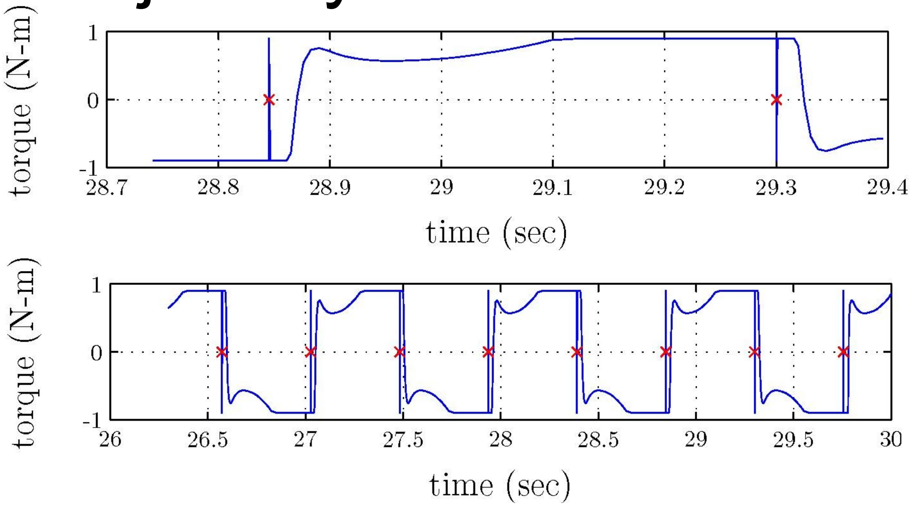
- FORTRAN!!
- DIRCOL (Oscar von Stryk, 2002) – Direct Collocation
  - Takes as inputs the differential equations, cost function, limits on the control and state, boundary conditions.
  - Can handle multiple phases, and adjust the timing of the transitions between the phases.
  - Handles the discretization of the problem and its transformation into a non-linear program which SNOPT can handle.
  - Helps to scale the problem for SNOPT.
  - Utilizes Hermite-Simpson discretization (Enright and Conway, 1992)
- SNOPT (Gill, Murray, and Saunders, 2005) – Sparse nonlinear optimization
  - Solves non-linear programs using a robust sequential quadratic programming method.
  - Deals relatively gracefully with infeasible constraints, which tend to pop up with the discretization of the equations of motion.
- MATLAB
  - Used for scripting and subsequent verification and further simulation of the results

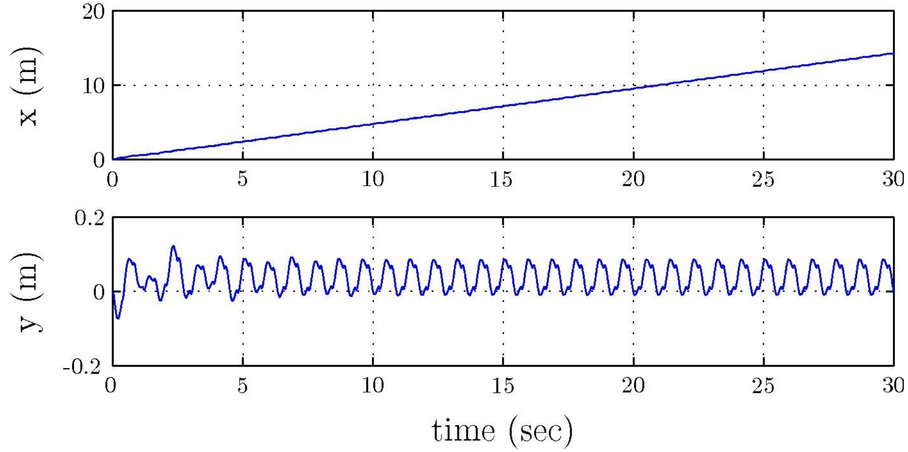
ROCR Model

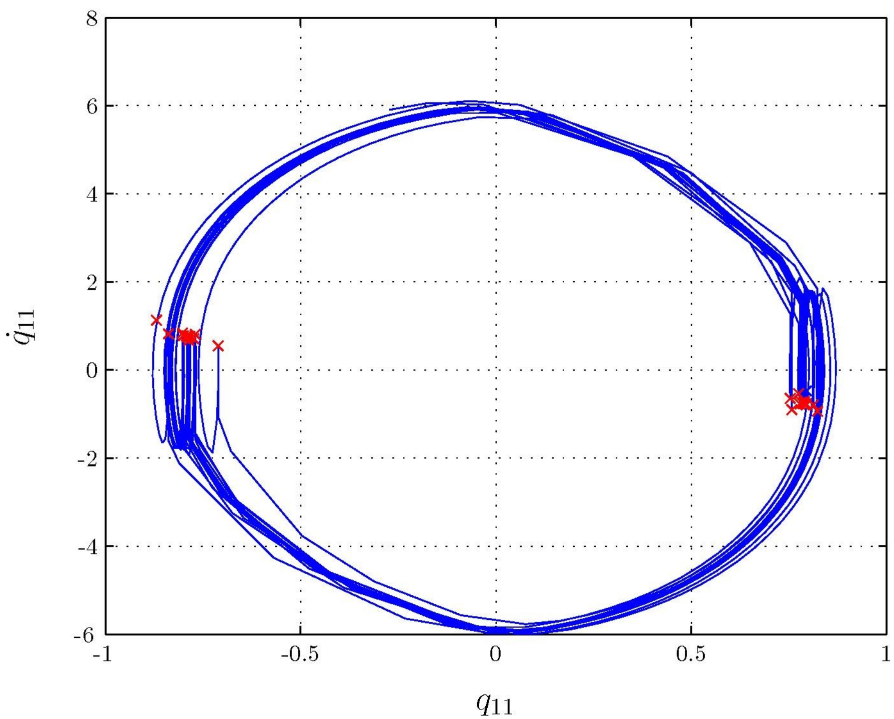


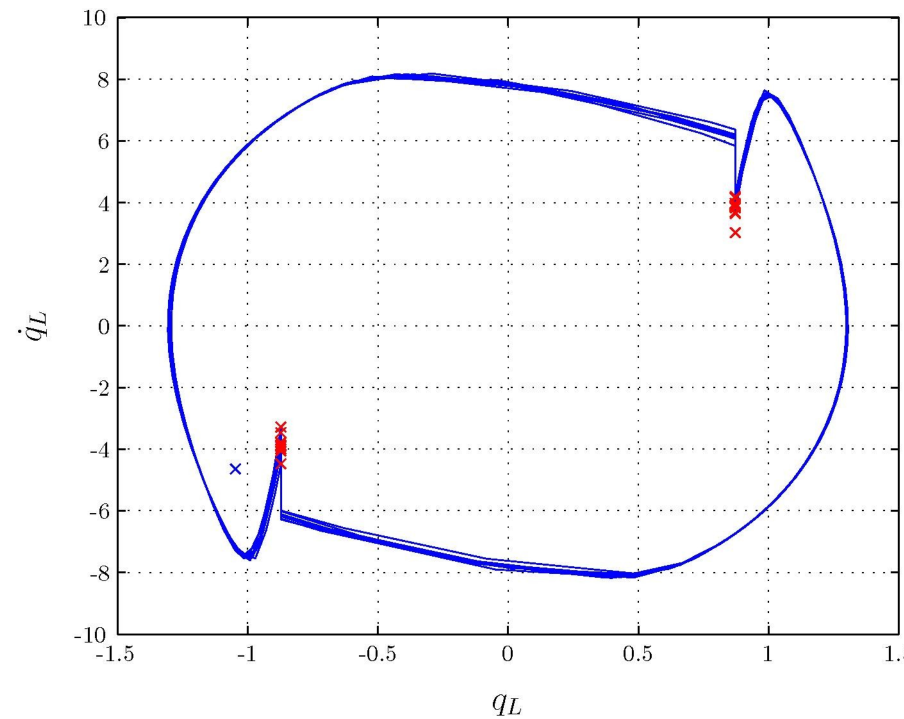
m_1	$0.335\ kg$
Izz_1	$0.0032\ kg\text{-}m^2$
m_2	$0.3\ kg$
Izz_2	$0.0\ kg\text{-}m^2$
L_b	$0.305\ m$
L_{me}	$0.07\ m$
L_t	$0.457\ m$

Trajectory-based control



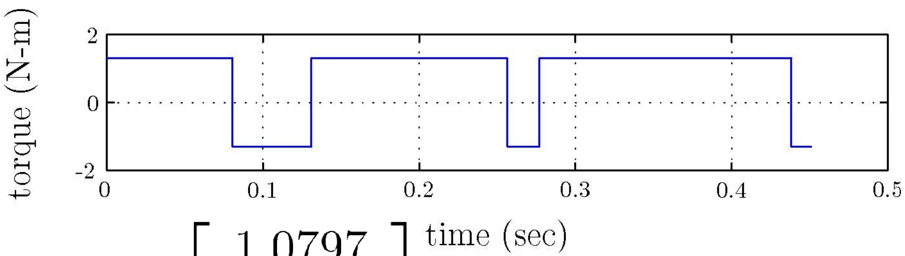




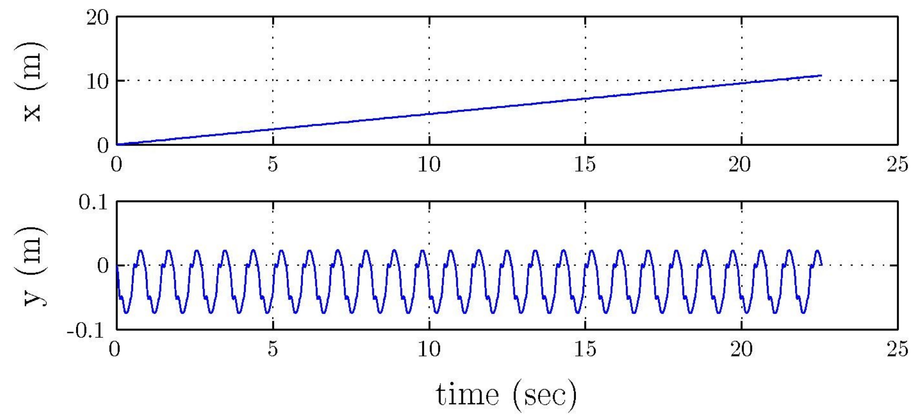


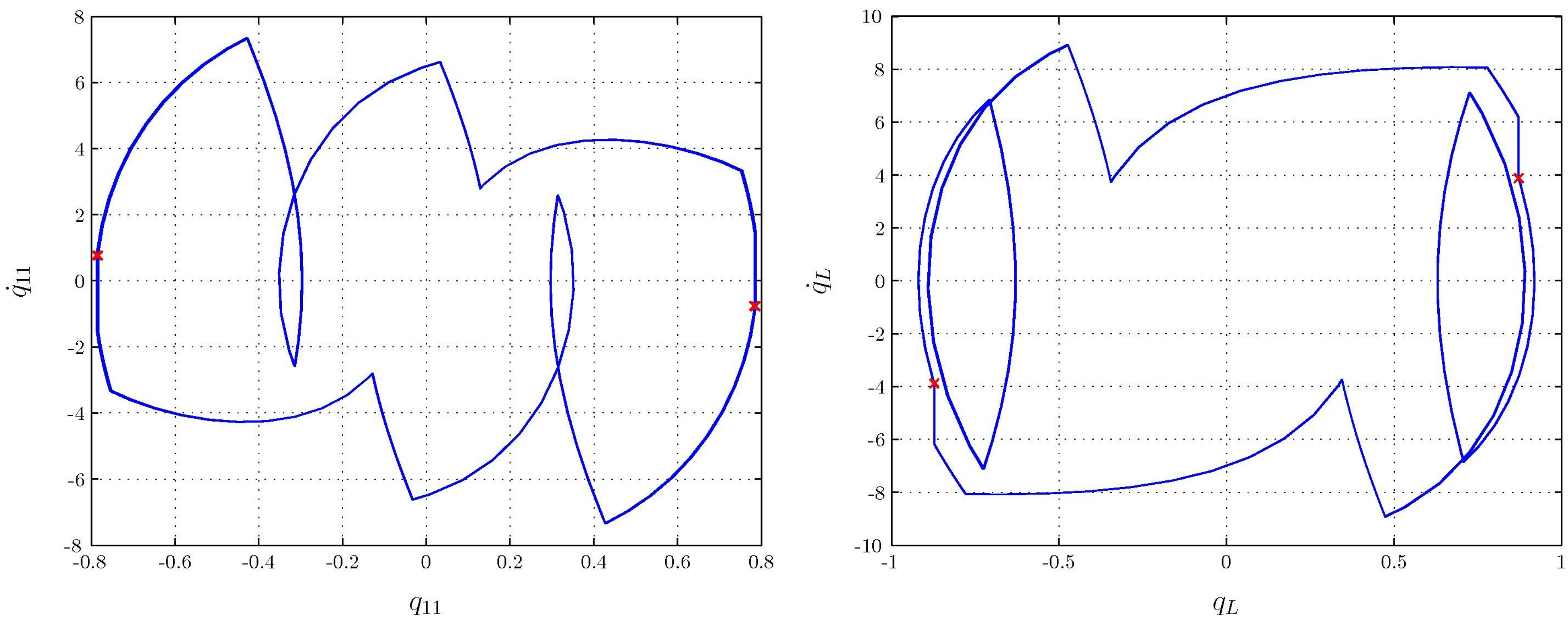
ROCR-1.10Hz-75deg-0.9Nm-0.05b-50deg_sw-20.00kp-1.00kd.mp4

Minimum time: q0 = [0.7854, -0.8727, -0.7679, -3.8746]



$$\text{eig} = \begin{bmatrix} 1.0797 \\ 1.0797 \\ 0.1730 \\ 0.3656 \end{bmatrix}$$





ROCR-spec-tau-0.05b-45--50--44--222-6n.mp4

Some questions to answer...

- Foot switches – some state based control is likely extremely helpful to overall gait stability.
- Can the continuous-torque optimization yield good results?
- Are there any rules-of-thumb for sizing the main body and tail? How does this affect actuator requirements and resulting rates of climb?
- How generalizable is this? (To RiSE?)
- How does one find the “sweet spot” of climbing:
  - Stable return map
  - Minimum torque required
  - Fast climbing
  - Actuator doing mostly positive work (always adding energy to the system).