# Optimization of Motions for Underactuated Systems that Interact with the Environment

Juanita Cecilia Vargas Albro

July 10, 2003

# Motivation

- Goal: to find motions for an underactuated robot that interacts with the environment by solving an optimal control problem
- applications in:
  - robotics
  - biomechanics
  - computer animation

# We assumed that the desired motions are the solution to an optimal control problem.

$$\text{Min } J = \Psi[q, \dot{q}, t_f] + \int_0^{t_f} L[q(t), \dot{q}(t), \tau]dt$$

$$\text{subject to: } M(q) + C(q, \dot{q}) + G(q) = Q + \sum J_M^T F$$

$$\underline{q} < q < \overline{q}$$

$$\underline{\dot{q}} < \dot{q} < \overline{\dot{q}}$$

# References

[1] B. J. Martin. *Robot Motion Optimization and Path Planning*. PhD thesis, University of California, Irvine, 1996.

[2] S. R. Ploen. *Geometric Algorithms for the Dynamics and Control of Multibody Systems*. PhD thesis, University of California, Irvine, 1997.

[3] G. A. Sohl. *Optimal motions for underactuated manipulators*. PhD thesis, University of California, Irvine, 2000.

[4] B. J. Martin and J. E. Bobrow. Minimum-effort motions for open-chain manipulators with task-dependent end-effect constraints. *The International Journal of Robotics Research*, 18(2):213–224, February 1999.

[5] C-Y. E. Wang, W. K. Timoszyk, and J. E. Bobrow. Weightlifting motion planning for a Puma 762 robot. In *Proceedings of the 1999 IEEE International Conference on Robotics and Automation*, Detroit, MI, 1999.

[6] J. E. Bobrow, J. M. McCarthy, and V. K. Chu. Time-optimal control of two robots holding the same workpiece. *Journal of Dynamic Systems, Measurement, and Control*, 115:441–446, September 1993.

[7] G. A. Sohl and J. E. Bobrow. Optimal motions for underactuated manipulators. In *Proceedings of Design Engineering Technical Conference*, Las Vegas, Nevada, September 1999.

[8] M. W. Spong. Swing up control of the acrobot. In *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, pages 2356–2361, San Diego, CA, May 1994.

[9] J. V. Albro, G. A. Sohl, J. E. Bobrow, and F. C. Park. On the computation of optimal high-dives. In *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, San Francico, CA, April 2000.

[10] J. V. Albro and J. E. Bobrow. Optimal motion primitives for a 5 dof experimental hopper. In *Proceedings of the 2001 IEEE*

*International Conference on Robotics and Automation*, Seoul, Korea, 2001.

[11] M. W. Spong and M. Vidyasagar. *Robot Dynamics and Control*. John Wiley and Sons, 1989.

[12] F. C. Park, J. E. Bobrow, and S. R. Ploen. A lie group formulation of robot dynamics. *The International Journal of Robotics Research*, 14(6):609–618, December 1995.

[13] R. M. Murray, Z. Li, and S. S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.

[14] R. Featherstone. The calculation of robot dynamics using articulated body inertias. *The International Journal of Robotics Research*, 2:13–29, Spring 1983.

[15] A. Jain and G. Rodriguez. An analysis of the kinematics and dynamics of underactuated manipulators. *IEEE Transactions on Robotics and Automation*, 9(4):411–422, August 1993.

[16] S-H Lee, J. Kim, F. C. Park, M. Kim, and J. E. Bobrow.

Newton-type algorithms for model-based motor learning. *IEEE Trans. Sys., Man, Cyber.*, 2001 (under review).

[17] A. M. Bloch, P. E. Crouch, J. E. Marsden, and T. S. Ratiu. Discrete rigid body dynamics and optimal control. In *Proceedings of the 37th IEEE Conference on Decision and Control*, pages 2249–2254, Tampa, FL, December 1998.

[18] E. G. Gilbert and D. W. Johnson. Distance functions and their application to robot path planning in the presence of obstacles. *IEEE Journal of Robotics and Automation*, RA-1(1):21–30, March 1985.

[19] J. T. Betts. Survey of numerical methods for trajectory optimization. *Journal of Guidance, Control, and Dynamics*, 21(2):193–207, March–April 1998.

[20] L. S. Brotman and A. N. Netravali. Motion interpolation by optimal control. *Computer Graphics*, 22(4):309–315, August 1998.

[21] A. Witkin and M. Kass. Spacetime constraints. In *Computer*

*Graphics (Proc. SIGGRAPH '88)*, volume 22, pages 159–168, 1988.

[22] M. F. Cohen. Interactive spacetime control for animation. *Computer Graphics*, 26(2):293–302, July 1992.

[23] Z. Liu, S. J. Gortler, and M. F. Cohen. Hierarchical spacetime control. In *SIGGRAPH 94*, Orlando, FL, July 1994.

[24] S. E. Engelbrecht. Minimum principles in motor control. *Journal of Mathematical Psychology*, 45(3):497–542, 2001.

[25] R. M. Alexander. A minimum energy cost hypothesis for human arm trajectories. *Biological Cybernetics*, 76:97–105, 1997.

[26] T. Flash and N. Hogan. The coordination of arm movements: an experimentally confirmed mathematical model. *The Journal of Neuroscience*, 5(7):1688–1703, 1985.

[27] N. Lan. Analysis of an optimal control model of multi-joint arm movements. *Biological Cybernetics*, 76:107–117, 1997.

[28] M. G. Pandy, F. E. Zajac, E. Sim, and W.S. Levine. An optimal

control model for maximum-height human jumping. *The Journal of Biomechanics*, 23(12):1185–98, 1990.

[29] T. Komura and Y. Shinagawa. A muscle-based feedforward controller of the human body. *Computer Graphics Forum, Special Issue*, 16(3):165–176, 1997.

[30] Z. Popović. Editing dynamic properties of captured human motion. In *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, pages 670–675, San Francisco, CA, April 2000.

[31] W. L. Wooten and J. K. Hodgins. Animation of human diving. *Computer Graphics Forum*, 15(1):3–14, 1996.

[32] L. S. Crawford and S. S. Sastry. Biological motor control approaches for a planar diver. In *Proceedings of the IEEE Conference on Decision and Control*, New Orleans, LA, December 1995.

[33] M. van de Panne, J. Laszlo, P. Huang, and P. Faloutsos. Towards agile animated characters. In *Proceedings of the 2000*

*IEEE International Conference on Robotics and Automation*, pages 682–687, San Francisco, CA, April 2000.

[34] W. L. Wooten and J. K. Hodgins. Simulated leaping, tumbling, landing, and balancing humans. In *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, pages 656–662, San Francisco, CA, April 2000.

[35] A. V. Hill. The heat of shortening and the dynamic constants of muscle. *Proceedings of the Royal Society*, B126:136–195, 1938.

[36] C-Y. E. Wang, J. E. Bobrow, and D. J. Reinkensmeyer. Dynamic motion planning for the design of robotic gait rehabilitation. *ASME Journal of Biomechanical Engineering*, to appear.

[37] K. Yoshida. A general formulation for under-actuated manipulators. In *Proceedings of the 1997 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1651–1957, Grenoble, France, 1997.

[38] M. Bergerman and Y. Xu. Optimal control sequence for

underactuated manipulators. In *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, pages 3714–3719, Minneapolis, MN, April 1996.

[39] F. Bullo and K. M. Lynch. Kinematic controllability for decoupled trajectory planning in underactuated mechanical systems. *IEEE Transactions on Robotics and Automation*, 17(4):402–412, August 2001.

[40] G. Oriolo and Y. Nakamura. Free-joint manipulators: Motion control under second-order nonholonomic constraints. In *IEEE/RSJ International Workshop on Intelligent Robots and Systems*, pages 1248–1253, Osaka, Japan, November 1991.

[41] S. Samuel and S. S. Keerthi. A nonlinear least squares approach to the numerical optimal control of non-holonomic systems. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, volume 2, pages 467–472, Piscataway, New Jersey, October 1993. IEEE.

[42] M. Hardt, K. Kreutz-Delgado, and J. W. Helton. Minimal

energy control of a biped robot with numerical methods and a recursive symbolic dynamic model. In *Proceedings of the 37th IEEE Conference on Decision and Control*, pages 413–416, Tampa, FL, December 1998.

[43] M. Buss, M. Glocker, M. Hardt, O. von Stryk, R. Bulirsch, and G. Schmidt. Nonlinear hybrid dynamical systems: Modeling, optimal control, and applications. In S. Engell, G. Frehse, and E. Schnieder, editors, *Modelling, Analysis and Design of Hybrid Systems*, volume 279 of *Lecture Notes in Control and Information Science*, pages 311–335. Springer-Verlag, Berlin, 2002.

[44] K. D. Mombaur, H. G. Bock, J. P. Schlöder, and R. W. Longman. Human-like actuated walking that is asymptotically stable without feedback. In *Proceedings of the 2001 IEEE International Conference on Robotics and Automation*, pages 4128–4133, Seoul, Korea, May 2001.

[45] K. Yamane and Y. Nakamura. Dynamics filter—concept and

implementation of online motion generator for human figures. *IEEE Transactions on Robotics and Automation*, 19(3):421–432, June 2003.

[46] David Baraff. Fast contact force computation for nonpenetrating rigid bodies. In *Computer Graphics Proceedings*, pages 23–34, 1994.

[47] M. G. Pandy and F. C. Anderson. Dynamic simulation of human movement using large-scale models of the body. In *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, pages 676–681, San Francisco, CA, April 2000.

[48] A. Pandolfi, C. Kane, J. E. Marsden, and M. Ortiz. Time-discretized variational formulation of non-smooth frictional contact. *International Journal for Numerical Methods in Engineering*, 2002.

[49] M. C. Çavuşoğlu, J. Yan, and S. S. Sastry. A hybrid system approach to contact stability and force control in robotic manipulators. In *Proceedings of the 12th IEEE International*

*Symposium on Intelligent Control*, Istabul, Turkey, July 1997.

[50] Y. Wang, V. Kumar, and J. Abel. Dynamics of rigid bodies undergoing multiple frictional contacts. In *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, pages 2764–2769, Nice, France, 1992.

[51] J. A. Reichler and F. Delcomyn. Dynamics simulation and controller interfacing for legged robots. *The International Journal of Robotics Research*, 19(1):42–58, January 2000.

[52] E. G. Gilbert, D. W. Johnson, and S. S. Keerthi. A fast procedure for computing the distance between complex objects in a three-dimensional space. *IEEE Journal of Robotics and Automation*, 4(2):193–203, April 1988.

[53] M. C. Lin and J. F. Canny. A fast algorithm for incremental distance calculation. In *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, pages 1008–1014, April 1991.

[54] J. F. Canny and M. C. Lin. An opportunistic global path

planner. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1554–1559, 1990.

[55] T. McGeer. Passive dynamic walking. *The International Journal of Robotics Research*, 9(2):62–82, April 1990.

[56] J. E. Marsden and J. Ostrowski. Symmetries in motion: Geometric foundations of motion control. *Nonlinear Science Today*, 1998.

[57] J. P. Ostrowski, J. P. Desai, and V. Kumar. Optimal gait selection for nonholonomic locomotion systems. *International Journal of Robotics Research*, 19(3):225–237, 2000.

[58] M. Raibert, S. Tzafestas, and C. Tzafestas. Comparative simulation study of three control techniques applied to a biped robot. In *Proceedings of the International Conference on Systems Engineering in Service of Humans*, volume 1, pages 494–502, October 1993.

[59] M. W. Spong, R. Lozano, and R. Mahony. An almost linear biped. In *IEEE Conference on Decision and Control*, pages

4803–4808, Sydney, Australia, December 2000.

[60] J. Schmitt and P. Holmes. Mechanical models for insect locomotion: stability and parameter studies. *Physica*, D(156):139–168, 2001.

[61] J. M. McCarthy. *An Introduction to Theoretical Kinematics*. The MIT Press, MA, 1990.

[62] Jr. A. E. Bryson and Yu-Chi Ho. *Applied Optimal Control: Optimization, Estimation, and Control*. Taylor and Francis, 1975.

[63] D. C. Luenberger. *Linear and Nonlinear Programming*. Addison-Wesley Publishing Co., 1989.

# Kinematics using the Product of Exponentials

Denavit-Hartenberg:

$$f_{i-1,i} = Rot_{z_i,\theta_i} Trans_{z_i,d_i} Trans_{x_i,a_i} Rot_{x_i,\alpha_i}$$

$$= \begin{bmatrix} \cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\sin\alpha_i & a_i\cos\theta_i \\ \sin\theta_i & \cos\theta_i\cos\alpha_i & -\cos\theta_i\sin\alpha_i & a_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \in SE(3)$$

Alternative: Matrix Exponentials

$$f_{i-1,i} = e^{Sq}M \quad \frac{df}{dq} = Se^{Sq}M \qquad (1)$$

1. $q$ is the joint angle (or distance for prismatic joint)
2. $S \in se(3)$ is the *joint screw* composed of unit vectors $\omega$ and $v$ which define the directino of the joint motion arranged in a 4x4

matrix:

$$S = (w, v) = \begin{bmatrix} [w] & v \\ 0 & 0 \end{bmatrix}, \ [w] = \begin{bmatrix} 0 & -w_z & w_y \\ w_z & 0 & -w_x \\ -w_y & w_x & 0 \end{bmatrix} \qquad (2)$$

3. $M \in SE(3)$ is the 4x4 transformation between adjacent joint frames in the home position

# The Manipulator Jacobian

$$V = \begin{bmatrix} w \\ v \end{bmatrix} = \begin{bmatrix} J_w(q,p) \\ J_v(q,p) \end{bmatrix} \dot{q} = J(q,p)\dot{q} \tag{3}$$

where $q$ are the joint values and $p$ is the point on the robot to which the Jacobian is being written.

$$v = \frac{dx}{dt} = \frac{dx}{dq}\frac{dq}{dt} = J_v \dot{q}, \;\; J_v = \frac{dx}{dq} \tag{4}$$

$$\tau = J^T F = [\; J_w \quad J_v \;] \begin{bmatrix} M \\ f \end{bmatrix} \tag{5}$$

where $V \in \Re^6$ is the generalized velocity, $\dot{q}$ is the vector of joint velocities, $M$ is a moment, $f$ is a force, so $F \in \Re^6$ is the generalized external force.

The Manipulator Jacobian is written:

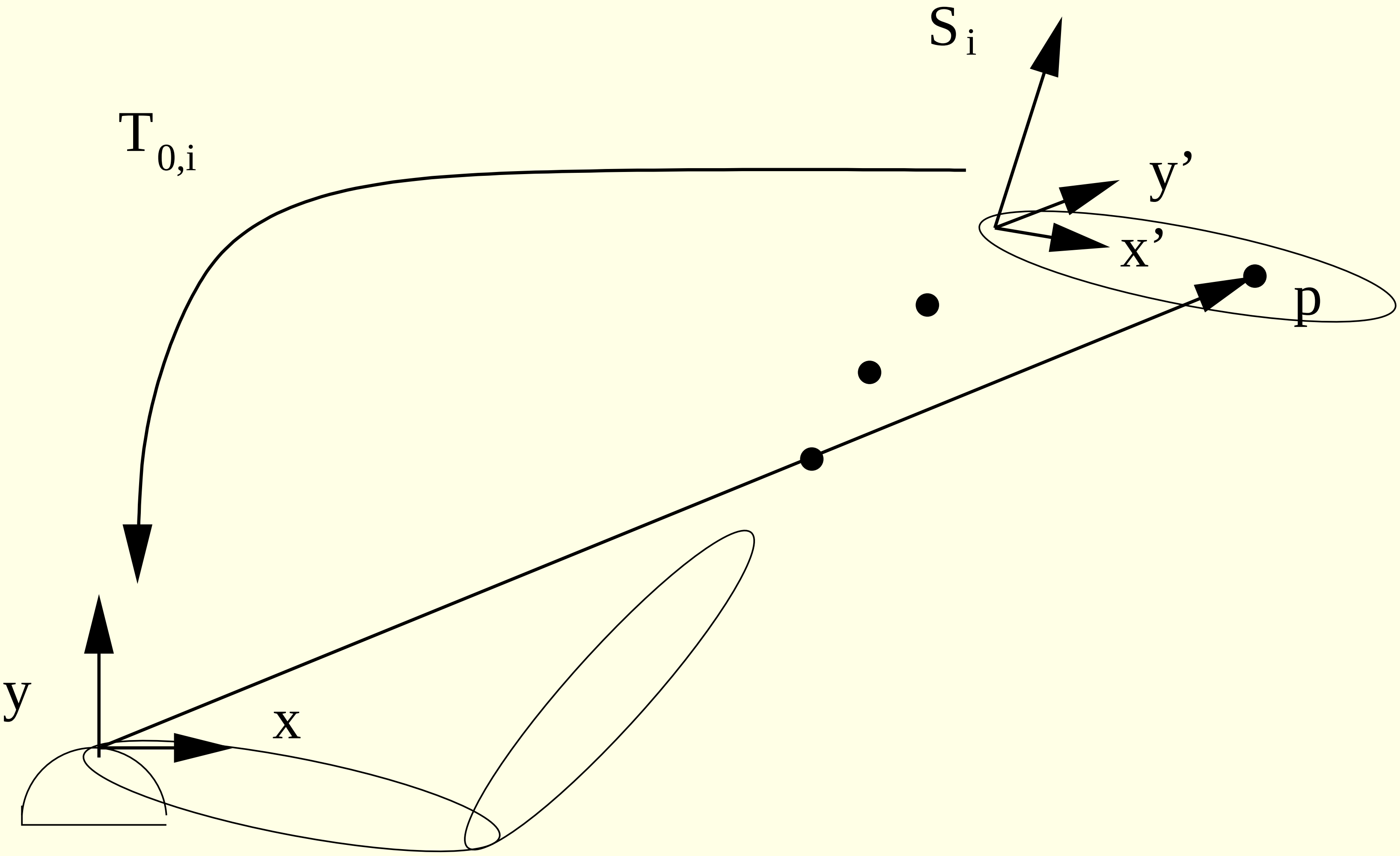
$$\begin{bmatrix} J_w \\ J_v \end{bmatrix} = \dot{T}_{0,n} T_{0,n}^{-1} + \begin{bmatrix} 0 \\ J_w \times p \end{bmatrix} \tag{6}$$

Later we will need the derivative of the Jacobian with respect to the joint angles $\frac{\partial J}{\partial q}$

$$J_w = [\dot{T}_{0,n} T_{0,n}^{-1}]_{upper\ \ half}$$

$$J_v = [\dot{T}_{0,n} T_{0,n}^{-1}]_{lower\ \ half} + J_w \times p$$

# Physical Interpretation of Jacobian Terms



# Derivative of the Manipulator Jacobian

$$\frac{\partial J_w}{\partial q_i} = \frac{\partial}{\partial q_i}[\dot{T}_{0,n}T_{0,n}^{-1}]_{upper\ half}$$

$$\frac{\partial J_v}{\partial q_i} = \frac{\partial}{\partial q_i}[\dot{T}_{0,n}T_{0,n}^{-1}]_{lower\ half} + \frac{\partial J_w}{\partial q_i} \times p + J_w \times \frac{\partial p}{\partial q}$$

1. Calculate $J_w, J_v$
   (a) find $\dot{T}_{0,n}T_{0,n}^{-1}$
   (b) extract upper half, $= J_w$
   (c) evaluate $J_w \times p$, add to lower half of $\dot{T}_{0,n}T_{0,n}^{-1}$, have $J_v$
2. Calculate $\partial(\dot{T}_{0,n}T_{0,n}^{-1})/\partial q$
   (a) perform ad on all combinations of columns of $\dot{T}_{0,n}T_{0,n}^{-1}$ for $j < i$:

$$\frac{\partial \dot{T}_{0,i}T_{0,i}^{-1}}{\partial q_j} = \frac{\partial}{\partial q_j}(Ad_{T_{0,i}}S_i) = ad_{Ad_{T_{0,j}}S_j}Ad_{T_{0,i}}S_i \tag{7}$$

(b)

$$\frac{\partial J_w}{\partial q} = \left[\frac{\partial}{\partial q_j}\dot{T}_{0,n}T_{0,n}^{-1}\right]_{upperhalf} \tag{8}$$

3. calculate $J_w \times \frac{\partial p}{\partial q_j}$ and $\frac{\partial J_w}{\partial q_j} \times p$
4. Add terms for $\frac{\partial J_v}{\partial q_j}$

# We assume that the desired motions are the solution to an optimal control problem. An example of this problem is the human diver.

$$\text{Min } J = \Psi[q, \dot{q}, t_f] + \int_0^{t_f} L[q(t), \dot{q}(t), \tau] dt$$

$$\text{subject to: } M(q) + C(q, \dot{q}) + G(q) = Q + \sum J_M^T F$$

$$\underline{q} < q < \overline{q}$$

$$\underline{\dot{q}} < \dot{q} < \overline{\dot{q}}$$

Find $\ddot{q}^a, \tau^p$. These will define the motion for an underactuated robot (that is, a robot with more degrees of freedom than actuators)

# Parametrize the Motion

1. Transform the optimal control problem into a parameter optimization problem.
2. Choose a set of parameters, $P(\ddot{q}^a, \tau^p)$ to define the motion.
3. We usually choose P to represent the active joint trajectories, $q^a(t)$. In the free-flying case, the passive joint torques are zero. When contact with the environment is made, the passive joint torques are $\sum J_M^T F$.
4. B-Splines are used to represent the active joint trajectories:

$$q^a(t) = \sum_{i=1}^{n} p_i B_{i,j} \tag{9}$$

5. With this definition, $\dot{q}^a$ and $\ddot{q}^a$ are easily found. The coefficients $P = (p_0, ...p_n)$ are the parameters over which the problem is optimized.

# We assumed that the desired motions are the solution to an parameter optimization problem.

Min $J(P) = \Psi[q(P), \dot{q}(P), t_f(P)] + \int_0^{t_f} L[q(P), \dot{q}(P), \tau(P)]dt$

subject to: $M(q(P)) + C(q(P), \dot{q}(P)) + G(q(P)) = Q(P) + \sum J_M^T F$

$$\underline{q} < q(t, P) < \overline{q}$$

$$\underline{\dot{q}} < \dot{q}(t, P) < \overline{\dot{q}}$$

initial conditions: $q(t_0), \dot{q}(t_0)$

Find the set of parameters $P$ that yield the desired motion $(\ddot{q}^a, \tau^p)$

# Numerical Solution of Optimization Problem

$$J(P) = (q^p(t_f) - q_d)^T W (q^p(t_f) - q_d) + \frac{1}{2}\int_{t_0}^{t_f} \|\tau^a\|^2 dt \tag{10}$$

To computer $J$ we need to integrate the system forward in time to determine the motion of the passive joints:

$$\frac{d}{dt}\left\{\begin{array}{c} q^p \\ \dot{q}^p \\ \int_0^t \frac{1}{2}||\psi||^2 dt \end{array}\right\} = \left\{\begin{array}{c} \dot{q}^p \\ \ddot{q}^p(q, \dot{q}, \ddot{q}^a, \tau^p) \\ \frac{1}{2}||\psi(q, \dot{q}, \ddot{q}^a, \tau^p)||^2 \end{array}\right\} \tag{11}$$

We solve this with C-STORM, which contains the hybrid active/passive dynamics algorithm:

$$(\tau^a, \ddot{q}^p) = g(q, \dot{q}, \ddot{q}^a, \tau^p). \tag{12}$$

# Hybrid Dynamics Algorithm, Input: $q, \dot{q}, \ddot{q}^a, \tau^p$

# Computing the Exact Gradient

$$\frac{\partial J}{\partial P} = 2(q^p(t_f) - q_d)^T W \frac{\partial q^p(t_f)}{\partial P} + \int_{t_0}^{t_f} \frac{\partial \tau^a}{\partial P} \tau^a dt \tag{13}$$

Recall: $(\tau^a, \ddot{q}^p) = g(q, \dot{q}, \ddot{q}^a, \tau^p)$. To compute $\frac{\partial J}{\partial P}$ we need the derivative of the hybrid active/passive dynamics algorithm:

$$\frac{\partial g(q, \dot{q}, \ddot{q}^a, \tau^p)}{\partial P} \tag{14}$$

and to integrate these equations forward in time.

# Computing the Gradient (cont'd)

Taking the derivative of the hybrid dynamics algorithm with respect to the set of parameters P:

$$\frac{\partial g(q,\dot{q},\ddot{q}^a,\tau^p)}{\partial P} = \frac{\partial g}{\partial q}\frac{\partial q}{\partial P} + \frac{\partial g}{\partial \dot{q}}\frac{\partial \dot{q}}{\partial P} + \frac{\partial g}{\partial \ddot{q}^a}\frac{\partial \ddot{q}^a}{\partial P} + \frac{\partial g}{\partial \tau^p}\frac{\partial \tau^p}{\partial P} \tag{15}$$

1. This equation requires the derivative of the hybrid dynamics equation with respect to joint angles, velocities, accelerations, and torques.

2. Since P parametrize $q^a$, can easily get $\dot{q}^a, \ddot{q}^a$.

3. The passive joint derivatives are integrated forward in time from an initial value of zero.

4. What about $\frac{\partial \tau^p}{\partial P}$ ? For the free-flying case, when the passive joint torques are zero, this term is zero.

5. If the passive joint torques are non-zero:

$$\frac{\partial \tau^p}{\partial P} = \frac{\partial \tau^p}{\partial q}\frac{\partial q}{\partial P} + \frac{\partial \tau^p}{\partial \dot{q}}\frac{\partial \dot{q}}{\partial P} + \frac{\partial \tau^p}{\partial \ddot{q}}\frac{\partial \ddot{q}}{\partial P} \tag{16}$$

   Later we'll discuss how the passive joint torques might come to be nonzero.

# Procedure for Solving Parameter Optimization Problem

1. Set initial conditions – $q_0^a$, $q_f^a$, $q_0^p$, $\dot{q}_0^p$
2. Begin constrained optimization in MATLAB with $constr$ :
    (a) Use hybrid algorithm to integrate system forward in time, obtain $\tau^a$, $\ddot{q}^p$
    (b) Evaluate the cost function using $\tau^a$, $\ddot{q}^p$, return value to *constr*
    (c) Use derivative of hybrid algorithm to generate data for the analytic gradient
    (d) Calculate the analytic gradient of the cost function, return it to *constr*
    (e) if local minimum is reached, exit loop
    (f) else
        i. vary parameters using the analytic gradient in the SQP algorithm
        ii. return to step (a)

# An Example of Our Approach – The Planar Diver

Cost function for diver:

$$J = c_1(q_3^p(t_f) - q_{3,desired})^2 + c_2 \int \frac{1}{2}\|\tau^a\|^2 dt \tag{17}$$

# Now we wish to solve for optimal motions for a system that makes and breaks contact with the ground, e.g. the Luxo hopping robot.

$$\begin{aligned}
\text{Min } J(P) &= \Psi[q(P), \dot{q}(P), t_f(P)] + \int_0^{t_f} L[q(P), \dot{q}(P), \tau(P)]dt \\
\text{subject to: } M(q(P)) + C(q(P), \dot{q}(P)) + G(q(P)) &= Q(P) + \sum J_M^T F \\
\underline{q} < q(t, P) &< \overline{q} \\
\underline{\dot{q}} < \dot{q}(t, P) &< \overline{\dot{q}} \\
\text{initial conditions: } q(t_0), \dot{q}(t_0)
\end{aligned}$$

Find the set of parameters $P$ that yield the desired motion $(\ddot{q}^a, \tau^p)$. Because the robot will make and break contact with the ground, the external forces $F$ will not be zero in this case.

# Numerical Solution of Parameter Optimization Problem

Same two algorithm calls as before:

$$(\tau^a, \ddot{q}^p) = g(q, \dot{q}, \ddot{q}^a, \tau^p). \tag{18}$$

$$\left(\tau^a, \ddot{q}^p, \frac{\partial(\ddot{q}^p, \tau^a)}{\partial q}, \frac{\partial(\ddot{q}^p, \tau^a)}{\partial \dot{q}}, \frac{\partial(\ddot{q}^p, \tau^a)}{\partial \ddot{q}}\right) = dg\left(q, \dot{q}, \ddot{q}^a, \tau^p, \frac{\partial \tau^p}{\partial q}, \frac{\partial \tau^p}{\partial \dot{q}}, \frac{\partial \tau^p}{\partial \ddot{q}}\right)$$

Only now the external forces $F$ are mapped onto the joint torques $\tau$ via the manipulator Jacobian $J_M$, so $\tau^p \neq 0$.

$$\tau = J_M^T F \tag{19}$$

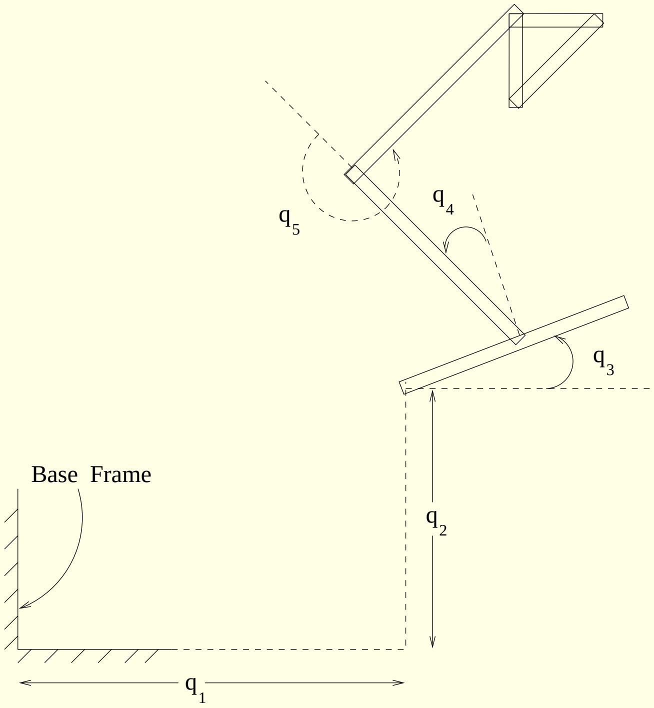
# Solving for Nonzero Passive Joint Torques

(Give example of what it means to map forces to joint torques?)
The joint torques are no longer zero, so in the expression:

$$\frac{\partial \tau^p}{\partial P} = \frac{\partial \tau^p}{\partial q}\frac{\partial q}{\partial P} + \frac{\partial \tau^p}{\partial \dot{q}}\frac{\partial \dot{q}}{\partial P} + \frac{\partial \tau^p}{\partial \ddot{q}}\frac{\partial \ddot{q}}{\partial P} \qquad (20)$$

we need $\frac{\partial \tau^p}{\partial q}$, $\frac{\partial \tau^p}{\partial \dot{q}}$, and $\frac{\partial \tau^p}{\partial \ddot{q}}$. How will we model the contact forces and calculate these derivatives?

# The Luxo Hopping Robot



The normal force exerted by the ground on the robot was:

$$N = \beta(y)(-ky - c\dot{y})$$

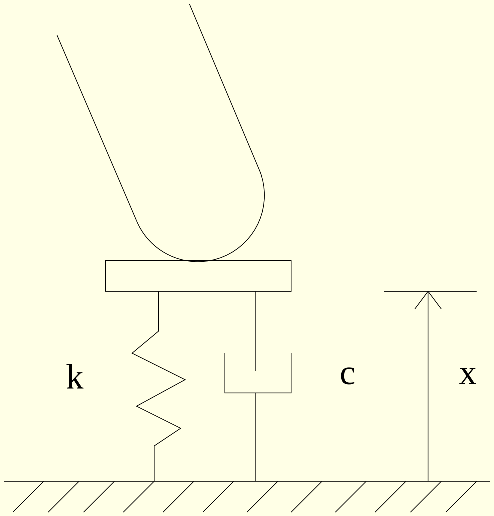
$$\beta(y) = \begin{cases} 1 & \text{if } y < p \\ \frac{3}{p^2}y^2 + \frac{-2}{p^3}y^3 & \text{otherwise} \end{cases}$$

$$\beta(0) = 0, \ \beta'(0) = 0$$

$$\beta(p) = 1, \ \beta'(p) = 0$$

Then $N \in C^1$ and it is calculated for points at either end of the base of the Luxo.

# The Contact Model



$$f_{fric,static} \leq \mu_s N$$
$$f_{fric,kinetic} = \mu_k N$$

To get some sense of the shape of this curve, but still have the friction be $C^1$, we use:

$$f = -.3N\gamma$$

$$\gamma(v_{t,norm}) = \begin{cases} sign(\dot{x}) & \text{if } |\dot{x}| \geq v_{max} \\ 1.5(\frac{\dot{x}}{v_{max}}) - 0.5(\frac{\dot{x}}{v_{max}})^3 & \text{otherwise} \end{cases}$$

$$\gamma(0) = 0, \ \gamma'(0) = 1$$
$$\gamma(v_{max}) = 1, \ \gamma'(v_{max}) = 0$$

# Mapping Normal and Friction Forces to Joint Torques

For the Luxo, the generalized force looked like:

$$F = [\ 0 \quad 0 \quad 0 \quad f_{fric} \quad N \quad 0\ ]^T \tag{21}$$

The Jacobian is a function of the joint variables but also the point for which you are writing it, so, for the Luxo first point is:

$$J_M^T = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \tag{22}$$

The resulting joint torques are:

$$\tau = \begin{bmatrix} f_{fric} \\ N \end{bmatrix} \tag{23}$$

For the Luxo second point, the Jacobian is:

$$J_M^T = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -l\sin q_3 & -l\cos q_3 & 0 \end{bmatrix} \tag{24}$$

So the joint torques due to external forces on point 2 are:

$$\tau = \begin{bmatrix} f_{fric} \\ N \\ -f_{fric} l \sin q_3 - N l \cos q_3 \end{bmatrix} \tag{25}$$

# Luxo Example Demo

Four motions: slide, rock, hop, flip

- $J = -c_1 q_1^p(t_f) + c_2(q_3^p(t_f) - 0)^2 + c_3 J_{penalty}$
- same as the slide, with more emphasis on first term, less on the torque penalty
- $J = c_1(q_3^p(t_f) - 0)^2 - \frac{1}{2} c_2 \int_0^{t_f} yTydt + c_3 J_{penalty}$
- $J = (q_3^p(t_f) - 2\pi)^2 + (q_2^p - 0)^2$

# General Contact and Collision Model

To implement $\tau = \sum J_M^T F$ generally, using virtual spring-damper systems, what do we need?

- a local coordinate frame to give a sense of the "normal" and "tangential" directions
- a general expression of distance, to replace $y$ in the expression for $N$
- velocity component in normal direction, for damper term in $N$
- velocity component in tangential direction, for $\gamma$ smoothing function

# General Distance

- A general expression for the distance between two rigid bodies is the distance $d$

  $$d = \|p_{rob} - p_{obs}\| = \|d_{vec}\| \tag{26}$$

  where $d_{vec}$ is a vector from the near point on the obstacle $(p_{obs})$ to the near point on the robot $(p_{rob})$.

- Also gives a definition for the normal direction, $\hat{u}_d$.

- $\beta(y)$ becomes $\beta(d)$, and need to define $d_{on}$, $d_{off}$i, where $d_{on} > d_{off}$

- $N = \beta(d)(k(d_{on} - d) - cv_{normal})\hat{u}_d$

Recall

$$f_{\textsf{fric}} = -.3N\gamma(\dot{x})$$

For replacement for $\dot{x}$, velocity component in tangential direction:

$$v_t = v - v_{\textsf{norm}}\hat{u}_d$$

$$f_{\textsf{fric}} = -.3N\gamma(\|v_t\|)\hat{u}_t$$

So, to map external forces to joint torques:

$$\begin{aligned}
\tau &= \sum J_M^T F \\
J_M^T F &= \begin{bmatrix} \dot{T}T^{-1} - \begin{matrix} 0 \\ J_w \times \vec{p} \end{matrix} \end{bmatrix} \begin{bmatrix} \vec{0} \\ f \end{bmatrix}^T \\
&= \begin{bmatrix} \dot{T}T^{-1} - \begin{matrix} 0 \\ J_w \times \vec{p} \end{matrix} \end{bmatrix} \begin{bmatrix} \vec{0} \\ N\hat{u}_d + f_{\textsf{fric}}\hat{u}_t \end{bmatrix} \\
J_M^T F &= J_v^T f = J_v^T (N\hat{u}_d + f_{\textsf{fric}}\hat{u}_t)
\end{aligned}$$

# Numerical Solution of Parametrized Optimization Problem

$$\begin{aligned}
(\tau^a, \ddot{q}^p) &= g(q, \dot{q}, \ddot{q}^a, \tau^p) \\
\left(\tau^a, \ddot{q}^a, \frac{\partial(\ddot{q}^p, \tau^a)}{\partial q}, \frac{\partial(\ddot{q}^p, \tau^a)}{\partial \dot{q}}, \frac{\partial(\ddot{q}^p, \tau^a)}{\partial \ddot{q}}\right) &= dg\left(q, \dot{q}, \ddot{q}^a, \tau^p, \frac{\partial \tau^p}{\partial q}, \frac{\partial \tau^p}{\partial \dot{q}}, \frac{\partial \tau^p}{\partial \ddot{q}}\right) \\
\tau^p &= J_v^T \left(\sum \left(N \hat{u}_d + f_{\text{fric}} \hat{u}_t\right)\right)
\end{aligned}$$

So, we need

$$\frac{\partial \tau^p}{\partial q} = \frac{\partial J_v^T}{\partial q} \sum \left(N \hat{u}_d + f_{\text{fric}} \hat{u}_t\right) + J_v^T \cdot \sum \left(\frac{\partial N}{\partial q} \hat{u}_d + N \frac{\partial \hat{u}_d}{\partial q} + \frac{\partial f_{\text{fric}}}{\partial q} \hat{u}_t + f_{\text{fric}} \frac{\partial \hat{u}_t}{\partial q}\right)$$

- To calculate the highlighted elements, we need $\frac{\partial d}{\partial q}$ which reduces to finding $\frac{\partial p_{\text{rob}}}{\partial q}, \frac{\partial p_{\text{obs}}}{\partial q}$

# Distance Derivatives

$$\begin{aligned} p_{\text{rob}} &= T(q)z \\ \frac{\partial p_{\text{rob}}}{\partial q} &= \frac{\partial T}{\partial q} z + T \frac{\partial z}{\partial q} \end{aligned}$$

- $z$ is $p_{\text{rob}}$ expressed in the local frame.
- As the robot moves, $z$ is expressed in local frame moves.
- Though the obstacle does not move, $p_{\text{obs}}$ is also a function of $p_{\text{rob}}$, and so is also a function of $q$.

For a pair of unique near points, they are the solution to the optimization problem:

$$\begin{array}{rcl} \min f(x) & = & \|p_{\textsf{rob}} - p_{\textsf{obs}}\|^2 = d \\ \text{s.t.} & & h(x) = 0, g(x) \leq 0 \end{array}$$

- $g_i(x) = 0$ are active inequality constraints, so we can treat the problem as just having equality constraints.
- The solution to the optimization is the solution to the following system of equations:

$$\begin{array}{rcl} \nabla f(x^*) + \lambda^T \nabla h(x^*) & = & 0 \\ h(x^*) & = & b \end{array}$$

$$\begin{aligned}\nabla f(x^*) + \lambda^T \nabla h(x^*) &= 0 \\ h(x^*) &= b\end{aligned}$$

Depending on the shapes used to describe robots and obstacles, we can factor this to get

$$\begin{aligned}Qx - D^T\lambda &= 0 \\ Ax &= b\end{aligned}$$

$$\underbrace{\begin{bmatrix} Q & -D^T \\ A & 0 \end{bmatrix}}_{M} \begin{bmatrix} x \\ \lambda \end{bmatrix} = \begin{bmatrix} 0 \\ b \end{bmatrix}$$

So near points are in $\vec{x}$, from

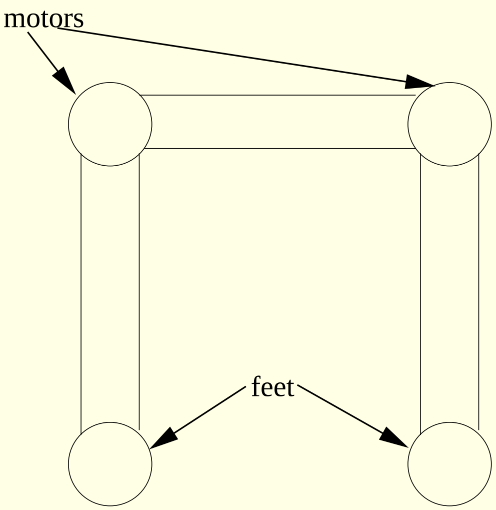
$$\begin{bmatrix} \vec{x} \\ \lambda \end{bmatrix} = M^{-1} \begin{bmatrix} 0 \\ b \end{bmatrix}$$

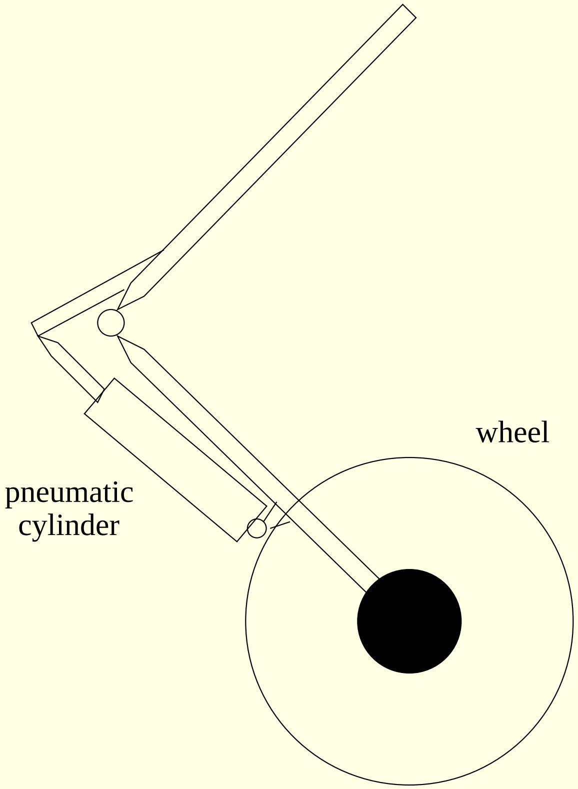
But, we have an algorithm to solve for $p_{\text{rob}}$ and $p_{\text{obs}}$, what we need is the derivatives of the $p_{\text{rob}}$ and $p_{\text{obs}}$ with respect to $q$:

$$
\begin{aligned}
M\begin{bmatrix} x \\ \lambda \end{bmatrix} &= \begin{bmatrix} 0 \\ b \end{bmatrix} \\
\frac{\partial M}{\partial q}\begin{bmatrix} x \\ \lambda \end{bmatrix} + M\begin{bmatrix} \frac{\partial x}{\partial q} \\ \frac{\partial \lambda}{\partial q} \end{bmatrix} &= \begin{bmatrix} 0 \\ \frac{\partial b}{\partial q} \end{bmatrix}
\end{aligned}
$$

- $\frac{\partial M}{\partial q}$ and $\frac{\partial b}{\partial q}$ can be defined from the expresions we have for them
- $\begin{bmatrix} x \\ \lambda \end{bmatrix} = M^{-1}\begin{bmatrix} 0 \\ b \end{bmatrix}$, so long as $M^{-1}$ exists
- We can solve for $\begin{bmatrix} \frac{\partial x}{\partial q} \\ \frac{\partial \lambda}{\partial q} \end{bmatrix}$, where $\frac{\partial x}{\partial q}$ will contain $\frac{\partial p_{\text{rob}}}{\partial q}, \frac{\partial p_{\text{obs}}}{\partial q}$

# Proposed Research





- fully generalize collision and contact model to general polytopes
- find planar gaits for tumbler and rocking pneumatic robot
- find a three-dimensional gait for the rocking pneumatic robot