Control Issues in Underactuated Robotic Manipulation

Kevin M. Lynch
Laboratory for Intelligent Mechanical Systems
Department of Mechanical Engineering
Northwestern University
Evanston, IL USA

Workshop on Dynamics and Control
Brussels, Belgium
July 2002

Northwestern University

Evanston: main campus

Chicago: medical school, law school

Evanston →

LIMS

Laboratory for Intelligent Mechanical Systems

With Ed Colgate and Michael Peshkin

Research areas:

- Robotic manipulation
- Motion planning for underactuated mechanical systems
- Human-robot interfaces
- Haptic interfaces

Robotic Manipulation

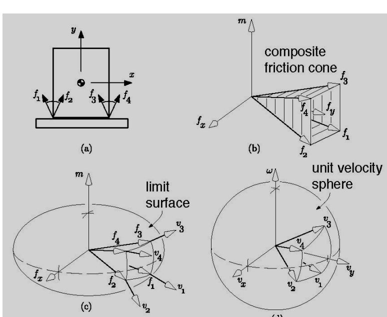
The process of controlling the position (state) of one or more objects through contact forces by a robot.

Q: Where can a robot place a part?

Standard answer: Pick-and-place
—kinematic workspace, dexterous workspace

Other answers: Allow pushing, rolling, throwing, striking...
—dynamic workspace?

Quasistatic Pushing Mechanics

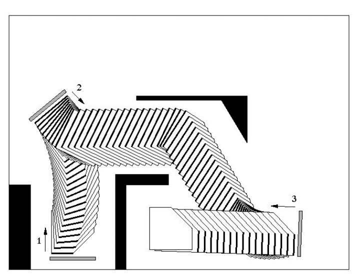


Limit surface determined by friction between object and support surface

(Goyal, Ruina, Papadopoulos 1991)

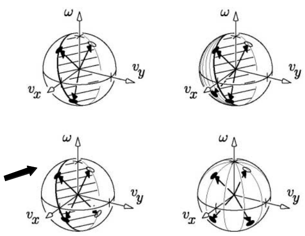
Motion Planning

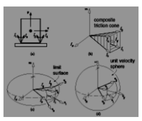
Allows placing parts by open-loop stable pushing
(Lynch and Mason 1996)



Local Controllability

Depends on:
- Part geometry
- Support friction (friction centroid)
- Pushing friction coefficient





Almost every part is locally controllable by pushing with a 2-DOF pushing point

Underactuated Manipulation

Underactuated robotic manipulation occurs when a robot controls more degrees-of-freedom of an object (or objects) than the robot has actuators.

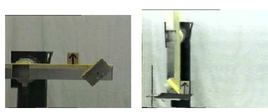
Extra object DOF: rolling, slipping, flight

Examples:
- —Robot assembly
- —Parts feeders
- —Batting, juggling, pushing, rolling, throwing
- —Flexible objects

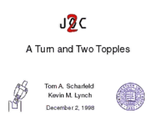
Why?
- —Inexpensive, low-DOF robots
- —Shift system complexity from hardware to motion planning and control

Examples

Sony APOS parts feeder



1-joint rolling and throwing (automatically planned, open loop)



Conveyor-based parts feeder

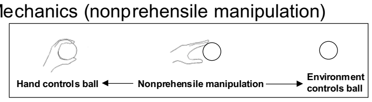
Planar juggling

Related Work

- Rolling
  - Montana (1988)
  - Dai and Brockett (1991)
  - Hristu-Varsakelis (2001)
  - Li and Canny (1990)
  - Bicchi and Sorrentino (1995)
  - Choudhury and Lynch (2002)
- Juggling
  - Buhler and Koditschek (1990)
  - Schaal and Atkeson (1993)
  - Brogliato and Zavala-Rio (2000)
  - Rizzi and Koditschek (1993)
  - Bishop and Spong (1999)
  - Lynch and Black (2001)
- Tapping
  - Higuchi (1985)
  - Huang and Mason (1998)
- Pushing
  - Mason (1986)
  - Alexander and Maddocks (1993)
  - Peshkin and Sanderson (1988)
  - Lynch and Mason (1996)
- Slipping
  - Trinkle (1992)
  - Erdmann (1996)

Underactuated Manipulation

- Mechanics (nonprehensile manipulation)



  - —Pushing, rolling, slipping, throwing, batting
  - —Friction, restitution, Newton s laws
  - —Object geometry, manipulator shape and motion constraints, unilateral constraints, changing dynamics (hybrid)
- Controllability
  - —Reachability, feedability
- Motion Planning
- Feedback Control

Controllability

Robot state: $z_R = (q_R, dq_R/dt) \in M_R = TC_R$

Part state: $z_P = (q_P, dq_P/dt) \in M_P = TC_P$

System state: $z = (z_R, z_P) \in M_R \times M_P$

Underactuated manipulation: $dim(C_R) < dim(C_P)$

Given initial state z and time T, what is the set of reachable states $R(z,T)$?

Part only: $R_P(z,T)$

Controllability (cont.)

- Accessible: $R_P(z,\dagger T)$ is a full-dimensional subset of $T_{z_P} M_P$ for some T>0.
- Feedable: $z_{Pg} \in R_P(z,\dagger T)$ for some T>0 and any $z \in U$, the set of initial possible states.
- Controllable: $z_{Pg} \in R_P(z,T)$ for some finite T and any z, z_{Pg}.
- Locally controllable: $z_P \in int(R_P(z,\dagger T))$ for all z_P and T>0. (Only possible at zero velocity.)
- Equilibrium controllable: $R_P(z,\dagger T)$ contains a neighborhood of q_P at zero velocity.

Controllability (cont.)

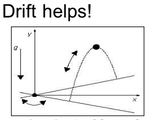
- Accessible: $R_P(z,\dagger T)$ is a full-dimensional subset of $T_{z_P} M_P$ for some T>0.
- Feedable: $z_{Pg} \in R_P(z,\dagger T)$ for some T>0 and any $z \in U$, the set of initial possible states.
- Controllable: $z_{Pg} \in R_P(z,T)$ for some finite T and any z, z_{Pg}.
- Locally controllable: $z_P \in int(R_P(z,\dagger T))$ for all z_P and T>0. (Only possible at zero velocity.)
- Equilibrium controllable: $R_P(z,\dagger T)$ contains a neighborhood of q_P at zero velocity.

Controllability (cont.)

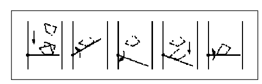
- Accessible: $R_P(z,\dagger T)$ is a full-dimensional subset of $T_{z_P} M_P$ for some T>0.
- Feedable: $z_{Pg} \in R_P(z,\dagger T)$ for some T>0 and any $z \in U$, the set of initial possible states.
- Controllable: $z_{Pg} \in R_P(z,T)$ for some finite T and any z, z_{Pg}.
- Locally controllable: $z_P \in int(R_P(z,\dagger T))$ for all z_P and T>0. (Only possible at zero velocity.)
- Equilibrium controllable: $R_P(z,\dagger T)$ contains a neighborhood of q_P at zero velocity.

Single Input Systems

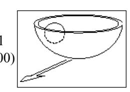
- Minimum actuator systems
- Often globally controllable but not locally controllable
- Drift helps!



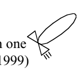
Planar juggler (Bühler and Koditschek 1990; Zavala-Rio and Brogliato 1999; Lynch and Black 2001)



1JOC conveyor parts feeder (Akella *et al.* 2000)



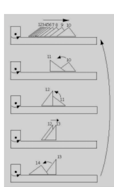
Ball in an asymmetric bowl (Choudhury and Lynch 2000)

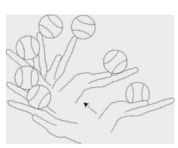


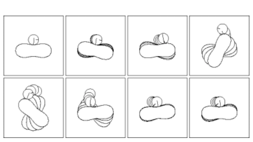
Planar body with one thruster (Lynch 1999)

Single Input Systems (cont.)

Repetitive throwing and catching







Butterfly

A Simple Model

Robot shapes the natural dynamics of the environment.

A simple single input model:

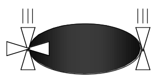
$$dz/dt = f(z) + g(z)u$$

z	system state
f	drift vector field (natural dynamics)
g	control vector field
u	control

(though often the systems are hybrid)

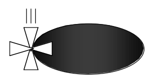
Controllability

planar body, three thrusters



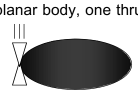
linearly controllable

planar body, two thrusters



locally controllable

planar body, one thruster



not locally controllable (Lewis 1997, Manikonda and Krishnaprasad 1997), but globally controllable (Lynch 1999)

Global Controllability

$dz/dt = f(z) + g(z)\,u, \quad z \in M$

Involutive closure of $\{f, g\} = Lie(\{f, g\})$

<u>*Theorem*</u> *(Lian et al. 1994)* If the drift vector field f is Weakly Positively Poisson Stable (WPPS) and $Lie(\{f, g\}) = T_zM \;\; \forall z \in M$, then the system is controllable.

Accessibility + Poisson Stability $\Rightarrow$ Controllability

(Jurdjevic and Sussmann 1972; Lobry 1974; Brockett 1976; Bonnard 1981; Jurdjevic 1997)

Poisson Stability

Flow of drift field: $\Phi^f : M \times \Re \to M; \quad (z,t) \to \Phi^f(z,t)$

The point z is *Positively Poisson Stable* (PPS) for f if for all $T>0$ and any neighborhood $B(z)$ of z, there exists a time $t>T$ such that $\Phi^f(z,t) \in B(z)$.

f is PPS if the set of PPS points is dense in M.

f is WPPS if for all $z \in M$, any neighborhood $B(z)$ of z, and all $T>0$, there exists $t>T$ such that $\Phi^f(U_z, t) \cap B(z) \neq \emptyset$.

Examples: a swing (no damping), satellite attitude, ball rolling in a bowl

Controllability

satellite attitude, two thrusters

.

locally controllable (Crouch 1984)

satellite attitude, one thruster

not locally controllable, but globally controllable (Crouch 1984, Jurdjevic 1997)

Extension

If the drift is not WPPS, global controllability can be established by:

<u>*Continuous fountain condition*</u> (Caines and Lemch 1999)

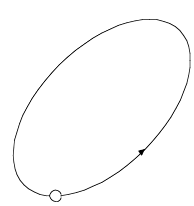
Locally accessible states form an open subset of the state space. (Neither stronger nor weaker than local accessibility.)

Plus some form of *control recurrence*, e.g.,

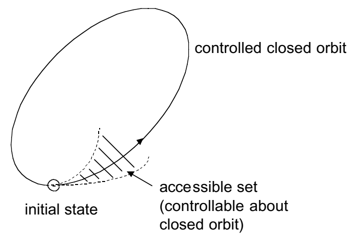
$z = \Phi(z, T, u)$ flow under the control u

for some control u and time T.

Global Controllability



Global Controllability



Motion Planning and Control

A trajectory of a drift-free, controllable system is a *nonsingular loop* if

1) trajectory returns system to initial state, and
2) system is linearly controllable about the trajectory.

(Sontag 1993; Sussmann 1993; Wen 1996)

- Similar to the controlled closed orbit of systems with drift.
- Generic loops are nonsingular for strongly accessible systems (Sontag 1993).

Control Algorithm

Given:

- Control parameterization $u = (u_1, u_2, \;, \; u_k) \in U$
- End-state map $z_2 = f(z_1,u)$
- Goal state z_g
- Cost function $V(z)$ (control Lyapunov function)

1. Calculate recurrent control $u^r(z)$.
2. $u = u^r - \alpha\, (\partial V(f(z,u)) / \partial u)\, |_{u=u^r}$, $\alpha > 0$.
3. Execute control u .
4. Go to 1.

Summary

D is an open connected subset of M such that $\exists\; u^r(z) \in int(U)\; \forall\; z \in D.$

Controllability on D

$z \in int \{ f(z,u) \mid u \in B(u^r(z)) \}$

$B(u^r(z))$ is any neighborhood of $u^r(z)$

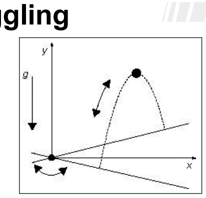
Stabilization of any point in D

- Define a distance between current and goal state.
- Perturb $u^r(z)$ to reduce the distance.
- Asymptotic stabilization if $u^r(z)$ gives nonsingular loops.

Example: Juggling

Point mass puck,
zero thickness batter.

$z = (x, y, x , y)$



(Buhler and Koditschek 1990; Zavala-Rio and Brogliato 1999)

One bat:

$z_{j+1} = f_1(z_j,u_1)$ $u_1 = (t_1, \omega_1, t_f)$

t = flight time, ω = impact vel

Two bats:

$z_{j+2} = f_2(z_j,u_2)$ $u_2 = (t_1, \omega_1, t_2, \omega_2, t_f)$

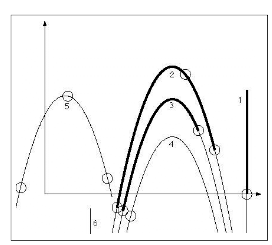
D is the set of reversible states; puck can be batted back and forth along the trajectory.

Reversible impact states: $x\, x + y\, y = 0$. (*)

Reversible States

Bold: reversible states

Circles: impact points



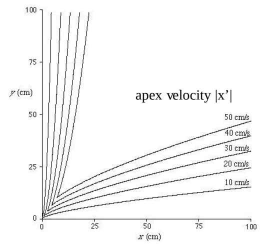
(*) is cubic in flight time. At most three real solutions to reversing impact states.

Reversible States

D_A 3-D open set of reversible apex states

D 4-D open set of reversible states

g = -981 cm/s^2



$(-gx^2/2|x\,|) > y > (-3(x|x\,|g)^{2/3} - 2|x\,|^2)/(2g)$

Controllability and Stabilization

Proposition The point mass puck is controllable on D.

Follows from rank$(\partial f_2 / \partial u)|_{u_2^r} = 4$.

Proposition Under the two-bat control law, the system asymptotically converges to $z_g = (x_g, y_g, 0, 0) \in D$ from any $z \in D$ for any positive definite $V(z)$ where $\partial V/\partial z = 0$ only at z_g.

Proof: $\partial V/\partial u = (\partial V/\partial z)(\partial f_2 / \partial u)$

One Bat Control Law

For a single bat, $u_1^r(z)$ is a reversing control if
$rev(z) = (x, y, -x, -y) = f_1(z, u_1^r(z))$.

Choose $V(z) = (z - z_g)^T W (z - z_g)$
$V(rev(z)) = V(z)$

Just three controls: rank$(\partial f_1 / \partial u)|_{u_1^r} = 3 < 4$.

May not be able to reduce distance to goal after a single bat.

Stabilizability

Lemma For all $z_0 \in D$, $z_0 \neq z_g$, if $(\partial V/\partial u_1)|_{u_1^r} = 0$ at z_0, then $(\partial V/\partial u_1)|_{u_1^r} \neq 0$ at $rev(z_0)$.

Proposition Under the one-bat control law, the system asymptotically converges to $z_g = (x_g, y_g, 0, 0) \in D$ from any $z \in D$.

Control by Optimization

- Object state z

¥ Finite parameterization of the control u
¥ Define an endpoint mapping f, $z_f = f(z_i, u)$
¥ Define an objective function $V(z_f)$
¥ Given an initial control, iteratively modify u to minimize $V(f(z_i, u))$ using the gradient $\nabla_u V$ and possibly the Hessian $\nabla^2_{uu} V$

Initial control guess: u^r (reversing control)

Variants of this approach (continuation, homotopy methods, MPC)
Divelbiss and Wen 1992
Sontag 1993
Sussmann 1993
Fernandes, Gurvits, and Li 1994
Zefran and Kumar 1995
Lizarralde and Wen 1997
Lynch and Mason 1997

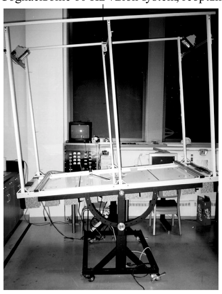
Details

- Goal apex state: $z_g = (x_g, y_g, x'_g, y'_g) = (x_g, y_g, 0, 0)$
- Control u_1: pre-impact flight time, impact speed, post-impact flight time
  - —implemented by 4th-order polynomial arm trajectory
- Endpoint mapping f_1 based on Poisson restitution (Wang and Mason 1992) with known restitution
- Quadratic cost function

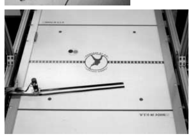
$$V(z_f) = (z_f - z_g)^T W (z_f - z_g)$$

Experimental Setup

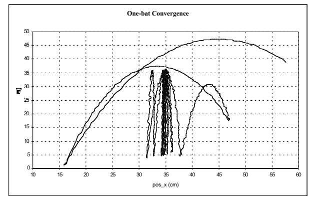
Cognachrome 60 Hz vision system; reoptimize control at 60 Hz



adjustable gravity



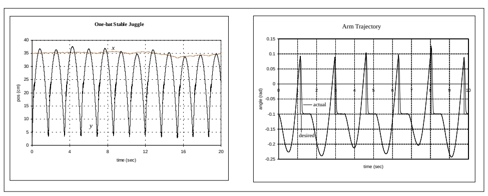
Experiment



raw vision data: goal apex (35 cm, 35 cm)

Plastic disk, radius = 3.8 cm, Gravity = *g sin 5º*
Arm width = 5 cm, length = 60 cm
Friction coefficient = 0.1, restitution coefficient = 0.45

Experiment



x(*t*), *y*(*t*) arm motion

Discussion

Advantages
- Provably stable
- Estimation of the basin of attraction
- Controllability and stabilization closely tied

Limitations
- Real-time calculation of forward dynamics and gradient
- How to find recurrent control?

Conclusion

- Minimize actuation and hardware
- Reduce cost
- Transfer complexity from hardware to control

Drawbacks
- Slower
- Heavier computational demand (and possibly sensory)
- More complex control