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

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
- 1 joint rolling and throwing arm (automatically planned, open loop)
- Conveyor-based parts feeder
- Sony APS parts feeder
- Planar juggling

Related Work

Rolling
- Montana (1988)
- Li and Canny (1990)
- Hristu-Varsakelis (2001)

Juggling
- Buhler and Koditschek (1990)
- Schaaf and Atkeson (1993)
- Bishop and Spong (1999)
- Lynch and Black (2001)

Pushing
- Mason (1999)
- Peshkin and Sanderson (1998)
- Lynch and Mason (1996)

Slipping
- Troccoli (1992)
- Eichmiller (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

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: $R_d(z, T)$ is a full-dimensional subset of $T_z M_p$ for some $T > 0$.
- Feedable: $z_p \in R_d(z, T)$ for some $T > 0$ and any $z \in U$, the set of initial possible states.
- Controllable: $z_p \in R_d(z, T)$ for some finite $T$ and any $z, z_p$.
- Locally controllable: $z_p \in \text{int}(R_d(z, T))$ for all $z_p$ and $T > 0$. (Only possible at zero velocity.)
- Equilibrium controllable: $R_d(z, T)$ contains a neighborhood of $q_p$ at zero velocity.

Single Input Systems

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

A Simple Model

Robot shapes the natural dynamics of the environment.

A simple single input model:

$$\frac{dz}{dt} = f(z) + g(z)u$$

- $z$ 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

\[
\frac{dz}{dt} = f(z) + g(z) u, \quad z \in M
\]

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

Theorem (Lian et al. 1994) If the drift vector field \(f\) is Weakly Positively Poisson Stable (WPPS) and \(\text{Lie}\((\{f, g\})\) = T_z M \forall z \in M\), then the system is controllable.

Accessibility + Poisson Stability \implies\ Controllability
(Jurdjevic and Sussmann 1972; Lobry 1974; Brockett 1976; Bonnard 1981; Jurdjevic 1997)

Poisson Stability

Flow of drift field: \(\Phi^f: M \times \mathbb{R} \to M; (z,t) \to \Phi^f(z,t)\)

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 \(\Phi^f(z,t) \notin B(z)\).

\(f\) 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 \(\Phi^f(U_z, f) \cap B(z) = \emptyset\).

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

Controllability

satellite attitude, two thrusters
locally controllable

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:

Continuous fountain condition (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

controlled closed orbit

initial state
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:
1. Control parameterization \( u = (u_1, u_2, \ldots, u_k) \in U \)
2. End-state map \( z_2 = f(z_1,u) \)
3. Goal state \( z_g \)
4. Cost function \( V(z) \) (control Lyapunov function)

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

Summary

\( D \) is an open connected subset of \( M \) such that \( \exists u(z) \in \text{int}(U) \forall z \in D \).

Controllability on \( D \)
\( z \in \text{int} \{ f(z,u) \mid u \in B(u(z)) \} \)
\( B(u(z)) \) is any neighborhood of \( u(z) \)

Stabilization of any point in \( D \)
1. Define a distance between current and goal state.
2. Perturb \( u(z) \) to reduce the distance.
3. Asymptotic stabilization if \( u(z) \) gives nonsingular loops.

Example: Juggling

Point mass puck, zero thickness batter.
\( z = (x, y, \dot{x}, \dot{y}) \)
One bat:
\( z_{j+1} = f_1(z_j,u_j) \quad u_j = (t_1, \omega_1, t_f) \)
\( t = \) flight time, \( \omega = \) impact vel
Two bats:
\( z_{j+2} = f_2(z_j,u_j) \quad u_j = (t_1, \omega_1, t_2, \omega_2, t_f) \)
\( D \) is the set of reversible states; puck can be battered back and forth along the trajectory.
Reversible impact states: \( xx + yy = 0 \). (*)

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
- $g = -981 \text{ cm/s}^2$

Controllability and Stabilization

**Proposition** The point mass puck is controllable on $D$.

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

**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 $\frac{\partial V}{\partial z} = 0$ only at $z_g$.

Proof: $\frac{\partial V}{\partial u} = (\frac{\partial V}{\partial z}) (\frac{\partial f_2}{\partial u})$

One Bat Control Law

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

Choose $V(z) = (z - z_g)^TW(z - z_g)$

$V(r(z)) = V(z)$

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

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

Control by Optimization

- Object state $z$
- Finite parameterization of the control $u$
- Define an endpoint mapping $f_1(z, u)$
- Define an objective function $V(z)$
- Given an initial control, iteratively modify $u$ to minimize $V(f_1(z, u))$ using the gradient $\nabla_u V$ and possibly the Hessian $\nabla^2_{uu} V$

Initial control guess: $u'$ (reversing control)

Variants of this approach (continuation, homotopy methods, MPC)
Divelbiss and Wen 1992
Sontag 1993
Sussmann 1993
Zeltzer and Kumar 1993
Lizarralde and Wen 1997
Lynch and Mason 1997

Stabilizability

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

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

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
  $V(z_2) = (z_2 - z_g)^TW(z_2 - z_g)$
Experimental Setup

Cognachrome 60 Hz vision system; reoptimize control at 60 Hz

Adjustable gravity

Experiment

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

One-bat Convergence

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