An Experimental Test Bed
For Robotic Grasping

John Behmer
Thesis Presentation

Rensselaer Polytechnic Institute
CS Robotics Lab
Outline

• Control System Design
  • Disturbance Rejection
  • Trajectory Formation
  • Selecting Control Gains
• Camera Design
  • Calibration Procedure
  • Coordinate Frame Transformations
• Database and Software Setup
System Dynamics

- Centrifugal/Coriolis negligible at low speeds
- Gravity is simple given the forward kinematics
- Friction is more difficult
  - Highly non-linear
  - Hard to estimate

\[ \tau = M(q) \ddot{q} + C(q, \dot{q}) + G(q) + B(q, \dot{q}) \]
Velocity Feedback

- Differentiate position using Backward Euler

\[ \dot{q}[k] = q[k] - q[k-1] \Delta t \]

- Filtering Methods
  - Simple Moving Average
  - Exponential Moving Average
  - Kalman Filter
Friction Compensation

- Difficult due to poor velocity feedback
- Tough to model non-linearities
Controller Design

- Gravity + PID, neglect friction for now
- PID Output
  - Torques
    \[ \tau = G(q) + (K_p e + K_d \dot{e} + K_i \int e \, dt) \]
  - Accelerations
    \[ \tau = G(q) + M(q)(K_p e + K_d \dot{e} + K_i \int e \, dt) \]
- Jacobian for Cartesian space
Gain Selection

- Manual Selection
- Inertia Parameters
  - Similarity Transformation
  - Parallel Axis Theorem

\[ M_i^j = R_i^j M_i^i (R_i^j)^T + m_i \begin{bmatrix} y^2 + z^2 & xy & xz \\ xy & x^2 + z^2 & yz \\ xz & yz & x^2 + y^2 \end{bmatrix} \]
Interpolating Orientation

• Using axis-angle method
  • Two solutions for each rotation matrix
    – Axis +, Angle + ↔ Axis -, Angle -
  • Singularities at 0, 180 degrees

• Eventually implement quaternions + SLERP
  • Two solutions for each rotation matrix
  • Can become unstable near 0, 180 degrees but solution stays finite
Camera Calibration

- Wanding and Ground Frame
  - Cameras determine position/orientation in space
- 2\textsuperscript{nd} calibration transforms data into WAM frame
  - Non-linear least squares
  - $i =$ marker index, $j =$ configuration index

\[
r_{ij} = HT^{-1}_{\text{base}} p_{ij} - HT^{1}_{0}(q_j) s_i
\]
Coordinate Frame Transformations

- Left to right handed system
  - Flip any one axis (we choose the z-axis)
  - Axis +, RH rule ↔ Axis -, LH rule (same angle)
- Use transformation matrix into WAM frame

\[
\epsilon = \begin{bmatrix}
\epsilon_1 \\
\epsilon_2 \\
\epsilon_3 \\
\epsilon_4
\end{bmatrix} = \begin{bmatrix}
k_x \sin(\frac{\theta}{2}) \\
k_y \sin(\frac{\theta}{2}) \\
k_z \sin(\frac{\theta}{2}) \\
\cos(\frac{\theta}{2})
\end{bmatrix}
\]

\[
\epsilon^r = \begin{bmatrix}
\epsilon_1^l \\
\epsilon_2^l \\
-\epsilon_3^l \\
\epsilon_4^l
\end{bmatrix}
\]

\[
p^r = \begin{bmatrix}
p_x^l \\
p_y^l \\
-p_z^l
\end{bmatrix}
\]
System Diagram

- RPI Network
- dVC3d
- MATLAB
- xPC Target
  - Serial Port
  - Softing CAN Ethernet Card
- Private Switch (1 Gbit/s)
- Grasp DB
  - Ubuntu, MySQL
- WAM
  - (torque commands)
- Hand
  - (velocity commands)
Grasp Database

- Experiments – description, time stamp, type
- Calibration – mean/max error, transformation matrix
- Frames – elapsed time, tracking frame captured
- Trackables – marker definitions; position, orientation, marker positions on each frame
- Raw markers – position on each frame
- WAM Data – joint/Cartesian space actual/desired positions/velocities, applied torques
Future Work

- Quaternions + SLERP
- Real-time Hand Controller
- Grasping Experiments
- dVC3d Simulations
- Improve Sample Rate/Camera Synchronization
- Velocity Blends in Trajectory
- Object Avoidance
The End

- Thanks! Questions?