WAM/Hand Simulink Controller

Rensselaer Polytechnic Institute
Robotics Lab

John Behmer

Overview

- Background on Barrett Arm and Hand
- Required Matlab Toolboxes
- System Diagram
- The Controller
- Some Challenges
- What's left to be done...

Background on Barrett Arm/Hand

- Barrett Arm
  - 4 DOF and 7 DOF models (7 DOF includes wrist)
  - Human like, back-drivable, low friction
- Barrett Hand
  - 3 Fingers and a “Spread Joint”
  - Spread is back-drivable, fingers are not
  - Strain gauges (optional) in fingers

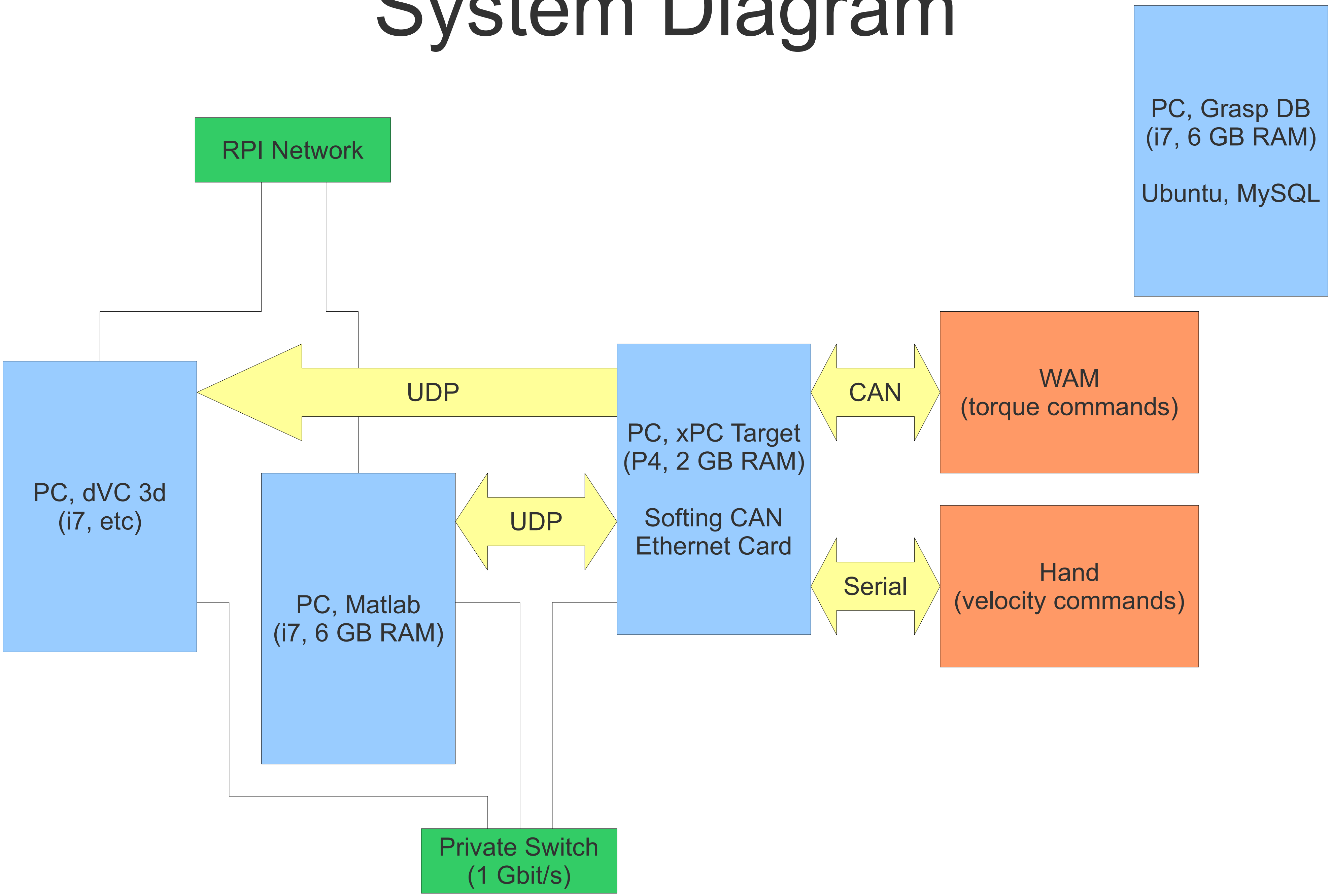
How did we get here?

- Barrett's code is ugly :-)
- UNM (Greg Starr and Matt Courtney)
  - Matlab/Simulink Controller for the WAM
  - Low level functionality
- Used this as our base
  - Gravity compensation
  - Convert 7 DOF to 4 DOF

Matlab Toolboxes

- Matlab – this one's easy
- Simulink – block diagram programming for dynamic systems, simpler than C code
- Real Time Workshop – Generates C code from Simulink models
- xPC Target – a very fast real-time OS which the compiled C code can be downloaded to

System Diagram



The Controller

- 500 Hz arm, ~30 Hz hand
- Gravity Compensation
- No friction/coriolis/centrifugal compensation
  - Instead use really high gains with really small steps
- 3 modes of operation
  - Joint Space
  - Cartesian Space
  - Impedance (a work in progress by Chris)

Joint Space

- Simple PID on top of gravity compensation
  - q = joint position
  - e = joint position error

$$e = q_{des} - q \qquad \dot{e} = \dot{q}_{des} - \dot{q}$$

$$tau = G(q) + K_p e + K_d \dot{e} + K_i \int e \, dt$$

Cartesian Space

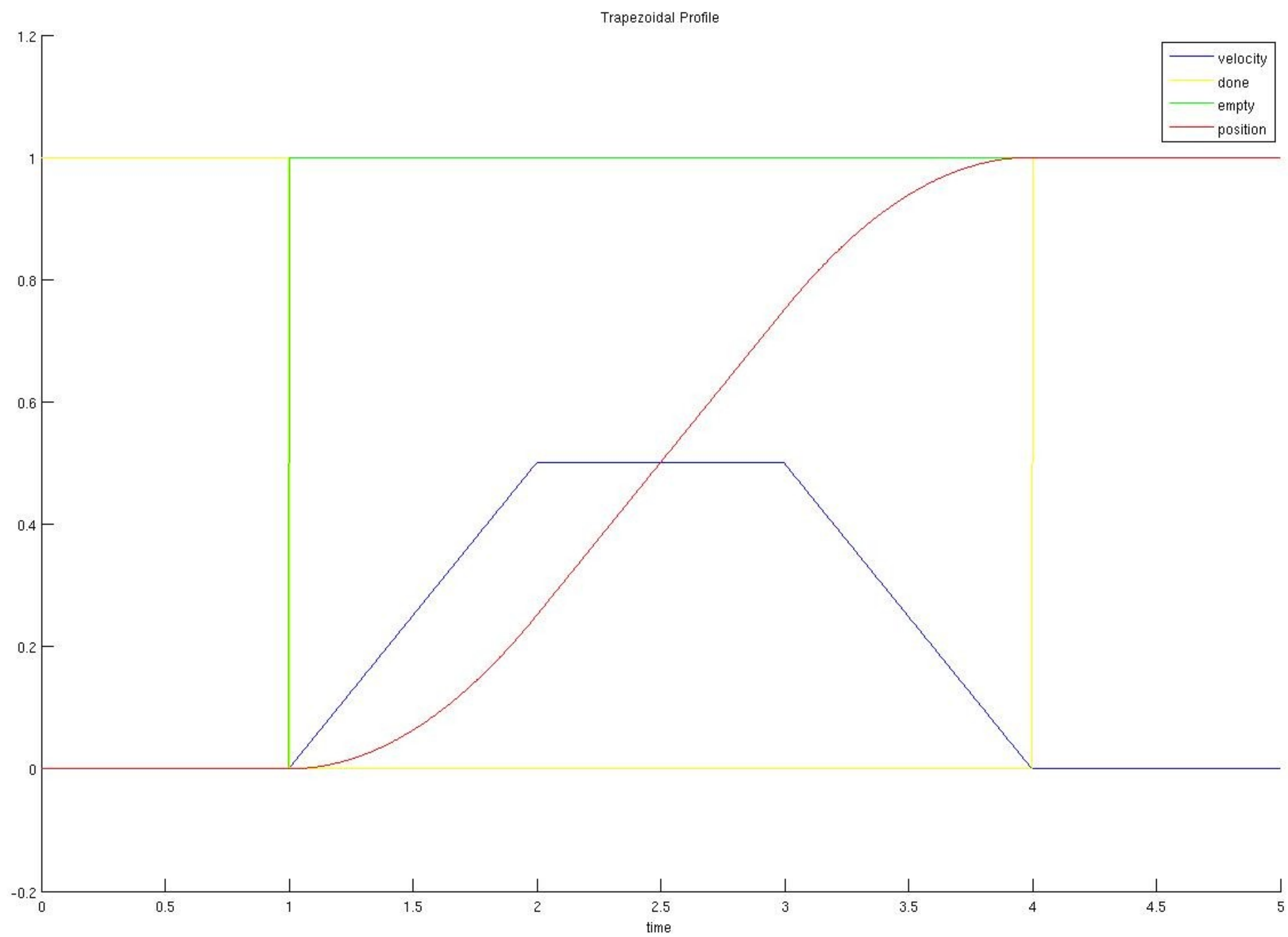
- We already have joint space control
- Convert Cartesian Space back into Joint Space
  - Using inverse kinematics? No
  - Using the Jacobian (4 dof) – with very small dt, the approximation is not too bad
- End result: change in q gives us our desired change in w

$$\dot{w} = J\,\dot{q} \qquad \frac{dq}{dt} = J^{-1}\frac{dw}{dt}$$

$$\dot{q} = J^{-1}\,\dot{w} \qquad dq = J^{-1}\,dw$$

Forming Trajectory

- Trapezoidal Profile
  - In Cartesian Space *or* Joint Space



Some Challenges

- Maximum Joint Torque
  - Solution: Set WAM property
- Velocity Feedback
  - Solution: TBD (Kalman filter?)
- Steady State Errors
  - Solution: Small time steps, high gains
  - Excellent tracking ability
  - Small vibration though

Where are we now?

- Very nice UDP interface from Matlab
  - 3 modes as described previously
- Matlab feeds high level control commands, and receives feedback
- Streams from cameras at the same time
- Saves all data in a very large struct
  - Can be pushed out to grasp database (almost)

What's left?

- A ton of things. Here are a few...
  - Real time controller for the hand
  - Filter velocity from arm, dampen vibrations
  - Setup grasp database to hold all the new data
  - Run camera system in a parallel thread
  - Grasping experiments comparing to dvc 3d
  - Camera calibration to align coordinate frames

The End

- Thanks! Questions?