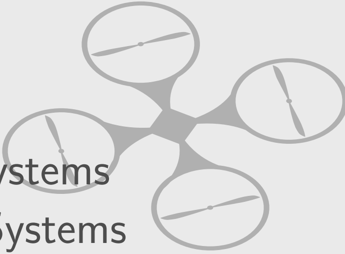
March 27, 2018

Hybrid Genetic Fuzzy Systems for Control of Dynamic Systems



Nick Stockton

Outline

Introduction

Motivation

- Nonlinear systems are hard to control
- But humans do it all the time
- Translate human intuition into automatic control
- Apply basic machine learning to improve on human intuition where possible

Problem Statement

Develop a methodology for creating and tuning controllers for complex dynamic systems. Wherever feasible, use evolutionary strategies to tune (or learn) the controller for better performance. Carefully define what "better" means.

Fuzzy Logic (1/5)

- Why Fuzzy?

Fuzzy Logic (1/5)

- Why Fuzzy?
  - Universal approximator (arbitrary non-linearity)

Fuzzy Logic (1/5)

- Why Fuzzy?
  - Universal approximator (arbitrary non-linearity)
  - Uncertainty handling baked in

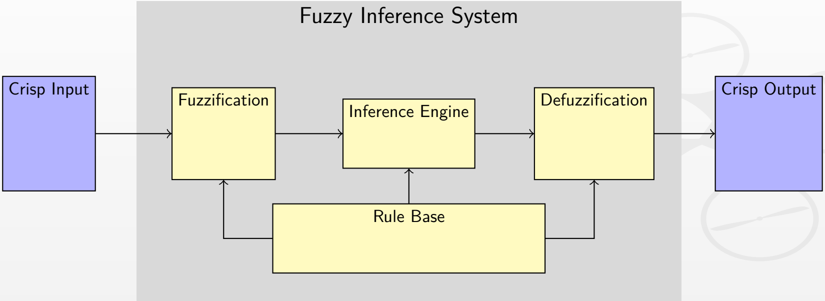
Fuzzy Logic (1/5)

- Why Fuzzy?
    - Universal approximator (arbitrary non-linearity)
    - Uncertainty handling baked in
    - Computationally inexpensive (if done correctly)

Fuzzy Logic (1/5)

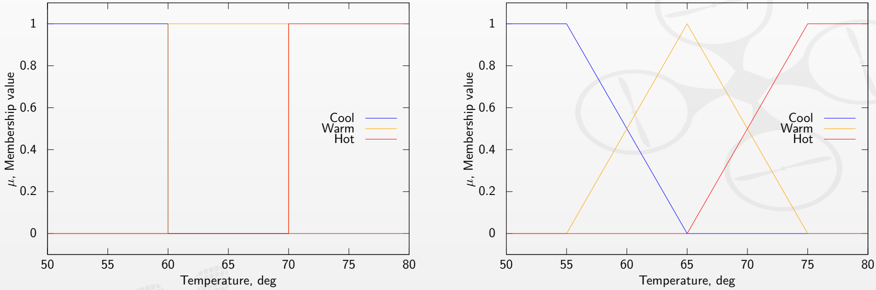
- Why Fuzzy?
  - Universal approximator (arbitrary non-linearity)
  - Uncertainty handling baked in
  - Computationally inexpensive (if done correctly)
  - Simple and intuitive to grasp concepts

Fuzzy Logic (2/5)

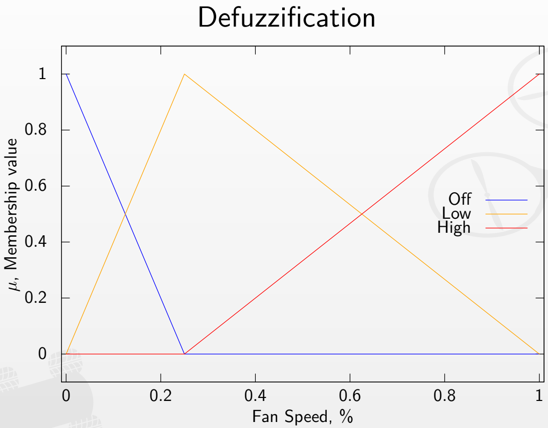


Fuzzy Logic (3/5)

Traditional vs. Fuzzy logic

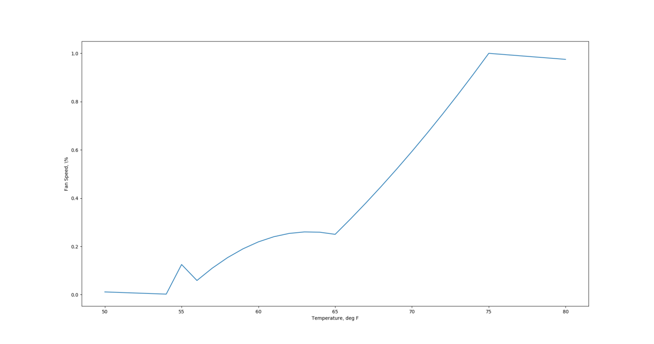


Fuzzy Logic (4/5)



Fuzzy Logic (5/5)

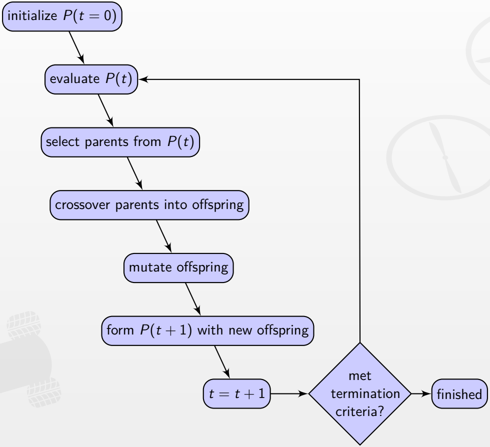
IF	*temperature* is	COOL,	THEN	*fan speed* is	OFF
IF	*temperature* is	WARM,	THEN	*fan speed* is	LOW
IF	*temperature* is	HOT,	THEN	*fan speed* is	HIGH



Genetic Algorithm

- To solve a problem with a GA you need:
  - A way to encode a solution as a genetic unit (chromosome)
  - A way to create an initial population of candidates
  - A grading or cost function to assess individual fitness
    - We are trying for "survival of the fittest", but what "fit" means is up to us
  - Genetic operators
  - General guidelines for the mechanics of the GA
    - How many individuals make up a population
    - How many generations until we stop looking
    - How do we apply the genetic operators to each population

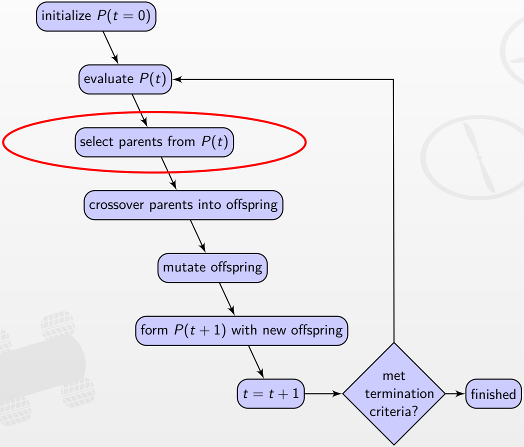
GA Operational Flow



Chromosomal Representation

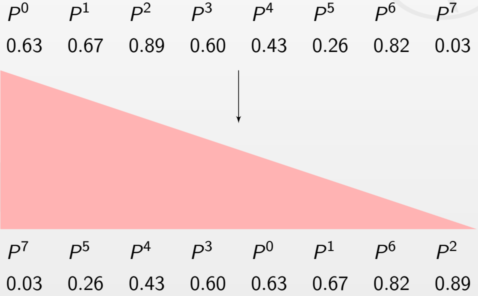
- Careful to obtain good mapping from phenotype to genotype
- For this work, most FISs are encoded with a mix of real values and discrete integers
- The fuzzy fan controller:
  - $(0,0,55,65)\,(55,65,75)\,(65,75,100,100)$ $|\,(0,0,0.25)\,(0,0.25,1)\,(0.25,1,1)\;\;[0,1,2]$

Selection

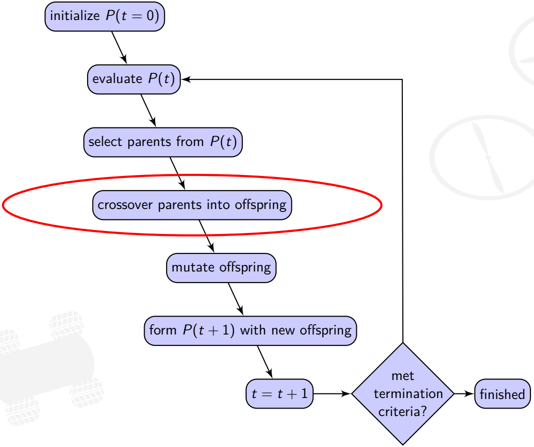

initialize $P(t=0)$
evaluate $P(t)$
select parents from $P(t)$
crossover parents into offspring
mutate offspring
form $P(t+1)$ with new offspring
$t=t+1$
met termination criteria?
finished

Selection Mechanism

- Want to retain good genes in future generations
- Favor fitter (harder, better, faster, stronger) parents to reproduce
- Main method used in this work is a triangular probability distribution



Crossover/Mating



Crossover/Mating

Operators depend on the method of encoding

- Binary Encoded Chromosomes
  - Single/Double/Multi Point Crossover
  - Hamming cliff
  - Representation does not reflect solution space
- Real-Valued Genetic Encoding
  - Simple/Flat/Blended Crossover
  - Recombination of parents in neighborhood
  - Often, chromosome represents actual solution – no encoding necessary!

Crossover/Mating

Operators depend on the method of encoding

- Binary Encoded Chromosomes
  - Single/Double/Multi Point Crossover ✓
  - Hamming cliff
  - Representation does not reflect solution space
- Real-Valued Genetic Encoding
  - Simple/Flat/Blended Crossover
  - Recombination of parents in neighborhood
  - Often, chromosome represents actual solution – no encoding necessary!

Crossover/Mating

Operators depend on the method of encoding

- Binary Encoded Chromosomes
  - Single/Double/Multi Point Crossover ✓
  - Hamming cliff×
  - Representation does not reflect solution space
- Real-Valued Genetic Encoding
  - Simple/Flat/Blended Crossover
  - Recombination of parents in neighborhood
  - Often, chromosome represents actual solution – no encoding necessary!

Crossover/Mating

Operators depend on the method of encoding

- Binary Encoded Chromosomes
  - Single/Double/Multi Point Crossover ✓
  - Hamming cliff×
  - Representation does not reflect solution space×
- Real-Valued Genetic Encoding
  - Simple/Flat/Blended Crossover
  - Recombination of parents in neighborhood
  - Often, chromosome represents actual solution – no encoding necessary!

Crossover/Mating

Operators depend on the method of encoding

- Binary Encoded Chromosomes
  - Single/Double/Multi Point Crossover ✓
  - Hamming cliff×
  - Representation does not reflect solution space×
- Real-Valued Genetic Encoding
  - Simple/Flat/Blended Crossover ✓
  - Recombination of parents in neighborhood
  - Often, chromosome represents actual solution – no encoding necessary!

Crossover/Mating

Operators depend on the method of encoding

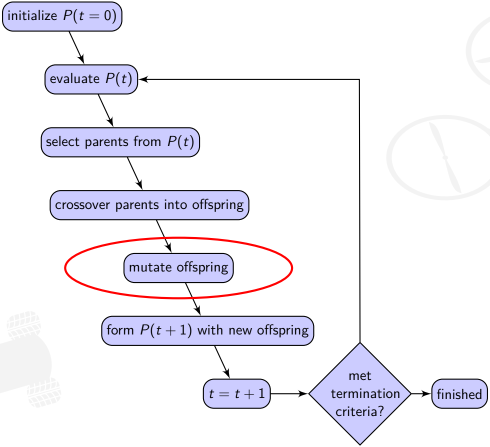
- Binary Encoded Chromosomes
  - Single/Double/Multi Point Crossover ✓
  - Hamming cliff×
  - Representation does not reflect solution space×
- Real-Valued Genetic Encoding
  - Simple/Flat/Blended Crossover ✓
  - Recombination of parents in neighborhood ✓
  - Often, chromosome represents actual solution – no encoding necessary!

Crossover/Mating

Operators depend on the method of encoding

- Binary Encoded Chromosomes
  - Single/Double/Multi Point Crossover ✓
  - Hamming cliff×
  - Representation does not reflect solution space×
- Real-Valued Genetic Encoding
  - Simple/Flat/Blended Crossover ✓
  - Recombination of parents in neighborhood ✓
  - Often, chromosome represents actual solution – no encoding necessary✓
- Recombination is the way in which the GA converges

Mutation


initialize $P(t = 0)$
evaluate $P(t)$
select parents from $P(t)$
crossover parents into offspring
mutate offspring
form $P(t + 1)$ with new offspring
$t = t + 1$
met termination criteria?
finished

Mutation

- Choose k number of genes to mutate
- Randomly choose k genes in each chromosome which is being mutated and change them

  Binary Encoding

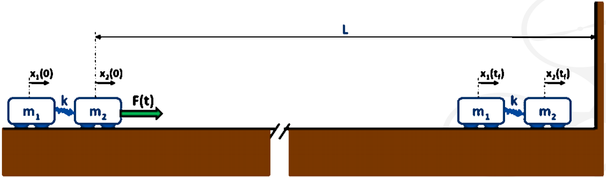
  - Flip the bit

  Real-Valued Encoding

  - Random disturbance or random generation
  - Sometimes advantageous to make it non-uniform with later generations
- Mutation is a factor which helps exploration of the solution space

Two Cart Flexible System

Problem Statement (1/3)



$$m_1 = 1kg, m_2 = 2kg, K = 250\frac{N}{m},\ \mathrm{L} = 100\mathrm{m}$$

Problem Statement (2/3)

- Initial Conditions:

$$\bar{y}(0) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

- Final Conditions:

$$\bar{y}(500) = \begin{bmatrix} 99 < x_1 < 100 \\ 99 < x_2 < 100 \\ 0 \\ 0 \end{bmatrix}$$

Input: $\bar{y}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix}$

Output: $|F(t)| \leq 1N$

Problem Statement (3/3)

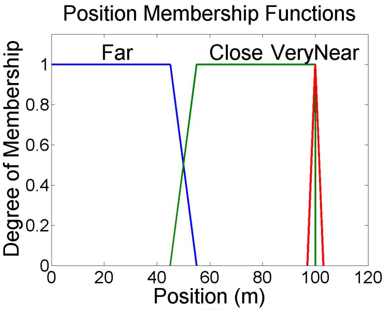
- Cost Function: (Minimize J)

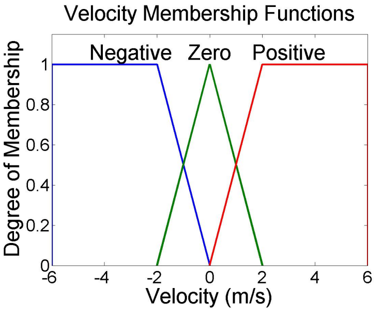
$$J = \frac{t_f}{100} + 2[L - x_2(500)],$$

$$t_f = \left|L - x_i(t_f)\right| \leq 1m \quad (settling\ time)$$

- Theoretical limit assumptions
  - True rigid body (no spring coupling)
  - $[L - x_2(500)] \ll 1 \Rightarrow J = \frac{t_f}{100}$
  - $\ddot{x}(t) = \begin{cases} \frac{1N}{m_{tot}},\ if\ x(t) \leq \frac{L}{2} \\ -\frac{1N}{m_{tot}},\ if\ x(t) > \frac{L}{2} \end{cases} \Rightarrow \quad t_f = t_{L-1}$

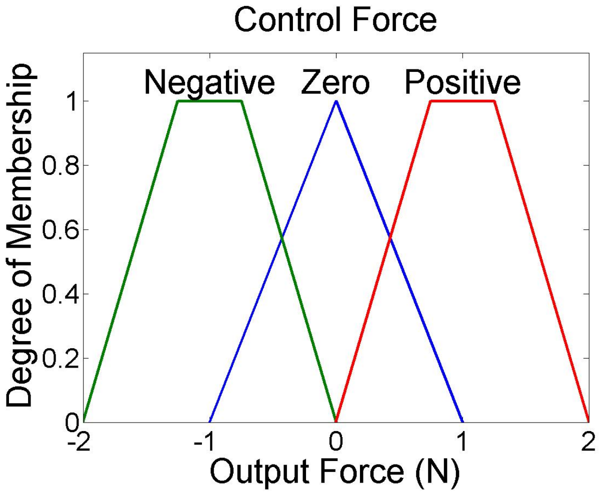
Inputs





UNIVERSITY OF Cincinnati

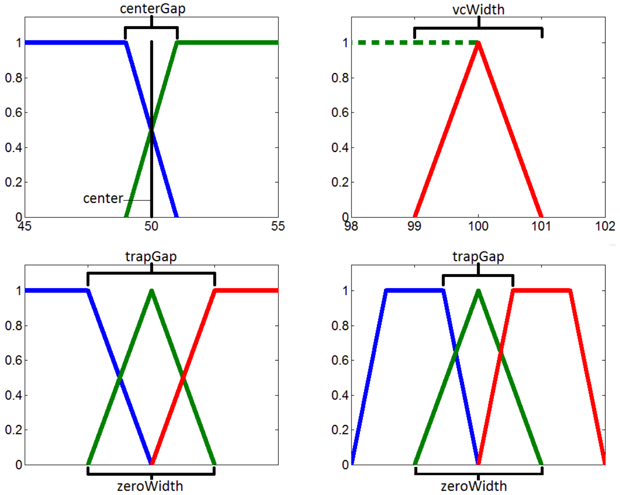
Output


Control Force
Negative
Zero
Positive
Degree of Membership
1
0.8
0.6
0.4
0.2
0
-2
-1
0
1
2
Output Force (N)

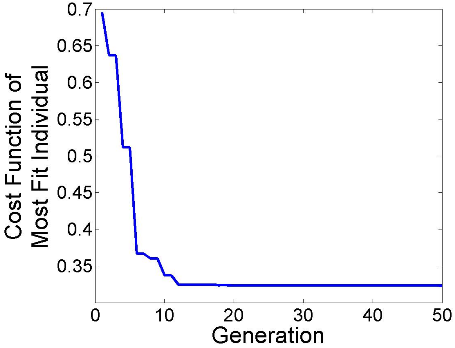
Rule Base

	Negative	Zero	Positive
Far	Positive		
Close	Positive	Positive	Negative
VeryClose	Positive	Zero	Negative

Search Space Reduction



Results



t_f	$x_2(500)$	J
32.308 s	99.999 999 m	0.32311

Robustness Results

	Mass 1 (kg), Mass 2 (kg)			
	1 kg, 2 kg	2 kg, 4 kg	4 kg, 8 kg	4 kg, 16 kg
Theoretical Limit	0.3191	0.4553	0.6438	0.8312
GA FIS	0.3231	0.4562	0.6579	0.8434
Hand-tuned FIS	0.3233	0.6125	5.3072	3.3240
GA Error	1.3%	0.2%	2.2%	1.5%
Hand-tuned Error	1.3%	34.5%	724.4%	299.9%

F-4 Pitch Attitude Control System

The Problem

- F4 Attitude Pitch Attitude Hold System
  - Chosen for interesting/challenging dynamics
    - Has zeros located close to origin
    - Poles close to imaginary axis
  - 4 Cases tested
    - Nominal approach
    - 50% degradation in aerodynamic derivatives
    - Nominal subsonic cruise
    - Nominal supersonic cruise
  - Covers large portion of F4 flight envelope

The Problem

- Traditional PID design
  - The good
    - Well understood
    - Relatively easy to tune
    - Computationally simple
  - The bad
    - Inflexible to changes in plant
    - Unintelligent, non-adaptive control
    - Change in plant may lead to unstable control behavior

The Solution (part I)

- Fuzzy PID
  - The good
    - Adaptive
    - Continuous, intuitive gain scheduling
    - Allows gains to ‘float’ around to best fit the situation
    - Best of both worlds
  - The bad
    - Hard to tune

The Solution (part II)

- Genetic Fuzzy
  - The good
    - Self-tuning

The Solution (part II)

- Genetic Fuzzy
  - The good
    - ~~Self-tuning~~ Self-learning

The Solution (part II)

- Genetic Fuzzy
  - The good
    - ~~Self tuning~~ Self-learning
    - Removes the iterative tedium from the control engineer

The Solution (part II)

- Genetic Fuzzy
  - The good
    - ~~Self-tuning~~ Self-learning
    - Removes the iterative tedium from the control engineer (mostly)
  - The bad

The Solution (part II)

- Genetic Fuzzy
  - The good
    - ~~Self-tuning~~ Self-learning
    - Removes the iterative tedium from the control engineer (mostly)
  - The bad
    - Non-deterministic

The Solution (part II)

- Genetic Fuzzy
  - The good
    - ~~Self-tuning~~ Self-learning
    - Removes the iterative tedium from the control engineer (mostly)
  - The bad
    - Non-deterministic
    - Resistant to validation and verification

The Solution (part II)

- Genetic Fuzzy
  - The good
    - ~~Self-tuning~~ Self-learning
    - Removes the iterative tedium from the control engineer (mostly)
  - The bad
    - Non-deterministic
    - Resistant to validation and verification
    - Highly dependent on cost function

The Solution (part II)

- Genetic Fuzzy
  - The good
    - ~~Self-tuning~~ Self-learning
    - Removes the iterative tedium from the control engineer (mostly)
  - The bad
    - Non-deterministic
    - Resistant to validation and verification
    - Highly dependent on cost function
    - Needs TIME/DATA

Generic Parameters

- Genetic Algorithm
  - Population Size: 100
  - Maximum Generation: 200
- Fuzzy Inference System
  - 3 input (e, $\sum e$, Δe), 3 output (k_p, k_i, k_d)
  - Membership functions per fuzzy partition: 5
  - Number of fuzzy rules: 125
- Cost Function
  - $J = T_s$

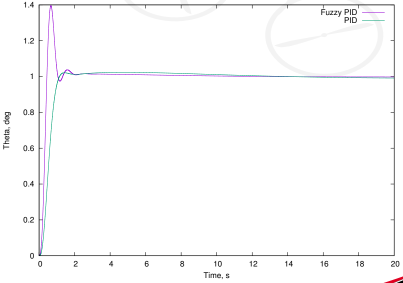
Nominal Condition

- Fuzzy PID Response

T_s	T_r	T_p	M_p	FV
1.86	0.26	0.66	1.396	1.00

- PID Response

T_s	T_r	T_p	M_p	FV
7.08	1.27	4.98	1.023	1.00



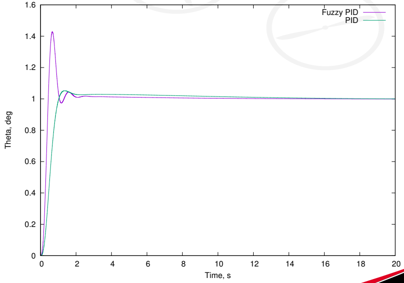
Degraded Condition

- Fuzzy PID Response

T_s	T_r	T_p	M_p	FV
1.87	0.25	0.66	1.428	1.00

- PID Response

T_s	T_r	T_p	M_p	FV
1.66	1.69	1.75	1.014	1.00



Adjusted Cost Function

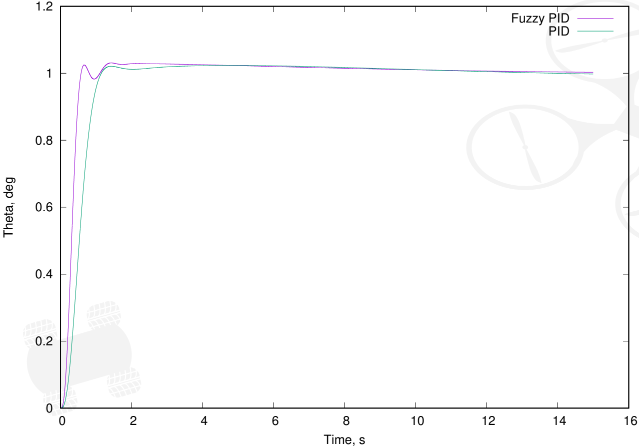
- $J = T_s + 3 \cdot OS\%$
- Fuzzy PID Response

T_s	T_r	T_p	M_p	FV
4.96	0.34	1.43	1.031	1.00

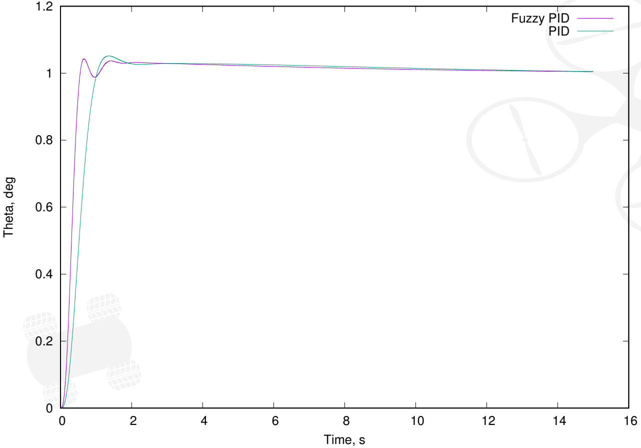
- PID Response

T_s	T_r	T_p	M_p	FV
4.47	0.33	0.66	1.043	1.00

Nominal Condition - Modified Cost Function



Degraded Condition - Modified Cost Function



Precision Landing System

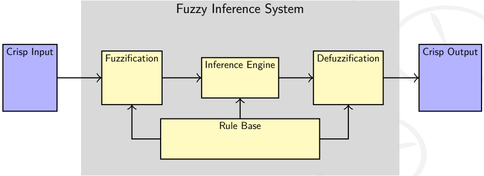
Background

- Develop control strategy for robust, precision landing
  - Place focus on implementable control
  - Approach from hardware constraints
  - Controller must be onboard
- Use vision-based sensors only for guidance
  - Ubiquitous
  - Cheap
  - No special sensors required
- Target available hardware in the lab
- Caveats
  - Target platform motion constrained to level plane

Flight Hardware

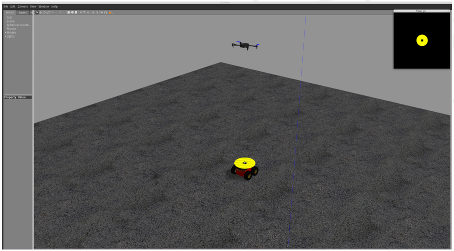
Control Architecture

- State vector
  - $\boldsymbol{x} = [x, y, z, \psi]^\top$
- Control
  - Input: $\Delta\boldsymbol{x}$, $\Delta\dot{\boldsymbol{x}}$
  - Output:
    $\boldsymbol{u} = \left[\dot{x}, \dot{y}, \dot{z}, \dot{\psi}\right]^\top$
- Fuzzy Logic Controller (FLC)
  - Computationally inexpensive
  - Tolerant of noisy data



Approach

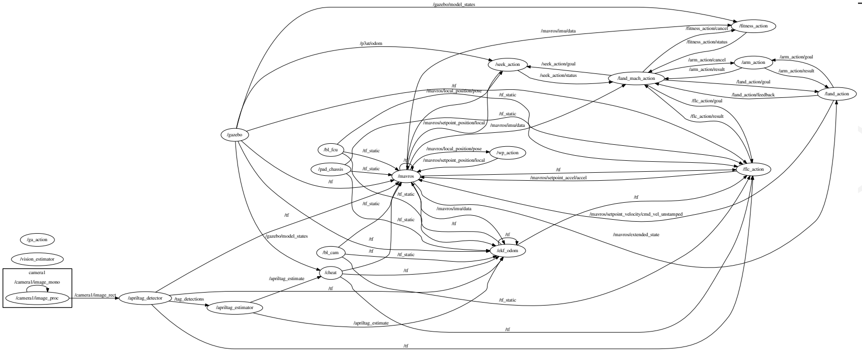
- Simulate
  - Gazebo
  - Full-featured simulation
  - Full hardware emulation
- Tune/Iterate
  - Evolutionary learning
- Build
  - Hardware implementation



Tools

- Robot Operating System
    - Distributed execution framework
    - Message marshalling
    - Publish/Subscribe model
- Gazebo
    - 3D dynamics simulator
    - Sensor simulation
- AprilTags
    - Visual fiducial system for robotics
    - Developed by Univ. of Michigan
    - Robust and lightweight (small data payloads)

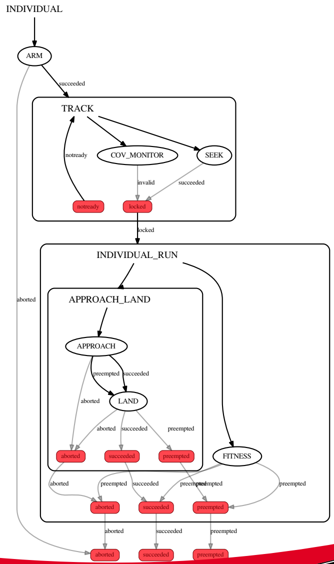
ROS (1/2)



ROS handles a lot of complexity for the end user. Allows building complex, scalable systems.

ROS (2/2)

- SMACH (State MACHine)
  - Decompose problem into states
  - Chain states with transitions
  - Allows to focus on micro behavior



Homography

- Altitude estimation
  - Use GPS/Baraometer until target is visible
  - Estimate altitude from known size of target
    - Circular Area (px) $\rightarrow$ Diameter (px) $\rightarrow \frac{d \cdot f}{m \cdot d_p} \rightarrow d_z$
  - Use AprilTag detection if available
    - Needed for target occlusion, frame saturation
- Position (2D) estimation
  - Relative to target position
  - EKF fusion of visual track, AprilTag track, and IMU information

Homography

- Visual estimate is obtained by mapping the projected target image onto world coordinates
- $\boldsymbol{R}(\phi, \theta, \psi) = \begin{bmatrix} c_\theta c_\psi & -c_\theta s_\phi & s_\theta \\ c_\phi s_\psi + c_\psi s_\phi s_\theta & c_\phi c_\psi - c_\psi s_\phi s_\theta & -c_\theta s_\phi \\ s_\phi s_\psi - c_\phi c_\psi s_\theta & c_\psi s_\phi + c_\phi s_\theta s_\psi & c_\phi c_\theta \end{bmatrix}$
- $\boldsymbol{R}_{cam}^{body} = R(\pi, 0, 0) \quad \boldsymbol{R}_{body}^{inert} = R(\phi, \theta, \psi)$
- Projected position of target center parallel to image plane:
  $p_i = \begin{pmatrix} d_x \\ d_y \\ d_z \end{pmatrix}$
- Relative position of target in inertial frame:
  $p_r = \boldsymbol{R}_{body}^{inert} \boldsymbol{R}_{cam}^{body} p_i$

Homography

- Visual estimate is obtained by mapping the projected target image onto world coordinates
- $\boldsymbol{R}(\phi, \theta, \psi) = \begin{bmatrix} c_\theta c_\psi & -c_\theta s_\phi & s_\theta \\ c_\phi s_\psi + c_\psi s_\phi s_\theta & c_\phi c_\psi - c_\psi s_\phi s_\theta & -c_\theta s_\phi \\ s_\phi s_\psi - c_\phi c_\psi s_\theta & c_\psi s_\phi + c_\phi s_\theta s_\psi & c_\phi c_\theta \end{bmatrix}$
- $\boldsymbol{R}_{cam}^{body} = R(\pi, 0, 0) \quad \boldsymbol{R}_{body}^{inert} = R(\phi, \theta, \psi)$
- Projected position of target center parallel to image plane:
  $$p_i = \begin{pmatrix} d_x \\ d_y \\ d_z \end{pmatrix}$$
- Relative position of target in inertial frame:
  $p_r = \boldsymbol{R}_{body}^{inert} \boldsymbol{R}_{cam}^{body} p_i$

Homography

- Visual estimate is obtained by mapping the projected target image onto world coordinates
- $\boldsymbol{R}(\phi, \theta, \psi) = \begin{bmatrix} c_\theta c_\psi & -c_\theta s_\phi & s_\theta \\ c_\phi s_\psi + c_\psi s_\phi s_\theta & c_\phi c_\psi - c_\psi s_\phi s_\theta & -c_\theta s_\phi \\ s_\phi s_\psi - c_\phi c_\psi s_\theta & c_\psi s_\phi + c_\phi s_\theta s_\psi & c_\phi c_\theta \end{bmatrix}$
- $\boldsymbol{R}_{cam}^{body} = R(\pi, 0, 0) \quad \boldsymbol{R}_{body}^{inert} = R(\phi, \theta, \psi)$
- Projected position of target center parallel to image plane:
$$p_i = \begin{pmatrix} d_x \\ d_y \\ d_z \end{pmatrix}$$
- Relative position of target in inertial frame:
$$p_r = \boldsymbol{R}_{body}^{inert} \boldsymbol{R}_{cam}^{body} p_i$$

Homography

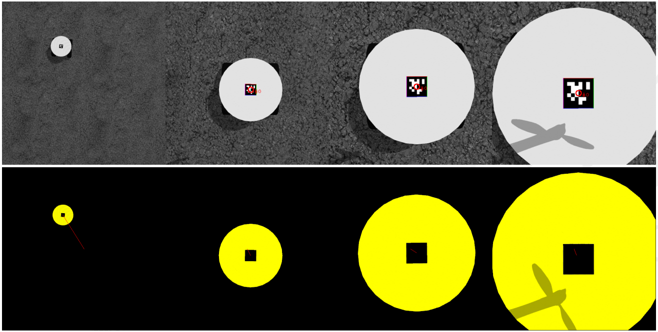
- Visual estimate is obtained by mapping the projected target image onto world coordinates
- $\boldsymbol{R}(\phi, \theta, \psi) = \begin{bmatrix} c_\theta c_\psi & -c_\theta s_\phi & s_\theta \\ c_\phi s_\psi + c_\psi s_\phi s_\theta & c_\phi c_\psi - c_\psi s_\phi s_\theta & -c_\theta s_\phi \\ s_\phi s_\psi - c_\phi c_\psi s_\theta & c_\psi s_\phi + c_\phi s_\theta s_\psi & c_\phi c_\theta \end{bmatrix}$
- $\boldsymbol{R}_{cam}^{body} = R(\pi, 0, 0) \quad \boldsymbol{R}_{body}^{inert} = R(\phi, \theta, \psi)$
- Projected position of target center parallel to image plane:

  $$p_i = \begin{pmatrix} d_x \\ d_y \\ d_z \end{pmatrix}$$

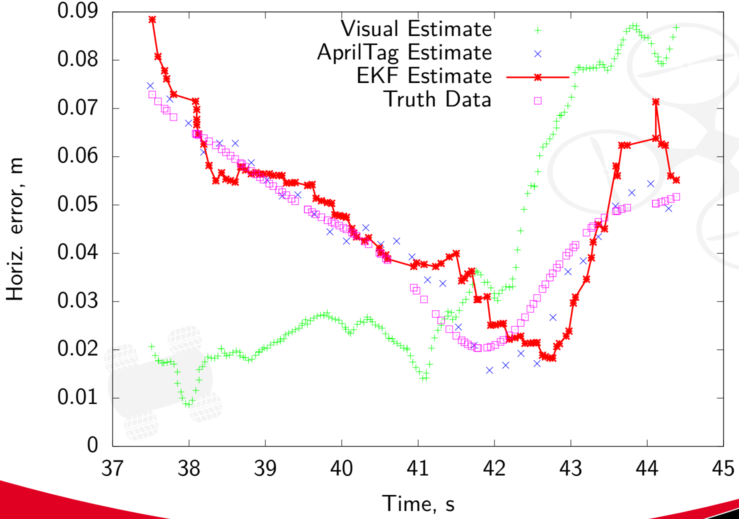
- Relative position of target in inertial frame:

  $$p_r = \boldsymbol{R}_{body}^{inert} \boldsymbol{R}_{cam}^{body} p_i$$

Pose Estimation



Extended Kalman Filter

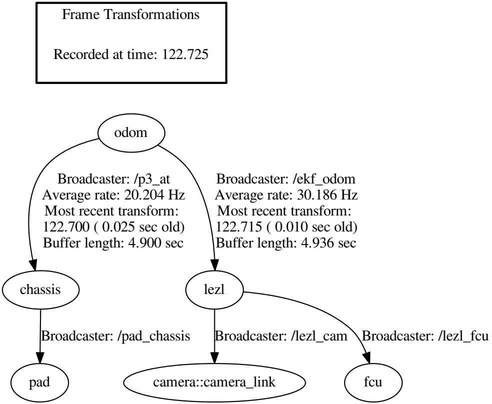


Extended Kalman Filter

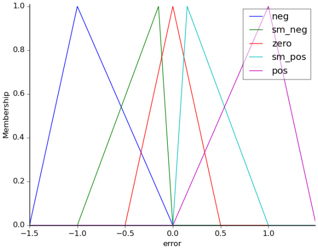
- Implemented using `robot_localization` ROS package
- Needs a great deal more tuning, but fusion is still better than discrete jumps

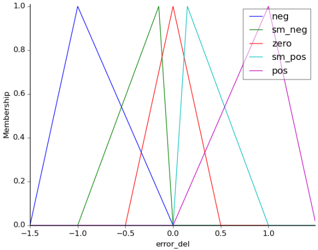
Transformation Tree

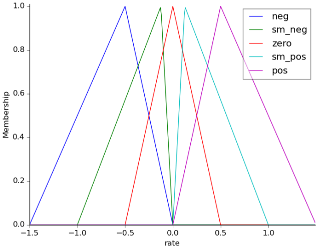
- Maintain a tree of transformations (tfs) at all times
- Can lookup tf from any frame to any other...
- ... as long as they share some common node



Membership Functions







Rule Base

error

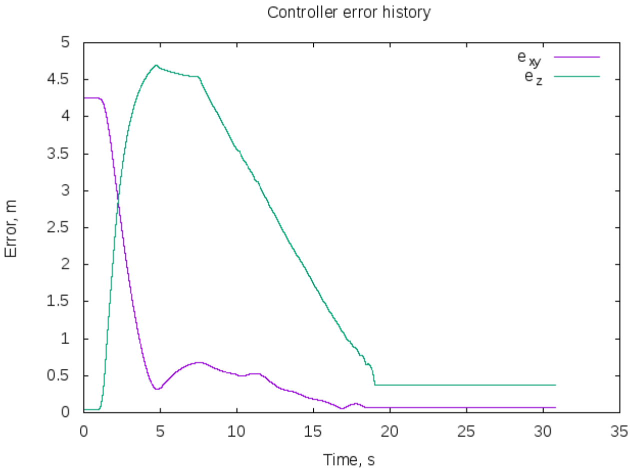
error rate	N	SN	Z	SP	P
N	P	P	SP	SP	Z
SN	P	SP	SP	Z	SN
Z	SP	SP	Z	SN	SN
SP	SP	Z	SN	SN	N
P	Z	SN	SN	N	N

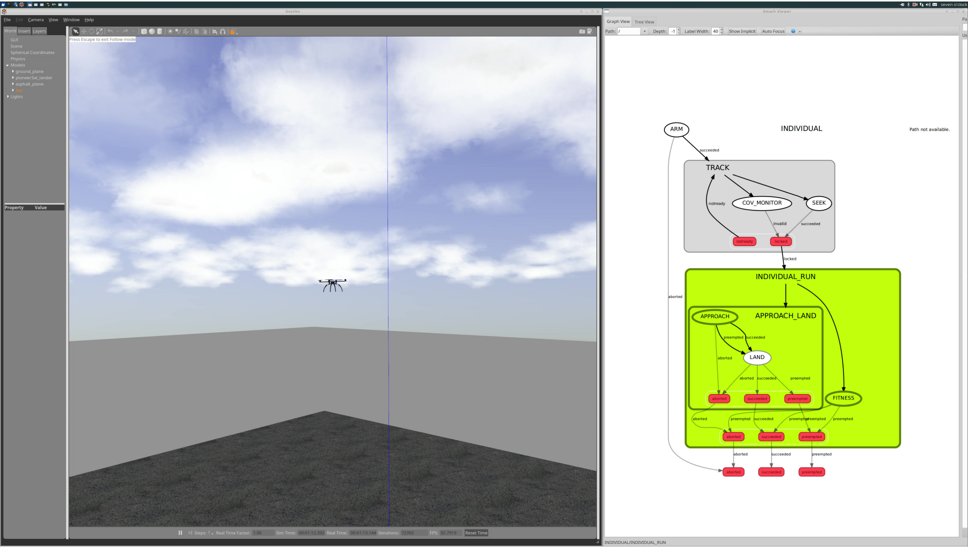
velocity

UNIVERSITY OF Cincinnati

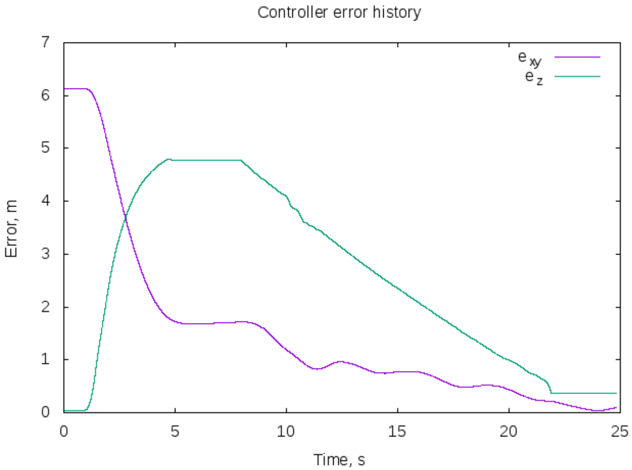
Results

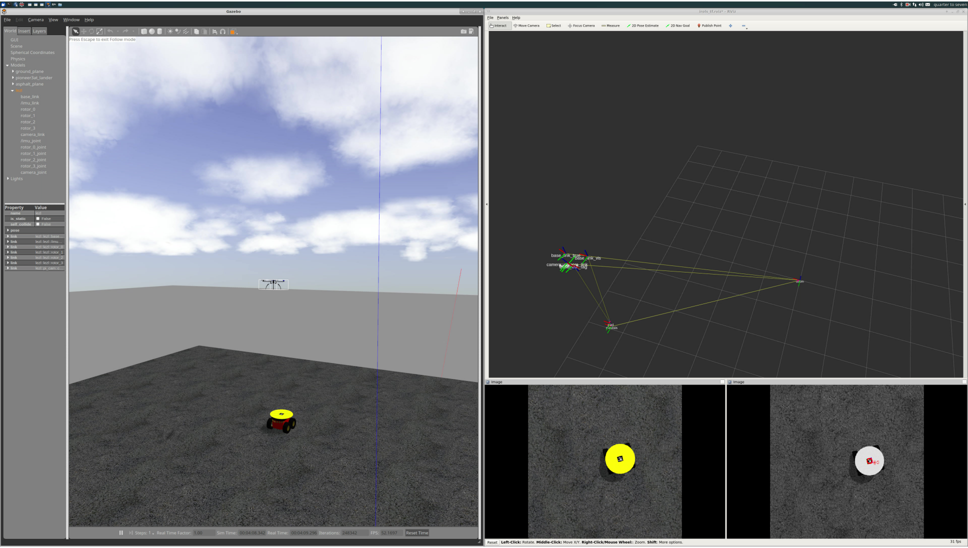
Static Target




Gazebo
File Edit Camera View Window Help
World Insert Layers
GUI
Scene
Spherical Coordinates
Physics
Models
ground_plane
pioneer3at_lander
asphalt_plane
Lights
Property Value
Press Escape to exit Follow mode
Steps: 1 Real Time Factor: 1.00 Sim Time: 00:01:12.392 Real Time: 00:01:13.144 Iterations: 72392 FPS: 55.7919 Reset Time
Smach Viewer
Graph View Tree View
Path: / Depth: -1 Label Width: 40 Show Implicit Auto Focus
ARM
INDIVIDUAL
Path not available.
succeeded
TRACK
notready
COV_MONITOR
SEEK
invalid
succeeded
notready
locked
locked
INDIVIDUAL_RUN
aborted
APPROACH
APPROACH_LAND
preempted
succeeded
aborted
LAND
aborted
succeeded
preempted
aborted
succeeded
preempted
FITNESS
aborted
preempted
succeeded
preempted
preempted
preempted
aborted
succeeded
preempted
aborted
succeeded
preempted
aborted
succeeded
preempted
INDIVIDUAL/INDIVIDUAL_RUN

Moving Target




Gazebo
File
Edit
Camera
View
Window
Help
World
Insert
Layers
GUI
Scene
Spherical Coordinates
Physics
Models
ground_plane
pioneer3at_lander
asphalt_plane
base_link
/imu_link
rotor_0
rotor_1
rotor_2
rotor_3
camera_link
/imu_joint
rotor_0_joint
rotor_1_joint
rotor_2_joint
rotor_3_joint
camera_joint
Lights
Property
Value
Press Escape to exit Follow mode
Reset Time
File
Panels
Help
Interact
Move Camera
Select
Focus Camera
Measure
2D Pose Estimate
2D Nav Goal
Publish Point
Image
Image
Reset
Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click/Mouse Wheel:: Zoom. Shift: More options.
31 fps

Conclusions

- Noisy data
- Different dynamic domains
- Comparable to gain-scheduled PID... without the need to schedule.

Future Work

- Get it on hardware!
    - MoCap functional
    - EKF tuning
    - Brave soul
- Send force/acceleration setpoints
    - Not yet implemented in firmware
    - Not trivial to add, but possible
- Integrate with other tools like $\mathbb{F}\mathbb{l}\mathbb{y}\mathbb{MASTER}$
    - Will make fuzzy control more accessible to others in the lab
    - Both projects focused on modular control
    - Recent work with capstone team proves possible