CRATES: Cognitive Robotics Architecture for Tightly-coupled Experiments and Simulation

Andrew Symington
University College London
a.symington@ucl.ac.uk

ROSCON 2014
Introduction

• My position in CompLACS
• Coordinating multiple agents in order to achieve some complex task (through decomposition)
• EU-based collaboration
  – Theorists
  – Engineers
  – ERC
• Quadcopters
• Theory → Simulation → Experiments
Simulation: QRSim

- Experiments = time + risk
- QRSIM (to appear in IROS'14)

- Features
  - Platform dynamics
  - Environment noise
  - Sensor noise

- Issues
  - CRATES: migration to ROS
Simulation: CRATES

**Topics**
- Estimate
- Control
- Truth

**Services**
- Waypoint
- Velocity
- VelocityHeight
- AnglesHeight

**Real platforms**
- hal_quadrotor
- platform_asctec
- hal_sensor_xyz

**Simulated platforms**
- hal_quadrotor
- hal_sensor_xyz
- sim

**WiFi**
- multimaster_fkie
- ros_client

**High level controller**
- High-level controller is completely platform agnostic.

**Platform-specific drivers that implement virtual methods from hal_quadrotor.**

**Simulator can spawn a number of models with plugins that provide hal_quadrotor.**
Simulation: Improved GNSS model

- GNSS performance → controller performance
- Time-correlated
  - Ephemerides
  - Satellite clock
  - Receiver clock
  - Ionosphere
  - Troposphere
- New GPS module
- IGS traces
Simulation: CRATES
Hardware: platform overview

• Ascending Technologies Pelican (5)
• Platform limits
  – Battery < 20 minutes
  – Max 5m/s velocity, 200 deg/s yaw rate
  – 650g max payload
  – 10 Hz < measurements / control < 10 Hz
  – Firmware flashing not permitted
• Supplied with manual transmitter
• Added a 2.0Ghz Atom “SBC”
Hardware: safety

- Levels
  - Manufacturer
  - Finite state model*
  - Safety module*
Hardware: ground truthing

- **Goal:** accurate position
- **Errors**
  - Code-phase: ~meters
  - Carrier-phase: ~decimeters
- **Cycle slip**
- **Hardware**
  - NV08C-CSM-BRD
  - Tallysman GPS/GLO
Hardware: ground truthing

Diagram showing GNSS error and cycle slips

Cycle slips

Histogram of Distance between Precise and Regular position (m)

Mean error
Hardware: putting it all together
Hardware: mechanical issues

- Propeller fatigue
- Leg fatigue
- CNC legs + fittings
Hardware: black box FCS

Control loop - Attitude / Position control (1000Hz)

- Attitude Control (CTRL)
  - Roll
  - Pitch
  - DesYawRate
  - DesThrust

- Attitude / Position Control
  - RX Decoder
  - RX
  - EN
  - MASK
  - OPTIONS
  - GPS
  - BARO

Attitude loop - Stabilization control (1000Hz)

- Direct Motor Control (DMC)
  - RollRate
  - PitchRate
  - YawRate
  - Thrust

- Stabilization Control
  - Orientation
  - GYRO
  - MAG
  - ACCEL

- Electronic Speed Controllers
  - RPM-Motor1
  - RPM-Motor2
  - RPM-Motor3
  - RPM-Motor4

- Motors

Autonomous loop (15Hz)

- Navigation Engine
  - State

- Control
  - Goal

- Possible goals
  - Waypoint, Yaw
  - 3D Velocity, Yaw
  - 2D Velocity, Height, Yaw
  - Angles, Height
  - Takeoff (to altitude)
  - Land
  - Hover (stay at current point)
Theory: Path Integral Control

- Coordinate actions of multiple quadrotors
- Path Integral Control
  - **COST**: Complex task is specified through a cost function
  - **DYNAMICS**: Controlled noisy movement
  - **CONTROL**: Given the current state the optimal control is computed with sampling methods, based on rollouts

- Approach
  - **FEEDBACK**: Control is re-computed after state update
  - **SAMPLING**: Computations based on a simplified model
Theory: Holding Pattern

• Task
  – Aircraft queue up for landing above a landing zone
  – They must maintain a safe distance from each other
  – Aircraft cannot hover; must maintain a minimum velocity

• Cost
  – Cost for too low or too high velocity
  – Cost for pair-wise distance
  – Cost for distance to the landing zone
  – Cost for collisions
Holding Pattern : 5 platforms
Holding Pattern : 10 platforms
Holding Pattern : 15 platforms
Results

• Demo
Experimental results
Conclusion

• Integration with mav_tools and hector_quadrotor
• HAL-based abstraction
  – Does this generalise?
• Some questions about ROS
  – Command-line tool latencies
  – Message inheritance
  – Multimaster integration plans
  – Quality of Service guarantees

All code available open source: https://bitbucket.org/asymingt/crates