Robot Control
(Planning and State Machines)

Maham Tanveer
Integrated Seminar Intelligent Robotics SoSe 2016
9th June, 2016
Motivation

Why is designing control system a complicated task?

Control system and behaviors

Approaches to control:
- Hierarchical Approach
- Behavioral Approach
- Comparison of Hierarchical and Behavioral approaches
- Hybrid
  - FSM
  - Subsumption architecture
  - SMACH

Conclusion

Bibliography
Motivation

- Early robotics dominated by machine tool industry
  - Limited sequence control (pick and place operations)
  - Playback with point to point control
  - Playback with continuous path control
  - Intelligent control
Why is designing control system a complicated task?

- Building a control system for robot: Complex task
  - Integrating mechanical, electrical and software components
  - Dynamic environment, sensor noise
  - Dynamic constraints of robot itself
  - Debugging and testing
  - Performance requirements
  - Maintainability
  - Performance improvement

Figure 1: Hardware, Electrical and Software components [1,2]
Control system and behaviors

- Control system and collection of behaviors
  - Well defined, self-contained and independently testable
  - Integrate behaviors to achieve goals

Figure 2: Behaviours independently defined and collection of behaviors define control system [3]
Approaches to control

- **Hierarchical: (Model-Plan-Act)**
  - Top-down approach
  - Not very flexible
- **Behavioral**
  - Bottom-up approach
  - Organization is difficult, messy approach
- **Hybrid (FSM, Subsumption architecture)**
  - Deliberative at high level; reactive at low level

Figure 1: Hardware, Electrical and Software components [1,2]
High level control systems approaches: Model-Plan-Act

Figure 3: Model Plan Act Flowchart [3]
High level control systems approaches: Model-Plan-Act

Figure 4: Finding the mouse-hole example [3]
High level control systems approaches: Behavioral

- Direct coupling of sensors and actuators
- Higher level behaviors are layered on top of lower level behaviors

Figure 5: Behavioural Approach Flowchart [3]
High level control systems approaches: Behavioral

Cruise behavior simply moves robot forward

Figure 6: Behavioural approach example [3]
High level control systems approaches: Behavioral

Figure 7: Behavioural approach example [3]

Left motor speed inversely proportional to left IR range
Right motor speed inversely proportional to right IR range
If both IR < threshold stop and turn right 120 degrees
High level control systems approaches: Behavioral

Escape behavior stops motors, backs up a few inches, and turns right 90 degrees

Figure 8: Behavioural approach example [3]
High level control systems approaches: Behavioral

The track ball behavior adjusts the motor differential to steer the robot towards the ball

Figure 9: Behavioural approach example [3]
High level control systems approaches: Behavioral

Hold ball behavior simply closes ball gate when ball switch is depressed

Figure 10: Behavioural approach example [3]
High level control systems approaches: Behavioral

The track goal behavior opens the ball gate and adjusts the motor differential to steer the robot towards the goal.

Figure 11: Behavioural approach example [3]
High level control systems approaches: Behavioral

All behaviors are always running in parallel and an arbiter is responsible for picking which behavior can access the actuators.

Figure 12: Behavioural approach example [3]
# Comparison of Model-Plan-Act & Behavioral Approaches

<table>
<thead>
<tr>
<th>Model-Plan-Act</th>
<th>Behavioural</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speed of response: Slower</td>
<td>Speed of response: Faster (Real time response)</td>
</tr>
<tr>
<td>Symbolic</td>
<td>Reflexive</td>
</tr>
<tr>
<td>Adaptive</td>
<td>More robust but forgetful</td>
</tr>
<tr>
<td>Requires complete model of the world</td>
<td>Requires complete design of the system</td>
</tr>
<tr>
<td>High level intelligence</td>
<td>Low level intelligence</td>
</tr>
</tbody>
</table>
High level control systems approaches: Finite State Machines

• Combination of above approaches
• Each state is a behavior, linked to form a close loop control system
• Pre, post and end conditions of state
• Sensor data used as feedback for transition to next state
• Debug and verify states for improved behavior

Software:
• Implement behaviors as functions in code
• Use switch statement to handle state transition and behaviors
High level control systems approaches: Finite State Machines

Figure 13: Finite state machine approach example [3]
High level control systems approaches: Finite State Machines

Figure 14: Finite state machine approach example [3]
High level control systems approaches: Finite State Machines

Figure 15: Finite state machine approach example [3]
High level control systems approaches: Finite State Machines

Figure 16: Finite state machine approach example [3]
High level control systems approaches: Finite State Machines

Figure 17: Finite state machine approach example [3]
High level control systems approaches: Finite State Machines

- **Subsumption Architecture:**
  - Generalization of finite state machine approach
  - Collection of modular controllers (FSMs) in a hierarchy where each module is treated as a light weight thread with priorities. These modules can share limited variables.
  - Implemented in code through nested if else statement
  - Common approach in many robots
High level control systems approaches: Finite State Machines

Figure 18: Finite state machine approach example [4]
Finite State Machines Implementation in ROS

- **SMACH:**
  - Python based – task level architecture
  - Each state is an executable task
  - Default outcome of each state:
    - Succeeded
    - Aborted
    - Preempted
  - States have database for calculations with input and output keys
  - Collection of containers and states
  - Aids in building complex behavior
Finite State Machines Implementation in ROS

• **States of SMACH:**
  - Generic State
  - CBState (Callback)
  - Simple Action State
  - Service State
  - Monitor State

• **Containers of SMACH:**
  - State Machine Container
  - Concurrence Container (Parallel execution, 2 callbacks)
  - Sequence Container
  - Iteration Container
Conclusion

- Many different types of control strategies exist for robots, choose one depending on complexity of the task to be achieved.
- Finite state machines can combine the model-sense-act and behavioral approaches and are modeled as closed loop control systems. FSMs offer a better approach for control.
- Finite state machines can be modularized and grouped into hierarchies to create a subsumption architecture.
Any Questions?
References

[1] : https://en.wikipedia.org/wiki/Robotics (Figure 1, Hardware and electrical component)
[2] : https://people.mech.kuleuven.be/~bruyninc/robotics/H02A4A/ (Figure 1, Software component)