

CPE 670: Autonomous Mobile Robots
Homework #2

Terence Henriod

November 30, 2014

1. [10 points] Describe the difference between robot control using a horizontal decomposition and a vertical decomposition.

Solution: Robot control that is horizontally decomposed exhibits several specific task-oriented competencies divided into self-contained modules that tend to run sequentially, while vertical decompositions tends to present behavior-oriented competencies layered on top of one another to produce complex behaviors built on top of primitive ones that all ideally run in parallel.

Horizontally decomposed controllers often follow the “SPA” paradigm, while vertically decomposed ones might follow the subsumption architecture.

2. [10 points] Describe how activities can be sequenced in the Subsumption Architecture, without using explicit communication “wires” between the task-achieving layers.

Solution: When using a subsumption architecture, by running all behaviors in parallel, the behaviors all wait for an appropriate set of sensor readings or input from other behaviors before they take certain actions. Because each behavior is waiting for a particular set of inputs to perform a given action, the behavior becomes dependent on other behaviors to bring the robot to a state where those inputs become reality. Through careful design of the behaviors, sequenced behavior can *emerge* without explicit tasking being programmed into the robot (in fact, subsumption controllers are incapable of being tasked).

3. [10 points] Describe at least two principles of good design of a behavior-based control system.

Solution:

- (a) BBC systems should be reactive.

BBC systems should have behaviors that execute concurrently and in parallel. This will give the behaviors the ability to operate in real-time.

- (b) BBC systems should use a uniform time scale and representations.

A uniform time scale or representation should be used so that behaviors that rely on inputs from other behaviors can be properly synchronized or informed.

- (c) Networks of behaviors can store state and construct world models

In a behavior-based system, one should never try to explicitly construct a symbolic representation of a world model and just rely on current sensor data for modeling the world (“the world is its own best model,” after all), but based on the states of each behavior, some state of the world is implicitly encoded (via the collection of information held by the behaviors) into the robot’s controller. Sometimes, these representations can be used to generate efficient behavior, or even make viable predictions about the future.

4. [10 points] Describe briefly how Toto the robot used a map to navigate from a location to another. Give one advantage of using such a method.

Solution: Toto would construct a topological map of “behaviors” for each landmark encountered. These behaviors would instruct Toto how to move around each landmark when it was the nearest landmark. Novel landmarks were added to the map and linked to the other nearest landmarks.

When Toto was instructed to move to a certain destination (landmark) messages were passed through the network from neighboring landmark to neighboring landmark. The messages would be passed until messages reached the landmark in the map representing Toto’s current location. Then the behavior

corresponding to the current location/landmark could decide which message had been routed along the shortest path (path-lengths were updated at each node along the message's path through the network), and then begin navigating towards the neighboring landmark along the shortest path. Messages were passed to continuously to make Toto update its navigation behavior continuously with a changing world or changing destination goals.

One advantage to using a method like this is that it has the potential to perform well in a dynamic environment. Should the destination/goal change, the navigation strategy can quickly change and a new path can be taken; should the location of landmarks change, once the changes are discovered, shortest path computation are quickly updated and the current path can be quickly abandoned if needed.