Exam #1 Sample Problems  
(Reminder: Exam is closed books, closed notes)  

[The problems given here are representative of the type of questions you will be asked,  
and give an indication of your expected level of knowledge for the material we have covered so far.  
They do not cover all the material you are expected to know.  
The exam will cover Ch. 1-4 of Murphy, the handouts, and material covered in class through Feb. 8.]  

1. **Historical Robots & Architectures.** We have discussed several examples of “historical” robots and robot  
control architectures that were implemented using different control approaches. In the list below, circle either  
“H” or “R” to indicate whether the specified robot or control architecture is/was more closely an example of  
hierarchical control (“H”) or reactive control (“R”).

<table>
<thead>
<tr>
<th>Robot/Architecture</th>
<th>H</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>Genghis</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Shakey</td>
<td>H</td>
<td>R</td>
</tr>
<tr>
<td>NASREM</td>
<td>H</td>
<td>R</td>
</tr>
<tr>
<td>Braitenburg’s vehicles</td>
<td>H</td>
<td>R</td>
</tr>
<tr>
<td>CMU Rover</td>
<td>H</td>
<td>R</td>
</tr>
<tr>
<td>Grey Walter’s tortoise</td>
<td>H</td>
<td>R</td>
</tr>
<tr>
<td>STRIPS</td>
<td>H</td>
<td>R</td>
</tr>
<tr>
<td>Hilare</td>
<td>H</td>
<td>R</td>
</tr>
<tr>
<td>Stanford Cart</td>
<td>H</td>
<td>R</td>
</tr>
<tr>
<td>Subsumption</td>
<td>H</td>
<td>R</td>
</tr>
</tbody>
</table>

2. **Three+ Robotic Paradigms**  

Name the three+ (i.e., four) robotic paradigms we described in class, and show the relationship between the  
SENSE, PLAN, and ACT modules for each.

3. **Innate Releasing Mechanisms**  

Many mammals exhibit a camouflage meta-behavior. The animal freezes when it sees motion (an affordance  
for a predator) in an attempt to become invisible. It persists until the predator is very close, then the animal  
flees. Write pseudo-code of the behaviors involved in the camouflage behavior in terms of innate releasing  
mechanisms, identifying the releasers for each behavior.
4. Match the term with its proper definition, writing the letter of the best correct definition in each blank in the second column of the table. Each definition should be used exactly once.

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Closed world assumption</td>
<td>a) A robot control architecture that achieves behavioral fusion via vector summation.</td>
</tr>
<tr>
<td>Subsumption</td>
<td>b) Dilemma of how to represent a real-world situation in a way that is computationally tractable.</td>
</tr>
<tr>
<td>Emergence</td>
<td>c) Response that continues for a longer duration than the stimulus.</td>
</tr>
<tr>
<td>Reflex</td>
<td>d) Perceivable potentialities of the environment for an action.</td>
</tr>
<tr>
<td>Computational theory</td>
<td>e) Robot has a physical body.</td>
</tr>
<tr>
<td>Open world</td>
<td>f) Appearance of novel properties in whole systems from interacting components.</td>
</tr>
<tr>
<td>Taxes</td>
<td>g) Behavioral responses that orient animal toward or away from a stimulus.</td>
</tr>
<tr>
<td>Fixed action pattern</td>
<td>h) World model contains everything the robot needs to know.</td>
</tr>
<tr>
<td>Motor Schemas</td>
<td>i) Robot is controlled primarily by a human.</td>
</tr>
<tr>
<td>Frame problem</td>
<td>j) Definitions of levels of commonality in intelligent agents.</td>
</tr>
<tr>
<td>Embodiment</td>
<td>k) Similar to a latch or Boolean variable that triggers a behavioral response.</td>
</tr>
<tr>
<td>Teleoperation</td>
<td>l) A robot control architecture that achieves behavioral fusion via fixed priorities.</td>
</tr>
<tr>
<td>Affordance</td>
<td>m) Situation in which it is not possible to fully define everything the robot needs to know.</td>
</tr>
<tr>
<td>Releaser</td>
<td>n) Response that lasts as long as the stimulus.</td>
</tr>
</tbody>
</table>

Definitions:

a) A robot control architecture that achieves behavioral fusion via vector summation.
b) Dilemma of how to represent a real-world situation in a way that is computationally tractable.
c) Response that continues for a longer duration than the stimulus.
d) Perceivable potentialities of the environment for an action.
e) Robot has a physical body.
f) Appearance of novel properties in whole systems from interacting components.
g) Behavioral responses that orient animal toward or away from a stimulus.
h) World model contains everything the robot needs to know.
i) Robot is controlled primarily by a human.
j) Definitions of levels of commonality in intelligent agents.
k) Similar to a latch or Boolean variable that triggers a behavioral response.
l) A robot control architecture that achieves behavioral fusion via fixed priorities.
m) Situation in which it is not possible to fully define everything the robot needs to know.
n) Response that lasts as long as the stimulus.

5. Multiple Behavior Combinations

We have studied three methods for competitive behavior combination – priority-based coordination, action-selection coordination, and voting-based coordination. In the following questions, the table below gives the magnitude of the behavior response for each of three behaviors over time. (In this figure, all behavior response values are multiples of 5, so values located between the y grid lines are interpreted to be exactly half-way between the grids. For example, the starting values of the behavior responses at time t=0 are 5, 25, and 50.) Using this figure, answer the following questions (labeled subparts a through n).
Priority-based (or fixed priority) coordination:

Assume the gain vector, \( \mathbf{G} = \begin{bmatrix} g_{behavior_1} \\ g_{behavior_2} \\ g_{behavior_3} \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \\ 0.6 \end{bmatrix} \) for all time values.

Also assume that the fixed priorities are Behavior 1 > Behavior 2 > Behavior 3, meaning that Behavior 1 has priority over Behavior 2, which has priority over Behavior 3.

At time \( t=20 \):
  a. What is the magnitude of the output response? ____________
  b. Which behavior generates this output response? ____________

At time \( t=60 \):
  c. What is the magnitude of the output response? ____________
  d. Which behavior generates this output response? ____________

Action-Selection coordination:

Assume the gain vector is still \( \mathbf{G} = \begin{bmatrix} g_{behavior_1} \\ g_{behavior_2} \\ g_{behavior_3} \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \\ 0.6 \end{bmatrix} \) for all time values.

Also assume that the activation levels of each behavior are as shown in the following figure:
Activation levels of behaviors

![Activation levels of behaviors](image)

(As before, in this figure, all activation level values are multiples of 5, so values located between the y grid lines are interpreted to be exactly half-way between the grids.)

In this exercise, we are still using the behavior response diagram given previously.

At time $t=20$:
- e. What is the magnitude of the output response? _____________
- f. Which behavior is generates this output response? _____________

At time $t=60$:
- g. What is the magnitude of the output response? _____________
- h. Which behavior is generates this output response? _____________

**Voting-based coordination:**

Assume we now change the gain vector to be $G = \begin{bmatrix} g_{\text{behavior}_1} \\ g_{\text{behavior}_2} \\ g_{\text{behavior}_3} \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$ for all time values.

We are still using the behavior response values given earlier.

We now have two predefined motor responses $M_1$ and $M_2$. Let $r^B_{\text{magniute}}(t)$ equal the magnitude of the response vector of behavior $B_i$ at time $t$. Then, the mappings of each behavior’s output response to votes for the two motor responses is given by:
Votes for $M_1$ | Votes for $M_2$
---|---
Behavior 1: | $0.1 \times r_{\text{magnitude}}^B(t)$ | $0.9 \times r_{\text{magnitude}}^B(t)$
Behavior 2: | $1.0 \times r_{\text{magnitude}}^B(t)$ | $0.0 \times r_{\text{magnitude}}^B(t)$
Behavior 3: | $0.4 \times r_{\text{magnitude}}^B(t)$ | $0.6 \times r_{\text{magnitude}}^B(t)$

At time $t=20$:

i. What is the number of votes for $M_1$? ______________

j. What is the number of votes for $M_2$? ______________

k. Which motor response is activated? ______________

At time $t=60$:

l. What is the number of votes for $M_1$? ______________

m. What is the number of votes for $M_2$? ______________

n. Which motor response that is activated? ______________


Recall that the five primitive potential fields are uniform, perpendicular, attraction, repulsion, and tangential.

a. Draw the potential field representation for each of these five primitive potential fields.

b. Suppose you were to construct a library of potential fields of the five primitives. What parameters would you include as arguments to allow a user to customize the fields?

<table>
<thead>
<tr>
<th>Primitive potential field</th>
<th>Parameters for customization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uniform</td>
<td></td>
</tr>
<tr>
<td>Perpendicular</td>
<td></td>
</tr>
<tr>
<td>Attraction</td>
<td></td>
</tr>
<tr>
<td>Repulsion</td>
<td></td>
</tr>
<tr>
<td>Tangential</td>
<td></td>
</tr>
</tbody>
</table>
7. Wall Follower Pseudocode

Let’s assume that your simulated robot has 16 sonar sensors which are numbered 0 to 15 in the following arrangement (where sonar 0 points toward the forward direction of motion of the robot):

![Diagram of sonar sensor arrangement](image)

Assume your Player/Stage code has the following definitions:

```c
PlayerClient robot ("localhost");
Position2dProxy pp (&robot, 0);
SonarProxy sp (&robot, 0);
```

Further assume have the following functions:

- `robot.Read()`: gets the current robot sensor data
- `pp.SetSpeed(speed, rate)`: sets the robot’s speed to `speed`, and its turn rate to `rate`
- `sp[i]`: returns the value of sonar number `i`.
- `pp.SetMotorEnable(true)`: enables the robot to move

Also assume:

- The valid translation velocities are: **fast, moderate, slow, 0** (where these terms translate to valid robot speed specifications)
- The valid rotation velocities are: **strong_left, left, 0, right, strong_right** (where these terms translate to valid robot turning rate specifications)
- The desired distance of the robot from the wall is pre-defined in a constant called `prefDist`.

This means that `pp.SetSpeed(fast, left)` will cause the robot to move forward quickly while turning somewhat left, and `pp.SetSpeed(slow, strong_right)` will cause the robot to move forward slowly while turning strongly to the right.

Write pseudocode using the above functions and pre-defined velocity values to enable a robot to use its sonar to follow a smooth, infinitely long wall, on its right. In this problem, the robot should maintain a distance from the wall of `prefDist`. Assume the starting position of the robot is facing parallel to the length of the wall, with the wall on the right, as follows: