Intro to Artificial Intelligence

C951- Task 2

James Revello

# A: Disaster Environment

In this instance, our disaster environment is representative of a burning building with structural deficiencies. Our robot is designed to navigate the building and locate which pillars, if any, are structurally unsafe so emergency services do not risk a collapse on entry.

The building is surrounded by flames and smoke elements as indicators of the intended scenario. While these smoke and flame objects can obscure sensors, I have opted to place them outside of the building for appearances only, to set the stage of the scenario.

There are five normal grey columns which represent structurally sound building support pillars; they are not damaged and therefore the robot will simply navigate around them without sounding the alarm. These pillars are static and immovable and will be obstacles the robot must account for.

There is a single red column which represents a structurally unsound pillar which the robot needs to locate.

The building itself is constructed of four immovable walls which will likewise be obstacles the robot will collide with and navigate around

Finally, there are several cuboid obstacles which represent debris from the burning building and *can* be pushed out of the way by the robot (with a bit of effort). They will impede movement of the robot and occasionally cause a failure. This is important, as not every emergency is predictable nor ideal.

# B: Improved Disaster Recovery

Emergency situations are innately chaotic and unpredictable. There is not always a perfect solution to every disaster response. It is crucial that a robot be able to navigate environments riddled with highly variable obstacles which one often finds in these disaster recovery situations. This is especially the case in fire responses like the ones detailed in this recovery environment, where walls and pillars can collapse, and debris needs to be navigated around and may or may not be moveable.

Our robot will enter a burning building to navigate walls, pillars, and fire debris to locate damaged structural pillars. This immediate first response will ensure the building is safe for fire personnel to respond shortly thereafter without fear of collapse. This robot will save lives and prevent injury, preventing additional casualties above and beyond the initial fire.

Upon locating a damaged pillar, the robot will stop and sound an audible alarm. Fire personnel can be assured that the route the robot took up until its alarm location is safe from collapse and they can enter that preceding area to administer aid or recover injured victims.

# C: Architecture

The robot will have two sensors, the first of which will be an object detection sensor. The camera thereon will allow emergency personnel to view the building from a safe distance. The second sensor will detect immoveable obstacles and “collide” with them, forcing the robot backwards and initiating a slight counterclockwise turn. This is important, as the robot needs to act intelligently without external input in the case of loss of connection with a remote operator.

The colliding sensor is relatively small so the recovery bot can navigate tighter areas without unnecessary collisions and their requisite turns. The pillar detection sensor, however, has a much larger view to allow for faster detection of damaged pillars and to avoid near-misses where the robot bypasses the target by a narrow margin.

When the detection sensor locates the damaged pillar, the robot stops in place, an alarm sounds, and the sensor switches to a bright green light to indicate where the robot is pointing. This will allow security personnel to more easily locate and observe the damaged pillar in variable light environments.

# D: Internal Representation of Environment

The robot is completely autonomous and maintains an internal representation of its environment so that it can better navigate the area. As seen in the attached simulation, the robot sensor represents the outer edges and faces of obstacles with a negative white-on-black visual representation of what it sees.

The robot is constantly in a default “moving” state, whereby it travels forward at a designated speed. When the smaller navigation cone collides with an object, it changes state variables in the robot’s code, causing it to stop, reverse, and perform a 30 degree turn counterclockwise. This means the robot will perform trial and error to navigate its environment based on its detection of external objects.

The robot can differentiate between (1) moveable objects (debris) which it will push until it is unable to advance forward and (2) immoveable objects which must be navigated around. This is another form of internal representation of the outward environment.

When the robot’s detection sensor locates a damaged pillar, the robot stops moving entirely and begins its alarm.

At face value, the robot could possess more advanced responses to its environment and more robust working memory of the path it had previously taken so it does not retake steps. However, in an environment where debris may continue to fall or previously navigable paths become unusable, sometimes “dumber” is smarter.

# E: Reasoning, Knowledge Representation, Uncertainty, Intelligence

Our robot collects knowledge about its environment through its two sensors. One sensor detects immoveable objects and triggers an active state where the robot reverses and turns to navigate around the object. The robot also collects a wider field of view with further range with its second sensor. The second sensor detects pillars and triggers a different state, whereby the robot stops, flashes a green light, and sounds an audible alarm for emergency responders. The robot knows the difference between moveable debris and immoveable walls and pillars.

The robot also reasons about its environment. It decides which objects are moveable and which must be navigated around. It determines if a pillar is damaged or structurally sound. It can even learn within its environment, pushing against debris until the debris is wedged and no longer moveable, then changing course.

The robot starts each scenario with a large degree of uncertainty. It possesses no previous knowledge about its surroundings. It is dropped into a blind scenario and responds accordingly based on the obstacles and objects it detects with its sensors. This is crucial, as no fire or disaster response is identical to another. Each have different obstacles and challenges in different arrangements. This autonomy in uncertain environments is precisely why the robot has very simple “stop, reverse, turn” responses to collisions with the inner navigation sensor. More functionality would create more bugs and require far more testing to be viable. There is wisdom in the old emergency response adage “KISS.” Sometimes it’s better to keep things “stupid simple.”

Hand-in-hand with the robots reasoning is its intelligence. It makes intelligent decisions when it reasons about which debris is moveable or not. It also makes simple reasoning on which pillars are structurally sound and must be navigated around, versus other pillars which are damaged and must be scanned for alarm. This intelligence is largely present in the robot’s choice of state based on its prior reasoning after gaining knowledge about its environment. It essentially observes the environment, orients its state based on the observations, makes a decision (pre-decided in the case of hard coding), and then acts based on the decision. This observation, orientation, decide, and act loop continues until a pillar is found and an alarm is sounded.

# F: Further Improvements

While the simple functionality of our robot makes it perform without failure in robust and variable problem scenarios, there is value in making further improvements. Our robot is simple, but my no means perfect.

Currently, the bot’s searching algorithm is a random stopping and turning pattern as it is presented with immovable obstacles. The simplicity inherent in the searching pattern design makes the robot function reliably. However, the robot tends to narrowly miss the target pillar and continue its random search, especially as more obstacles are introduced in a given scenario. This could be improved by a searching algorithm which incorporates a 360 degree turning scan after the robot has traveled a certain distance. Doing so would capture large circles of scanned zones and prevent these narrow misses.

The robot could also have enhanced intelligence with the introduction of reinforcement learning. This is especially true when the robot gets wedged between two nearby objects and begins an interminable loop of backing and turning. By punishing the robot when it encounters frequent collisions within a short amount of time and rewarding it when it travels long distances without collisions, we can effectively reinforce the robot performing more calibrated turns beyond the typical 30 degrees. In this way, it could more fluidly respond to its environment and navigate more quickly, thereby locating damaged pillars in a timelier fashion.

# G: Panopto Recording

Please see the attached recording for a video overview of the disaster robot in action.