

## I. DISPLAY MODES AND BUTTONS

### A. Command heard

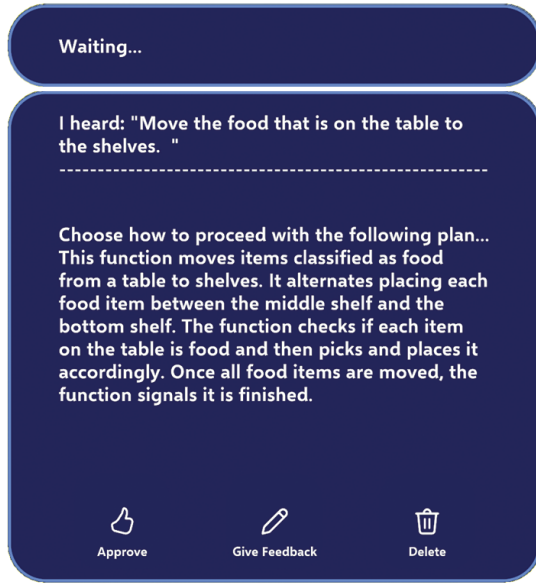


Fig. 1: After hearing a command such as “Move the food that is on the table to the shelves.”, the display shifts to command heard mode.

The heard command is displayed on the top. The LLM then plans a function for the command and displays a natural language description for the function. The buttons on the display can be triggered by speech or by tapping gesture. “Approve” accepts the function and sets the rule. “Give feedback” prompts the user for verbal feedback. “Delete” deletes the command.

### B. Verbal Feedback

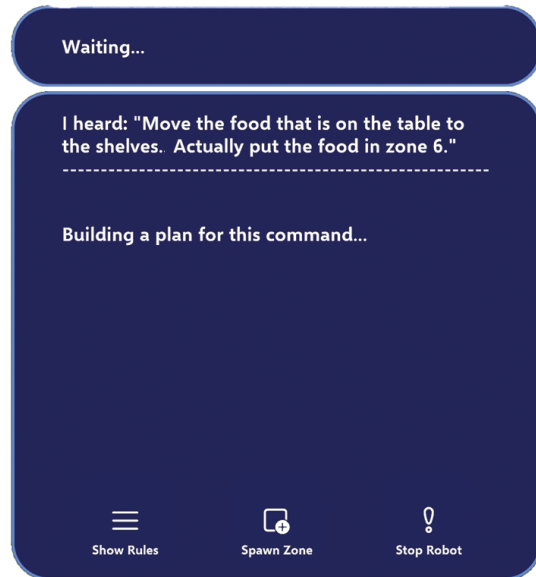


Fig. 2: Users can give verbal feedback that gets added to the previous command.

Users can provide verbal feedback similar to giving a command. The feedback gets included in the prompt to replan the function. The “Show rules” button displays all active rules. “Spawn zone” creates a new virtual zone. “Stop robot” halts the robot’s execution.

### C. Replanning

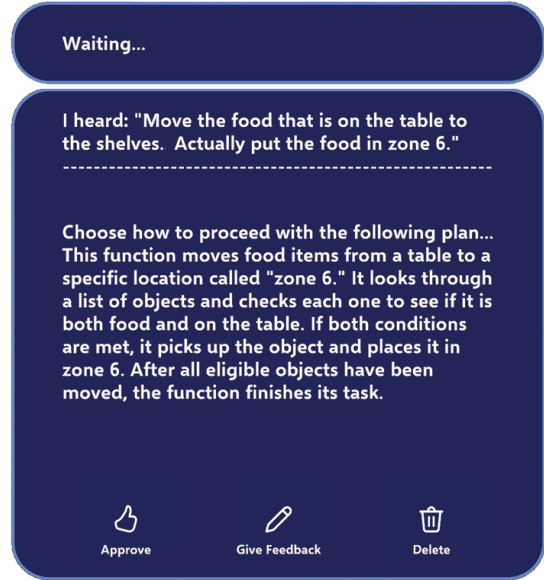


Fig. 3: The LLM replans the function based on the additional feedback.

Users can choose to accept the rule, again provide verbal feedback, or delete the rule.

### D. Rule triggered



Fig. 4: The display shows the robot state when the rule is triggered.

After a rule is triggered, the display shows the current action being planned or executed.

## II. LLM PROMPT EXAMPLES

### A. Trigger detection prompt

**Command:** “Pour the water from the bottle into the glass”

**System prompt:** “You are a helpful assistant. You will be given a task and you have to identify the trigger for the task.”

**User prompt:**

“Here are a few examples of tasks with their respective triggers -

wipe the table

object\_list = glass, pringles, mustard, bottle, sponge

location\_list = table on the left, Zone 1

object\_location\_list = 'bottle is on Zone 1', 'pringles is on Zone 1', 'glass is on Zone 1', 'mustard is on Zone 1', 'sponge is on the table on the left'

No trigger

When the cheezit is in zone 1, Move the cheezit to the top shelf

object\_list = glass, pringles, mustard, bottle, sponge

location\_list = table on the right, table on the left, bottom shelf, top shelf, zone 1

object\_location\_list = 'glass is on the table on the right', 'pringles is on the table on the right', 'mustard is on the table on the right', 'bottle is on the table on the right', 'cheezit is on the table on the right', 'cheezit is on zone 1'

Zone trigger, zone 1

After 2 minutes, Wave hi!

object\_list = glass, pringles, mustard, bottle, sponge

location\_list = table on the right, table on the left, bottom shelf, top shelf, zone 1

object\_location\_list = 'glass is on the table on the right', 'pringles is on the table on the right', 'mustard is on the table on the right', 'bottle is on the table on the right', 'cheezit is on the table on the right', 'cheezit is on zone 1'

Time trigger, 2 minutes

These are a few things to consider -

1. If a task requires waiting for some time, then it has a time trigger. Answer with 'time trigger < time >'. Where the < time > token should be replaced by a measure of time in minutes.

2. If a task involves objects being in a specific location before execution, then it has a zone trigger. Answer with 'zone trigger < zone >'. Where the < zone > token should be replaced by a surface.

3. If a task involves a 'zone trigger' and a 'time trigger'. Answer with 'zone and time trigger, < zone >, < time >'. Where the < zone > token should be replaced by a surface and < time > token should be replaced by a measure of time

in minutes.

4. If a task has neither of the triggers then it does not have a trigger. Answer with 'no trigger'

Given the following task what trigger does it have?"

Pour the water from the bottle into the glass

object\_list = glass, pringles, mustard, bottle, sponge

location\_list = table on the right, table on the left

'glass is on the table on the right', 'pringles is on the table on the right', 'mustard is on the table on the right', 'bottle is on the table on the right', 'cheezit is on the table on the right'

### B. Function Generation prompt

**Command:** “Pour the water from the bottle into the glass”

**System prompt:** “You are an action plan generator. You will be given a task to perform and you have to answer with a python function to complete the task. Your function can internally call the python functions from the function\_list to generate the python function.

function\_list = [pick(object), place(surface), pour(object), wipe(surface), dance(), wave(), check\_function(condition), done()]

The < object > token should be replaced by an object in object\_list that the action refers to. You see the objects from object\_list in the same room as you.

object\_list = [glass, pringles, mustard, bottle, cheezit]

The < surface > token should be replaced by a surface in location\_list that the action refers to. You see the location from location\_list in the same room as you.

location\_list = [table on the right, table on the left]

The < condition > token should be replaced by a condition to check.

**User prompt:**

Here are a few examples of tasks with their respective action plans -

wipe the table

object\_list = glass, pringles, mustard, bottle, sponge

location\_list = table on the left, Zone 1

object\_location\_list = 'bottle is on Zone 1', 'pringles is on Zone 1', 'glass is on Zone 1', 'mustard is on Zone 1', 'sponge is on the table on the left'

“python

def wipe\_table(self):

self.pick('sponge')

self.wipe('table on the left')

self.done()

return

““

Here are a few examples of tasks with their respective action plans -

These are a few constraints to consider -

1. If an object is picked up, it needs to be put down before

picking another object.

2. An object needs to be picked up before it can be placed on anything.
3. An object needs to be picked up before it can be used.
4. If an object is picked up, it needs to be put down before waving.

Given the following task what is the action plan for it?

Pour the water from the bottle into the glass

object\_list = glass, pringles, mustard, bottle, sponge

location\_list = table on the right, table on the left

'glass is on the table on the right', 'pringles is on the table on the right', 'mustard is on the table on the right', 'bottle is on the table on the right', 'cheezit is on the table on the right'

### C. Description Generation Prompt

**Function:** “python  
def wipe\_table(self):  
self.pick('sponge')  
self.wipe('table on the left')  
self.done()  
return  
”

**System prompt:** “You will be given a python function. You have to respond with a summary of the function.”

**User prompt:** “Summarize the following Python function in simple but specific terms for a non-programmer. Focus only on what it will do when it runs, using no more than five clear and concise sentences.”

“python  
def wipe\_table(self):  
self.pick('sponge')  
self.wipe('table on the left')  
self.done()  
return  
”

## III. TASK ERRORS

Users encountered five different errors that caused stoppages or hindered the participant during the study.

- T - Vicon Tracking Error: Trackers on the objects were occasionally knocked out of position, forcing the experimenters to re-calibrate the object.
- R - Robot Manipulation Error: In some cases, the robot joint controller stopped working, requiring the experimenter to restart the program.
- L - Large Language Model: The LLM could sometimes produce hallucinations, where it would output invalid code.
- H - Headset: The headset ran into two errors, one where it stopped receiving information and needed to be switched

out, and another where the virtual world was misaligned from the real world causing the program to fail.

- M - Microphone: In one case, the microphone stopped picking up verbal commands and needed to be restarted.