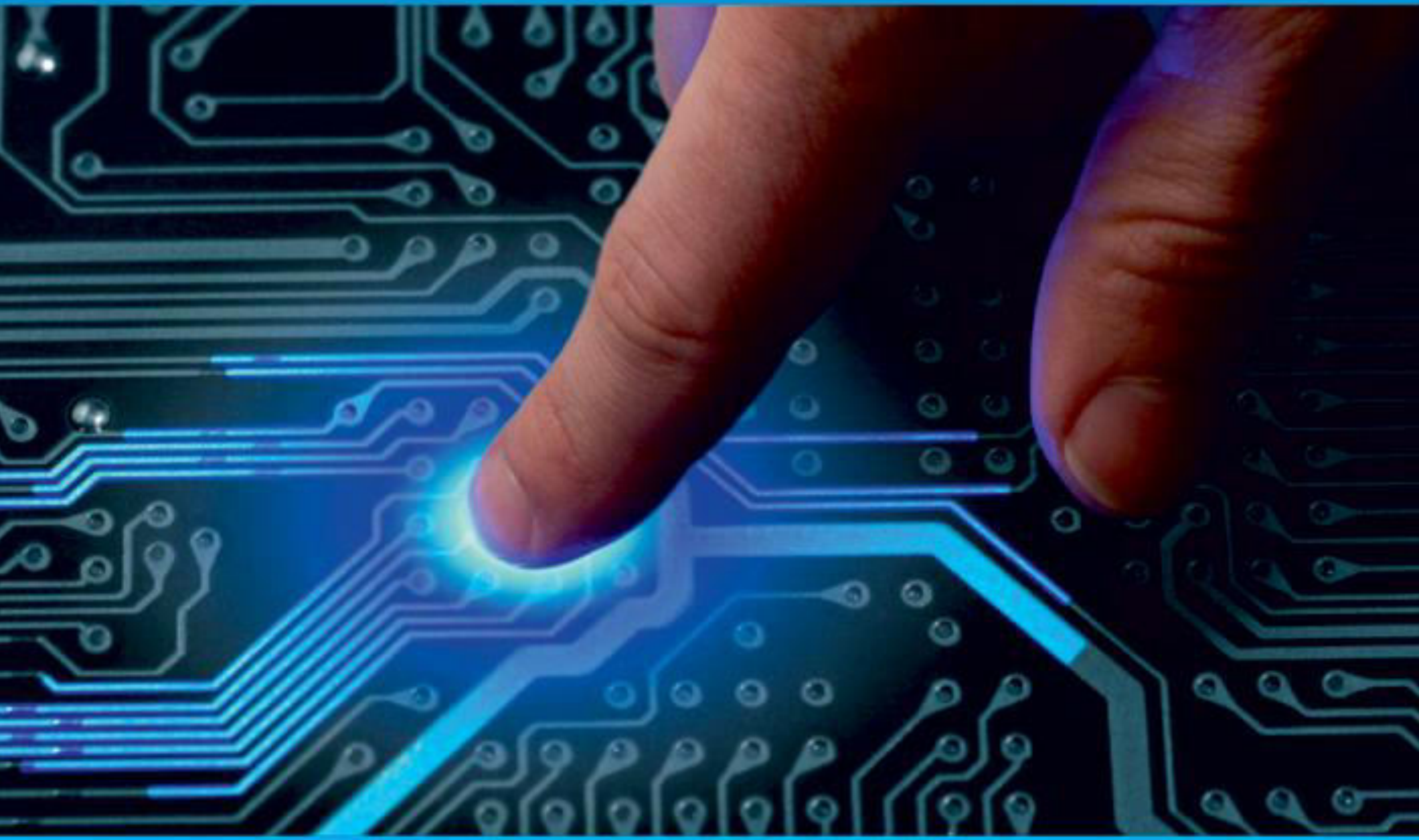
IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



# INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 12, Issue 8, August 2024

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.625

© 2024 IJIRCCE | Volume 12, Issue 8, August 2024| DOI: 10.15680/IJIRCCE.2024.1208016

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

# Improving LLM-based Robot Control via Human-Robot Cooperation

**Shri Ranjani S M[1], Vinay Kumar[2], Milan Srinivas[3], Shravya Bhat[4]**

BE, Department of AI and ML, BNM Institute of Technology, Bangalore, India[1]

BE, Department of AI and ML, BNM Institute of Technology, Bangalore, India[2]

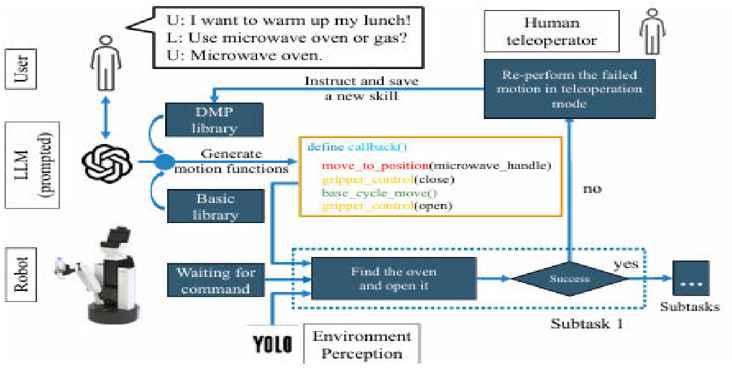
BE, Department of Computer Science and Engineering, VKIT, Bangalore, India[3]

BE, Department of AI and ML, Nitte Meenakshi Institute of Technology, Bangalore, India[4]

**ABSTRACT**: In the field of robotics, large language models, or LLMs, are becoming more and more common. However, because of the inadequate connection between LLM and motion, LLM-based robots can only perform simple, repetitive actions. The environment, robotics, and language models. This research presents a unique method using Human-Robot Collaboration (HRC) to improve the performance of LLM-based autonomous manipulation. The method entails breaking down high-level language commands into motion sequences that the robot can do using a suggested GPT-4 language model. Additionally, the system uses a YOLO-based perception algorithm to give the LLM visual signals that help it design possible motions within the given environment. Furthermore, an HRC approach that combines Dynamic Movement Primitives (DMP) with teleoperation is put forth, enabling the LLM-based robot to pick up knowledge from human instruction.

## I. INTRODUCTION

The idea of self-governing robots that can converse with people in a natural language has long fascinated the robotics community. Large Language Models (LLMs), which are driven by Transformer technology, have recently emerged and offer a workable way to achieve this goal [1]. Integrating environmental awareness is essential for LLMs to effectively guide robot behavior in real-world contexts. Because of this synergy, the robot can understand and respond to user commands in real-world situations on its own. Several important technologies make this possible, such as computer vision algorithms.


User
LLM (prompted)
Robot
U: I want to warm up my lunch!
L: Use microwave oven or gas?
U: Microwave oven.
Human teleoperator
Re-perform the failed motion in teleoperation mode
Instruct and save a new skill
DMP library
Generate motion functions
Basic library
no
yes
Success
Waiting for command
Find the oven and open it
Subtask 1
Subtasks
YOLO
Environment Perception

```
define callback()
    move_to_position(microwave_handle)
    gripper_control(close)
    base_cycle_move()
    gripper_control(open)
```

Figure 1 shows a summary of a human-robot collaboration system based on LLM.

Including user interaction, a DMP library for the creation and storing of adaptive motion functions, and a basic library for pre-programmed motion functions complete a challenging assignment in the real world. For example, "Heat my lunch".

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Only Look Once (YOLO) [5], as well as creative strategies such as the Segment Anything Model (SAM) [7] and Contrastive Language-Image Pre-Training (CLIP) [6]. The combination of LLMs with cutting-edge computer vision has sparked the creation of highly developed robotic manipulation platforms that can carry out exacting, complex actions in response to human directions expressed in natural languages.

The LLM uses natural language commands to choose motion functions from the motion library. The chosen functions are then combined with environmental data that is interpreted by a perception module based on YOLOv5, allowing for the autonomous completion of a variety of activities. The suggested LLM system makes use of GPT4's prompt function to implement a hierarchical planning framework. Complex jobs can be divided into smaller tasks by the LLM, which can then divide those smaller tasks into several motion functions and carry out each motion function one after the other. The LLM-based system's ability to carry out complicated tasks is greatly improved by the suggested HRC structure. These tasks can be effectively completed by including human examples, even if they frequently require complex trajectory planning and reasoning over settings.

## II. RELATED WORKS

### A. Natural Language Manipulation in Robotics

When it comes to studying autonomous robotics control, LLMs have a lot of promise. Liang et al.'s [10] "Code as Policies" approach leverages a language model to produce robot policy code from natural language commands; the resulting code is then executed to govern the robot's motion. The contributions of Liang and other relevant pioneers have served as the foundation for significant improvements and optimizations. The authors of [11] introduce the SoftGPT, a language model for effective manipulation of soft objects and learning from human demonstrations in home environments. It blends graph representations with LLM-based dynamics. Additionally, [12] suggests a multimodal strategy to improve the robot's comprehension and performance of natural language commands for robot manipulation. Additionally, we make use of the LLM to help us generate the code required to operate the robot. In this instance, choosing the motion functions is entirely within the purview of the LLM.

### B. Teleoperation System based on VR

Goertz proposed the first robot teleoperation system in 1940 [13]. Currently, a vast array of highly manipulable and intelligent robot teleoperation systems have been developed. A VR device was suggested as part of an easy-to-use teleoperation system by Nakanishi et al. [14]. Furthermore, Zhu and colleagues introduce a shared control framework and a force feedback system to enhance the teleoperation's transparency and manipulability [15], [16]. The authors present a robotic avatar system with sophisticated force-feedback telemanipulation and immersive 3D visualization in [17]. In this work, we employ the teleoperation system as a teaching link between the LLM-based robot manipulation system and the human supervisor.

### C. Primitives of Dynamic Movement for Trajectory Learning

A mathematical framework called Dynamic Movement Primitives (DMP) is used in robotics and machine learning to simulate and replicate intricate joint behaviors. The authors of [18] and [19], who revised this approach, present DMP as a technique for simulating intricate robotics behaviors and highlight its adaptability, versatility, and potential for integration with statistical learning techniques. Probabilistic movement primitives (ProMP), which allow for motion blending, task variable adaptation, and co-activation of movement primitives with benefits of probabilistic functions, were introduced by Paraschos et al.

## III. METHODS

Our suggested approach builds an LLM-based autonomous system by integrating an LLM with environmental data into the Robot Operating System (ROS). To improve the LLM-based system's ability to perform intricate tasks, we additionally use an HRC approach to direct robot movements while human guidance is provided. The translation of human commands into precise robotic motions is made possible by this integration. The basic library and the DMP library are the two main libraries used by the system to execute motion. Standard library: There are pre-programmed motion functions in this library. Based on the specifications of the task, the LLM chooses motion functions and combines them with contextual data to produce Pythonic code. The robot is controlled by this code to make the movements necessary to complete the tasks.

**A. Comprehensive Language Model for Self-Driving Robot Control**

As part of our methodology, we formulate the robot control hub using the GPT-4 Turbo network, with the temperature set to 0. The purpose of this hub is to transform user input into motion functions that may be used with Python programming. We give the prompt "Your mission is to control the robot to complete the task assigned by the user, utilizing the motion functions from the basic library" to make the LLM's role Additionally, the following prompts are offered to introduce the elements of the basic library: ". The object's size determines its range of motion. These modifications allow the LLM to achieve 99.4% executability by allowing it to choose and employ motion functions from the basic library in response to user requirements (Table I).

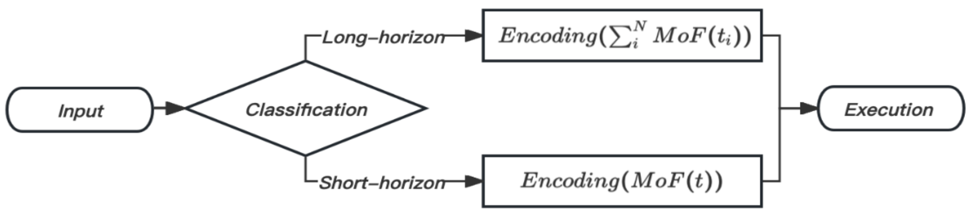


Figure 2 shows an example of how the LLM interprets, categorizes, breaks down, and performs various activities.

The LLM is instructed to "break them down into subtasks, if possible, divide them into subtasks with motion functions, if not, directly output the motion functions." The LLM converts short-horizon jobs directly into motion functions MoF(t), as shown in Fig. 2, whereas long-horizon tasks are split into subtasks, each of which consists of several motion functions PNi M oF (ti).

In actuality, Fig. 3 illustrates how user commands for long-horizon operations are processed, which involves the following steps:

1) The user gives an instruction (Clean the table, for example).
2) The directive is interpreted by the LLM, which categorizes it as a long-horizon assignment. This work can be broken down into multiple smaller tasks based on the classification, like "Put the left cup in storage."
3) The LLM parses the first subtask and translates it to the motion routines in the basic library.
4) After analyzing sensory data, the environmental perception system publishes the spatial coordinates of pertinent things in the operating environment, such as the position of the bowl and bottle in Figure 3.
5) An executable Pythonic code incorporates both the motion function sequence and the processed sensory input. The robot is directed to complete the first subtask by running this Python script. The previously indicated procedure will be followed to complete the remaining subtasks.

**B. Sensation of the Environment**

In the section on environmental perception, we identify, quantify, and locate the target objects in the picture using the Xtion depth camera on the robot, based on YOLOv5. In response to user commands, the LLM's motion planning prompts are dynamically updated based on the names and quantities of these objects. The following formula can be used to get the target's coordinates to the depth camera:

$$\boldsymbol{P}_{obj}^{c} = d \cdot \boldsymbol{K}^{-1} \cdot \boldsymbol{i} \qquad (1)$$

Where $\boldsymbol{i} = \begin{bmatrix} u & v & 1 \end{bmatrix}^T$ The target object's image coordinate point (u,v) is represented by T, which stands for the homogeneous coordinates in the picture. The depth value at point (u,v) obtained from the depth map is indicated by the term d, while the depth camera's intrinsic matrix is shown by the term K. The object's location near the depth camera is indicated by $P^c_{obj}$.

The world coordinate of the target is given by the following:

$$P^w_{obj} = T^w_c \cdot P^c_{obj}$$

where Pc obj indicates the object's position in camera coordinates, and Pw obj indicates the object's position in world coordinates. The rotation Rw c and translation c matrix from the camera coordinate system to the world coordinate system are contained in the transformation matrix Tw c.

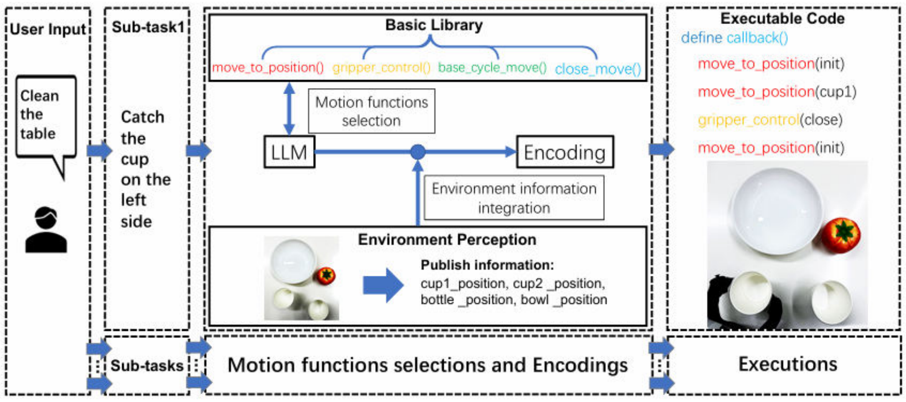

User Input
Clean the table
Sub-task1
Catch the cup on the left side
Basic Library
move_to_position() gripper_control() base_cycle_move() close_move()
Motion functions selection
LLM
Encoding
Environment information integration
Environment Perception
Publish information:
cup1_position, cup2 _position, bottle _position, bowl _position
Executable Code
define callback()
move_to_position(init)
move_to_position(cup1)
gripper_control(close)
move_to_position(init)
Sub-tasks
Motion functions selections and Encodings
Executions

Fig. 3. An overview of LLM-based autonomy for a real-world long-horizon task. This process encompasses sub-task identification, motion function selection from the basic library, environment perception integration, and executable code generation.

Names are attached to the target positions that have been determined and are regularly released. As shown in Fig. 3, the LLM integrates these computed positions with motion functions for the ongoing task, directing the robot's motions. For instance, to complete the job "Pick the middle cup," one must select the object 'cup2' that has been suitably labeled. Algorithm 1 provides the position information for this item ('cup2' in a sorted sequence of three cups, for example). Algorithm 2 deals with removing obstacles before task completion. Upon task execution, the LLM retrieves the names of the targets and sends them to Algorithm 2. After that, Algorithm 2 assesses each target's workspace to find any impediments and provides the LLM with a list of those obstacles.

## C. Human-Robot Collaboration

When combined with environmental perception, LLMs can help with simple tasks, but in complicated, real-world situations, their limits show. For instance, the LLM can operate a microwave with a standard design with ease, but it would have trouble opening an oven door with a horizontal axis design. This challenge arises from the LLM's restricted code generation capabilities, specifically in managing a restricted motion library and comprehending intricate trajectories. We support a synergistic framework of autonomy based on LLM for human-robot collaboration to tackle these issues. By utilizing teleoperation, this suggested architecture allows users to easily train the robot and intervene proactively. Using the user interface depicted in Figure 5, the revised motion function sequences are recorded in the DMP library and can be substituted for the original motion functions in the basic library.

**Algorithm 1** Sort and Label Objects Detected

```
 1: Input: List of objects $O$ detected by YOLOv5
 2: Output: List of labeled objects $L$
 3: Initialize $L = []$
 4: for each unique object name $n$ in $O$ do
 5:     Filter objects in $O$ with name $n$, store in $O_n$
 6:     Sort $O_n$ from left to right based on their positions, store in $O_{n\_sorted}$
 7:     for each object $o$ in $O_{n\_sorted}$ do
 8:         Label $o$ with its index in $O_{n\_sorted}$
 9:         Add labeled $o$ to $L$
10:     end for
11: end for
12: Return $L$
```

**Algorithm 2** Identify Obstacles for Task Execution

```
 1: Input: Target coordinates $TCoord$, robot base coordinates $RCoord$, workspace objects $WObjs$
 2: Output: Coordinates sequence of obstacles $ObsCoord$
 3: Initialize $ObsCoord$ as an empty list
 4: Define an equilateral triangular detection space with one vertex at $RCoord$ and the base midpoint at $TCoord$
 5: if $WObjs$ is empty then
 6:     Return 'No objects'
 7: end if
 8: for each object $Obj$ in $WObjs$ do
 9:     Check if $Obj$ is within the triangular detection space
10:     if $Obj$ is inside then
11:         Add $Obj$'s coordinates to $ObsCoord$
12:     end if
13: end for
14: Return $ObsCoord$
```

Two components of the teleoperation system [14] are a headset and an operator's controller. The headset replicates a real-world perspective as seen through the robot's eyes and gives stereo-vision feedback. It also reflects the user's head movements to the robot. The controller is held in the user's hand and is used to translate the hand movements of the user into the hand movements of the robot.

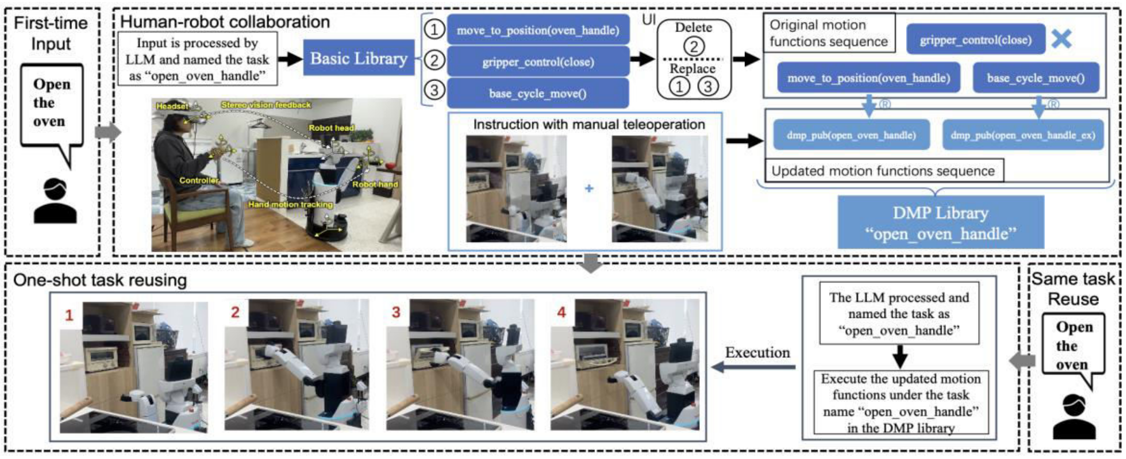

First-time Input
Open the oven
Human-robot collaboration
Input is processed by LLM and named the task as "open_oven_handle"
Basic Library
① move_to_position(oven_handle)
② gripper_control(close)
③ base_cycle_move()
UI
Delete ②
Replace ① ③
Original motion functions sequence
gripper_control(close)
move_to_position(oven_handle)
base_cycle_move()
dmp_pub(open_oven_handle)
dmp_pub(open_oven_handle_ex)
Updated motion functions sequence
DMP Library "open_oven_handle"
Headset
Stereo vision feedback
Robot head
Controller
Robot hand
Hand motion tracking
Instruction with manual teleoperation
One-shot task reusing
1
2
3
4
Execution
The LLM processed and named the task as "open_oven_handle"
Execute the updated motion functions under the task name "open_oven_handle" in the DMP library
Same task Reuse
Open the oven

Fig 4. The LLM processes user input to select motion features from the fundamental Library. The user interface with teleoperation is then used to change these chosen motions. The updated motion functions, which have a specific name like "open oven handle," are saved in the DMP Library. The LLM records the action "open" and the target "oven

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

handle," integrating them as "open oven handle" for later use (re-entering the same task or reusing it in the long-horizon task), which leads to the successful completion of the one-shot task.

In order to save and record these customized motion trajectories in the DMP library and expand the robot's capabilities for a wider range of activities, DMP technology is essential. The following equations capture the fundamental idea of DMP:

$$\ddot{y} = \alpha(\beta(g - y) - \dot{y}) + f(x)(g - y) \tag{3}$$

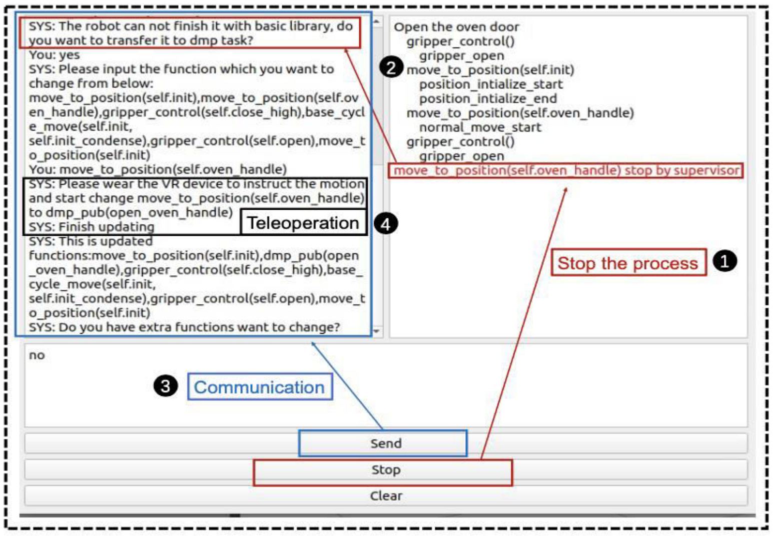
When g is the objective, y is the system's current state, and α and β are the constant gain terms. The behavior of the trajectory is described by this equation. For f(x), the forcing term:

$$f(x) = \frac{\sum_{i=1}^{N} \psi_i w_i x}{\sum_{i=1}^{N} \psi_i} \tag{4}$$

where x is the non-linearly decaying phase variable from the canonical system, with the appropriate weights, and ψi are the Gaussian basis functions. Weighted linear regression is used to extract the weights wi from the training trajectory, and N denotes the number of Gaussian basis functions—which was set at 15 for our trials. By acquiring the weights from a demonstrated trajectory obtained by teleoperation, DMP accomplishes trajectory storage and replay. Once learned, these weights specify the form of the forcing function f(x), which generates the desired movement trajectory when coupled with the spring-damper system in Equation (3).

Figure 5: A user interface is used to monitor and control a robot's work sequence, encompassing user-system communication and the stages involved in teleoperation to change robotic motions.

The system allows an LLM-based robot to perform previously difficult tasks more effectively by using teleoperation for trajectory recording to access human-defined motion trajectories kept in the DMP library.



```
SYS: The robot can not finish it with basic library, do you want to transfer it to dmp task?
You: yes
SYS: Please input the function which you want to change from below:
move_to_position(self.init),move_to_position(self.oven_handle),gripper_control(self.close_high),base_cycle_move(self.init,
self.init_condense),gripper_control(self.open),move_to_position(self.init)
You: move_to_position(self.oven_handle)
SYS: Please wear the VR device to instruct the motion and start change move_to_position(self.oven_handle) to dmp_pub(open_oven_handle)
SYS: Finish updating
SYS: This is updated functions:move_to_position(self.init),dmp_pub(open_oven_handle),gripper_control(self.close_high),base_cycle_move(self.init,
self.init_condense),gripper_control(self.open),move_to_position(self.init)
SYS: Do you have extra functions want to change?
```

```
Open the oven door
  gripper_control()
    gripper_open
  move_to_position(self.init)
    position_intialize_start
    position_intialize_end
  move_to_position(self.oven_handle)
    normal_move_start
  gripper_control()
    gripper_open
move_to_position(self.oven_handle) stop by supervisor
```

```
no
```

Using the procedures listed below, Fig. 4 depicts the real-world HRC process, which is carried out via the user interface depicted in Fig. 5:

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

1) The user gives an order, like "Open the oven," to start the operation.
2) As in Fig. 3, the LLM might process the instruction, map it into a set of fundamental motions found in the robot's basic library, and combine it with environment data.
3) Nevertheless, the horizontally hinged oven door in Fig. 4 cannot be opened by the first motion sequence, which is appropriate for a door with a vertical axis. Using the teleoperation and user interface depicted in Fig. 5, the user detects it, halts the operation, and adjusts motion functions.
4) After that, the revised new motion functions sequence is The new motion trajectory dictated by manual teleoperation can be called by the function 'DMP pub', and it is saved in the DMP library. These files are stored in the DMP library that the LLM has called "open oven handle."
5) After retrieving the task "Open the oven," the LLM renames it "open oven handle," takes the revised motion functions sequence out of the task's DMP library section, and gives the robot permission to carry out the task as shown in Fig. 4.

## IV. EXPERIMENTS AND DISCUSSION

Toyota's Human Support Robot (HSR) has been used in real-world testing of the HRC framework, which is based on the LLM. A 3-DoF mobile base, a 5-DoF arm (4-DoF for rotating joints and 1-DoF for a torso lift joint) with a gripper, and a 2-DoF head make up the HSR's total of 10-DoF. Teleoperation is possible with Oculus VR. Fig. 6 displays the objects of the experiment. The studies were conducted in a well-lit kitchen, as depicted in Fig. 6(b), and the robot was given orders in natural language from users to carry out a range of household chores. The studies are designed to assess the system's performance in routine zero-shot activities as well as other areas of complex and unique one-shot jobs.

### A. Zero-shot Basic Tasks

1) Put and Stack: The Put motions can be used to put an object above or inside of a target at a given distance. The customized LLM appends a suffix, "above," for placement 10 mm above the target, and "inside," for insertion, such as "object above," to describe the target context. The following are the Put&Stack motion functions:

```
# Move to the first object's position
move_to_position(object1)
# Close gripper
gripper_control(close)
# Move to the second object's above position
move_to_position(object2_above/inside)
# Open gripper
gripper_control(open)
```

2) Open: This motion's impact To open a door is to have its axis perpendicular to the ground; otherwise, it cannot be opened.

```
# Move to the door handle's position
move_to_position(microwave_handle)
# Close gripper
gripper_control(close)
# Open the door
base_cycle_move(radius_door2axis)
# Open gripper
gripper_control(open)
```

3)Close: Only doors with their axis perpendicular to the ground may be closed with this motion.

```
# Close the door
close_move(object)
```

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

4) Power on To power the item, this motion involves rotating the waist and moving to the target.

```
# Move to the object
move_to_position(microwave_knob)
# Rotate waist to power on the object
rotate_waist(degree)
```

5) Warm-up: This long-term goal, which consists of multiple smaller actions, is to warm up the apple.

```
# Generate sub-tasks
sub-tasks = ['open the microwave', 'put the apple
into the microwave', 'close the microwave', 'power
on the microwave']
# Execute the functions of the sub-task in order
for sub-task in sub-tasks:
   execute(sub-task)
```

**B. Discussion**

For all of the one-shot and zero-shot fundamental tasks Table 1 lists the DMP-based activities that are used to assess their performance.

Seven unique tasks were evaluated in total, each requiring 23 trials for a total of 161 trials.

TABLE I
EXECUTABILITY, FEASIBILITY, AND SUCCESS RATES OF LLM-BASED AUTONOMY AND HUMAN-ROBOT COLLABORATION

| Tasks | Num of trials | Executability | Feasibility | Success rate |
|---|---|---|---|---|
| Put&Stack | 23 | 100.0% | 100.0% | 91.3% |
| Open microwave | 23 | 100.0% | 100.0% | 82.6% |
| Open oven (HRC) | 23 | 100.0% | 100.0% | 91.3% |
| Open cabinet (HRC) | 23 | 100.0% | 100.0% | 87.0% |
| Clean table | 23 | 100.0% | 95.7% | 87.0% |
| Warm up apple | 23 | 100.0% | 100.0% | 60.9% |
| Roast apple (HRC) | 23 | 95.7% | 87.0% | 56.5% |
| Total | 161 | 99.4% | 97.5% | 79.5% |

TABLE II
COMPARISON OF EXECUTABILITY, FEASIBILITY, AND SUCCESS RATES FOR SUB-TASKS WITH AND WITHOUT HRC

| Tasks | Num of trials | Executability | Feasibility | Success rate |
|---|---|---|---|---|
| Open oven | 1 | 100.0% | 0.0% | 0.0% |
| Open oven (HRC) | 23 | 100.0% | 100.0% | 91.3% |
| Open cabinet | 1 | 100.0% | 0.0% | 0.0% |
| Open cabinet (HRC) | 23 | 100.0% | 100.0% | 87.0% |

The integrated Pythonic code's "Executability" score indicates if it can be executed and adheres to a preset format. The percentage of tasks for which the code is successfully executed to the total number of tasks assessed is used to quantitatively measure this score. The job "Roast apple (HRC)" was the sole deviation from the average score of 100%, with executability falling to 95.7%.This was ascribed to the LLM's infrequent creation of smaller tasks without designating corresponding motor functions, which resulted in responses that were not interpretable.

Table II provides proof of HRC's vital role, which is highlighted by this score. Particularly, after adding HRC, tasks like "Openoven" and "Open cabinet," which were previously judged unfeasible with a feasibility of 0% (impossible to be accomplished using motion functions from the basic library), showed notable gains, highlighting HRC's improvement on task execution. Furthermore, several jobs, like "Cleantable" and "Roast apple (HRC)," received scores of 95.7% and 87.0%, respectively. The margin is within the discrepancy range for certain activities, like inserting an apple that is 0.063 meters high into an oven that is 0.086 meters high, with a margin of 0.0115 meters and a discrepancy range of 0.0095 meters to 0.0122 meters, with a median discrepancy of 0.01 meters to 0.012 meters. Furthermore, mistake accumulation will certainly cause the success rates of the long-horizon tasks to decline, even while the success rates of all sub-tasks in our tests are above 80%.

## V. CONCLUSION

By integrating the LLM with robotic systems through an interface that allows task execution through natural language, this research advances robot autonomy. Additionally, a novel hierarchical planning framework based on LLM effectively handles long-horizon jobs, demonstrating advanced task planning. Additionally, by integrating DMP and teleoperation into the HRC framework, it is possible to improve LLM-based robotic manipulation through human demonstrations. The efficiency of the suggested strategy is demonstrated by the experimental findings, which indicate an average success rate of 79.5%, 99.4% executability, and 97.5% feasibility across a range of tasks. These findings highlight the system's resilience in converting language commands into robot movements and incorporating operator instructions to complete tasks that were previously unachievable, marking a significant advancement in the LLM-based robot's ability to perform tasks that are complex in the real world. However, future research will concentrate on addressing a drawback of existing LLM-based robots that mostly rely on visual inputs on combining tactile sensing technologies with point clouds generated by LIDAR to improve the suggested LLM-based robot performance in real-world settings.

## REFERENCES

[1] A. Vaswani et al., “Attention is all you need,” arXiv preprint arXiv:1706.03762, 2023.
[2] H. Touvron et al., “Llama: Open and efficient foundation language models,” arXiv preprint arXiv:2302.13971, 2023.
[3] M. Ahn et al., “Do as I can and not as I say: Grounding language in robotic affordances,” in arXiv preprint arXiv:2204.01691, 2022.
[4] J. Wu et al., “Tidybot: Personalized robot assistance with large language models,” Autonomous Robots, 2023.
[5] J.Redmon, S.Divvala, R.Girshick, and A.Farhadi, “You Only Look Once: Unified, real-time object detection,” arXiv preprint arXiv:1506.02640, 2016.
[6] A. Radford et al., “Learning transferable visual models from natural language supervision,” arXiv preprint arXiv:2103.00020, 2021.
[7] A. Kirillov et al., “Segment anything,” arXiv preprint arXiv:2304.02643, 2023.
[8] I. Singh et al., “Prog Prompt: Generating situated robot task plans using large language models,” in 2023 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2023, pp. 11 523–11 530.
[9] Z. Zhao, W. S. Lee, and D. Hsu, “Differentiable parsing and visual grounding of natural language instructions for object placement,” in 2023 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2023, pp. 11 546–11 553.
[10] J. Liang et al., “Code as policies: Language model programs for embodied control,” in 2023 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2023, pp. 9493–9500.
[11] J. Liu, Z. Li, W. Lin, S. Calinon, K. C. Tan, and F. Chen, “Softgpt: Learn goal-oriented soft object manipulation skills by generative pre- trained heterogeneous graph transformer,” in 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2023, pp. 4920– 4925.
[12] W. Wang, X. Li, Y. Dong, J. Xie, D. Guo, and H. Liu, “Natural language instruction understanding for robotic manipulation: a multisensory per- ception approach,” in 2023 IEEE International Conference on Robotics and Automation (ICRA), 2023, pp. 9800–9806.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

[13] R. C. Goertz, "Manipulator systems developed at anl," in Proceedings, The 12th Conference on Remote Systems Technology, 1964, pp. 117–136. [14] J. Nakanishi, S. Itadera, T. Aoyama, and Y. Hasegawa, "Towards the development of an intuitive teleoperation system for human support robot using a vr device," Advanced Robotics, vol. 34, no. 19, pp. 1239–1253,