

Chief Scientist Office 2024 Team Description Paper

Takaaki Numai Airi Yokochi Joshua Supratman
Tatsuro Sakaguchi Yushi Kaida Gakuto Okamoto*

November 21, 2023

Abstract. This paper presents SoftBank Corp. Japan team Chief Scientist Office's new mobile manipulator "SOAR" for RoboCup 2024 @Home Open Platform League. Differing from our previous participation with "Cuboid-X," a robot assembled from commercial products, SOAR integrates both custom-developed and commercial components, a significant step toward developing a customizable and affordable robotic platform. Our main achievement this year has been the successful integration of various hardware components and software development to tackle RoboCup challenges.

Our approach includes detailed hardware design and calibration, such as the arm and the sensor-rich head unit. The software focuses on developing systems, such as object and speech recognition, tailored to the dynamic environment of RoboCup.

SOAR embodies our vision of a robotic system that combines modularity with cost-effectiveness. As we continue to refine SOAR's capabilities further, we are optimistic about its potential to create an adaptable, cost-effective robotic platform for a variety of applications and fields.

1 Introduction

This paper details the RoboCup @Home Open Platform League team Chief Scientist Office from SoftBank Corp. Japan. This is our second appearance at RoboCup @Home Open Platform League. Since our last participation in 2022, we have made significant improvements, which include the introduction of our new mobile manipulator "SOAR."

At SoftBank Corp. Chief Scientist Office, we aim to pioneer a customizable and affordable commercial robot platform akin to the modular ease found in computer assembly or the historical T-Ford model. Our mission is to democratize

*The order of authors is based on age.

robotics, making it accessible and adaptable for research and commercial use. We believe in a future where integrating robotics into business is as straightforward as selecting components for a custom PC build. With SOAR as a stepping stone, we aim to develop a versatile platform that celebrates modularity in both hardware and software, allowing seamless integration and upgrades through a suite of interchangeable modules.

1.1 Focus of research

Our research is motivated by the ambition to develop a customizable and affordable robot platform. We recognize the constraints of current technologies and are committed to exploring their limits and potential. Key areas of our interest include but are not limited to:

- Modularity and Configurability: Investigate how to design a system that allows for easy integration and customization of hardware modules.
- Technological Feasibility: Assessing the capabilities of existing technologies to meet our affordability and performance criteria.
- Device Integration: Tackling the hardware challenges related to the seamless integration of diverse devices and components
- Software Optimization: Addressing the limitation of current software modules, such as MoveIt, by enhancing computation speed and completing critical functions.

We believe that these research areas are crucial to bridging the gap between the aspirational and the practical, laying the groundwork for a robot that can adapt to various needs.

2 Hardware

SOAR is a mobile manipulator comprising of four main components; the mobile base, the manipulator, the torso unit, and the head unit. The mobile base and the manipulator are commercially available products while the torso and head units are developed in-house and designed by our team. The hardware is designed with a modular approach so that it not only facilitates customization but also simplifies maintenance and upgrades, allowing for the interchange of components as needed. Fig. 1 shows the the image of SOAR.

2.1 Mobile Base

SOAR's mobile base is a 2-wheel differential drive mobile base developed by F-Design¹. It is supplemented with two casters and each driving wheel and caster are equipped with suspensions for smoother movement and stability. The base has a maximum speed of 3.0 km/h. We made additional modifications to the unit, equipping it with a LiDAR and an IMU sensor for navigation. We also integrated a docking charging connector, enabling SOAR to recharge its battery autonomously once depleted, enhancing its operational autonomy and efficiency.

¹ <https://f-ds.jp/>

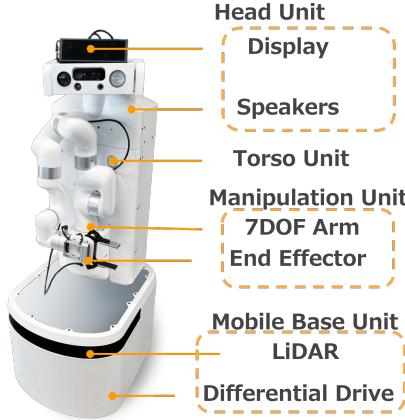


Fig. 1: Image of SOAR.

2.2 Torso and Manipulator

SOAR’s arm is composed of two units; the torso and the manipulator. The manipulator is xArm 7², a commercially available product developed by uFactory. It has seven degrees of freedom with a payload of 3.5 kg. The end-effector is equipped with a two-finger gripper and a RGB-D hand camera.

The torso unit, developed by our team, is a linear actuator with one degree of freedom equipped with a constant-force spring to provide gravity compensation. The unit elevates the head and manipulator units, providing a clearance of 410 mm, significantly enhancing the latter’s reach and versatility.

2.3 Head Unit

The head unit has two main purposes: recognizing and understanding the environment and facilitating social interaction with people. The head unit is a hub of sensory and interactive technology, equipped with various devices including a RGB-D camera, a zoom camera, a thermal camera, a LED light, a head display, a microphone, and speakers. The device placement is shown in Fig. 2 Additionally, the head has 2 degrees of freedom for pan and tilt movements, enhancing its SOAR’s ability to survey diverse environments and engage in social interaction more effectively.

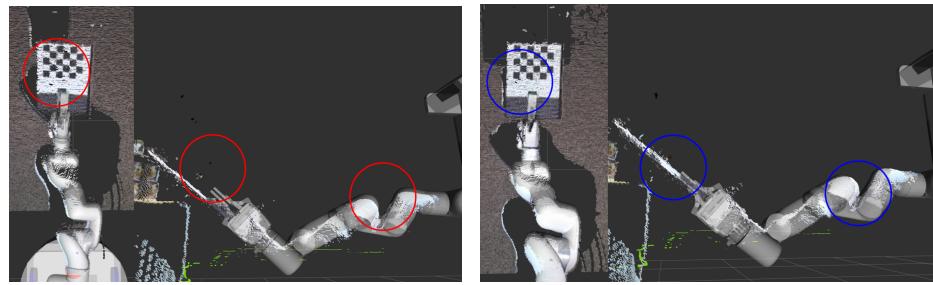
2.4 Calibration

SOAR’s design incorporates interchangeable modules like the manipulation unit. This, however, brings forth unique calibration challenges. These challenges arise

² <https://www.ufactory.cc/xarm-collaborative-robot/>



Fig. 2: Device placement on the head unit.



(a) Before calibration. Red circles show the discrepancies between the measured joint and the actual joint.
(b) After calibration. Blue circles show the alignment between the measured joint and the actual joint, indicating improved accuracy and precision in the measurement.

Fig. 3: Calibration results.

due to alignment issues, accuracy demands, adaptability needs, and the importance of consistent performance across different modules. Calibration, therefore, becomes essential to ensure that each module functions correctly within the system.

To address this, we employ the `robot_calibration[1]` ROS package. This package allows us to fine-tune the arm, aligning the measured joint values with their actual physical counterparts, ensuring accuracy. While we achieved good calibration results for the manipulator unit, as shown in Fig. 3, the task is more challenging for the head unit and is still undergoing refinement to reach the desired level of accuracy.

3 Software

The overview of our software system, shown in Fig. 4 is comprised of three main layers:

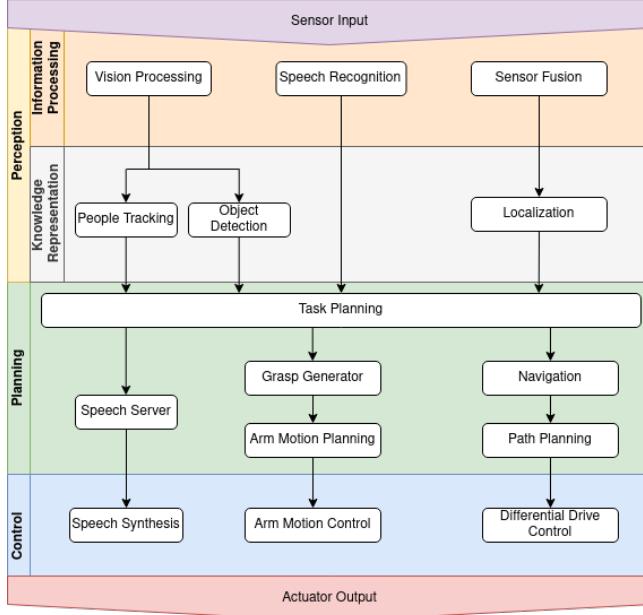


Fig. 4: SOAR software overview.

1. The perception layer, a collection of software modules that take the robot’s sensor data and process them to create a symbolic representation of the environment,
2. the planning layer, a collection of software modules that take the state of the environment to evaluate and generate a sequence of appropriate robot behaviors, and
3. the control layer, the collection of software modules that translates the planned behavior and sends appropriate commands to the actuators for the robot to interact with the environment

The primary framework for software module interaction is Robot Operating System (ROS)³, with each software module representing one or several ROS nodes.

3.1 Object Recognition

Object detection and 3D localization are handled through the seamless integration of existing technologies. We employed ultralytics’s YOLOv8[2] for its object recognition capabilities, producing bounding boxes with labels for each detected object in the robot’s field of view. To understand the object’s spatial orientation and structure, we used the jsk_pcl_ros⁴ package for its robust point

³ <https://www.ros.org/>

⁴ https://github.com/jsk-ros-pkg/jsk_recognition

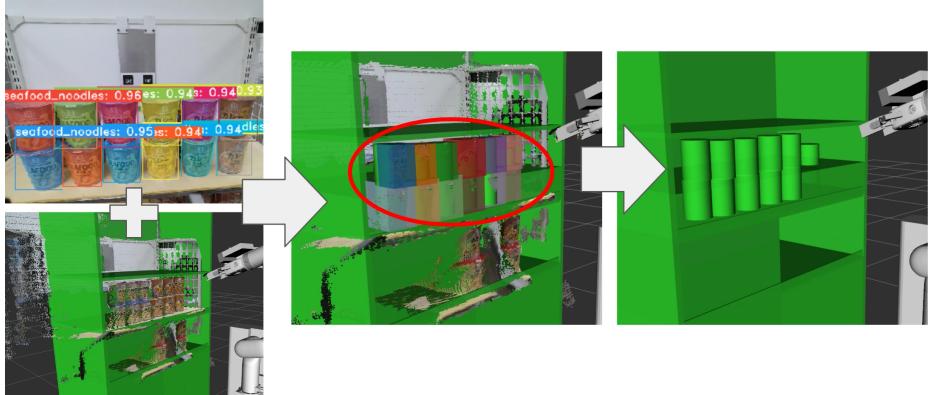


Fig. 5: Object detection pipeline. YOLOv8’s bounding boxes and the point cloud data are given to jsk_pcl_ros package to generate 3D collision boxes, circled in red, which are passed to MoveIt planning scene.

cloud processing. We used the EuclideanClustering and the ClusterPointIndices-Decomposer functionalities within this package to segment and cluster point cloud data within the detected bounding boxes from YOLOv8 and generate 3D collision boxes along with its pose. The pipeline is illustrated in Fig. 5.

3.2 Creating dataset

To train YOLOv8 for object recognition in the dynamic nature of RoboCup competitions, where new objects are often introduced on the day of the event, our team has developed a robust method for dataset preparation. The process involves capturing images of objects from multiple angles and capturing background images on the RoboCup field. We then crop the object images using a GUI tool and overlay them onto the background images, creating composite images. We then generate a COCO-style dataset with the composite images. An example is shown in Fig. 6. This method significantly simplifies the annotation process, allowing us to create datasets with new objects within a day, thus enabling rapid adaptation and expansion of our object recognition system.

We are continuously working to enhance our model’s capability to recognize challenging objects, such as transparent objects or surfaces with reflective properties like silver. We are also refining our data collection techniques to consider factors such as varying camera angles and light intensities, which are critical not only for realistic and effective annotation but also for creating 3D simulation models. These improvements are aimed at increasing the robustness and accuracy of our object recognition system under varied and unpredictable competition conditions.

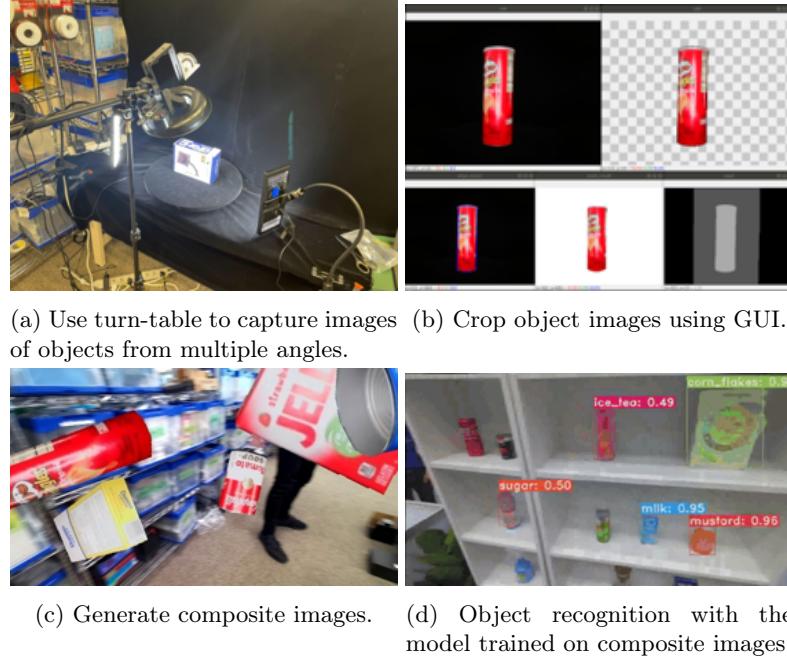


Fig. 6: Dataset creation process.

3.3 Speech Recognition And Interaction

We use EfficientWord-Net[3] to detect hotword to initiate the conversion. Once engaged, we employed Google Dialogflow for conversation, which can interpret spoken phrases and generate relevant responses based on our predefined conversational settings. For vocal output, we use Google Text-To-Speech.

In addition to the current pipeline, we are also exploring the integration of ChatGPT[4] as a potential replacement for Google Dialogflow to enrich SOAR’s conversational capabilities. To facilitate this transition, we use Silero VAD[5] to segment the voice signals and Whisper[6] to convert speech to text.

4 Contribution

SoftBank Corp. has continued to sponsor various robotic competitions⁵⁶ and conferences⁷⁸ to advocate human-robot interaction research. Our sister company, SoftBank Robotics Corp. is a long-term global sponsor for RoboCup, which further exemplifies our commitment to advancing robotic technologies and educa-

⁵ <https://wrs.nedo.go.jp/en/sponsor2020/>

⁶ <https://tsukubachallenge.jp/2023/organization/sponsors>

⁷ <https://ac.rsj-web.org/2023/>

⁸ <https://roscon.ros.org/jp/2023/>

tion. Despite our low direct scientific contribution, SOAR, our current prototype, represents our venture into supporting the research community and industry by offering a platform for research and study. While it is still in development, SOAR is designed with the future in mind, a testament to our initiative in paving the way for affordable and customizable robots, which we believe will make significant contributions to both academic research and practical applications.

5 Conclusions and future work

In this paper, we presented our currently developing robot SOAR which will participate in the RoboCup@Home Open Platform League. We detailed the hardware specification and introduced several software modules, such as object recognition and speech recognition, which we will continue to refine to tackle the tasks in RoboCup @Home for our future work. By enhancing SOAR’s capabilities in real-world interaction, we aim to contribute not just a flexible and affordable platform but also a robust platform for research and commercial use.

References

1. Michael Ferguson and Niharika Arora. Robust and Efficient Calibration of Mobile Manipulators. Technical report, Fetch Robotics, Tech. Rep, 2015.
2. Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics YOLOv8, 2023.
3. R. Chidhambararajan, Aman Rangapur, S. Sibi Chakkavarthy, Aswani Kumar Cherukuri, Meenalosini Vimal Cruz, and S. Sudhakar Ilango. EfficientWord-Net: An Open Source Hotword Detection Engine Based on Few-Shot Learning. *Journal of Information & Knowledge Management*, 21(04):2250059, 2022.
4. OpenAI. GPT-4 Technical Report, 2023.
5. Silero Team. Silero VAD: pre-trained enterprise-grade Voice Activity Detector (VAD), Number Detector and Language Classifier. <https://github.com/snakers4/silero-vad>, 2021.
6. Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust Speech Recognition via Large-Scale Weak Supervision, 2022.

SOAR Hardware Description

Specifications are as follows:

- Base: differential drive, 0.83 m/s max speed.
- Dimensions: (w) 550 mm (d) 650 mm (h) 1360 mm
 - 1770 mm
- Weight: 92 kg
- Arm: 8 DOF (1 DOF torso, 7 DOF manipulator)
- Sensors:
 - Mobile Base:
 - * Hinson LE-50621 LiDAR
 - * N100 9-axis IMU
 - Head:
 - * Azure Kinect RGB-D camera
 - * ECM-SP10 microphone
 - * JM800S-30-360X zoom camera
 - * Seek Thermal CompactPRO Fast Frame
 - Arm:
 - * Realsense D435 RGB-D camera
- PC: Intel NUC (NUC11PHKi7C)
- Display: THANKO C-79D21B
- Speaker: YAMAHA VXS1MLB



SOAR Software Description

SOAR uses the following software components:

- Platform: Ubuntu 20.04 ROS Noetic
- Navigation: ROS Navigation Stack, EBand local planner
- Arm Motion Planner: MoveIt ROS package
- Object Recognition:
 - YOLOv8
 - jsk_pcl_ros ROS package
- Behavior Architecture: SMACH ROS package
- Hotword: EfficientWord-Net
- Speech Recognition: Google Dialogflow
- Speech Generation: Google Text-To-Speech
- Person Recognition: spencer_people_tracking ROS package