# LASR 2024 Team Description Paper

Matteo Leonetti, Gerard Canal, Nicole Lehchevska, Pawel Makles, and Jared Swift

Department of Informatics, King's College London, WC2B 4BG London, UK
https://www.sensiblerobotsresearch.org/lasr/

**Abstract.** The Learning Autonomous Service Robots (LASR) team is an emerging team, which participated in RoboCup@Home 2022 for the first time. The team is part of a larger research group whose focus is on using machine learning to improve the ability of robots to integrate with the dynamic everyday environment, and the people that inhabit it. We describe our research results and the capabilities of the robotic platform for participation in RoboCup@Home.

## 1 Introduction

The LASR team was founded in 2019 as the Leeds (now Learning) Autonomous Service Robot team, and has since participated in three competitions of the European Robotics League (ERL), namely, two Sciroc Challenges[1] (2019 and hybrid 2021), and the Smart Cities Challenge[2] (2023); a virtual RoboCup@Home (2021) and RoboCup@Home 2022 in Thailand. We achieved first place in the Coffee Shop episode in 2019. During the pandemic, we ranked 5th out of 10 in the virtual RoboCup@Home, and achieved third place in the remote (stremed from our lab) pick-and-place manipulation Sciroc Challenge. After the pandemic, like most teams, we had to reboot. We took part in RoboCup@Home 2022, our first in-person RoboCup, and restored a fully working team for the ERL Smart Cities Challenge 2023, where we achieved first place in both the Coffee Shop and the Elevator task, in addition to the award for best team overall, shared with B-it-bots.

The team also runs a robot club society at King's College London, where any students interested in robotics can come, learn, and participate. We use our current research goals and competitions to provide direction and output-focused projects to help drive engagement and relevance. The team is therefore both an education and research platform, which several of the previous members have described as the highlight of their university experience. We currently have members at all levels: undergraduate, master's, and PhD students.

[1] https://sciroc.org/challenge-description-2019/
[2] https://eu-robotics.net/erl-smart-city-competition-in-2023/

## 2 Research

The group's research is centred on adaptation for decision-making in autonomous agents, with robots as one of the natural applications. Some of the research lines have an immediate use in RoboCup, while others are more oriented towards the future of service robots. This is particularly the case for online learning, which has not been part of the competition yet, but we consider nonetheless central to the field.

In this section, we will first briefly describe the research on planning and learning that we believe will have a future role in autonomous robots. Then, we will present our research on robot manipulation and human-robot interaction, which tackles more directly some of the challenges of the last RoboCup@Home.

We expect future robots to be versatile, being able to solve a variety of tasks in an environment also inhabited by humans. Of the three main techniques to define an agent's behaviour—programming, planning, and learning— programming (often in the form of state machines) is by far the most popular, but also the least prone to versatility. Our research is aimed at transitioning from state-machine representations towards a combination of planning and learning over the upcoming years, largely based on the results described in the next section.

### 2.1 Adaptation in Planning

Reinforcement Learning (RL) agents learn behaviours through exploration, that is, trying new actions and evaluating their long-term effects. Exploration is necessary for learning but also difficult to harness in robotics, since costs in terms of time, energy and wear and tear can be prohibitive. This phenomenon has almost entirely relegated robot learning to simulations. With the goal of making online learning practical for real robots, we study how planning and learning can complement each other.

Reasoning over models allows the agent to strongly limit the exploration to actions that lead towards the goal. Since "all models are wrong, but some are useful", the interaction of planning and learning can meaningfully drive exploration greatly reducing the sample complexity of RL agents, while the adaptation provided by model-free RL allows to overcome the inevitable inaccuracies of the models. We developed methods to make use of action languages while learning action costs from the real world [16], or that integrate with Answer Set Programming [18] to constrain the exploration to safe and explainable behaviours, while adapting to the unmodeled aspects of the environments.

Planning is a notoriously computationally hard problem in general, but effective heuristics can make planning feasible in a number of scenarios of practical interest. We developed learned heuristics from meta-reinforcement learning, so that previously solved tasks can inform the search on new related tasks [13,14]. We also developed a method to reduce, over time, the planning horizon, so that the agent behaviour gradually transitions from model-based to model-free [10]. Beyond being hard to compute, executing plans is also a difficult task due to the high uncertainty that robotics domains present. We studied monitoring users to preventively replan when errors may occur [15] and also considered how to model planning problems better to prevent artificial dead-ends [4].

At King's College London, there is a long tradition on task planning applied to robotics, with contributions such as the ROSPlan framework [9]. ROSPlan has been actively in development since then, with upgrades that greatly simplify the use of different planners as well as their integration with robot sensors [7], and providing tools to implement low-level action execution with intermediate state machines [3].

### 2.2 Adaptation in Human-Robot Interaction

We recently started a new research line in adaptation to users with different abilities. We consider fully collaborative tasks, in which a robot and a person share a common goal. In defining robot actions, the designer further defines whether the action depends on human capabilities. For instance, a robot may be able to move at different speeds, with the action `move_fast` depending on the human collaborator to be able to `walk_fast`. For any new collaborator, the robot cannot know, beforehand, what capabilities they have. However, it starts from a prior, and through reinforcement learning and interaction, it estimates the capability level of the collaborator. If the robot collects sufficient evidence that the person does not have an ability necessary for a given action, the robot adapts by disabling the corresponding action and finding a new way to carry out the task. The robot can, therefore, tailor its level of support from minimal, for fully-abled people, to carrying out most of the task when assisting a disabled person. We demonstrate the adaptation on several tasks, including a real-world experiment using our TIAGo robot [23].

We also carry out research in assistive robotics and robot adaptation to preferences [8,6], as well as efforts towards explainability of the robot's motions and behaviour [5,25].

### 2.3 Learning for Manipulation

We tackled two manipulation problems for which efficient planners are not available: manipulation in clutter, and with deformable objects. Most consolidated manipulation strategies for rigid objects compute collision-free trajectories, which cannot be used in clutter. Positioning and retrieving objects from shelves are examples of manipulation often involving clutter, also recognized at RoboCup@Home. Considering the interaction with other objects makes the trajectory planning problem significantly more complex, especially if, in addition to grasping, other physics-based actions (such as pushing and sliding) are taken into account, whose effects are difficult or expensive to predict accurately. We developed a learning-based Receding Horizon Planner, which tackles two challenges: the computational complexity of the problem when considering interactions between all objects, and the inaccuracy of models, whose predictions accumulate errors and become invalid after a small number of actions. We used a learned value function in simulation as a heuristic for planning, both influencing action probabilities during rollouts and providing a cost-to-go estimate for states at the end of the short-horizon plan. The short horizon enables quick reaction times. Rather than planning for each problem as if that was the first one ever encountered, experience is accumulated in the value function so that previously solved problems provide a heuristic for the new ones. The system has been extended [1] to retrieve objects in the more realistic scenario of partial observability, with the robot looking at the shelf from the side, and also demonstrated on a real robot [2].

Recently, we developed a planning algorithm to simplify the actions when planning for deformable objects, such as for cloth folding [26]. We expect deformable object manipulation to play an increasing role in RoboCup@Home, given the natural application in the home setting.

### 2.4 Curriculum Learning

Knowledge transfer between related tasks is another approach to agent versatility, increasing the range of capabilities of the autonomous system, while learning new tasks increasingly faster. Curriculum learning consists in learning through tasks of growing complexity, towards one or more final tasks, so that learning is either faster, or results in a better learned behaviour than from scratch. The automatic generation of curricula involves a number of interesting challenges: in the definition of tasks at the appropriate level of difficulty for the agent, in the knowledge transfer methods that allow the agent to take advantage of previous tasks, and in the sequencing of tasks once they have been generated. Nonetheless, curriculum learning is widespread in any level of human learning, from motor control to higher education, and there is no doubt that the order in which we learn matters. Our team, with collaborators, contributed to the problem of optimal curriculum generation: a set of strategies to create intermediate tasks for artificial agents [19], a method to estimate the transfer potential between tasks [22], the first algorithm to generate curricula that require no learning in the process [24], a formalization of the problem in the framework of combinatorial optimization [12], and an algorithm for task sequencing in critical, real-world problems [11]. The field has grown significantly under the pressure that deep learning has put on sample complexity, to the point that most deep learning applications employ some form of curriculum, often implicitly defined by hand.

## 3 System Architecture and Capabilities

The research group owns a TIAGo Steel robot from PAL Robotics[3], as seen in the addendum. The robot has a mobile base with a differential drive mechanism, battery pack, laser range finder, rear sonar sensors and an onboard computer. The torso has a lifting mechanism, houses the onboard microphone array and supports a 7 degree of freedom (DOF) arm with gripper and a 2 DOF head. The head houses an RGB camera and depth sensor setup.

[3] `http://pal-robotics.com/`

The TIAGo robot comes with the ROS middleware on top of which PAL has developed their own proprietary middleware. We have then integrated our own software either directly through ROS, or through PAL's middleware layer.

### 3.1 Current Capabilities

Some of the currently implemented capabilities are described below, and in most cases can be found within our GitHub Organisation[4], available to the public and in particular to the RoboCup@Home community.

**Task Architectures** Many of the below capabilities are implemented as standalone ROS packages, which are intended to be robot-agnostic. Cohesive uses of these capabilities are implemented through *robot skills*, which are individual States or small Finite State Machines (FSMs), which can be easily dropped into larger Hierarchical Finite State Machines (HFSMs), due to their well-defined interfaces. As is often the case, when solving complex tasks, such as those at RoboCup, we tend to develop by hand a HFSM. In the future, we plan to enable the robot to perform its own reasoning about how best to solve the task at hand by utilising its aforementioned *robot skills*.

**Geometric Navigation** Any robot operating in the real world will need to be able to navigate autonomously, quickly adapting to and localizing within a dynamic environment. It will also need to be able to detect obstacles and understand its own infrastructure to be able to navigate successfully. We make use of the ROS package *move_ base* with proprietary PAL planners for both global and local planning.

**Social Navigation** It is often the case that robots navigating in the wild look unnatural or break social conventions, especially in situations where crowded spaces are involved, such as waiting for and subsequently riding an elevator. We developed two key approaches to deal with these problems, mainly focusing on how the robot should statically position itself. To contribute to decision-making about acceptable waiting positions outside of elevators, we constructed a dataset of laser readings represented as 2D images collected whilst the robot was waiting for the elavator, and finetuned a Keypoint RCNN model[5]. The navigation planner was used to filter out positions that couldn't be reached. For positioning the robot whilst riding the elevator, we use heightmaps - an approach borrowed from terrain representations. We use laser readings to construct a heightmap and select the least busy position, again using the navigation planner to filter infeasible positions.

[4] `https://github.com/LASR-at-Home/`

[5] `https://pytorch.org/vision/main/models/keypoint_rcnn.html`

**Object Detection and Recognition** Object detection has been a hot topic in computer vision for many years, with many competing solutions vying for the top spot. After testing a number of implementations we have settled on the popular YOLO framework [21] for object detection. In addition to YOLO, we had previously fine-tuned a pretrained Mask R-CNN model on the YCB[6] dataset for object detection and 3D object segmentation. However, more recent versions of YOLO, namely YOLOv8, perform both object detection and 2D segmentation - which through further computation on the PointCloud, we scale to 3D. Whilst pretrained weights for YOLO exist that are trained on large datasets, encompassing many classes, such as COCO[7], often there is a need to detect specific object classes that are less general. Thus, we developed our own training pipeline[8]. We begin by collecting 2D images from the robot's camera of target objects at varying (but not exhaustive) rotations about each axis, through the use of a turntable with a uniform background. We segment the objects using SegmentAnything [17] to generate masks and generate a synthetic dataset by superimposing these masks onto random and realistic backgrounds. Our pipeline only takes as input the 2D images, generates the synthetic dataset, and trains a model using it (bootstrapping from pretrained weights), without the need for manual intervention. However, we found it quite useful to supplement our synthetic dataset with manually labelled, in-context images of the objects, again collected through the robot's camera.

**Person Detection and Recognition** The pretrained weights available to YOLO incorporate both objects and people. We have taken a slightly different approach where we train separate networks for objects and for people, and then contextually select which model to apply at runtime. However, more recently we implemented a ROS wrapper for BodyPix 2.0, which is specifically aimed at person detection, segmentation and joint-pose estimation. We apply the same method as we do to objects to produce 3D detections. For re-identifying people, we use a fairly simple solution. We maintain a database of images for each individual, and given a target image (cropped to only contain a single person) we perform a simple lookup in our database, using DeepFace for comparison. DeepFace verifies a match by evaluating a distance metric in facial-embedding space.

**Person Pose Estimation** Person pose estimation is a general problem in computer vision to deduce a person's behaviour from the position and orientation of their body. Previously, we used OpenPose to estimate people's poses - we have utilised it to recognise gestures, such as waving, alongside determining whether people are standing or sitting, and inferring what object someone is pointing at. As described above, we are now using BodyPix 2.0 for estimating people's poses, but the same methods for deducing behaviour apply.

[6] `https://www.ycbbenchmarks.com/`
[7] `https://cocodataset.org/`
[8] `https://github.com/insertish/yolov8-auto-trainer`

**Object Manipulation** We use the *MoveIt!* motion planning framework for object manipulation. It integrates 3D sensors with the Octomap, which implements 3D occupancy grid mapping to model arbitrary environments. This allows our robot to execute planning motions free of collisions to grasp the target object. In Robocup 2021, we used Grasp Pose Detection (GPD) to generate 6-DoF grasps that were executed by MoveIt. GPD generalizes well to unknown models because it takes in a pointcloud of an object and produces viable grasps. In SciRoc 2021, we used MoveIt to execute geometrically inferred grasps. More recently, we used Contact Graspnet, which specialises in grasp pose generation in cluttered scenes, and similarly to GPD just takes an input pointcloud.

**Social Interaction** Dialogue is a natural medium for humans to interface with robots. Previously, we used Google's Dialogflow cloud platform for transcribing audio and reasoning about the intents and entities present in the text - however, this requires an internet connection due to its cloud-based nature, which is not always available. Thus, we now utilise Whisper [20] for transcribing audio into text and then various context-dependent natural language understanding (NLU) models trained with Rasa for intent recognition and entity extraction. Our speech processing pipeline thus performs end-to-end audio to intent recognition and entity extraction. Communication through dialogue is not always possible, particularly when the human cannot speak. Thus, we have also implemented methods of communicating with our robot through various interfaces implemented on the tablet which is mounted on our robot's head.

## 4 Conclusion

We introduced the research and current capability of the LASR team. We believe that our research in adaptive decision making and reinforcement learning in the real world will bring a new perspective to the competition, strongly contributing to the development of service robotics for the home.

## References

1. Bejjani, W., Agboh, W.C., Dogar, M.R., Leonetti, M.: Occlusion-aware search for object retrieval in clutter. In: IROS (2021)
2. Bejjani, W., Leonetti, M., Dogar, M.R.: Learning image-based receding horizon planning for manipulation in clutter. Robotics and Autonomous Systems (2021)
3. Bezrucav, S.O., Canal, G., Cashmore, M., Corves, B.: An Action Interface Manager for ROSPlan. In: ICAPS PlanRob Workshop (2021)
4. Bezrucav, S.O., Canal, G., Coles, A., Cashmore, M., Corves, B.: Towards Automatic State Recovery for Replanning. In: ICAPS Intex Workshop (2022)
5. Brandão, M., Canal, G., Krivić, S., Magazzeni, D.: Towards providing explanations for robot motion planning. In: ICRA (2021)
6. Canal, G., Alenyà, G., Torras, C.: Adapting robot task planning to user preferences: an assistive shoe dressing example. Autonomous Robots (2019)

7. Canal, G., Cashmore, M., Krivić, S., Alenyà, G., Magazzeni, D., Torras, C.: Probabilistic Planning for Robotics with ROSPlan. In: TAROS (2019)
8. Canal, G., Torras, C., Alenyà, G.: Are Preferences Useful for Better Assistance?: A Physically Assistive Robotics User Study. ACM Transactions on Human-Robot Interaction (2021)
9. Cashmore, M., Fox, M., Long, D., Magazzeni, D., Ridder, B., Carrera, A., Palomeras, N., Hurtos, N., Carreras, M.: Rosplan: Planning in the robot operating system. In: ICAPS (2015)
10. Dunbar, L., Rosman, B., Cohn, A.G., Leonetti, M.: Reducing the planning horizon through reinforcement learning. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML-PKDD) (2022)
11. Foglino, F., Christakou, C.C., Gutierrez, R.L., Leonetti, M.: Curriculum learning for cumulative return maximization. In: IJCAI (2019)
12. Foglino, F., Christakou, C.C., Leonetti, M.: An Optimization Framework for Task Sequencing in Curriculum Learning. In: ICDL-EpiRob (2019)
13. Gutierrez, R.L., Leonetti, M.: Information-theoretic task selection for meta-reinforcement learning. In: NeurIPS (2020)
14. Gutierrez, R.L., Leonetti, M.: Meta reinforcement learning for heuristic planing. In: ICAPS (2021)
15. Izquierdo-Badiola, S., Canal, G., Rizzo, C., Alenyà, G.: Improved Task Planning through Failure Anticipation in Human-Robot Collaboration. In: ICRA (2022)
16. Khandelwal, P., Yang, F., Leonetti, M., Lifschitz, V., Stone, P.: Planning in action language BC while learning action costs for mobile robots. In: ICAPS (2014)
17. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., Dollár, P., Girshick, R.: Segment anything (2023)
18. Leonetti, M., Iocchi, L., Stone, P.: A synthesis of automated planning and reinforcement learning for efficient, robust decision-making. Artificial Intelligence (2016)
19. Narvekar, S., Sinapov, J., Leonetti, M., Stone, P.: Source task creation for curriculum learning. In: AAMAS (2016)
20. Radford, A., Kim, J.W., Xu, T., Brockman, G., McLeavey, C., Sutskever, I.: Robust speech recognition via large-scale weak supervision. In: Proceedings of the 40th International Conference on Machine Learning (2023)
21. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: CVPR (2016)
22. Sinapov, J., Narvekar, S., Leonetti, M., Stone, P.: Learning inter-task transferability in the absence of target task samples. In: AAMAS (2015)
23. Sun, C., Cohn, A., Leonetti, M.: Online human capability estimation through reinforcement learning and interaction. In: Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2023)
24. Svetlik, M., Leonetti, M., Sinapov, J., Shah, R., Walker, N., Stone, P.: Automatic Curriculum Graph Generation for Reinforcement Learning Agents. In: AAAI (2017)
25. Wachowiak, L., Tisnikar, P., Canal, G., Coles, A., Leonetti, M., Celiktutan, O.: Analysing Eye Gaze Patterns During Confusion and Errors in Human–Agent Collaborations. In: RO-MAN (2022)
26. Wang, S., Papallas, R., Leouctti, M., Dogar, M.: Goal-conditioned action space reduction for deformable object manipulation. In: 2023 IEEE International Conference on Robotics and Automation (ICRA) (2023)

## PAL Robotics TIAGo Steel Hardware Description [OPL]

PAL Robotics TIAGo is a customisable robot used commercially and for research. We have the Steel version of the robot with an additional Windows tablet. The specifications are as follows:

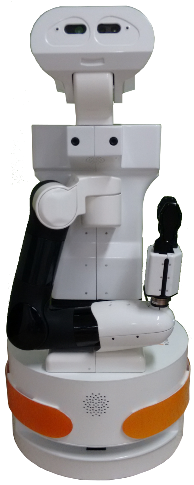

**Fig. 1.** PAL Robotics TIAGo Steel robot

- Base: differential drive base, 1m/s max speed.
- Torso: lifting, stroke 35cm.
- Arm: 7 DOF with gripper.
- Head: 2 DOF with sensors.
- Dimensions: height: 110 - 145cm, base footprint: 54cm diameter
- Weight: 72kg.

*Our robot incorporates the following devices:*

- External laptop with graphics card
- Touch screen Windows tablet (head mounted)
- External microphone array (potential)
- Nvidia Jetson TX2 (potential)
- Raspberry Pi 5 (potential)

## Robot's Software Description

| | |
|---|---|
| OS: | Ubuntu 20.04<br>http://releases.ubuntu.com/20.04/ |
| Middleware: | ROS Noetic + PAL<br>http://wiki.ros.org/noetic |
| Simulation: | Gazebo<br>http://gazebosim.org/ |
| Visualisation: | RViz<br>http://wiki.ros.org/rviz |
| Navigation: | move_base & pal_planner<br>http://wiki.ros.org/move_base |
| Manipulation: | MoveIt!<br>https://moveit.ros.org/<br>GPD<br>https://github.com/atenpas/gpd<br>Contact Graspnet<br>https://github.com/NVlabs/contact_graspnet |
| Depth Analysis: | PCL<br>http://pointclouds.org/ |
| Speech Analysis: | Dialogflow<br>https://dialogflow.com/<br>Whisper<br>https://github.com/openai/whisper<br>Rasa<br>https://rasa.com/ |
| Object & Person Recognition: | YOLO<br>https://pjreddie.com/darknet/yolo/<br>YOLOv8<br>https://github.com/ultralytics/ultralytics<br>SegmentAnything<br>https://github.com/facebookresearch/segment-anything<br>BodyPix 2.0<br>https://github.com/tensorflow/tfjs-models/tree/master/body-segmentation<br>https://github.com/de-code/python-tf-bodypix |
| Facial Recognition: | DeepFace<br>https://github.com/serengil/deepface |
| Complex Robot Planning: | SMACH<br>http://wiki.ros.org/smach<br>actionlib<br>http://wiki.ros.org/actionlib |
| Pose Estimation | OpenPose<br>https://github.com/CMU-Perceptual-Computing-Lab/openpose<br>BodyPix 2.0<br>https://github.com/tensorflow/tfjs-models/tree/master/body-segmentation<br>https://github.com/de-code/python-tf-bodypix |