Newsletter of the Robotics Society Of India

Vol. 1, No.2, 2013

From the Editor's Desk

"We take off from where we left in the previous Newsletter as we showcase the research and development activities pertaining to various aspects of robotics in various parts of the country. Considering brevity of space this newsletter is dedicated to few of the several labs in IIT Madras that have contributed to robotics in the country. We expect to cover the remaining in future versions. We request the alert reader to make us aware of new labs coming up in various parts and send descriptions of the same.

Here's wishing a safe and peaceful new year that enhances our research and development productivity"

Dr. Madhavs Krishna
IIIT Hyderabad

Real-Time Verification of 3D Obstacle Avoidance Algorithm for Autonomous Underwater Vehicles Using Hardware-In-Loop Setup

S. Saravanakumar, T. Asokan

Robotics Laboratory, Department of Engineering Design
Indian Institute of Technology Madras, Chennai 600 036

In this article, we present an overview of the development of Hardware-in-the Loop (HIL) simulation setup used to test and verify control algorithms for autonomous underwater vehicles. Obstacle avoidance algorithm for an Autonomous Underwater Vehicle has been taken as an example and the HIL setup developed for testing this algorithm has been explained.

Autonomous Underwater Vehicle (AUVs) is a robotic device that is driven through the water by a propulsion system, controlled and piloted by an onboard computer, and maneuverable in three dimensions. In recent years, AUVs have become more popular due to their autonomous nature of operation, long-duration exploration of the ocean, ability to gather information, and the potential in both military and civilian applications. The main objective of the HIL simulation is to verify the real-time operation of the model without developing the actual system. Real-time testing of control algorithms are costly and time consuming. In HIL simulation, the simulation model is embedded into a real-time controller and the hardware components are physically connected with a real-time simulation model. A multi-point potential field algorithm has been developed by the authors for obstacle avoidance applications in static and dynamic obstacle environments. The main idea of the potential field method is to generate attraction and repulsion potentials for the target and the obstacles. The target has an attraction potential and the obstacles have repulsion potential. In the multi-point potential field method, the total potentials are generated at multiple points. By determining the point at which the minimum potential exists among the total potentials, the vehicle can be commanded to that point. The developed obstacle avoidance algorithm is interfaced with the AUV dynamic model and other necessary systems such as sensor signal processing module, trajectory planner, controllers and actuators (thrusters and control planes).

In the developed HIL simulation system, the control algorithm and AUV dynamic model are developed in Simulink and executed using dSPACE DS1104 controller board. dSPACE DS1104 R&D controller board is a complete real-time control system based on 603 PowerPC floating point processor running at 250MHz. From the real-time interface (RTI) I/O library from the implementation software "RTI", the standard I/O channels blocks such as ADC and DAC and more complex I/O devices blocks such as incremental encoder and RS-232 are picked up and attached to the Simulink model. The serial receiver is set to receive the obstacle sensor data at 9600 baud rate. The input channel ADC 5 is used to read the analog voltage corresponding to the actual speed of the thruster motor. The output channels DAC 1, 2 and 3 are used for sending out the desired analog signals to the thruster, pitch (stern) and yaw (rudder) motors respectively. The angular position of pitch and yaw motors are read by the encoder channels 2 and 1. Then the Simulink model is converted into real-time interface (RTI) model using "RTI". Once RTI model is built, the real hardware can be connected as shown in Fig. 1. Another software, also from dSPACE, "ControlDesk" is used to develop the Graphical User Interface (GUI). The various parameters to be displayed and plotted are selected from the instrument and data acquisition panels of the "ControlDesk" software. Their corresponding data are captured for further analysis of the results. Figure 2 shows the AUV on its development bench.

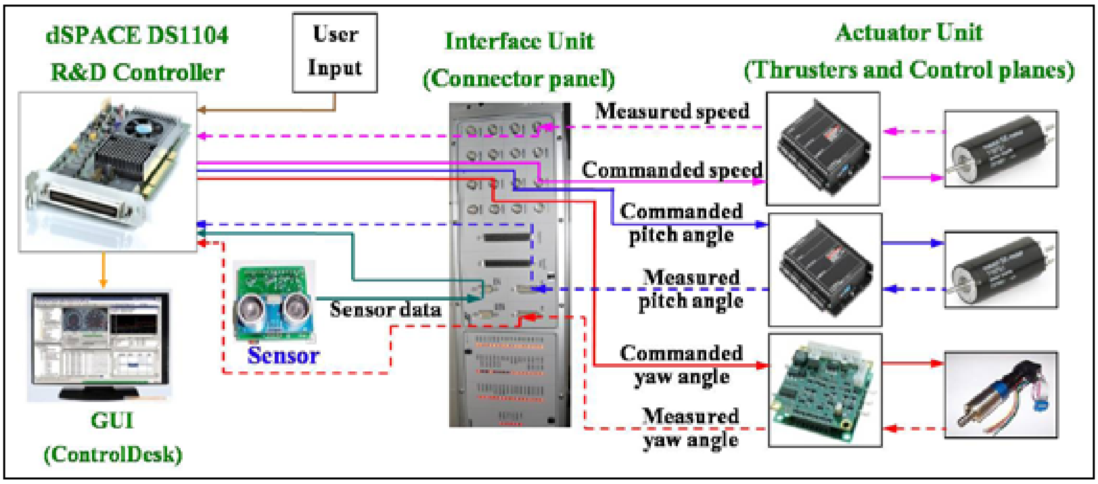


Fig. 1. HIL simulation setup

Ultrasonic sensors are used on the AUV to detect the obstacles. Brushed DC servo motors are used as thrusters and brushless DC servo motors are used for stern and rudder control planes. The thruster motor is used to control the motion in x (surge) direction. The stern motor is used for pitch and z (heave) direction control whereas rudder motor is used for yaw and y (sway) direction control. A typical result from the HIL study is shown in Fig. 3 for a commanded AUV path with static obstacles. The HIL simulation results show that the developed test bench setup is very much useful and effective for verifying the control algorithms of real time systems. With this developed dSPACE environment necessary improvements and changes can be done before testing the actual vehicle in underwater and the probability of test-bed vehicle collisions with obstacles can be greatly reduced. It also reduces the time required for validating the developed control algorithm for real-time implementation.

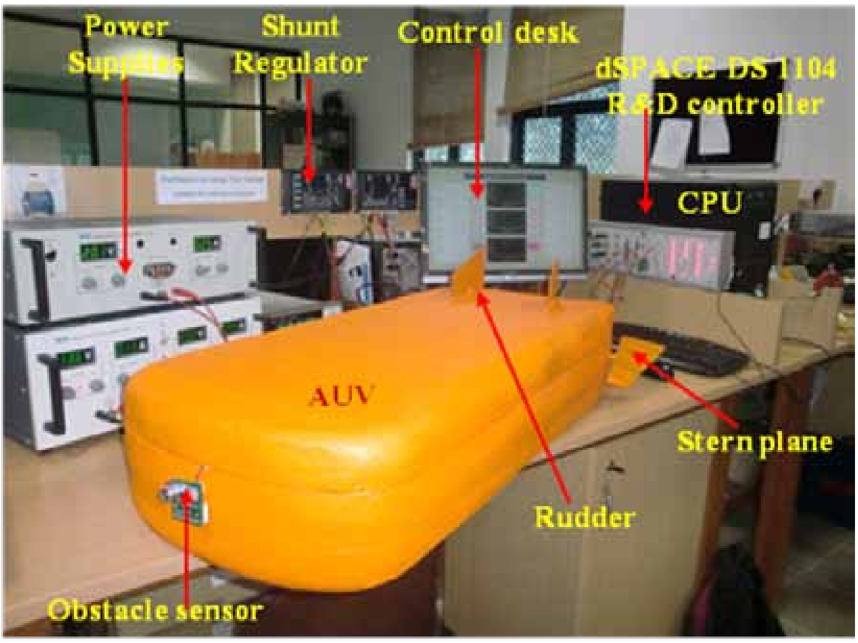


Fig. 2. AUV on its development bench

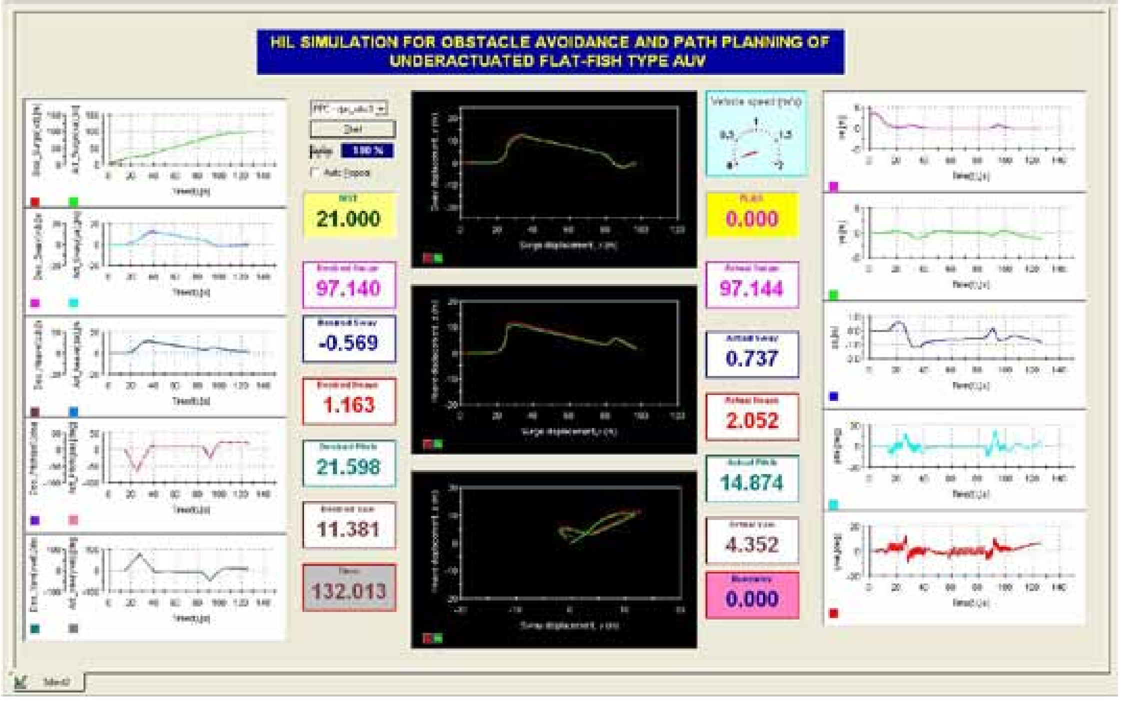


Fig. 3. GUI using ControlDesk layout

Acknowledgement: The authors acknowledge the contributions from Mr. Thomas George, summer internship student from IIT Hyderabad, in developing the HIL setup.

References

1. T. I. Fossen, *Marine Control Systems: Guidance, Navigation and Control of Ships, Rigs and Underwater Vehicles*, Marine Cybernetics AS, Trondheim, Norway, 2002.
2. "HIL Testing", http://www.marinecybernetics.com/hil-testing.
3. A, Martin and M. R Emami, "An architecture for robotic hardware-in-the-loop simulation", in *Proc. IEEE International Conference on Mechatronics and Automation*, 2006, pp.2162-2167.
4. http://www.dSPACE.com
5. http://www.dynafusiontech.com

Introducing MaPaMan: the Madras Parallel Manipulator

R Arun Srivatsan, Sandipan Bandyopadhyay

Robotics Laboratory, Department of Engineering Design
Indian Institute of Technology Madras

Parallel robots, or "parallel manipulators" as they are often called, form an interesting group of robots that are rapidly gaining popularity in diverse fields of applications. These robots are routinely used for pick-and-place and assembly operations in the industries; as motion platforms in various vehicle and aircraft simulators; in robot-assisted surgery as well as in medical rehabilitation. The classification "parallel" derives from the fact that these devices have multiple "limbs" connecting a fixed base to a moving platform, which serves as the *end-effector*. These limbs have one or more actuators in each – thus the end-effectors are actuated "in parallel", justifying the nomenclature. More details on these manipulators, their design, performance and applications can be found in the book by Jean-Pierre Merlet [1], and also at the website dedicated to parallel robots: http://www.parallemic.org.

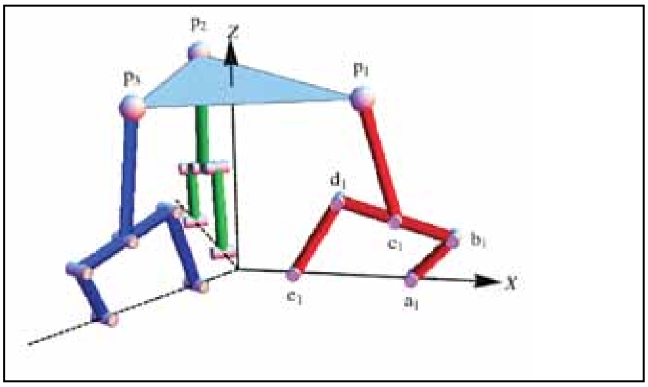
Arguably, the earliest parallel manipulator is the Stewart-

Gough platform. Introduced in 1954, it is still the most popular in its class. It can carry large payloads, while executing a wide range of manipulations with excellent accuracy and precision -- thanks to its six-legged architecture. However, the six actuators together with their drives make the manipulator very expensive, and its kinematics very difficult. Owing to such reasons, in the past two to three decades a lot of attention has been paid to manipulators with *lower,* i.e., three degrees of freedom (DoF). They are less expensive, and they serve specific purposes compatible with their DoF. For instance, the 3-RPS has a roll-pitch-heave motion, and as such it finds applications in ankle rehabilitation. Likewise, the Agile Eye, being a spherical robot, is used in laparoscopic surgery, as well as mounts for cameras. There are other robots in this category of “spatial parallel manipulators with lower mobility”, such as the DELTA, CaPaMan, 3-UPU, and so on.

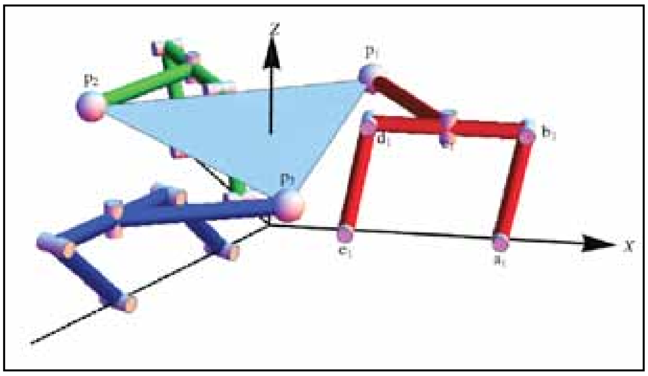
Recently, a manipulator belonging to this category has been developed in the Robotics lab of the Department of Engineering Design, IIT Madras (see Figure 1).

Figure 1: Fully functional prototype of MaPaMan-I

The manipulator has been named as MaPaMan, relating it to its place of origin. This new manipulator was conceptualised to address two different issues. Firstly, a disadvantage of a lower mobility manipulator is that often its design is too focussed on a particular application, and as such the scope of other applications is limited. MaPaMan alleviates this problem by having a *reconfigurable* architecture, by virtue of which the same manipulator can have two different combinations of its three DoF: roll-pitch-heave as MaPaMan-I, and roll-pitch-yaw as MaPaMan-II. Thus, it has a broader scope of application. As seen in Figure 2, this is achieved by simply changing the orientation of the axis of rotation of the joint situated at c_1 etc.



(a) MaPaMan-I



(b) MaPaMan-II

Figure 2: The two configurations of MaPaMan

The second major aspect of the new design is the elimination of prismatic joints or actuators. Existing manipulators, e.g., 3-RPS, CaPaMan have prismatic actuators/joints. Prismatic actuators usually produce a lot of force for their size, but in general, they are heavier and costlier. The present design can produce motions similar to the 3-RPS manipulator, but it has only rotary and spherical joints. Figure 1 shows how the leg architecture of the 3-RPS was modified to create the legs of the MaPaMan.

The advantages of the MaPaMan come at a cost, too. The position kinematics of the manipulator is fairly complicated. Fortunately, this problem has been solved by the extensive use of symbolic computations. The forward and inverse kinematics of the manipulator has been reduced to the solution of univariate polynomials. The coefficients of this polynomial are functions of the design parameters, which have been obtained in closed form. Thus, the kinematic analysis covers all possible parametric variations of MaPaMan-I and II. Details of the position kinematics of the manipulator are available in [2].

Several prototypes of the MaPaMan have been built so far. The authors are thankful to the Centre of Artificial Intelligence and Robotics (CAIR), Bangalore, for letting them use the RP facilities at CAIR to fabricate the first

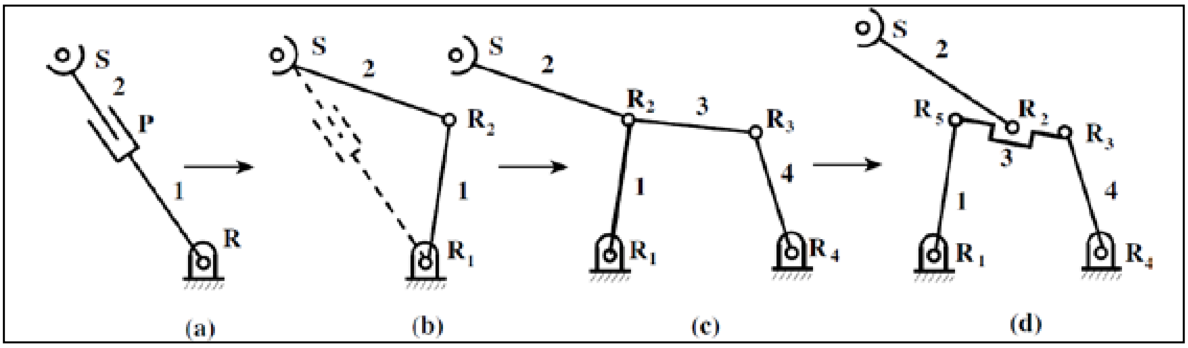


Figure 3: Stages of evolution of the leg architecture of MaPaMan

ABS prototypes. These prototypes were demonstrated in the "2011 ASME Student Mechanism and Robot Design Competition", where they won the overall third prize for the first author. A fully functional prototype with MS links was developed later in IITM, as part of the M. Tech. projects of the first author and his batch-mate Tarun S. Mehta, under the guidance of the second author. This prototype has a unique feature -- it has additional sensors at its passive joints, which eliminate the need for computing the pose of the top platform through expensive forward kinematic computations. Instead, the pose is computed directly from the sensed angles at the passive joints, as well as the motors encoders. These aspects of the manipulator have been described in detail in a provisional patent [3].

Further development of MaPaMan is in progress at the time of writing this article. At IIT Madras, MaPaMan is being developed as a motion platform for rehabilitation applications. Plans are also afoot to develop vehicle simulators for the Indian Army based on the MaPaMan platform. Interested readers can find more details on the manipulator, as well see the manipulator in action in the research website of the second author: http://www.ed.iitm.ac.in/~sandipan/research/mapaman.html.

References

1. Jean-Pierre Merlet, *Parallel Robots,* 2nd. Ed., Springer, 2006.

2. R. Arun Srivatsan and S. Bandyopadhyay, "On the position kinematic analysis of MaPaMan: a reconfigurable three-degrees-of-freedom spatial parallel manipulator," 2012, accepted for publication in *Mechanism and Machine Theory*.

3. Sandipan Bandyopadhyay, R Arun Srivatsan and Tarun S Mehta, "A reconfigurable parallel manipulator", Indian provisional patent 3277/CHE/2012, August 2012.

Robotics at RISE, IITM

Modern day robots possess the physical capabilities and versatility to perform several complex tasks. One of the main focus of research in our group is on building situated learning agents that can incrementally solve larger and larger problems using structures built from prior experience. We look at this both from a spatial and temporal perspective, with motivations drawn from cognitive theories of representation. However, due to the complex nature of the tasks and the stochastic nature of the environments the robots work in, pre-programming a robot to learn several tasks from scratch and deal with uncertainty at the same time becomes highly infeasible. Thus a robot must possess the capabilities to learn to solve new tasks by itself and from human teachers, throughout its lifetime. In the RISE lab we are working on cognitive models for visual representations of objects as well as models of lifelong learning from different modalities of instructions–from humans, other robots, as well as feedback from the environment.

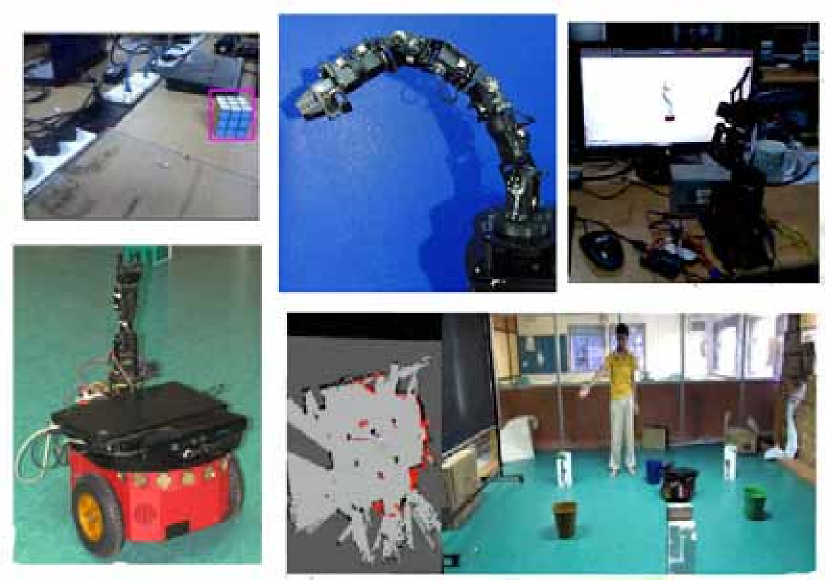


Figure 1: Counter clockwise from top left : Perceiving a cube, Cyton 7 DOF arm, Modeling simulation and control of a 5 DOF arm using ROS-OpenRAVE, Instructing robots to solve tasks, Mobile robot with Kinect and arm

Robots and related technology

Mobile Robot P3DX Mobile robot platform, used as the base in experiments on the instruction taking robots.

Microsoft Kinect Popular and cheap RGB-D sensors Forms the basis of various experiments involving object detection and gesture recognition. Associated ROS nodes are OpenNI and PCL.

Cyton Veta light arm 7 DOF robotic arm made from Dynamixel servos - high performance, intelligent actuators give feedback on position, speed, load, voltage, and temperature. Control using Dynamixel ROS node. Planning and simulation using Actin SE. Currently OpenRAVE models are being designed for the same.

iRobot Create Platform Simple platforms based on iRobot's Roomba design. Easy and extensive interfacing in ROS using the turtle-bot node. A laser-tag like game using two creates was developed using the ROS framework.

ROS : Simulation and Control Robot Operating System is a software framework for robot software development, providing operating system-like functionality on a heterogeneous computer cluster. All projects use the ROS framework.

Projects

Reasoning with incomplete visual evidence for robots

Visual perception is a key function for an embodied agent to interact with its environment to perform complex object manipulation tasks. The visual routines theory suggests a framework to solve high level vision tasks in a cognitive way. But a major challenge is the incompleteness of the data obtained from the visual sensors, causing uncertainty in perception and inference of the objects. We propose a novel approach for object categorization from incomplete visual evidence, using probabilistic graphical models and active vision in a hierarchical framework. The object models are built from shape properties and spatial relations and are inferred using a set of Markov logic networks (MLN) organized as layers. Given that the information is incomplete, active vision is a mechanism for focused gathering of additional information. Inspired by the ideas of active vision, in the event of missing evidence, this framework restricts the selective visual processing to specific regions of the input image and further inference is carried out incorporating the new evidence.

Transfer between Heterogenous Robots

Transfer learning refers to reusing the knowledge gained while solving a task, to solve a related task more efficiently. Much of the prior work on transfer learning, assumes that identical robots were involved in both the tasks. In this work we focus on transfer learning across heterogeneous robots while solving the same task. The action capabilities of the robots are different and are unknown to each other. The actions of one robot cannot be mimicked by another even if known. Such situations arise in multi-robot systems. Theobjective then is to speed-up the learning of one robot, i.e., reduce its initial exploration, using very minimal knowledge from a different robot. We proposed a framework in which the knowledge transfer is effected through a pseudo reward function generated from the trajectories followed by a different robot while solving the same task. The framework can effectively be used even with a single trajectory. We extend the framework to enable the robot to learn an equivalence between certain sequences of its actions and certain sequences of actions of the other robot. These are then used to learn faster on subsequent tasks. [2]

Instruction Taking Robots

Consider a human learning to throw a ball into a basket. The evaluative feedback will depend on how far the ball misses the target by. Whereas, instructive feedback will be a coach instructing him to *throw harder or slower.* Instructions could be of various forms. For example, consider the agent searching for a key. The agent could be instructed to “look in the key stand”. The effect of this instruction is to reduce the agent’s search space considerably. Take the case of an agent navigating an obstacle course. When it is obstructed by a puddle of water, the agent is instructed to “jump”. This instruction can then be reused by generalizing it over puddles of various locations, liquids, colors, shapes, etc. Thus efficiently using the information in the instruction. We incorporate learning from such instructions into traditional RL methods. We give a mathematical formulation for instructions and outline two kinds of instructions, π-instructions and Φ-instructions. We also provide algorithms that utilize both instructive and evaluative feedback. These are then empirically shown to perform better than traditional RL methods[1].

Deciding between various kinds of instructions We assume a human instructs the agent in a manner that can be a potential $\pi - instruction$ or a $\Phi - instruction$. Using pointing gestures is one such instructing mechanism, where pointing to an Object can either be interpreted as *“Goto(Object)”* or $F' = \{Objectfeatures\}$. Every instruction, $I(s)$, is interpreted as an action ($\pi - Ins$) and as a binding on the state space ($\Phi - Ins$). The learner accumulates instructions, if available, to be used to train an instruction model. These instruction model $\hat{\Pi}_\pi$ and $\hat{\Pi}_\Phi$ are built using supervised learning. We use Markov Logic Networks (MLN) to perform the generalization as they can succinctly represent these dependencies resulting in sample-efficient learning and inferring. The available action models are $\hat{\Pi}_{\pi Ins}$, $\hat{\Pi}_{\Phi Ins}$ and $\hat{\Pi}_Q$ ($\pi - Ins, \Phi - Ins$ and regular $Q - Learner$). The action suggested by the most confident model is used. The confidence of a model is measured using a specified metric. The selected action is performed and updates are made to the Q-learner. [3].

Associating the instruction with a robot specific meaning We consider the specific case of $\pi - instruction$, each instruction tells the robot to perform a specific action. Gestures are a method of communication that can be used to convey several types of instructions. Most research on gesture learning only concentrate on recognizing them, and do not cognize on the semantics or the intention of the instructor when making the gestures. The mapping of gestures to the robot’s action space is usually manually programmed. Different instructors might use different gestures to convey their intention. The robot’s action space could also potentially keep varying as it picks up new skills or macro actions during its lifetime. Thus reprogramming the interpretations manually becomes infeasible, especially in the scenario where the robot works with non-expert users. We propose a framework where a robot learns the correct interpretation of gestures by generalizing its past experiences. Such a model can be refined and used across all tasks the robot might perform during its lifetime. This provides the robot with a powerful medium of knowledge transfer, enabling it to learn efficiently from human instructors. Such a module can fit inbetween the human and the learning framework such as the one used in earlier work [3].

Videos and more details can be found online at *http://rise.cse.iitm.ac.in/wiki/*

References

[1] Pradyot Korupolu V N, Manimaran Sivamurugan S, and Balaraman Ravindran. Instructing a reinforcement learner. In *Proceedings of the 25th FLAIRS Conference.*

[2] B. Lakshmanan and B. Ravindran. Transfer learning across heterogeneous robots with action sequence mapping. In *Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2010)*, pages 3251–3256. IEEE Press, 2010.

[3] K. V. N. Pradyot, S. S. Manimaran, B. Ravindran, and S. Natarajan. Integrating human instructions and reinforcement learners : An srl approach. In *Proceedings of the UAI workshop on Statistical Relational AI (StarAI 2012).*

FIRST CALL FOR PAPER

ADVANCES IN ROBOTICS

(AR - 2013)

INTERNATIONAL CONFERENCE OF ROBOTICS SOCIETY OF INDIA

PUNE, INDIA

July 4th to July 6th, 2013 at R&DE (Engrs), DRDO

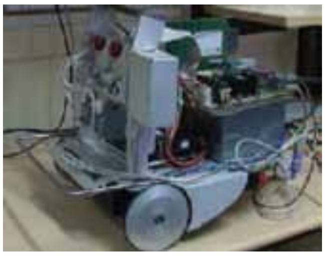
Patron: Dr. V. K. Saraswat, SA to RM ,
Secretary, Department of Defence Research and Development and DG DRDO

Advisor: Dr. S. Guruprasad, Director, R&DE (Engrs)

Robotics Society of India is initiating a new conference series to be held on a regular basis for creating a forum to present and exchange new ideas by researchers and developers from India and abroad working in the area of robotics and allied fields. It will have plenary talks, oral and poster presentations and special industry oriented sessions.

Scope of the conference will include (representative and non-exhaustive):

1. Kinematics, dynamics, control, and simulation of robots and autonomous intelligent systems;
2. Design of robotic mechanisms;
3. Man-machine interface and integration;
4. Robotics-related computer hardware, software, and architectures;
5. Vision and other non-contact sensory systems;
6. Tactile and other contact sensory technology;
7. Active sensory processing and control;
8. Machine learning and artificial intelligence for robotics;
9. Medical and Assistive Robotics
10. Bio-mimetic and Bio-inspired Robotics
11. Swarm Robotics



GENERAL CO-CHAIRS

ALOK MUKHERJEE, R&DE(ENGRS), DRDO

ANUJ KAPURIA, Hi-tech Robotic Systemz Ltd

PROGRAMME CO-CHAIRS

TOM SHIBATA, NARA INSTITUTE OF SCIENCE AND TECHNOLOGY, JAPAN

SANTANU CHAUDHURY, IIT DELHI, INDIA

PUBLICITY CHAIR

ASHISH DUTTA, IIT KANPUR

PUBLICATIONS CHAIR

SUBIR KUMAR SAHA, IIT DELHI

Venue: Pune

Queen of the Deccan, Oxford of the East, and cultural capital of Maharashtra, Pune as a city is as interesting as its history. Long-standing bastion of the Maratha empire and home to the legendary Raja Shivaji, the Pune district is marked by magnificent land and sea forts, testimony to its glorious past.

Last Date for Submission of Papers: 28.02.13

FOR FURTHER INFORMATION CONTACT

Prof. Ashish Dutta,
Secretary, Robotics Society of India
Dept. of Mechanical Engg.,
IIT Kanpur.
adutta@iitk.ac.in

Prof. Santanu Chaudhury,
Vice President (Academic), Robotics Society of India
Dept. of Electrical Engg.,
IIT Delhi. santanuc@ee.iitd.ac.in

Additional Information and instructions about paper submission will be available at the web page
http://www.rsindia.org/