Human-Robot Teaming with Sliding Autonomy

Jacob Longazo, Michael Heid; **Computer Science**
Mentor: Dr. Daisy Tang
Kellogg Honors College Capstone Project

PROBLEM

Multi-robot systems have numerous applications in the modern world but often suffer problems in control and coordination. This project seeks to solve the problems that plague the traditional control methods of full autonomy and teleoperation by combining them into a new control method known as sliding autonomy. When paired with a task allocation algorithm to improve coordination, we should see a drastic improvement in the team's performance.

APPROACH

SLIDING AUTONOMY

Sliding autonomy is a control method that allows a robot to switch between autonomy, teleoperation and various mixtures of the two [3]. To initiate a switch in control, a robot can either request teleoperation or an operator can manually switch the robot to teleoperation. In this project, we test two different switch methods:

System Initiative - Only the robot can initiate a switch when it detects that is in an erroneous state.

Mixed Initiative - Either the robot (error detection) or the human can initiate a switch (judgment).

TASK ALLOCATION

This project uses a special algorithm to help coordinate robots in order to help them accomplish the overall mission goal optimally. The task allocation algorithm uses an auction-based approach [2], in which the task allocator broadcasts tasks that robots "bid" on using their location to the task and strength. The task allocator then picks which robot can best perform the task and assigns it to that robot. Task allocation is a secondary aspect to this project that does not act as a variable. It does, however, provide a more effective means to autonomously coordinate the team.

FIGURE 1.1

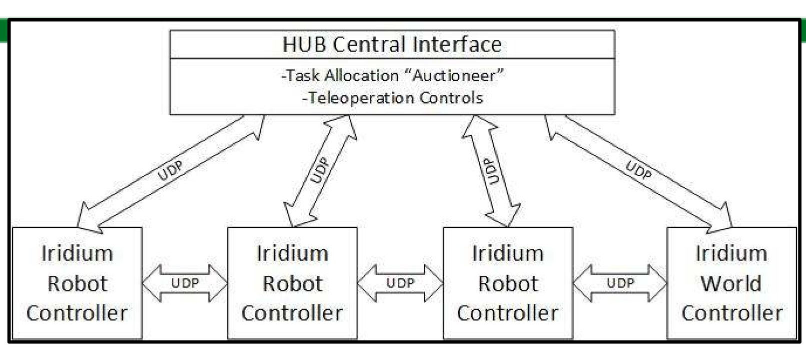


FIGURE 1.2

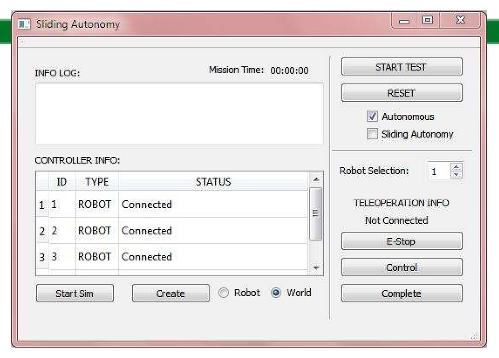


FIGURE 1.1:
Architecture diagram showing software hierarchy

FIGURE 1.2:
The "HUB", or central program developed as the core interface

EXPERIMENTS

THE MISSION

Site-Clearing - A team of robots must work together to push scattered boxes up against the room's walls. Each box is considered a task and has a weight, meaning only robots with a certain strength can push it.

THE ENVIRONMENT

USARSim (Unified System for Automation and Robot Simulation) [1] – This project uses a simulator to create the environment and the robots.

THE TEAM

The mission team for this project consists of 3 robots and 1 human operator. The three robots are identical, consisting of 2 sensors. The first sensor is the INS sensor, used to help the robot determine its position and direction using wheel encoders, but is often subject to drifting, causing accumulated errors over time. The other sensor is a range scanner, used to help a robot determine what's in front of it, whether it be an obstacle to avoid, or a box to push. The human operator is meant to act as an observer and controller.

FIGURE 2.1

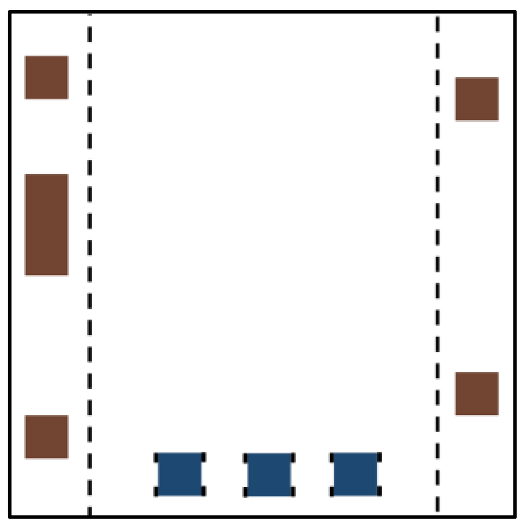
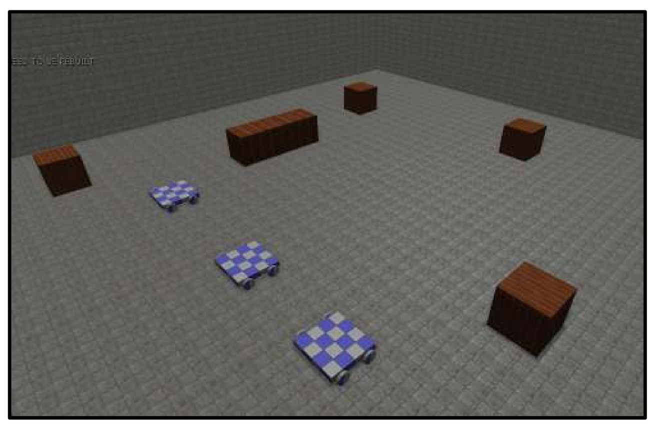


FIGURE 2.2



ENVIRONMENT:
The test environment consists of 3 robots and 5 boxes

FIGURE 2.1:
Simulation Diagram (birdseye). Dashed zones indicate area boxes should be in after the mission.

FIGURE 2.2:
Simulation Screenshot

RESULTS & ANALYSIS

TABLE 1 – EXPECTED RESULTS

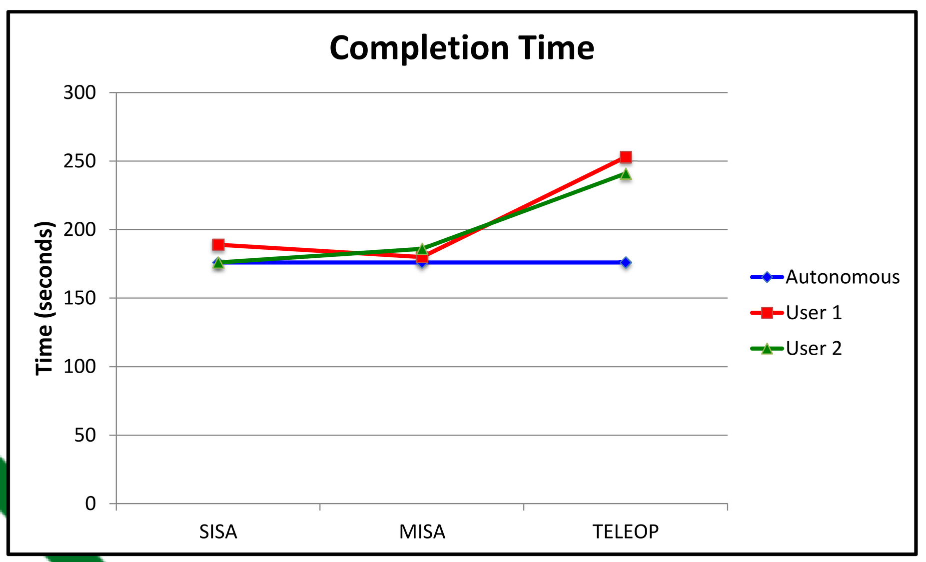
Test Type	Average Completion Time	Average Solution Quality	Average Workload
Teleoperation	worst	best	worst
Autonomous	best	worst	best
System Initiative	good	good	better
Mixed Initiative	better	better	good

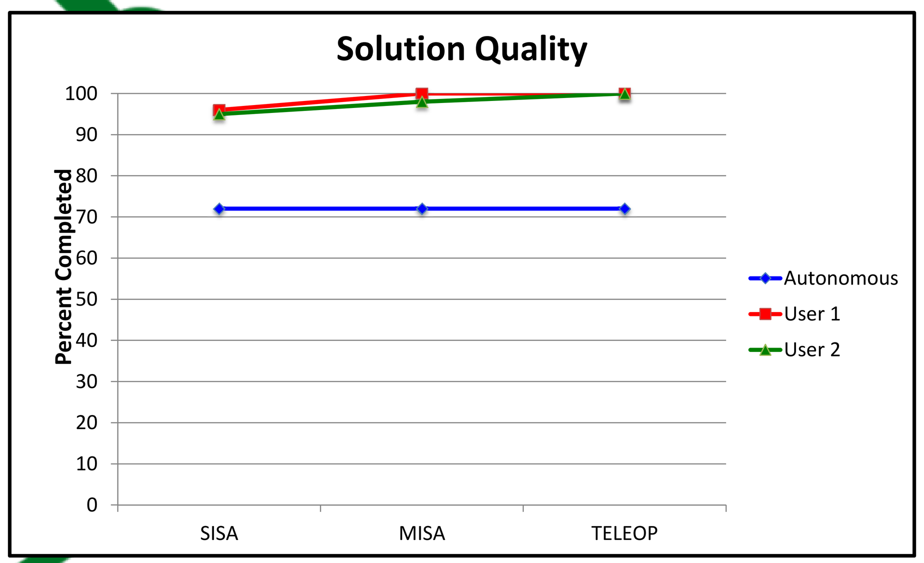
TABLE 2 – USER 1 RESULTS **Experienced User**

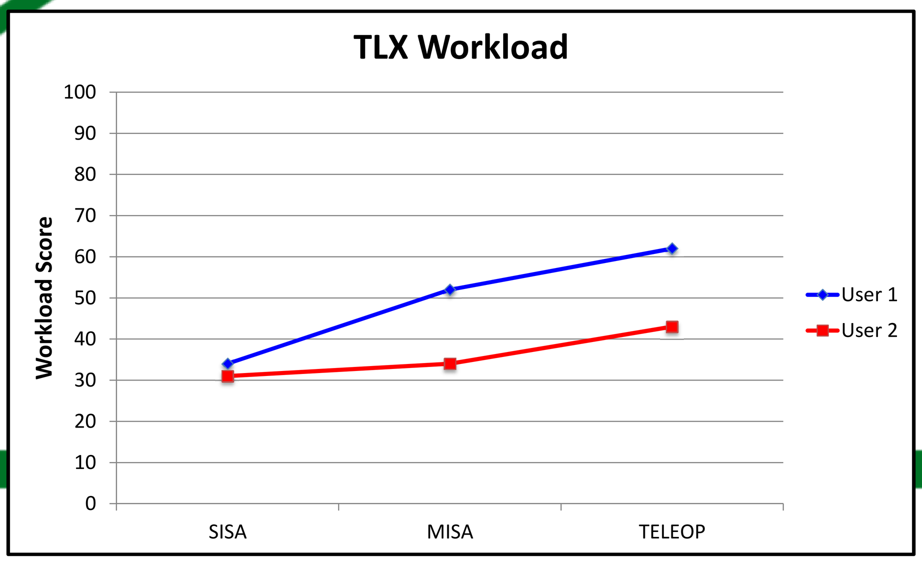
Test Type	Average Completion Time	Average Solution Quality	Average Workload
Teleoperation	253 seconds	100%	62
Autonomous	176 seconds	72%	0
System Initiative	189 seconds	96%	34
Mixed Initiative	180 seconds	100%	52

TABLE 3 – USER 2 RESULTS **Inexperienced User**

Test Type	Average Completion Time	Average Solution Quality	Average Workload
Teleoperation	241 seconds	100%	43
Autonomous	176 seconds	72%	0
System Initiative	176 seconds	95%	31
Mixed Initiative	186 seconds	98%	34







MEASURING PERFORMANCE

COMPLETION TIME

How quickly can the team complete all objectives? Time is measured in seconds from the point when the mission is begun until either the user or the system determines all objectives are complete.

SOLUTION QUALITY

How well is the job done? 1 point is awarded for a box being pushed close enough to the wall (at least one box length), .5 points awarded for incomplete pushes, and 2 points for completion time of under 5 minutes, leaving a total of 7 points.

WORKLOAD

How much work is required from the human operator? Workload determined by having the user fill out the NASA TLX (Task Load Index) Survey, with scores ranging between 1 and 100. The data in this area is very subjective, but still shows expected trends.

CONCLUSIONS & FUTURE WORK

Results were very consistent with what was expected, showing sliding autonomy as a viable and effective method to controlling multi-robot teams. While artificial intelligence may someday grow to a point that no longer needs a human supervisor, that day is far off. Today, sliding autonomy can be used in a wide variety of situations that ordinary AI can't solve on its own. Of course this system could use several improvements, namely improvements to the task allocation algorithm as well as the addition of more partially autonomous control options for the user. These issues as well as others are open for exploration in the near future.

REFERENCES

[1] M. Lewis, J. Wang, and S. Hughes. Usarsim : Simulation for the study of human-robot interaction. *Journal of Cognitive Engineering and Decision Making*, 2007:98–120, 2007

[2] L. E. Parker. Distributed intelligence: Overview of the field and its application in multi-robot systems. *Journal of Physical Agents, Special Issue on Multi-Robot Systems*, 2(2):5–14, 2008.

[3] B. Sellner, F. Heger, et al. Coordinated multi-agent teams and sliding autonomy for large-scale assembly. *Proceedings of the IEEE*, 94(7), 2006.

[4] R. Smith. The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver, *IEEE Transactions on Computers*, C-29, 1980.

AKNOWLEDGEMENTS

This project would not have been possible without the guidance and support given from my mentor, Dr. Daisy Tang as well as my research assistant Michael Heid, who assisted in communications and teleoperation controls.

SOFTWARE USED

[1] USARSim (Unified System for Automation and Robot Simulation) – SourceForge.net

[2] UDK (Unreal Development Kit) – Unreal Engine

[3] Qt – Qt Project