Teleoperation of LARS Robot

600.446: Computer-Integrated Surgery II

Project Proposal

Students: Seth Billings
Ehsan Basafa

Feb. 12, 2009

Mentors: Dr Russell Taylor
Marcin Balicki

Possible Collaborators: Min (Networked Robotics)
Gorkem Sevinc & Ali Uneri (Teleoperation of Eye Robot)

Stated Topic and Goal

The main goal of this project is to develop a teleoperation system for the LARS robot. The robot and a Phantom Omni haptic device will be integrated into a workstation and with the implementation of a master-slave control algorithm the Phantom will be used to control the movements of the robot. Adding virtual fixtures to the movements of the robot is also considered as a potential outcome of this project. When equipped with a dexterous tool, the robot may be used in various minimally invasive surgical procedures, e.g. laparoscopy, to provide the surgeons with high precision and dexterity. Implementing this feature is also a potential outcome.

Motivation & Significance

The motivation for this project is to provide a teleoperated robot development platform suitable for use in various medical robotic research applications. The Laparoscopic-Assisted Robot System (LARS) is an ideal platform for this due to its mobility, dexterity, and versatility of use with various end-effectors (Figure 1). LARS control software will be developed as integrated with the Surgical Assistant Workstation (SAW) architecture developed at JHU. Using SAW as the software framework provides the added benefit of presenting a trial system to demonstrate the SAW's capabilities, which may enable future funding for continued SAW development or for research efforts dependent on SAW.

Following completion of our project, the LARS robot is planned for use as a research platform to investigate minimally invasive osteolytic bone surgery. In this application, a dexterous robotic tool will be mounted on the LARS and controlled in conjunction with the LARS. While the LARS positions the tool shaft, the dexterous end-effector will extend the robot work-space to include regions outside the tool shaft's reach.

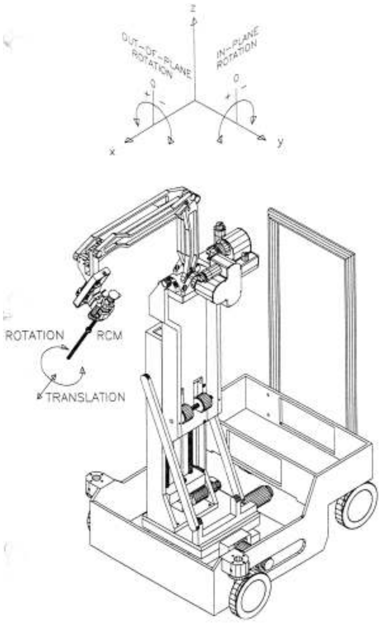

Figure 1. LARS Robot and Kinematics. From LARS Development Folder.

Background

The LARS robot was developed in the early 90's as a joint study between IBM Research and the JHU Medical School. More recently, in the summer of 2008, the LARS underwent some level of rehabilitation where it was re-wired and outfitted with new electronic hardware, including new encoders and a Galil motor controller / amplifier. The current status of the LARS requires some work to integrate the new motor controller and to test both new and old hardware for proper functionality. Current control software for the LARS is outdated and requires complete overhaul to be compatible with the SAW framework.

Technical Summary of Approach

Before implementing teleoperation, we must do some foundational work on the robot since the controller board is changed and the software previously used to control the robot is no longer useful. This stage can be summarized as below;

- Complete wiring of the controller and test for circuit/hardware problems
- Calibrate the encoders and set the limit angles on the joints
    - Some experimental work will be done to determine the gearbox ratios (for the unknown joints) and a homing procedure will be devised.
- Tune the motor controller constants
    - The controller uses a conventional PID scheme. There are several methods for tuning the three controller parameters, among which the trial-and-error and the Ziegler-Nichols are the most popular. The Galil software interface will be used for this purpose.

The next step will be to have a library of simple position/velocity control commands to control the movements of the individual robot joints. These routines will be integrated with the SAW architecture. Marcin's code for Eye Robot 2 will provide a starting point for this effort.

Implementing the forward and inverse kinematics calculations will form the next step. The forward kinematics is straightforward and can be used to have a graphical representation of the robot for visualization and debugging purposes. It also serves as a starting point for inverse kinematics calculations at each step of calculations. At this point, we will use a simple inverse kinematics algorithm. One way of doing this is to decouple the base motions (x-y-z translation) from the rotations of the arm (redundancy resolution scheme).

Finally the Phantom Omni haptic device will be added to the system and a master-slave control scheme will be implemented to move the LARS robot according to the movements of the Phantom Omni device. To teleoperate a robot with another robot, there are two main schemes. One is to use a joint-to-joint controller between the corresponding joints of the two robots and is used when the robots are identical, or at least have similar mechanisms or configurations. The second approach is the end-effector-to-end-effector control, in which the motion of the end-effector of the master robot is used to control the slave robot end-effector. This approach is especially useful when the two robots have different degrees of freedom (DOF) and/or configurations (as in our system). To implement the second approach, the forward kinematics map of the master (Omni) robot and the inverse kinematics map of the slave (LARS) robot are needed.

One possible extension to the system is to implement an optimization algorithm for computing the inverse kinematics of the robot. Since the robot has 7 DOFs the inverse kinematics problem has several solutions in general. Specifically, for a 6 DOF robot, the inverse kinematics may have up to 8 solutions. One way of solving this is to use the current forward kinematics map and find a solution to the inverse kinematics problem that minimizes the sum of the joint motions that take the robot to the desired position. This means that an optimization problem needs to be solved for the inverse kinematics. The algorithm is currently developed at JHU and we expect to implement that on the robot for our purpose.

Another interesting feature that can be added to the system is virtual fixture control. Generally, virtual fixtures come in two modes: forbidden region and guidance virtual fixtures. As the names suggest, forbidden region virtual fixtures prevent the robot from entering some regions of the workspace and guidance virtual fixtures help guide the robot in a pre-defined path. Both types of virtual fixtures can be implemented on the robot. Since the Omni device is capable of providing force feedback to the user, virtual fixtures imposed on the LARS robot can also generate forces when the user tries to penetrate the forbidden regions or tends to deviate from the guiding path.

Deliverables

- Minimum
    - Teleoperate the LARS robot with a Phantom Omni haptic device
        - Integrate the control scheme with the SAW architecture
        - Devise a simple inverse kinematics for motion control
- Expected
    - Implement advanced position control capability
        - Use the JHU optimization algorithm for motion control
- Maximum
    - Add various virtual fixtures
    - Alternative Maximum: attach and control dexterous (da Vinci) tool

Management Plan

- Collaboration with Mentor
    - Marcin
        - weekly meetings and help as-needed

  - Dr. Taylor
    - optional attendance to weekly meetings
    - inform of any challenges / problems that arise
- Collaboration with Eye Robot Teleoperation Group
  - meet on as-needed basis
  - share knowledge of SAW architecture and position control optimization algorithm
  - share ideas for software framework
- Responsibilities
  - We intend to work together on all stages / aspects of this project and as such share equal responsibility for every task.
- Development Timeline
  - Refer to Project Timeline at end of this document

Dependencies & Plan for Resolving

- Functionality of LARS Robot
  - Most electromechanical components can be replaced off-the-shelf if needed
  - If part not avail, redesign/replace with equiv part
  - Possibly borrow parts from other LARS
- Computer for Software Development / Robot Control
  - Use personal laptop (lab desktop also available)
- Availability of Master Controller
  - Dr. Taylor and Dr. Kazanzides to purchase Phantom Omni (~$2400 each)
  - May require PCMCIA ENET card for networking if (~$30)
- Small Budget for Unforeseen Expenses ($200 - $300)
  - new motor controller box, misc hardware needs, etc.
- Understanding of SAW architecture
  - reference online documentation
  - reference / start from Marcin's code for Eye Robot
  - support from knowledgeable personnel in the lab (Marcin, Anton, etc.)
  - collaboration with Eye Robot Teleoperation group
- Understanding of JHU motion control optimization algorithm
  - reference code from existing applications (NeuroMate, etc.)
  - reference JHU publications
  - support from knowledgeable personnel in the lab (Tian, Paul)
  - collaboration with Eye Robot Teleoperation group
- People
  - Marcin and Dr. Taylor for continuing help and guidance
  - less frequent support from others in the lab (Anton, Tian, Paul)

Reading List

1. J. Funda, R. Taylor, B. Eldridge, S. Gomory, and K. Gruben, "Constrained Cartesian motion control for teleoperated surgical robots," IEEE Transactions on Robotics and Automation, vol. 12, pp. 453-466, 1996.

2. Galil Motion Control, Inc. *DCM-40x0 User Manual*, Rev. 1.0c. Dec, 2008. www.galilmc.com

3. Galil Motion Control, Inc. *DCM-40x0 Command Reference*, Rev. 1.0d. Dec, 2008. www.galilmc.com

4. G. Hamlin and A. Sanderson. *A Novel Concentric Multilink Spherical Joint with Parallel Robotics Applications*. IEEE, pp. 1267-1272. 1994.

5. A. Kapoor, M. Li, and R. Taylor. *Constrained Control for Surgical Assistant Robots*. IEEE Int'l Conf. on Robotics and Automation. pp. 231-236. May 2006.

6. A. Kapoor. *Motion Constrained Control of Robots for Dexterous Surgical Tasks*. Johns Hopkins University Ph.D. Thesis. Sept, 2007.

7. P. Marayong, et. al. *Spatial Motion Constraints: Theory and Demonstrations for Robot Guidance Using Virtual Fixtures*. IEEE Int'l Conf. on Robotics & Automation. pp. 1954-1959. Sept. 14-19, 2003.

8. R. Taylor, et. al. *A Telerobotic Assistant for Laparoscopic Surgery*. IEEE Engineering in Medicine and Biology. pp. 279-288. May/June 1995.

9. Additional articles concerning teleoperation will be read as well (currently looking for good references)

Project Timeline

Task	9-Feb	16-Feb	23-Feb	2-Mar	9-Mar	16-Mar	23-Mar	30-Mar	6-Apr	13-Apr	20-Apr	27-Apr
Background Reading	🟨											
Project Proposal Presentation	🟨											
Milestone 1: Functional Robot												
Integrate Motor Controller & Complete Wiring		🟧										
Test, Tune & Calibrate Robot		🟧	🟧									
Milestone 2: Develop Control Software												
Develop Low-Level Routines for Control Commands				🟩	🟩	🟩						
Spring Break						🟪						
Develop Simiple Inverse Kinematic Algorithm						🟩	🟩					
Implement Master Control							🟩	🟩				
Checkpoint Presentation								🟪				
Milestone 3: Optimized Position Control Algorithm / Advanced Teleopertion Modes												
Understand Optimization Algorithm									🟦	🟦		
Implement Optimization Algorithm in Control Software										🟦	🟦	🟦
Milestone 4: Virtual Fixtures												
Advanced Teleoperation Modes (Virtual Fixtures, etc)												🟦
Report												🟫