Engineering 90: Senior Design Project Proposal

Linear Positioner

Taylor Hamilton
Swarthmore College
Department of Engineering
Submitted 06 December 2006

Abstract

This paper outlines my proposal for my Engineering 90 senior design project, focusing mainly on the specific tasks that I plan to undertake and their organization into a Critical Path framework from which I can properly allocate my time. I intend to work on a linear positioner. The bulk of my time will be spent writing and debugging a sophisticated GUI to provide simple, yet powerful, interaction with the positioner, but I will also dedicate a significant portion of my efforts towards implementing a mechanical upgrade, testing the positioner's accuracy, and creating a video taken by a camera on the positioner for use as a metric for scoring the accuracy of different spatial location algorithms created by the computer vision community.

Background

The Linear Positioner

The linear positioner in Professor Maxwell's lab consists of three metal rods attached to a metal base that are free to slide along one another in three independent directions. One of the rods has a small platform attached to it. All three rods are connected to servos which are controlled by a nearby computer. The effect of this arrangement is that the platform can be positioned or moved through a cubic meter of space with extremely high precision in each of its three coordinates.

Also connected to the linear positioner are six Hall effect sensors, one at each end of each rod. These sensors vary their output voltage based on changes in the density of the magnetic field surrounding them. In conjunction with magnets on the linear positioner's rods, the sensors provide a robust, high SNR measure of the proximity of each rod to the edge of the positioner. Thus the sensors safeguard against rods running off of the edges of their rails as well as providing a reliable 'position 0' from which to anchor measurements.

In the positioner's current state, the Hall effect sensors are held in place only by duct tape. This arrangement is suboptimal for two reasons: first, the duct tape is subject to wear, and the sensors can potentially fall off of the positioner. If this happens, the bars of the positioner can fall off of the edges, damaging or destroying the positioner and endangering the safety of those around it. The second disadvantage of the duct tape is that the sensors' locations are not fixed in a well-defined place. This limits the accuracy and repeatability of the positioner's operation, both of which are crucial in the proper creation of a video to be used for testing spatial location algorithms.

GUIs

Graphical User Interfaces, or GUIs, are what allow a user to interact with more than a blank screen and a blinking cursor. Good GUIs make using even unfamiliar software quick, simple, and intuitive. They accomplish this by making the functionality of the software immediately visible and easily manipulable through simple components such as buttons and sliders, rather than typed commands.

GUIs are of particular interest to me because computers are not inherently user-friendly, but I have a firm belief that they not only can be, but should be whenever possible. The world is becoming increasingly computer-literate, but most people's knowledge of a computer stops when the blinking cursor starts: the combined inconveniences of typing long strings and memorizing lists of commands discourages many users from exploring any further into the capabilities of their computer and its programs. This is especially unfortunate when a little more effort on the part of the programmer could have spared them this inconvenience.

Another reason that GUIs interest me is that they are so often implemented poorly. Instead of considering the concerns of the end-user, the programmer creates whatever interface is easiest to program. I intend to use this project as a careful counterexample to that kind of sloppy programming, focusing constantly on keeping the design simple and intuitive, even for people who have never used the software before. This kind of approach is one of the keys to the success of companies like Google, and is also the reason that computers are becoming more accessible even to people without any technical background at all. This project should teach me good practices for continuing this trend towards simpler computing.

A related issue in GUI design is the problem of user-friendliness versus robustness. The more flexibility and functionality is required from a program, the more complicated it necessarily gets. These two competing tendencies necessitate a careful balance in the design of GUIs, requiring the programmer to take care not to make too many sacrifices in either domain in order to accommodate the other. I intend to be conscious of this balance over the course of my project, ensuring that each new feature added to the program is worthwhile, integrates well with existing components, and avoids cluttering the design. I should be able to accomplish the last goal by employing a hierarchical design that allows components to remain hidden until they are needed.

Computer Vision

The final aspect of my project concerns the computer vision community. As robots equipped with cameras move through the world, they need to be able to process the visual data that is being provided to them. One useful computation is to calculate the position of the robot as it moves through a scene based solely on a video taken by a camera attached to the robot – an analysis known as ego-motion. A related set of algorithms that also attempt to draw a map of the area based on video data are known as SLAM – Spatial Location and Mapping. Several algorithms exist for performing these computations, but there is as far as I know no standard by which different algorithms can compare their accuracy.

If the linear positioner is designed to operate with sufficient precision and accuracy, then it can remedy this problem. If a camera is mounted onto the positioner's platform and then moved around while video is being taken, then the result will be a video for which there exists precise, accurate position information. Spatial location algorithms can then be run against the video and compared to the correct locations, yielding the ability to compare the effectiveness of different algorithms.

<u>Tasks</u>

I will next describe in some detail each of the tasks that I plan to accomplish over the course of this project. This discussion will be followed by a Critical Path analysis of the tasks presented.

1. **Obtain Machine-shop certification**. In order to make the fixtures I need for holding the Hall effect sensors in place, I will need machine shop certification. The fact that I failed to take the certification class before the Spring is not an issue because this task does not lie on the critical path, as will be seen later.
2. **Sketch potential mechanisms for semi-permanently fixing the Hall effect sensors in place. Discuss these mechanisms with my advisor and the machine shop coordinator, and decide on a final design**. I use the word 'semi-permanent' because I feel that the final design should allow sensors to be swapped in and out easily should the need arise. This should of course be accomplished without sacrificing the consistency and repeatability that these fixtures can provide.
3. **Build the fixtures, and attach them to the linear positioner**. This task should be relatively straightforward once the design is complete, but it would be unwise not to allow for unforeseen complications here.
4. **Obtain familiarity with the current program and GUI for controlling the linear positioner.** I am told that the current GUI for the positioner is neither robust in its capabilities not intuitive in its use. Spending some time learning it should provide me with both knowledge of the positioner interface (on the back end) and an example of how not to make a GUI.
5. **Research GUI programming paradigms and recommended languages for implementation. Choose which language I will use.** As I mentioned above, it is important to me that the GUI be as smooth and user-friendly as possible. In order to accomplish this goal, I believe that at least a rudimentary knowledge of GUI programming paradigms will be invaluable in efficiently implementing my program. I also want to make an informed decision in my choice of programming language.
6. **Learn how to program GUIs in my chosen language.** This task will probably be accomplished through either an online tutorial or a hardcopy guide. It will also probably bleed into task 8, as I seek to refine my design and need the tools to do so.

7. **Discuss / create design goals for the GUI.** I plan to perform this task both alone and with my advisor, focusing on ways to simultaneously achieve the two conflicting goals of user-friendliness and robustness.
8. **Implement the GUI**. This is the portion of the project to which I have allotted the greatest percentage of my time. In accordance with the design goals mentioned above, I seek to provide an interface which allows the user to access the positioner at different levels of detail, with increasing amounts of power and flexibility in controlling its operation.
9. **Test and Debug the GUI.** Of course, such a large undertaking will unavoidably have bugs that need to be found and corrected. This task will operate largely in conjunction with Task 8, but is being described separately to provide explicitly for testing and debugging time and to emphasize its importance in the programming process.
10. **Determine the precision and accuracy of the linear positioner.** This is a necessary step towards creating a useful dataset for evaluating spatial location algorithms. It is not, however, clear to me how exactly this will be accomplished. The motors that control the positioner are supposed to be accurate to within a fraction of a millimeter, and I'm not aware of any instruments capable of measuring distances on the order of meters with such high precision. If necessary, the positioner will be evaluated in different modes of operation to determine the effect that these have on its precision and accuracy.
11. **Research / discuss the type of data required to provide a useful dataset for spatial orientation algorithms.** This task just serves as a reminder that a video of a blank white wall is probably considerably less useful than a video of various brightly colored shapes. I need to learn if there are any special considerations to be taken into account when preparing the video to be used for evaluation.
12. **Take the video.** This task also involves storing the positioner's location at each frame as a parallel data set and using the previously calculated precision measurements to estimate the error in the location measurements.
13. **Send the video data to the Computer Vision community. Accept and score the output of all spatial orientation algorithms.** I will be treating tasks 11 through 13 as optional extras to be completed only if time permits, and this one especially is dependent upon the speedy cooperation of people from other

colleges and universities, and thus introduces a great deal of uncertainty into the analysis of its feasibility. If there is time for this task, then it will also be necessary to define a scoring metric for the results. The main advantage of performing the task is that it would allow my project to demonstrate through completion a real-world application of the capabilities that I added to the positioner. This is certainly a worthwhile goal, and thus will be undertaken if at all possible.

14. **Prepare and Deliver a mid-semester presentation on project status.** Mid-semester presentations will take place on Monday and Tuesday March 26-27.
15. **Create a final project report.** Final reports are due Thursday, May 10th. While the presentations take place before this, I will probably need to start preparing the report before giving my presentation.
16. **Prepare and deliver a final project presentation.** Final project presentations are on Monday and Tuesday, May 7th and 8th. In addition to presenting my work and my results, I hope to make the GUI available for others to experiment with.

Critical Path

Table 1 shows each task described above along with estimated duration in days, effort in hours, tasks needed, and tasks fed. It is the basis for the Critical Path diagram included as Appendix A. The diagram itself also includes the Earliest Start and Latest Start above each task, and indicates with thick lines those tasks for which the two values are the same: the critical path. Numbers along a line connecting two nodes indicate task duration.

#	Task	Dur. (days)	Eff. (hrs)	Needs	Feeds
1	Obtain Machine-shop certification	7	21		3
2	Sketch potential mechanisms for semi-permanently fixing the Hall-Effect sensors in place. Discuss these mechanisms with my advisor and the machine shop coordinator, and decide on a final design.	2	6		3
3	Build the fixtures, and attach them to the linear positioner	2-4	6-12	1,2	10

#	Task	Dur. (days)	Eff. (hrs)	Needs	Feeds
4	Obtain familiarity with the current program and GUI for controlling the linear positioner.	2	8		8
5	Research GUI programming paradigms and recommended languages for implementation. Choose which language I will use.	3	12		6
6	Learn how to program GUI's in my chosen language.	4	16	5	8
7	Discuss / create design goals for the GUI.	2	8		8
8	Implement the GUI	14-20	50-90	4,6,7	9
9	Test and debug the GUI	7-14	35-60	8	10
10	Determine the precision of the linear positioner.	2	6	3,9	12, 15, 16
11	Research / discuss the video data required to provide a good dataset for spatial orientation algorithms	1	4		12
12	Take the video.	2-4	6-10	10,11	13
13	Send the video data to the Computer Vision community. Accept and score the output of all spatial orientation algorithms.	4-8	10-20	12	15, 16
14	Prepare and deliver a mid-semester presentation on project status	2	6		
15	Create a final project report	8	35	9, 10, 13	
16	Prepare and deliver a final project presentation	4	16	9, 10, 13	

Table 1: Project Activities

Conceptually, the Critical Path Diagram is broken up into three sub-projects: the design and creation of the fixtures for the Hall effect sensors, the design and creation of the GUI, and the design of the video data for spatial orientation. As can be seen, the critical path runs through the GUI subproject.

The three sub-projects are relatively independent until they merge at the end of the diagram for the actual creation and distribution of the video. I have marked this part of the project as optional, since I will be treating it as a non-critical part of the project, to be completed only if there is sufficient time.

In order to more easily visualize my timeline and flexibility, I have included a GANTT chart as Appendix B. It depicts for each activity not on the critical path the span of time in which it can be completed, shown with a thin line, and the actual time required within that span, shown as a line of moderate thickness. Activities on the critical path are shown with a thick line. One drawback of Critical Path planning is that it fails to account for timing restrictions in its evaluation, assuming instead that any task can be begun or finished based solely on whether the tasks that it depends upon have been completed. My first task, obtaining machine shop certification, does not adhere to this assumption: I am required by the constraint of the certification program's start date to begin obtaining certification immediately upon commencement of the spring semester. I have chosen to indicate this on the GANTT chart by moving the line indicating duration to the beginning of the timespan. All other durations are included in arbitrary locations near the middle of their allotted timespans.

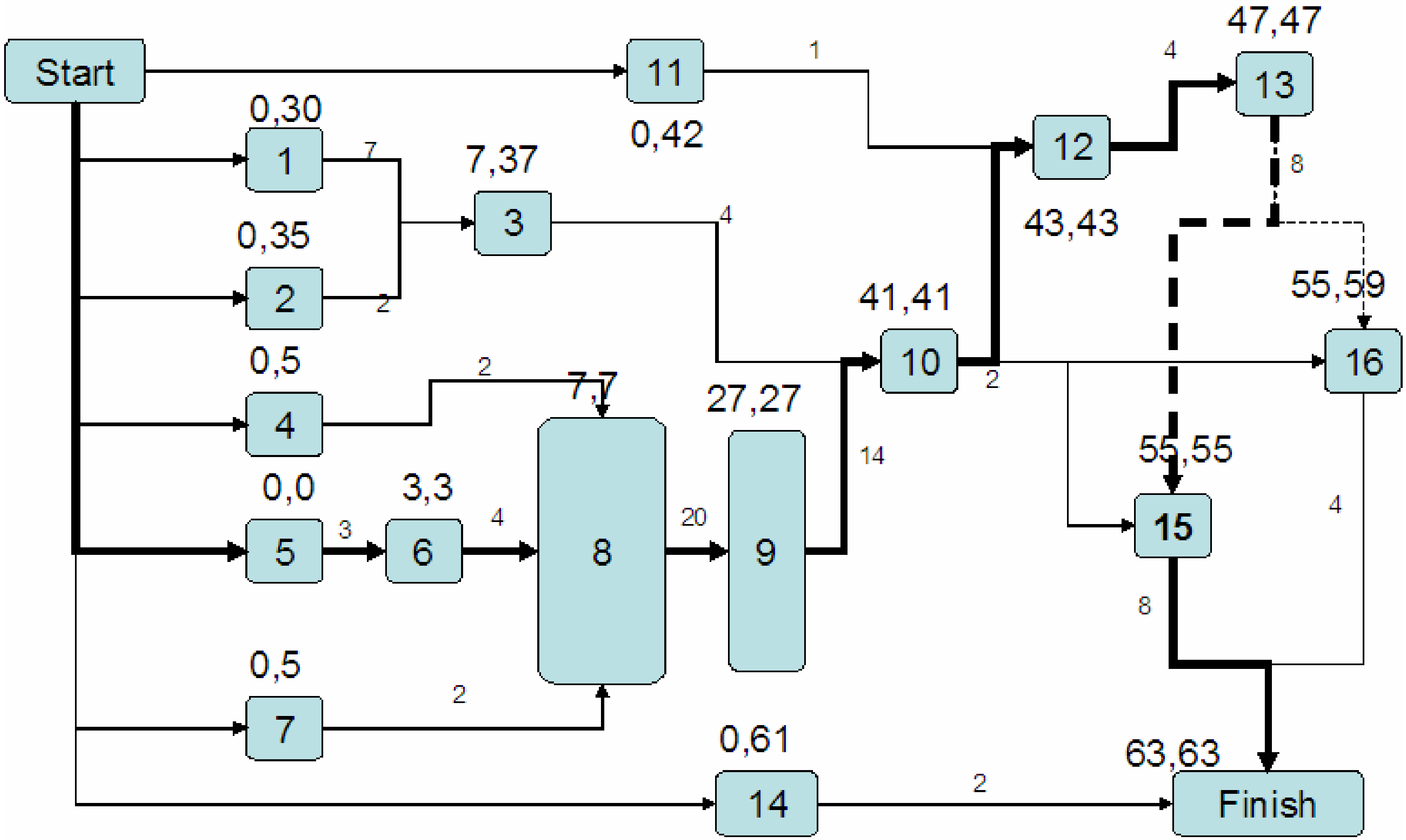
Real-World Design Considerations

Because the vast majority of this project will be implemented in software, it will leave only a very small environmental footprint. The footprint of the software itself will be no larger than the electrical requirements of the computer on which it was created and the space taken on the hard disk storing it. The fixtures for the Hall effect sensors will most likely be metal, which means that they needed to be mined somewhere and will eventually need to be disposed of, but these drawbacks are mitigated by three observations. First, the fixtures will be relatively small, and therefore have a less significant environmental impact. Second, nothing in this project is intended to be scaled to a production level: it is sufficient in its uniqueness. Third, the fixtures can be either reused or recycled once they have outlived their usefulness on the linear positioner.

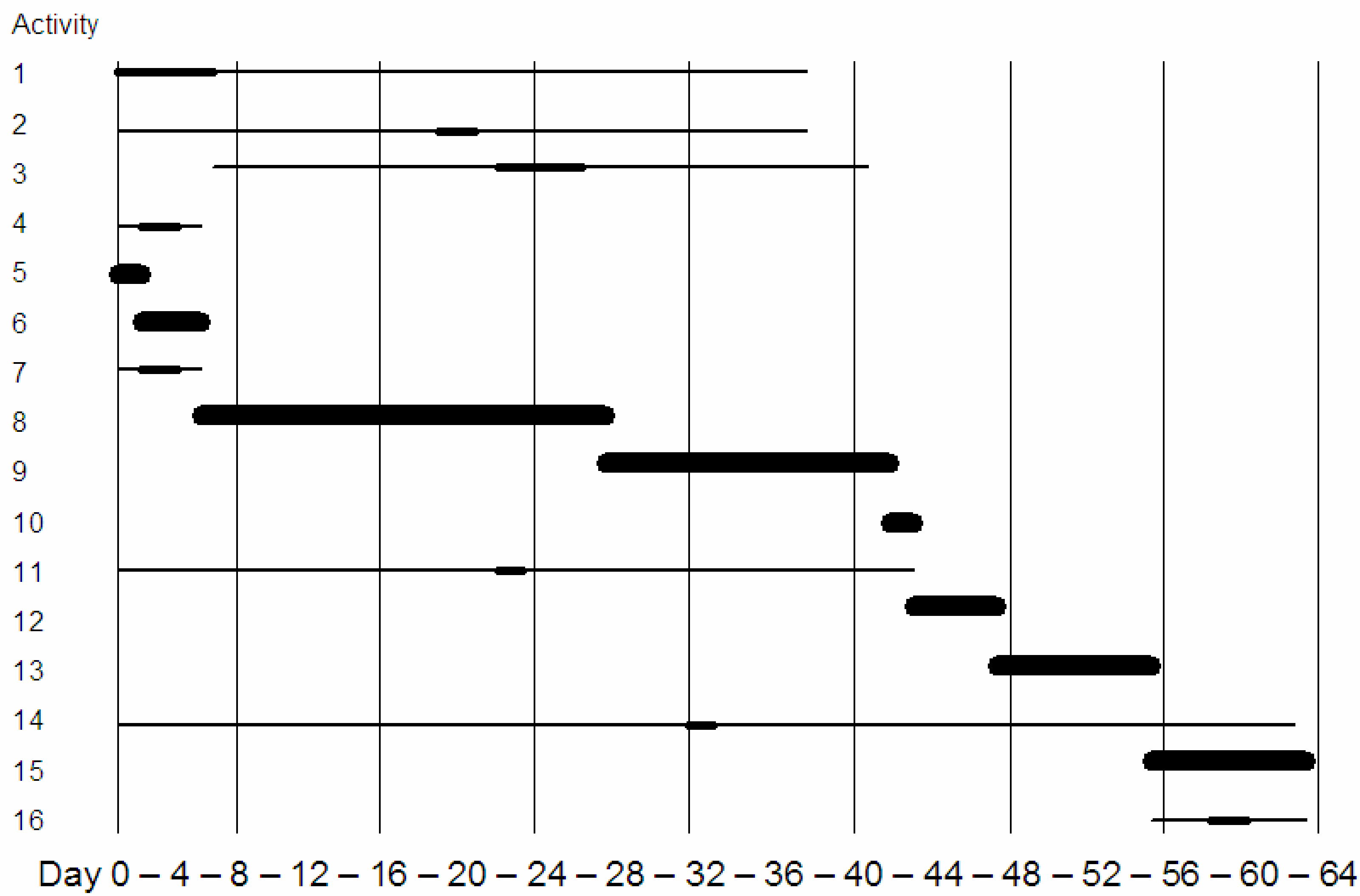
More salient to the design of software is the consideration of the software's usefulness to future individuals after I leave the project. To facilitate a smooth transfer and ensure that I leave behind a useful product, I intend to create a user manual that describes in depth the full capabilities of my software. In addition, I will ensure that my code is well-designed and well-commented so that it can be debugged, expanded, or improved upon in the future.

The final consideration is of my project’s usefulness to the computer vision community. As I mentioned above, video data taken by a camera attached to the linear positioner can provide a useful metric by which to measure the accuracy of existing ego-motion and SLAM algorithms.

These considerations serve as a reminder that my project does not exist in a vacuum, but instead has an impact on the environment, on future students who wish to use or build on it, and on the world at large for whom it can be beneficial.



Appendix A: Critical Path Diagram



Appendix B: GANTT Chart