Robone: Next Generation Orthopedic Surgical Device

Table of Contents

Team Members

Andrew Hundt
Alex Strickland
Shahriar Sefati

Mentors

Prof. Peter Kazanzides
(Prof. Taylor)

Topic and Goal

This plan is to develop a prototype integrating several subsystems of a next generation robot for orthopedic surgery

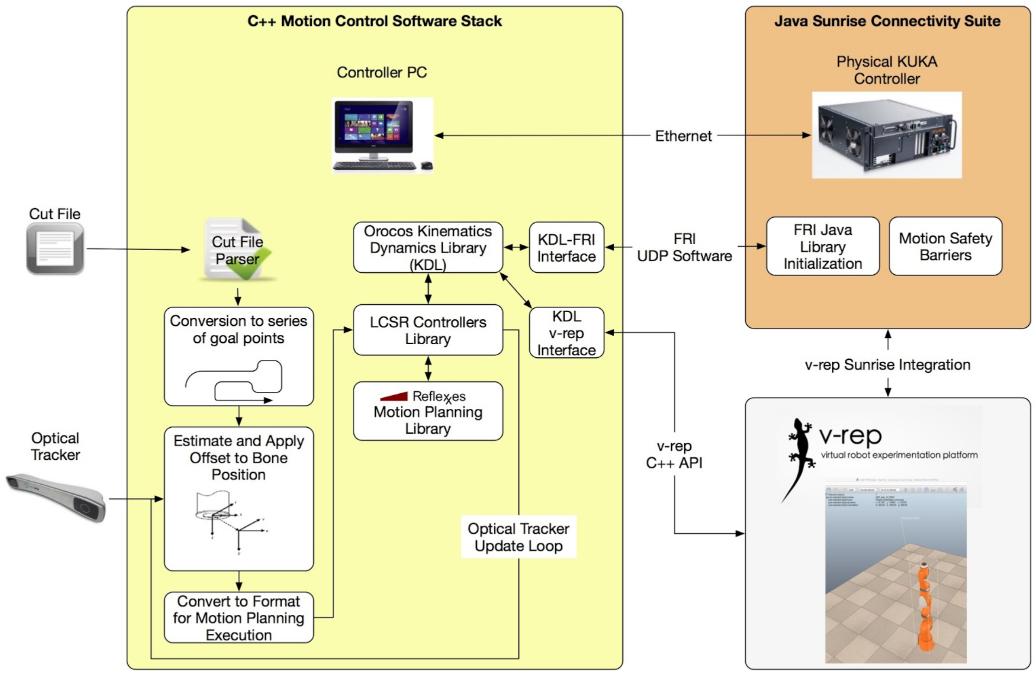
Relevance/Importance

Robodoc, now called TCAT, is an existing computer assisted orthopedic surgical device for hip replacement surgery. The existing device isn't perfect and requires the patient to be fixed to the operating table, an invasive and time consuming process. A next generation system will make real time position adjustments using a device such as an optical tracker so fixation is no longer necessary. The goal is to enable faster and less invasive surgery with this type of computer assisted tool.

Technical Summary of Approach

Below we outline the technical approach we will take to completing our project.

Initial Software Design Concept



The initial software design is to integrate existing tools to create a first working prototype. Then, if those first pass implementations don't meet our requirements, we can improve or replace components as needed. Design, implementation, evaluation, testing and iteration of these components will be done using the scrum process management method, which is described later in this document.

Outline of Project Plan

There are also some items of note to put this outline into context. The time in brackets [1] is measured in Ideal Full Time Equivalent (FTE) days. The names are of the people who will take leadership roles on those tasks, others may contribute substantially. Also, the scrum method allows reallocation of tasks, including the time, order and leadership roles during sprint planning. Finally, each level of deliverables (min, expected, and max) includes all previous ones.

Team startup

- Introduce software tools. [0.5][Andrew, Alex, Shahriar]
- Introduce programming and controlling kuka arm. [0.5][Andrew, Alex, Shahriar]
- Introduce simulation software, V-REP [0.5][Andrew, Alex, Shahriar]
- Create shared repository for software development. [1][Andrew]

- Evaluate additional methods to measure accuracy of results [1][Andrew, Alex, Shahriar]

Initial Simulation

Create a simulation of the system in V-REP without optical tracker

- Add arm to simulation [1] [Andrew]
- Integrate motion planning,
    - Most likely implemented using the reflexxess type II or type IV planning library. [2] [Andrew]
- Add milling simulation of simple object [2] [Shahriar]
- Milling simulation of bone shaped object [2] [Alex]
- Add simulation with optical tracker. (may go in step 5) [5] - [Andrew, Alex]

Initial Arm Integration - *min deliverable*

- Implement basic Java setup on Sunrise Connectivity suite. [2] [Alex]
- Implement and test read arm state over Fast Robot Interface (FRI). [3] [Andrew]
- Implement and test commanding arm motion over FRI. [2][Andrew]
    - Will send joint commands to the robot using this interface.
- Collaborate with other kuka groups, if possible on path planning. [3][Andrew, Alex, Shahriar]
- Move along simple series of points (joint and/or cartesian space) [5][Andrew, Shahriar]

Cut file integration - *min deliverable*

- Acquire ascii cut files [0.5] [Andrew]
- Implement ascii parsing and conversion to format amenable to sending to planner or arm as commands. [4] [Alex, Andrew]
- Test parsing and motion commands in simulation [2] [Shahriar]
- Test parsing and motion commands on physical robot [2] [Shahriar]

Optical tracker integration - *expected deliverable*

- Estimate bounds of acceptable system response time with respect to bone motion. [2] [Alex]
- Acquire optical tracker [0.5] [Andrew]
- Setup of optical tracker with kuka [1] [Andrew]
- Implement reading of optical tracker data into software, using existing saw components [3] [Alex]
- Integrate optical tracking data into cut file and arm commanding loops [5] [Andrew]
- Reaction time testing [5] [Alex]
    - draw a straight line on an object, move object and check response time
    - see “physical simulation of cutting” idea below

    - Characterize response time
    - Improve response time if necessary
- Simple redundancy checking of arm base and end effector position against optical tracker for consistency, safety, and reliability. [3] [Shahriar]
- Make sure the motion of the robot is consistent with the motion of the optical tracker. [2] [Shahriar]
- Trigger alarm if optical tracking system says that the robot is out of the planned motion. [2] [Shahriar]

Milling Physical simulation - *max deliverable*

We will create a physical simulation of cutting, as opposed to a computer simulation.

The initial concept is to put an optical tracker fiducial on the end effector and have a clear box to simulate “bone”. We can then use the optical tracker to generate a simulated estimate of actual cutting. This avoids the complexity of acquiring materials to cut and dealing with the dust created by milling foam, wood or other test cutting materials.

- Design and Create fiducial mounting attachment [5] [Shahriar]
- Implement logging of physical simulation [5] [Andrew]
- Integrate logging with V-REP to visualize execution of simulation [5] [Alex]
- Implement method to evaluate planned vs actual path within error bounds of sensors. [5] [Andrew, Alex]
- Create evaluation analysis [2] [Shahriar]

Investigate arm motion planning - *max deliverable*

- Investigate and estimate other motion planning tasks more accurately. [3][Andrew]
- Minimize elbow movement during cutting scenario [13][Andrew]
- Adjust cutting speed based on torque resistance of cutting [13][Shahriar]
- Investigate response and pre planning to avoid going beyond joint configuration limits. Consider stopping in real time and asking user to move bone and arm to a new position within the workspace. [13][Alex]
- Response to human touch, stopping path, following human pressure, and then resuming motion when human moves it approximately back on path. [13][Alex, Shahriar]

Milling integration - *max deliverable*

Install physical milling equipment on arm and demonstrate real output shape cut from a demo material such as wood.

- Create or acquire milling attachment (or whatever other mechanism we will use) [5-10][Shahriar, Andrew]
- Investigate and estimate design and fabrication steps more accurately. [3][Andrew, Shahriar]
- [Many steps may be here for design and fabrication [unknown][Shahriar, Andrew]
- Test milling attachment [3] [Shahriar, Andrew]
    - If needed, find a machine shop on campus which is properly equipped to handle dust [1] (may take weeks of wall time)
- Integrate control of milling attachment with rest of system [5][Alex]

System testing and iteration

- Test complete system [3][Andrew, Alex, Shahriar]
- Improve issues found [5][Andrew, Alex, Shahriar]
- Repeat steps A,B as necessary

Deliverables scale

Please note that each level includes all previous levels.

Min

- Receive arm state in real time
- Read in cut file specifying shape of implant
- Drive both simulated and physical KUKA arm along cut file path

Expected

- Receive optical tracker position in real time and adjust cut path accordingly to maintain consistent final cut volume to the extent possible, considering algorithm latency and arm speed limits.
- Characterize performance

Max

- Create a plan for executing selected max deliverables after the expected deliverables are complete
- Allow human adjustment of elbow or removal of arm during motion
- Milling Integration
- Improve motion planning to minimize elbow movement
- Evaluate configuration and joint limits, especially if object cut path exits workspace of arm

Possible side deliverable (no commitment)

- CISST component for kuka arm control

Dependencies and plan for resolving them

Optical Tracker

Estimated Resolution: February 27

We are depending on acquisition of the Atracsys optical tracking device, which is out for repair. If necessary, we will first utilize simulations to build out the software infrastructure then implement integration with another model of optical tracker such as a polaris tracker.

Estimated Resolution: April

Integration with arm software and controller is another dependency for using the optical tracker with the system. We expect we will be able to implement this within a sprint of receiving the optical tracker. If we are not, we can consider integrating another optical tracker using CISST.

KUKA Robot

Estimated Resolution: April

We are also assuming the KUKA robot arm itself will be available. We expect this will not be a problem, but if it is we can ask the various labs in LCSR that have an identical arm for time on their arm. If this also fails we plan to utilize simulation to continue development.

Milling

Estimated Resolution: April 30th

The creation or acquisition of actual milling device is the primary obstacle. Since it is a max deliverable, it will not be started until later in the semester, so the timeline for acquiring, designing, or creating the device will be very short. Integration of milling device as an end effector for the KUKA robot arm could include challenges such as incompatibility of the connectors or attachment not being to specifications. Additionally, special cabling and interfaces may be required but not available. If the above milling dependencies block progress, we will simply simulate the system and forgo fully completing the max goal. Alternately, Andrew Hundt may finish this component of the plan after the course has ended.

Access to space and materials for milling is one constraint. Milling involves dust and we would need a suitable location and equipment, plus availability of the robot. As a result, we might need to move the robot which would require planning and permissions as the project progress to max deliverables. If we are unable to secure space, materials, permission or time to physically test the robot we can test the system without the actual milling device running in soft pliable materials.

An extreme last resort backup plan is to acquire a simple electric drill and fasten it to the arm's end effector using a simple hand made coupling device. A a safety switch where the drill connects to the wall can be used for cutting power between the drill and the wall. A cable tie can be used to manually enable the drill before starting the arm's motions. We could use a drill that is around the lab, acquire the funds to purchase one, or use one of the team members' personal drills.

Software

Estimated Resolution: April

Some higher quality level integration of arm control software, such as real time torque control depends on planned software updates by the manufacturer, KUKA. This is expected by Quarter 2 of 2015. If it does not arrive we will implement control using the existing position interface.

Logistics

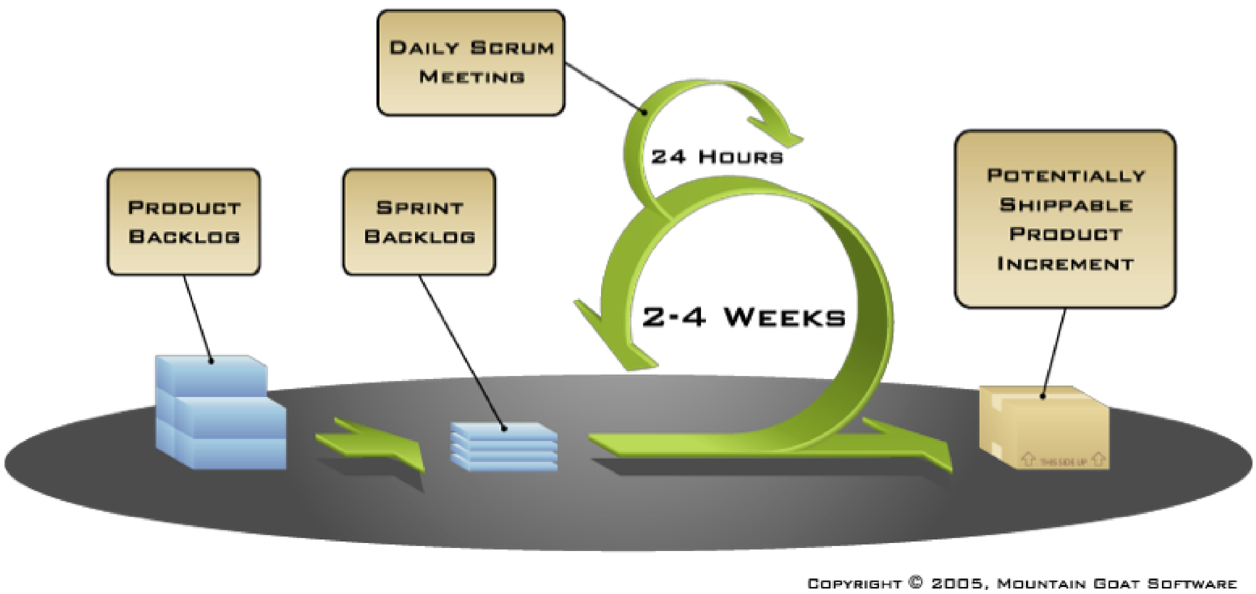
Estimated Resolution: February 13

Access to mentors is a logistics issue. We have a preliminary plan in place to meet with our mentor biweekly during the sprint planning meetings.

Management Plan

We've organized our whole plan on github + huboard. We are also following the "scrum" project management method with some tailoring to our purposes. The key to this system is iterative project development, where testing is continuously integrated in the development of the project, and the project is broken into small chunks which are followed through to completion, including testing, during two week increments called "sprints". At the end of each sprint the components that are completed should fit the definition of a "potentially shippable product".

Here is a link to our github repository which outlines each sprint and its corresponding tasks: https://github.com/ahundt/robone/milestones



There is a plethora of resources available that describe scrum. Since we cannot cover every detail here, we will refer to those resources so that interested parties can get additional information.

Here is a 1.5 minute explanation video for scrum: http://is.gd/scrum90sec
More detailed 10 minute video explanation of scrum: http://is.gd/scrum10min
Additional resources for scrum: http://www.mountaingoatsoftware.com/
Scrum intro presentation: http://is.gd/downloadScrumPPT
Explanation of Sprint Planning meetings:
http://www.mountaingoatsoftware.com/agile/scrum/sprint-planning-meeting

Scrum Management Meetings

Sprint planning for the next sprint occurs between sprints. Ideally we would all meet for:

- **Sprint Review** (15-30 mins, every 14 days)
  - demo completed functionality to the group (5 min each)
  - no slides
- **Sprint retrospectives** (15-30 mins, every 14 days)
  - Discuss what the team should start, stop, and continue doing
- **Sprint planning** (30-60 min, every 14 days)

    - Collaboratively identify tasks, estimate time to complete in "ideal days", and reorganize tasks
    - Team members select tasks each commits to leading/completing
    - High level plan is considered
- **"Daily" scrum meetings** (5-10 mins, every 2-3 days)
    - Describe what we did (1 min each)
    - Describe if we are blocked from continuing work, aka a dependency (1 min each)
    - Commit to progress for next scrum meeting (1 min each)
    - Note: Not for problem solving
        - Helps others avoid unnecessary meetings
        - Can only agree to schedule a separate meeting later

Also, one other key to all this is the standard by which we all agree upon **what it means for a task to be complete**, and there are several components:

- "Conditions of Satisfaction" (aka acceptance criteria)
    - requirements specific to the task. ex: "demonstrate moving to a series of points in simulation"
- "Definition of Done"
    - applies to every task. ex: code checked in, reviewed by another party, unit tested, documented, etc.
    - More details at: http://is.gd/scrumDefinitionOfDone

- Someone other than the person assigned verifies a "Completed" task before it is "Accepted". Preferably this would be Dr. Kazanzides, or the other team members.

- Sprint Review and Sprint Planning meeting: Team members and perhaps Dr. Kazanzides.
- Daily scrum meetings: Team members only.

One possible weakness in scrum was summarized by Dr. Kazanzides:

> *"To me, it seems one weak point may be relying on engineers to estimate how much time is left on a task. I managed a software team for 12 years in industry, and I knew one engineer who had an interesting use of the word "done". There were conversations like the following:*
>
> *Manager: What are you working on now?*
> *Engineer: I'm working on X*
> *Manager: Didn't you say that X was done last week?*
> *Engineer: Yes, but now I am doning it to get it doner (pronounced "dunning" and "dunner")."* -Dr. Kazanzides

Several controls exist to make sure what is meant by "done" is consistent across individuals, including the demos, the agreed "definition" of done, and testing/review by another party. Additionally, the retrospective serves as an occasion to correct any misunderstandings or inconsistencies among individual team members as time goes on.

Additionally, there is another process to more accurately assign and plan the work for each sprint:

- Instruct everyone to estimate time as consistently as possible from one week to another, with no deliberate day inflation or deflation
- Assume a group as a whole will generally be consistently inaccurate (imperfect, but better than nothing)
- Assign points for each task each sprint and record how many were actually done overall
- Over time it becomes possible to measure stats about what can really be accomplished in a single sprint based on past results.
- With practice, real measurements, and feedback people become better able to estimate the time required more accurately.
- Ultimately as more data driven progress is made both progress and long term completion dates can be estimated more accurately.

There are additional tools in scrum for when it isn't initially possible to estimate a task accurately. "Spikes", are quick first pass attempts at a tiny subset of a problem so it becomes possible to estimate the larger job.

Of course, none of this is perfect. Availability, team members, etc change over time as does the skill set of each person. Furthermore, unseen bugs and certain research tasks come up that are nearly impossible to estimate accurately. Nonetheless, these methods are widely used and generally considered effective.

> *"Having a process is better than not having one. Also, unless you initially try to estimate how long a task will take, you won't ever be able to compare your estimate to reality and thus more finely tune your estimation skills."* -Dr. Kazanzides

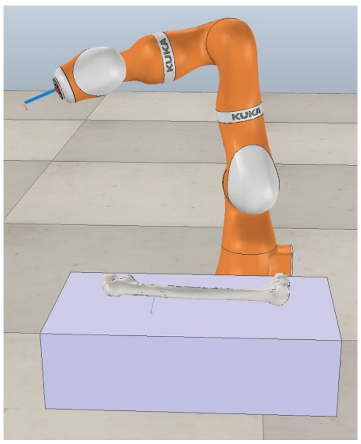
Key dates and assigned responsibilities

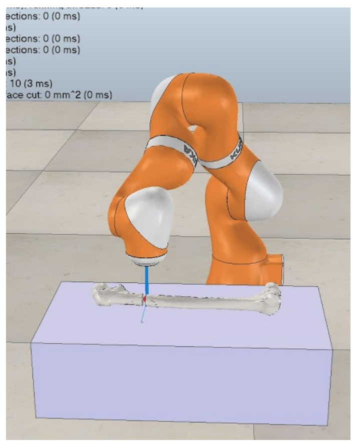
Please note that testing is a specified requirement of scrum and will be done for each component of the system before the end of each sprint.

Sprint	**Date**	**Task**	**Deliverable Level**
1	**Feb 19**	Initial Simulation	min
2	**Mar 5**	Initial Arm Integration	min
3	**Mar 19**	Cut file integration Optical tracker integration	min expected
4	**Apr 2**	Milling Physical Simulation	max
5	**Apr 16**	Investigate arm motion planning	max
6	**Apr 30**	Milling integration	max
7	**May 7**	Additional system testing and iteration, Poster Session	expected

Initial Simulation Progress

We have created an initial simplified proof of concept of bone cutting using the v-rep robot simulator. Currently this includes a real bone model, the kuka arm, and an attached milling object. The simulation uses the reflexxes library to navigate to the start part of a simple proof of line cutting path, then using the cutting path component of V-REP to follow the path from start to finish. Finally, it switches back to reflexxes to return to the path start and begin again.





Reading List

- G. Boiadjiev, K. Delchev, T. Boiadjiev, K. Zagurski, R. Kastelov, V. Vitkov. Controlled Trust Force Influence on Automatic Bone Drilling Parameters in the Orthopedic Surgery. Int. J. of Pure and Applied Math. 2013; 88(4):577-592.
- J. Pransky. ROBODOC -surgical robot success story. Industrial Robot. 1997; 24(3):231-233.
- Taylor, Russell H., et al. Computer-integrated revision total hip replacement surgery: concept and preliminary results. Medical image analysis 3.3 (1999): 301-319.
- Kazanzides, Peter, et al. Force sensing and control for a surgical robot. Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference on. IEEE, 1992.
- Kazanzides, Peter, et al. An integrated system for cementless hip replacement. Engineering in Medicine and Biology Magazine, IEEE 14.3 (1995): 307-313.
- Kazanzides P, Fichtinger G, Hager GD, Okamura AL, Whitcomb LL, Taylor RH. Surgical and Interventional Robotics - Core Concepts, Technology, and Design [Tutorial]. Robotics & Automation Magazine, IEEE 15.2 (2008): 122-130.
- Albu-Schäffer, A., et al. The DLR lightweight robot: design and control concepts for robots in human environments. Industrial Robot: an international journal 34.5 (2007): 376-385.
- Kroger, T. "Opening the door to new sensor-based robot applications—The Reflexxes Motion Libraries." Robotics and Automation (ICRA), 2011 IEEE International Conference on. IEEE, 2011.
- Kroger, T. “Online Trajectory Generation Algorithms as an Intermediate Layer between Robot Motion Planning and Control.” Workshop on Robot Motion Planning: Online, Reactive, and in Real-time. 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2012
- Taylor RH, Mittelstadt BD, Paul HA, Hanson W, Kazanzides P, Zuhars JF, et al. An Image-Directed Robotic System for Precise Orthopaedic Surgery. IEEE Trans on Robotics and Automation. 1994 Jun;10(3):261-275
- Kazanzides P, Zuhars J, Mittelstadt B, Taylor RH. Force Sensing and Control for a Surgical Robot. In: IEEE Intl. Conf. on Robotics and Automation. Nice, France; 1992. p. 612-617
- Zuhars J, Hsia TC. Nonhomogeneous material milling using a robot manipulator with force controlled velocity. In: IEEE Intl. Conf. on Robotics and Automation. Nagoya, Japan; 1995. p. 1461-1467
- Stocco L. Path Verification for Unstructured Environments and Medical Applications. In: ASME Design Automation Conf., Symp. on Mechanisms and Devices for Medical Applications. Pittsburgh, PA; 2001. p. 1103-1108