

# Assignment 1: Kinematics of Medical Robotics

ME 328: Medical Robotics  
Stanford University ♦ Winter 2019

Due in the drop box outside Allison's office by 4:00 pm on Wednesday, January 16

Note: Discussions or collaborations to help you come up with approaches or ideas for assignments are fine. You can work on this assignment with your classmates, but *no copying* -- what you submit should be your own work.

**0. Guest Speaker:** After the seminar on Friday, January 11, look for a question posted to Canvas about the seminar, and answer it online by the time this assignment is due.

**1. Readings:** You can download these papers from <http://www.stanford.edu/class/me328/#readings>. These papers are from the late 1990's and early 2000's -- a time when the field of surgical robotics was being defined.

G. S. Guthart and J. K. Salisbury, Jr. The Intuitive<sup>TM</sup> telesurgery system: overview and application. In Proceedings of the IEEE International Conference on Robotics and Automation, pp. 618-621, 2000. (DOI: 10.1109/ROBOT.2000.844121)

*This paper succinctly describes the design goals and function of the da Vinci Surgical System from Intuitive Surgical, Inc. (It is an invited conference paper, which explains its somewhat less formal style.)*

A. J. Madhani, G. Niemeyer, and J. K. Salisbury, Jr. The Black Falcon: a teleoperated surgical instrument for minimally invasive surgery. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 936-944, 1998. (DOI: 10.1109/IROS.1998.727320)

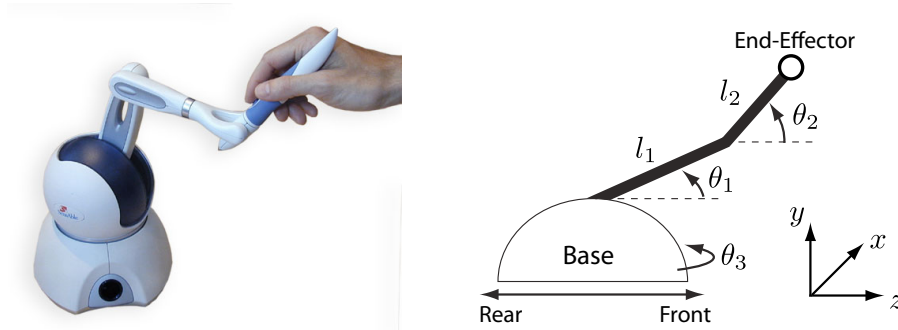
*A conference paper describing the design of a remote-center-of-motion robot; this design is similar to the commercial da Vinci Surgical System. Unlike some other fields, a number of important contributions in robotics appear in peer-reviewed conferences, rather than journals.*

R. H. Taylor and D. Stoianovici. Medical Robotics in Computer-Integrated Surgery. IEEE Transactions on Robotics, 19(5):765-781, 2003.

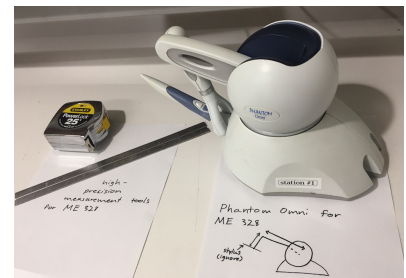
*This review paper provides a broad overview of the field of medical robotics, with particular attention to kinematic architectures and mechanical design considerations. This paper is rather long, so you can just skim and consider it a useful reference.*

- A. (Guthart and Salisbury) What do the authors say are the four sources of complexity in minimally invasive surgery (MIS) procedures?
- B. (Guthart and Salisbury) What are the foot switches on the da Vinci master console used for?
- C. (Madhani et al.) Where are the actuators of the Black Falcon located?
- D. (Madhani et al.) How many degrees of freedom does the Black Falcon have? There is a requirement for a minimally invasive surgery robot that the instrument must enter the tissue through an incision point. How many degrees of freedom are constrained by this requirement?
- E. (Taylor and Stoianovici) Do the authors think that surgical robot drives should (1) exhibit high stiffness and low backdrivability, or (2) be easily back-drivable using direct drive or lightly geared actuation? Can you think of a reason why the method *not* preferred by the authors would have advantages?

**2. Phantom Omni kinematics:** A 3-degree-of-freedom (dof) robot (also a haptic interface), the Phantom Omni, is shown below. In this device, two dof are created with a revolute-revolute serial arm, and the third dof simply rotates the serial arm. The diagram below (at right) shows a configuration of the device that is not achievable due to joint limits (observe for yourself by exploring the Omni movements without forcing it past designed limits). However, this is a convenient choice of coordinates for computing the kinematics. Also, note that the joint angles are absolute (measured with respect to ground). The end-effector is the point at the center of the “wrist” where the pen/stylus is attached to the main part of the device. Note that the wrist is not actuated, so we will not consider it or the stylus in our analysis. The coordinate system shown is just to define the  $x$ ,  $y$ , and  $z$  directions. You can place the origin of the coordinate system at a location that is convenient for your calculations – just be sure to explain where it is.



A. Go to the lab in Bldg. 550, Room 108. Access the room using the keypad and the code provided in class. Examine the Phantom Omni sitting on the bench just to the right of the door. To the best of your ability (without taking apart or modifying the Omni in any way!), make measurements of the link lengths of the Omni using the measuring tools that the CA has laid out for you (see image at right). After you are done, be sure to return the materials to their original locations for the next student.



B. Develop a forward kinematic model of the Omni. In other words, compute the  $x$ ,  $y$ , and  $z$  positions of the end effector as functions of  $\theta_1$ ,  $\theta_2$ , and  $\theta_3$ .

C. Determine the Jacobian of the Omni. Don't forget the chain rule when taking partial derivatives! Write out the matrix so that it can be used in an equation of the form  $[\dot{x} \ \dot{y} \ \dot{z}]^T = J[\dot{\theta}_1 \ \dot{\theta}_2 \ \dot{\theta}_3]^T$ .

D. Assume the device currently has joint angles  $\theta = [45^\circ \ -45^\circ \ 0^\circ]^T$ , which is near the center of its workspace. If the joint velocities are  $\dot{\theta} = [90^\circ/\text{s} \ 180^\circ/\text{s} \ 0^\circ/\text{s}]^T$  degrees per second, what is the vector of Cartesian endpoint velocities?

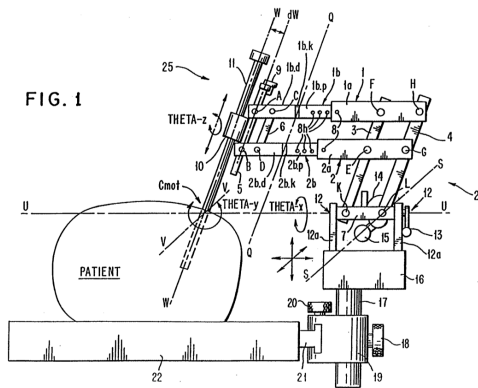
*Hint: Enter the forward kinematics and Jacobian (Parts B and C) into a program such as Matlab, where you can change the angles and observe the change in output. A reality check for seeing if the forward kinematics are being calculated correctly is to have your Matlab script construct a simple stick drawing of the device. Turn in a hardcopy of any code you write.*

E. Suppose that you want the robot end-effector to push on tissue with a Cartesian force vector of  $f = [2 \ 1 \ 0.5]^T$  N. Your device is at the same position as stated above. What vector of joint torques would be needed to create this force at the end-effector?

F. Now you are at a position such that the joint angles are  $\theta = [45^\circ \ 10^\circ \ 0^\circ]^T$ . You would like to output the same Cartesian force vector,  $f = [2 \ 1 \ 0.5]^T$  N. What joint torques are required? Are the torques higher or lower than at the previous position? Why is this so?

G. Are there any singularities of this device within its reachable workspace? You don't have to calculate them from the Jacobian (unless you want to); just use your intuition about the geometry of the device. You can play with the real Phantom Omni to help you figure this out.

**3. Remote Center of Motion kinematics:** There are a number of different mechanisms that can provide a remote center of motion. The Black Falcon presented in the paper by Madhani et al. is one example. Another is a device, shown below, patented by Taylor, Funda, Grossman, Karidis, and LaRose from IBM. (Yes, IBM did medical robotics!) The patent number is US5397323, feel free to look it up for details.



A. Create your own sketch of the device, showing the relevant parameters for 3D ( $x$ ,  $y$ , and  $z$ ) positioning of the endpoint. The three joint variables are  $\theta_x$ ,  $\theta_y$ , and  $d_z$ . ( $d_z$  is not labeled in the drawing above, but it is the translation of the instrument, which is labeled 11. Make  $d_z = 0$  when the endpoint is at the RCM point.) Label the near-vertical links  $l_1$  (both these links have the same length) and the horizontal links  $l_2$  (again, both these links have the same length).

B. Develop a forward kinematic model of this device. In other words, write equations that give the  $x$ ,  $y$ , and  $z$  positions of the end effector as functions of the joint variables  $\theta_x$ ,  $\theta_y$ , and  $d_z$ , and link lengths  $l_1$  and  $l_2$ .

C. Develop an inverse kinematic model of this device: Write equations that give the joint variables  $\theta_x$ ,  $\theta_y$ , and  $d_z$  as functions of the  $x$ ,  $y$ , and  $z$  positions of the end effector as well as the link lengths  $l_1$  and  $l_2$ .

*Hint: RCM robots do not translate with respect to their remote center. Because of this, the end-effector of the robot can be described with spherical coordinates about this point. For performing the inverse kinematics, consider that the mapping of a point from Cartesian to spherical coordinates is one-to-one.*

D. Create a simulation of the robot kinematics in Matlab. Your Matlab script should:

- 1) Draw a simple 3D skeleton of the device in an initial configuration similar to that shown above.
- 2) Repeatedly draw the skeleton as it sweeps from  $\theta_x = 0^\circ$  to  $45^\circ$  to  $-45^\circ$  and back to zero again (while the other joint variables remain fixed).
- 3) Repeatedly draw the skeleton as it sweeps through a similar range for  $\theta_y$  (while the other joint variables remain fixed at their original value).
- 4) Repeatedly draw the skeleton as it sweeps through a reasonable (considering the size of the patient shown in the drawing) set of values for  $d_z$  (while the other joint variables remain fixed).
- 5) Create an “interesting” path (you be the judge) for the endpoint, and move the robot accordingly.
- 6) Save the whole thing as a single movie with enough resolution of motion to appear pretty smooth but not create a file greater than 5 MB. Save **only** the movie from the workspace with the following format: `save('FirstnameLastnameAssignment1Movie', 'MyMovieWorkspaceName')` and upload it along with your accompanying code (called 'FirstnameLastname') to Canvas under Assignment 1.

*Hint: You can save any sequence of graphs and play the sequence back in a short movie. There are two steps to this process: (1) Use `getframe` to generate each movie frame. Be sure that your computer is not in screen saver mode when you call `getframe`. In the event that you are using several virtual desktops, make sure that the desktop on which the MATLAB application is running is visible on your monitor. (2) Use `movie` to run the movie a specified number of times at the specified rate. Typically, you use `getframe` in a `for` loop to assemble the array of movie frames.*