## Assignment for Week 6 readings: (due Tues 23 Nov)

Francesco Iori, Anna Astolfi, Carlo Bosio, Jacopo Quaglierini

Contattaci: [#week6 sul TMM Slack](https://topicsinmulti-b5g1638.slack.com/archives/C02LJB34X6E)

Goal: Learn how to compute when and how objects will slide under the influence of Coulomb friction. Get familiar with Limit Surfaces.

* [TMM Week 6 slides](https://docs.google.com/presentation/d/1GY-98dIG9XqLvJrMZzk5VU37yvZ4nCM7Y_M29waYbt8/edit?usp=sharing)

### Q1.

One of the ways to determine when and how an object under Coulomb friction will start sliding is to use the Maximum Power inequality. A nice example comes from the PhD thesis of H. Sakurai at MIT (1990) who, like Soo-Hong Lee at Stanford, considered the problem of a clamped part subject to cutting forces. In pages 90-100 ([linked here](https://drive.google.com/file/d/1x0wEtEKTKYEQApwHQfFiGffua446lKmX/view?usp=sharing)), starting on page 92, he works an example with a part held by 3 clamps. The (x,y) locations of the clamping points are known. Each results in a frictional contact. The location and direction of the external force are also assumed known.

1.1 How many *unknowns* are there and what are they? It will probably help to review the [SakuraiExampleSlides.pdf](https://drive.google.com/file/d/1dzP7t9Xiqla2cIWByag4Mvc72FbMYe72/view?usp=sharing) from class.

1.2 How many equations (equality and inequality) are there and what are they?

1.3 Use the Maximum Power inequality to solve the problem. Sakurai treats it as a linear programming problem and he calls it the “Maximum Friction Wrench Theorem.” The script, “**SakuraiFriction.py**” gives you a generous head start on setting up the problem for linprog( ). Please finish the script and see how your results compare with Sakurai’s in Figure 4.2.7 (b). To what do you attribute differences? [If you have any trouble determining how to compare your results to those in the Figure 4.2.7b, please post a question to the Slack channel.](https://topicsinmulti-b5g1638.slack.com/archives/C02LJB34X6E)

1.4 Suppose we move the location of pc (where the external force, fc, is applied) to pc = (0,0). What should happen? If you modify the script for this case does it give an appropriate result? What is the magnitude of the external force, fc, needed to initiate sliding in this case? Finally, why do you suppose the answer in this case might be more accurate than for Q1.3 above?

### Q2.

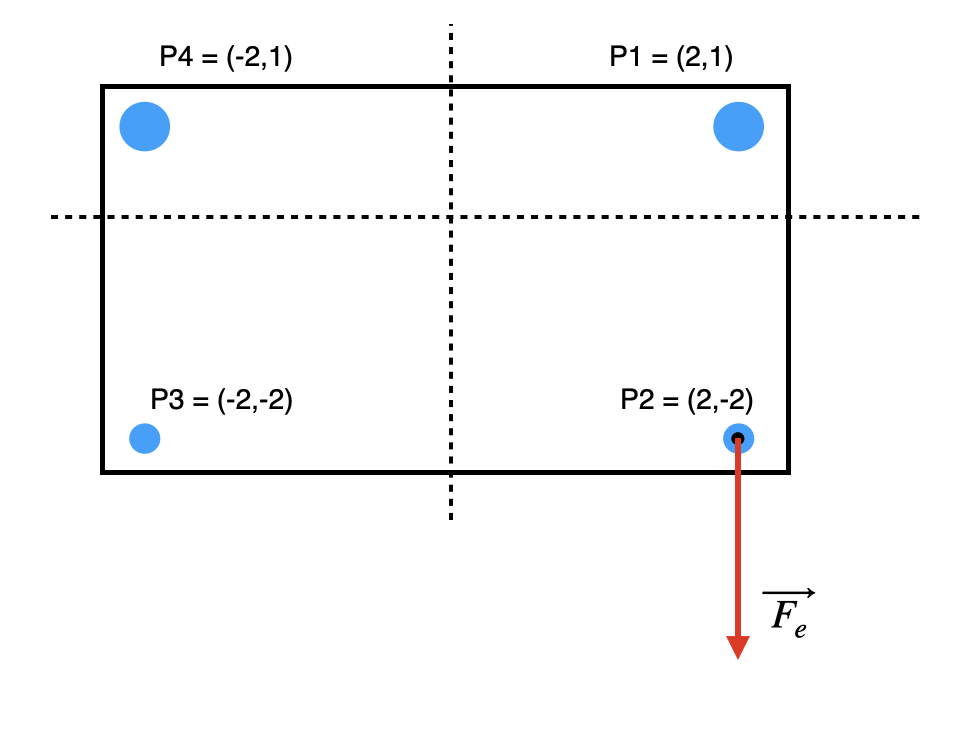
Let us try the same problem using the ellipsoidal Limit Surface approximation from Howe & Cutkosky. The script “LScalcs.py” pretty much does everything, so it’s more a matter of walking through the file and checking that it all makes sense.

2.1 The particular situation addressed in Sakurai’s example is a special case of the more general planar sliding behavior. What is going on and why does it lead to linear equations in this particular case? Why might solving the linear equations for this special case give a more accurate answer than using the ellipsoidal approximation?

### Q3.

A FlyCroTug[[1]](#footnote-2) is a small quadrotor that can attach a cable to an object, fly some distance away, anchor itself using microspines or adhesives, and pull on the cable. An ambitious FlyCroTug has decided to move a divan by attaching a cable to one of its legs. The dimensions and forces are shown in the schematic figure below. Let us assume a realistic coefficient of friction of mu = 0.5 and that the front legs each have a normal force of fn = 1 unit while the rear legs each have a normal force of fn = 2 units (a “unit” is presumably around 50N in this example).

Compute the necessary force units in the cable and the instantaneous unit twist (direction of motion) using the Ellipsoidal Limit Surface method, adapting the script from Q2. Is it rotating about a leg P4, or do you need to do the general case for an ellipsoid?



1. Estrada, M. A., et al. (2018). [Forceful manipulation with micro air vehicles. *Science Robotics*, *3*(23), eaau6903](https://www.science.org/doi/10.1126/scirobotics.aau6903). [↑](#footnote-ref-2)