**Plans:**

1/ Finding the global static feedback law (with T=fixed) is very difficult if not impossible because the solution migh not be time-invariant as Dr. Zeng has found a time-varying state feedback law for the harmonic oscilator aircraft, i.e. u(t) = k(x,t).

Consider a point **x** and its energy optimal controlled trajectory **X**=[x1,....,xn] as well as a second point **y=xi** (i.e **y** is an arbitrary point on the controlled trajectory X) and its energy optimal controlled trajectory **Y**=[y1,....,yn], we can see the optimal control signal and the optimal controlled trajectory for **x** at xi is usually different from those for **y=xi.**

One way to think about this is to think about the amount of effort vs the budget of time. Even if we start at the same location, the difference in time budgets significantly dictates the discrepancy in how much effort should be utilized to reach a common goal.

--> Therefore, a time-energy objective might yield a global feedback law. Specifically, given a same starting location, there should be a unique steering signal and a single resulted controlled trajectory to the target if the objective is to both minimize time and energy. This intuition can be linked to the Bellman principle of optimality and the famous HJB equation.

2/ Then for each point, we have a control signal U=[u1,....,un]. This generates a data set to train a NN whose role is a global controller to stablize a nonlinear control system in a time-energy optimal way.

3/ Additionally, we can explore the robustness of this controller by changing the parameters of the system. Try to give a rigorous analysis of robustness (maybe via gradient/jacobian of the NN). If it is not possible, we ought to qualitatively show the robustness by ploting the tolerant region of parameter variations.

4/ Finally, we can apply this method to the Furuta pendulum (physical platform). To see how robust it is.

**Try:** swing up the Furuta pendulum from bottom with the time-energy minimum, then take a point on the controlled trajectory, and apply minimum time-energy synthesis to see if it yields the same result. Maybe try this first with the Van-der-pol.

**Plans:**

1/ Discretize the optimal controlled trajectory to get more training data: divide an i-th interval (t\_i) into n smaller intervals (n=mod(t\_i/dt)) so that each approximately equals some desired fine step (e.g. dt=0.1ms). This process results in a much more number of training states and inputs. In addtition to getting more training data, this procedure helps to dictate what would be an appropriate time step for the later control implementation of the trained NN static state-feedback controller.

--> Since the minimum value of an t\_i interval of all the optimal controlled trajectories is extremely small, i.e., about e^-15, the 0.1ms discretization in task 1 might not be enough. Therefore, some constraints on the minimum interval need to be imposed during the optimal trajectories synthesis.

2/ Provided with the training data from task 1, we now train a NN to map state x\_i to the corresponding optimal control input u\_i.

3/ Validation: Plot an optimal controlled trajectory from the synthesis against the resulted controlled trajectory from the feedback implementation of the trained NN. Hopefully, they are vey similar.

**--> This works locally sofar!**

**Plans:**

1/(different dt-interval idea) generating data (synthesizing) remotely for x=-2:0.25:2 and y=-4:0.25:4. Once finish, patch it with previous global data for x=-4:0.25:-2 U x=2:0.25:4 and y=-4:0.25:4. Then train.

**Plans:**

1/ Showed that altering dt (although dt is the same accross whole trajectory) yields a less energy solution than that of the standard iterative method.

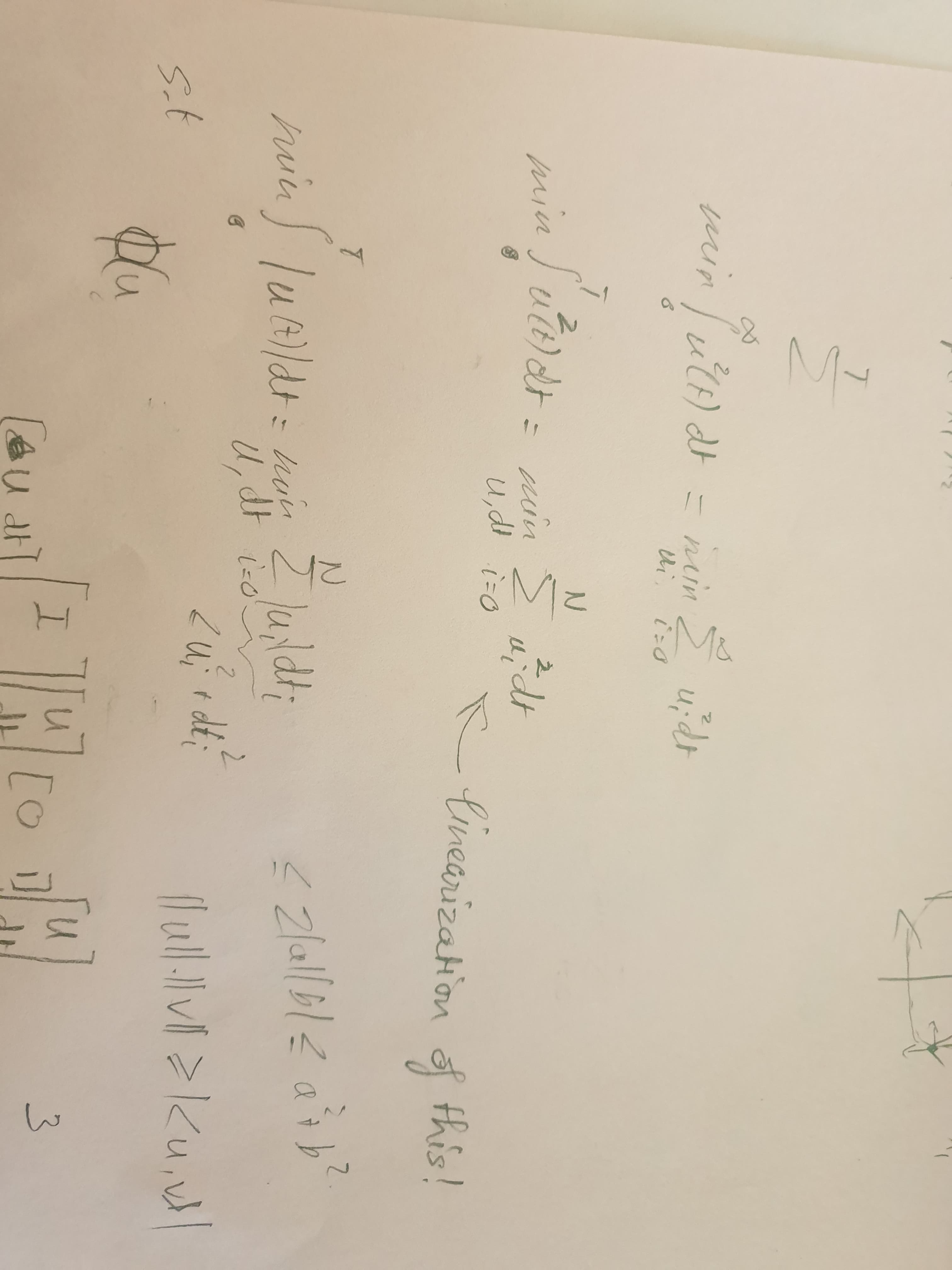
For x0=(-2.00;3.00):

T = 8.00; dt = 0.02; U = 13.9331 --> U\*dt = 0.2787 (iterative: approach the further manifold)

T = 8.00; dt = 0.02; U = 11.7078 --> U\*dt = 0.2342 (altering dt: use the closer manifold)

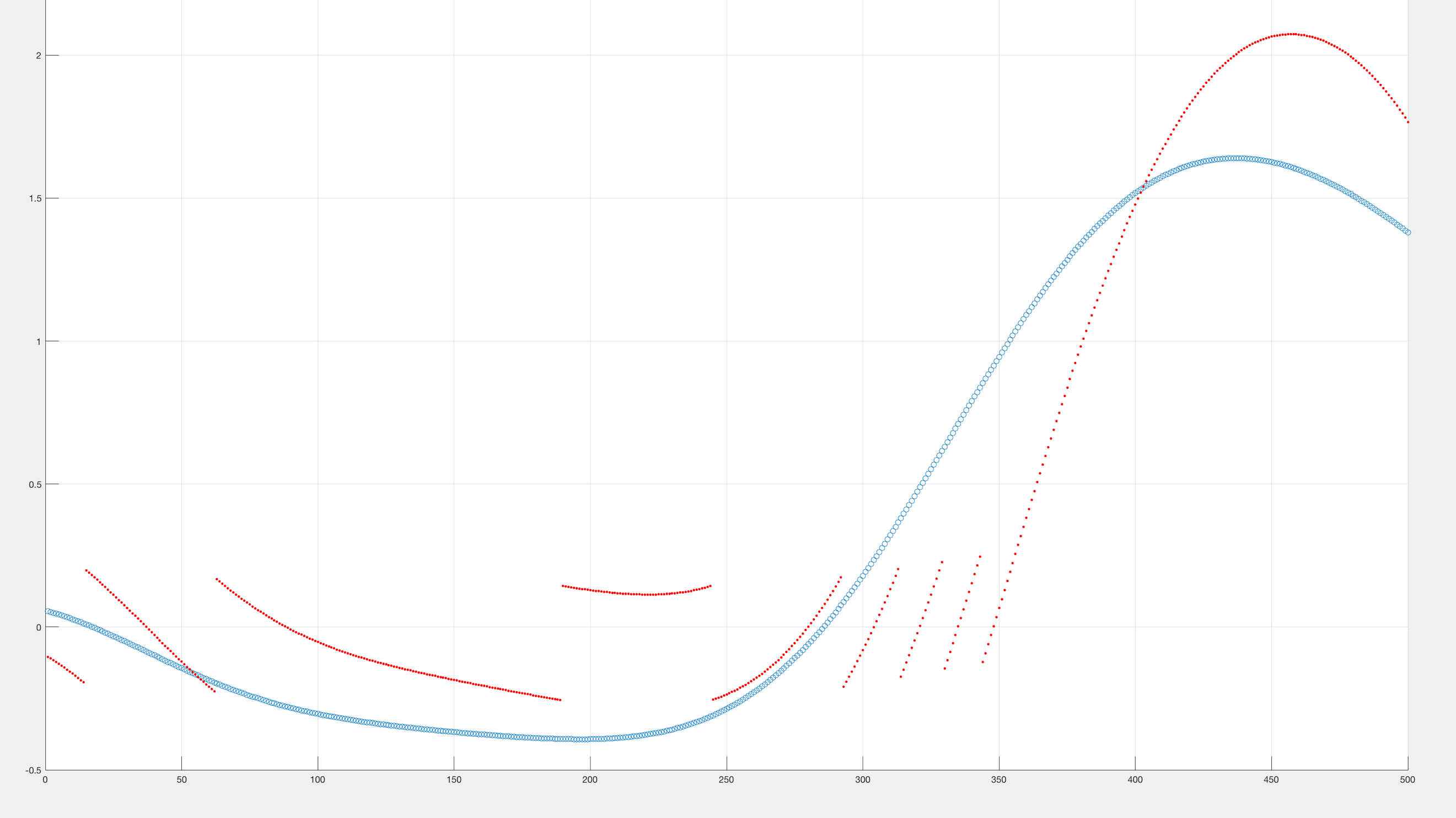
The idea behind this is that more flexibility in time let to less aggressive controls, resulting in less energy. This mimics the infinite time minimum energy in the sense that we now don't have any time-budget thus only focus on finding the control with the minimum norm.

2/ However, only by focusing on finding the minimum norm control, we are actually not solving the minimum energy problem because due to the extension of time (dt is getting bigger) the minimum norm control can easily have a much higher energy (i.e., norm(u)\*dt) than other control options. The more time we allow, the algorithm will try to maximize that horizon only in order to find the minimum norm control instead of finding the minimum energy solution.

3/ The aforementioned problem leads us to a different formulation of the objective. That is 

We usually use 2-norm for minimum energy because 2-norm forces the control signal to overally stay close to 0. One the other hand, not only the trajectory of 1-norm minimum energy consideration is nondifferentiable (due to the non-smooth nature of absolute values), but the control signals are also allowed to vary drastically and can be very large at some points (e.g.,

1-norm([1;2;3])= 1-norm([0;0;6]), as shown |U|\_1:red, |U|\_2:blue.



--> 3/ works pretty well. For an arbitrary point x0 on the controlled trajectory, its optimal controlled trajectory coincide with the first one (see min\_udt.fig). Also, the original performance index (energy cost) of the overall trajectory equals the sum of cost of the 2 parts (i.e., U1^2\*dt +U2^2\*dt = U\_total^2\*dt), confirming the correctness of our analysis.