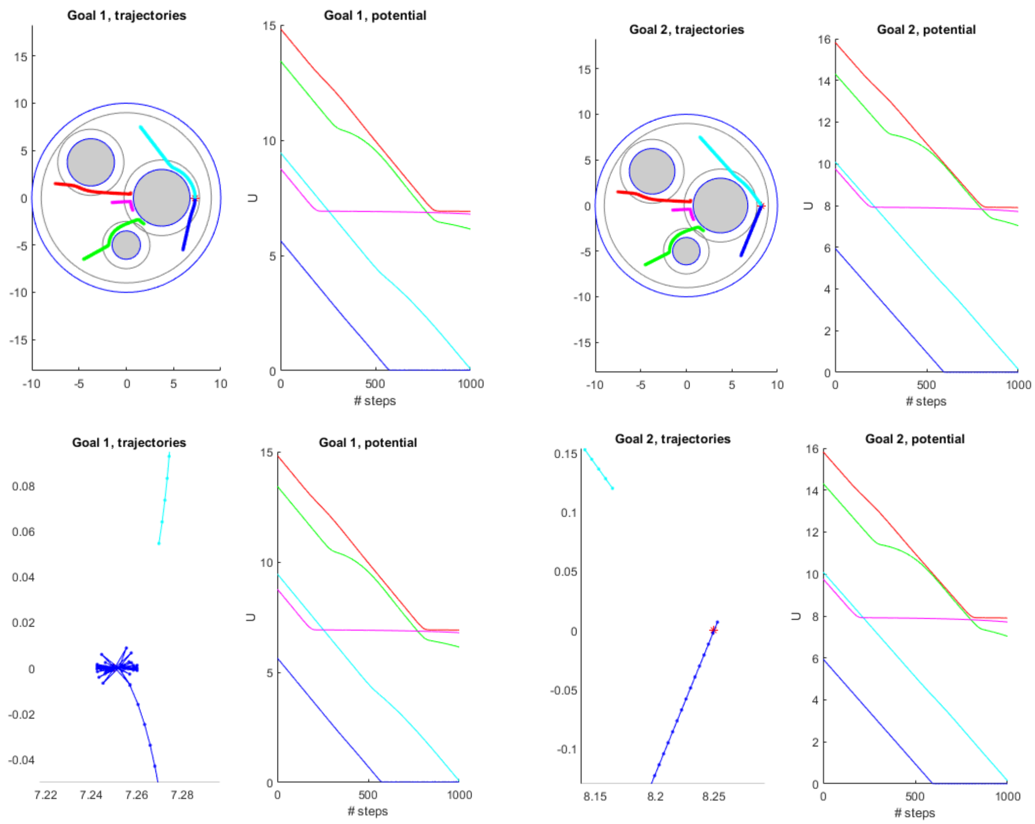
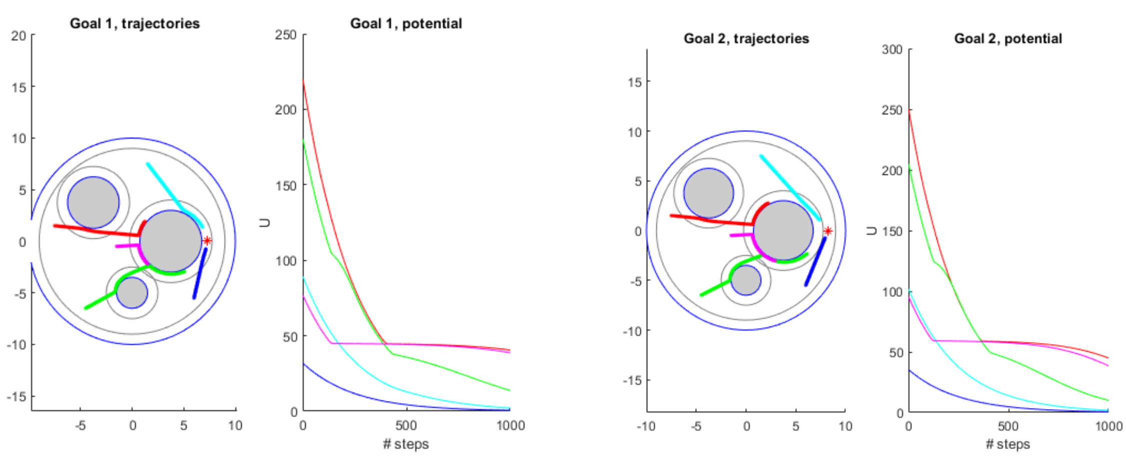
Zachary Weiss
ME570 HW3
Professor Tron
2 November 2020

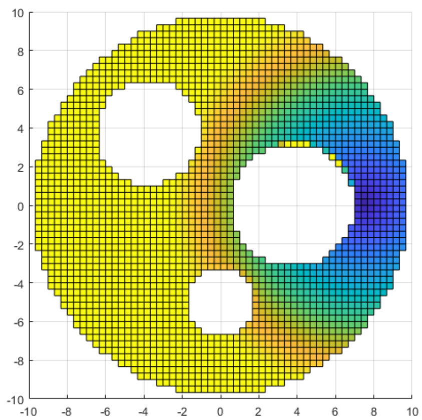
Q2.1:
With a conic shape, an epsilon of 1e-2, and a repulsive weight of 5e-2, we get the following results:

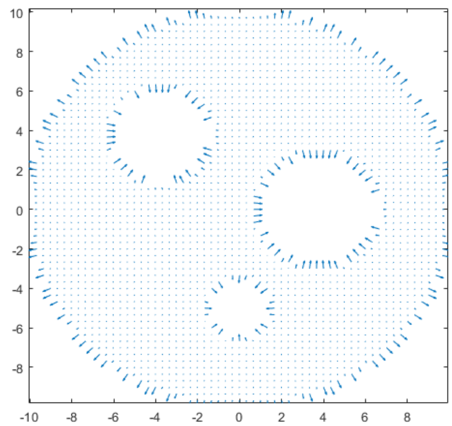


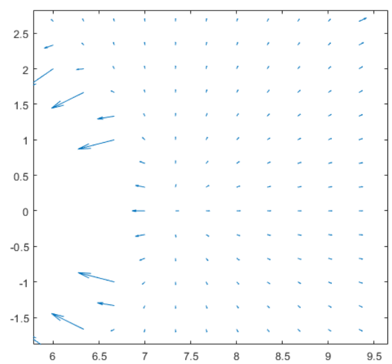
Similarly, for a quadratic shape, epsilon of 1e-3, and a repulsive weight of 5e-2, we get (no convergence in finite time, but approaches ever closer:



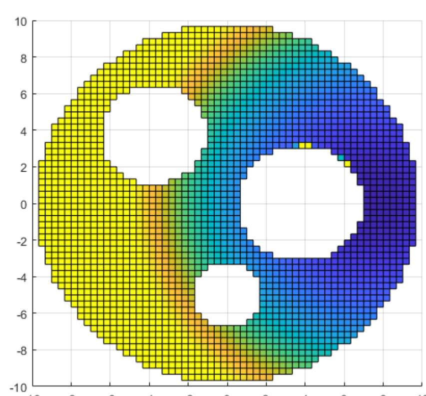
Q2.2:
With a conic shape, at a repulsiveWeight of 5e-2, we get the potential field and gradient field as follows:

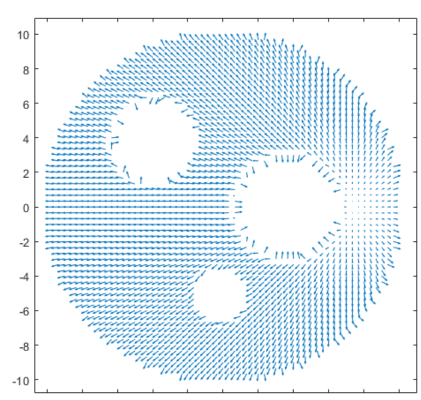






As the gradient vectorfield is hard to see due to scaling at edges of obstacles, the third image is a zoom of the vectorfield about the goal location. Similarly, the potential and gradient of the potential for the quadratic shape at the same repulsiveWeight is as follows (with the threshold for the potential map increased from the default of 10 to 150, for more meaningful visualization):





Q2.3:
With epsilon, the balance to be struck is between significantly over/undershooting due to a large timestep, or taking computationally longer (and similarly, more steps to convergence) with a small epsilon. As one varies repulsiveWeight, epsilon must similarly be modified so as to account for the first factor: over/undershoots, with overshoots being the more disastrous, potentially resulting in impossible / illegal moves (i.e., into an obstacle, before the repulsive influence has a chance to take significant effect on the gradient/potential). When varying repulsiveWeight, one also must be cognizant of the possibility of the creation of local minima, in which case a gradient-based planner will not (necessarily, dependent on initial conditions) converge to the goal.

Q2.4:
If the planner has succeeded (within the allotted number of steps), the potential U will be zero at the end of the iterations, whereas if it converges to another value greater than zero, it has fallen into a local minima (or requires more iterations to reach the goal, if what appears to be convergence is simply truncation / a change in slope).

Q2.5:
With the two goals in the dataset, one is within the distance of influence of an obstacle, whereas another is not. Dependent on the repulsiveWeight and shape, the goal within the range of influence may be unreachable. Generally, if the goal is reachable, in the conic-shaped planner, the goal is reached in finite time (within the limitations of stepsize, as such in practical cases, it oscillates about the goal once "there" due to overshoot). With a quadratic-shaped planner, the goal is converged to but never reached within finite time, due to the smoothly-decreasing gradient to zero as it nears the goal (not discontinuous at the goal).

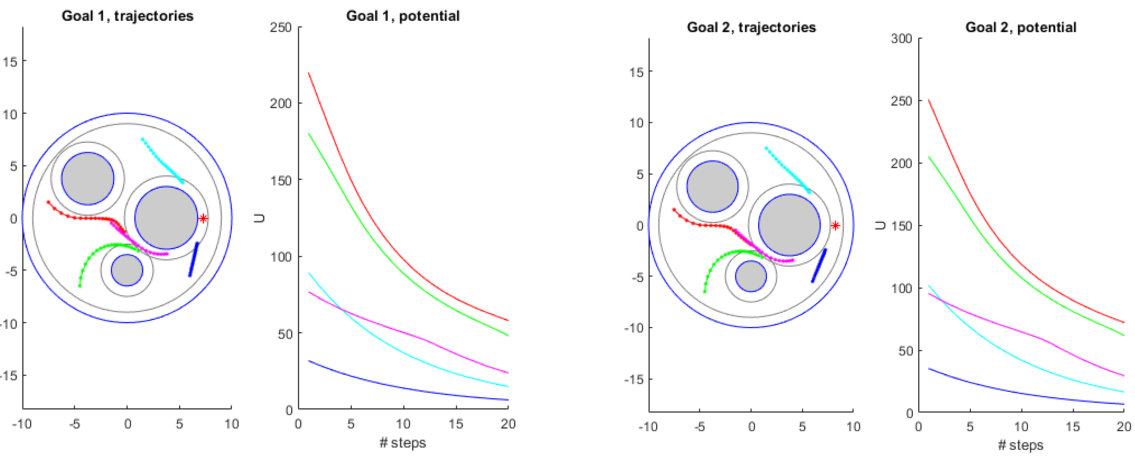
Q3.1:
The QP will have the form:

$$u^*(x) = \underset{u \in \mathbb{R}^2, \delta \in \mathbb{R}}{\operatorname{argmin}} \|u\|^2 + m\delta^2$$

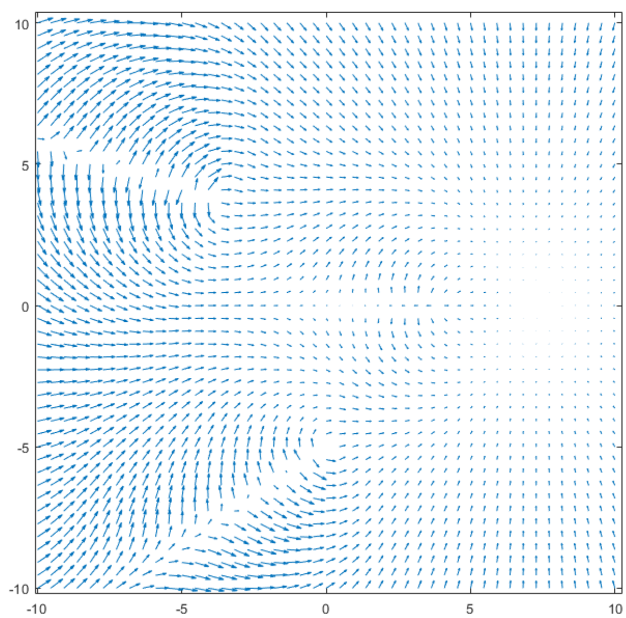
Subject to:

$$(\nabla \mathrm{U}_{Attr})u + \mathrm{U}_{Attr} \leq \delta$$
$$\big(-\nabla \mathrm{d}_i(x)\big)u - \mathrm{d}_i(x) \leq 0$$

Q3.2: CLF-CBF with an epsilon of 0.1 and a repulsiveWeight of 0.1, NSteps = 20 (given more, converges, just takes a long time computationally).



Q3.3:
CLF-CBF field on a 45x45 grid, for a repulsiveWeight of 0.1 (same as above in Q3.2).



Q3.4:
A key tradeoff observed in just producing the above images is the computational intensity of a CLF-CBF approach. Luckily, there are “tricks” one can use to minimize that tradeoff, such as passing a smaller grid (for the visualization of the field), as well as the fact that the CLF-CBF method enables higher epsilons than gradient methods. A key potential benefit of CLF-CBF is that it, by definition, minimizes control “effort” / input, whereas gradient methods follow wherever the gradient slope takes them without (direct) regard for input or distance metrics.

Q4.1:

We know from HW2, Q6.1 that $\frac{d}{dt}\left({}^{W}p_{eff}\right) =$

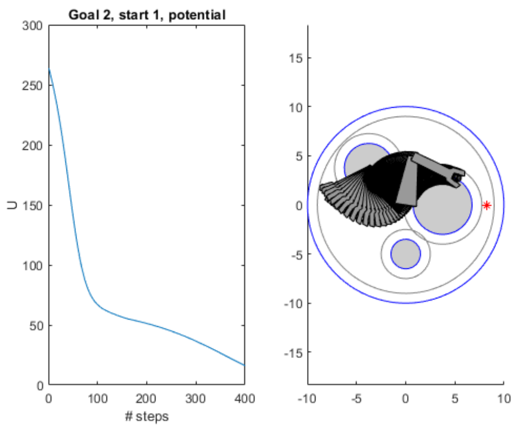
$$\begin{bmatrix} -5\sin(\theta_1)\,\dot{\theta}_1 - 5\cos(\theta_1)\sin(\theta_2)\,\dot{\theta}_1 - 5\cos(\theta_2)\sin(\theta_1)\,\dot{\theta}_1 - 5\cos(\theta_1)\sin(\theta_2)\,\dot{\theta}_2 - 5\cos(\theta_2)\sin(\theta_1)\dot{\theta}_2 \\ 5\cos(\theta_1)\,\dot{\theta}_1 + 5\cos(\theta_1)\cos(\theta_2)\,\dot{\theta}_1 + 5\cos(\theta_1)\cos(\theta_2)\,\dot{\theta}_2 - 5\sin(\theta_1)\sin(\theta_2)\,\dot{\theta}_1 - 5\sin(\theta_1)\sin(\theta_2)\dot{\theta}_2 \end{bmatrix}$$

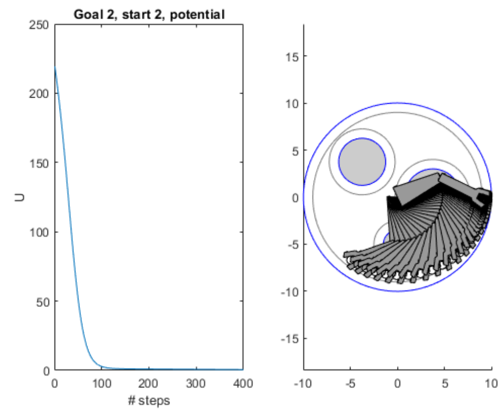
As such, to find J such that $\frac{d}{dt}\left({}^{W}p_{eff}\right) = J(\theta)\dot{\theta}$, we simply must factor out the $\dot{\theta}$ vector, yielding the 2-by-2 matrix:

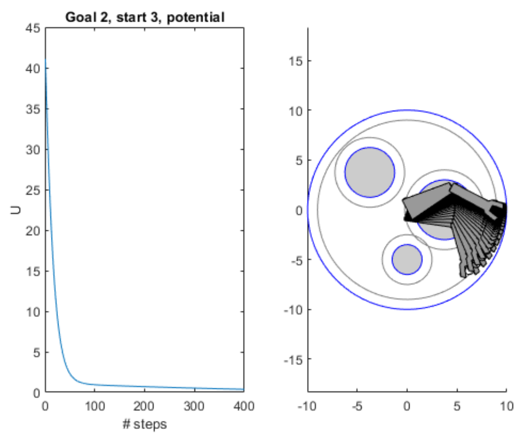
$$J(\theta) = \begin{bmatrix} -5\sin(\theta_1) - 5\cos(\theta_1)\sin(\theta_2) - 5\cos(\theta_2)\sin(\theta_1) & -5\cos(\theta_1)\sin(\theta_2) - 5\cos(\theta_2)\sin(\theta_1) \\ 5\cos(\theta_1) + 5\cos(\theta_1)\cos(\theta_2) - 5\sin(\theta_1)\sin(\theta_2) & 5\cos(\theta_1)\cos(\theta_2) - 5\sin(\theta_1)\sin(\theta_2) \end{bmatrix}$$

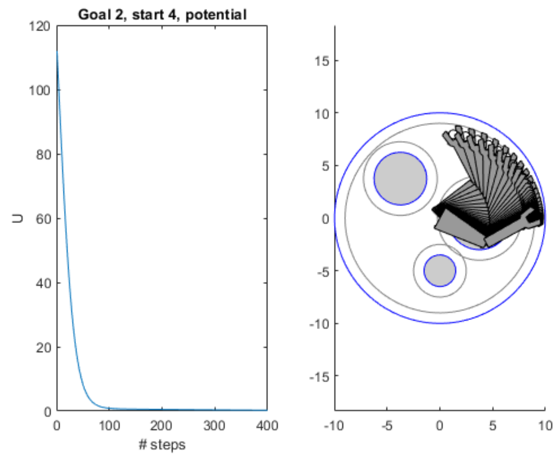
Q4.2:

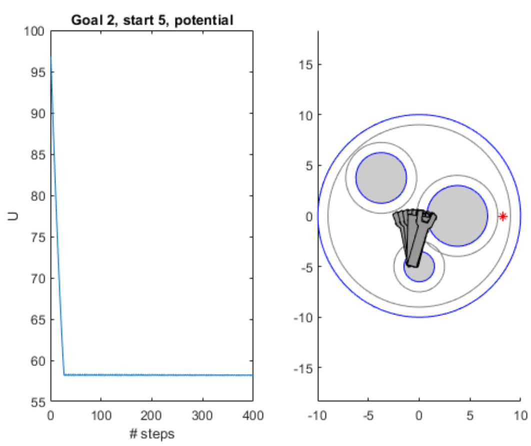
There did not seem to be a combination of epsilon and repulsiveWeight settings that made start position #5 feasible (within approx. NSteps = 400, given infinite time it would 'unstick', minor rotation CCW seen in the period allotted), but epsilons in the range of 2e-4 to 1e-3 and repulsiveWeights in the range of 5e-3 to 1e-1 worked for the other start positions well. At an epsilon of 2e-4, and a repulsive weight of 1e-2, the following results were obtained (placed on following page for ease of comparison):











Q5.1:
It’s harder to estimate the total time spent on this homework as I spread it across many sessions of work, but all total it was somewhere around a (waking / working) days’ worth of time.