Project of Optimization and Algorithms

João Xavier
Instituto Superior Técnico
September 2022

Contents

1 Tracking a moving target

We want to control a vehicle so that it tracks a given moving target over a finite discrete-time horizon $\{1, 2, 3, \ldots, T\}$.

1.1 Deterministic target

Target. The trajectory of the target is assumed known from the beginning of the horizon. The trajectory is denoted by $q(t) \in \mathbf{R}^2$ for $1 \leq t \leq T$ and is shown in Figure 1.

The state of our vehicle. The position of our vehicle at time t is denoted by $p(t) \in \mathbf{R}^2$; its velocity, by $v(t) \in \mathbf{R}^2$. The *state* $x(t)$ of our vehicle at time t is

$$x(t) = \begin{bmatrix} p(t) \\ v(t) \end{bmatrix}. \tag{1}$$

Note that $x(t)$ is a four-dimensional vector: $x(t) \in \mathbf{R}^4$, for $1 \leq t \leq T$.

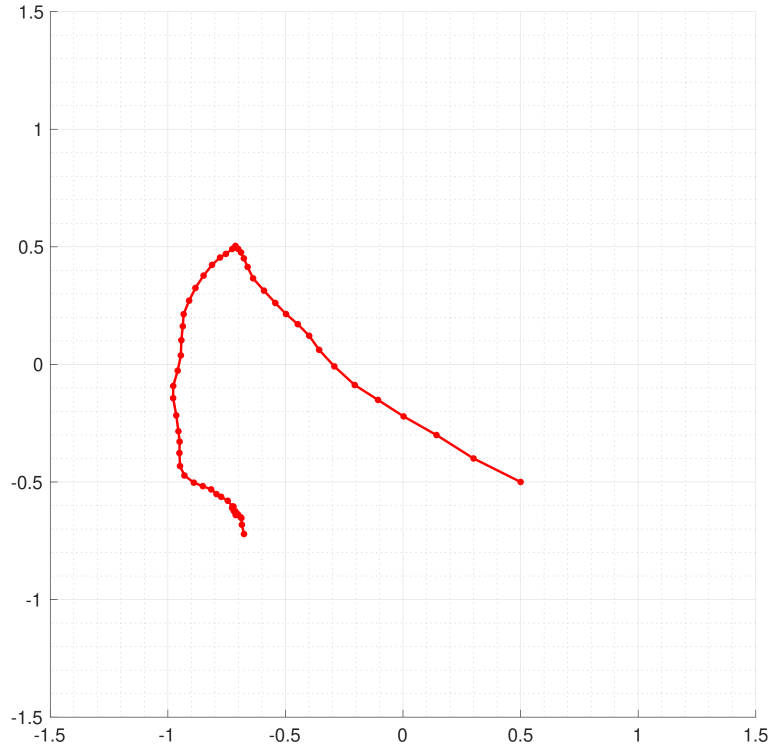

Figure 1: Trajectory of the target, which starts at $(0.5, -0.5)$, for an horizon of $T = 60$ time steps.

The control signal. We act on our vehicle by applying a force at each time t in $\{1, \ldots, T\}$. (These forces are applied in fact by the motor in the vehicle) The force that we apply at time t, denoted by $u(t)$, is two-dimensional, that is, $u(t) \in \mathbf{R}^2$ for $1 \leq t \leq T - 1$. The sequence of forces applied throughout time, $\{u(t) \colon t = 1, \ldots, T-1\}$, is called the *control signal*. This is the signal we want to design.

The control signal changes the state of our vehicle. Each time we apply a force to the vehicle, we push it a bit, thereby changing its state (position and velocity). More precisely, if $x(t)$ is the current state and the force $u(t)$ is applied, then the state changes to

$$x(t+1) = Ax(t) + Bu(t), \tag{2}$$

where $A \in \mathbf{R}^{4\times 4}$ and $B \in \mathbf{R}^{4\times 2}$ are known matrices, depending on physical constants such as the mass of the vehicle and the drag coefficient of the environment. For this project, take

$$A = \begin{bmatrix} 1 & 0 & 0.2 & 0 \\ 0 & 1 & 0 & 0.2 \\ 0 & 0 & 0.8 & 0 \\ 0 & 0 & 0 & 0.8 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0.2 & 0 \\ 0 & 0.2 \end{bmatrix}.$$

The initial state (initial position and initial velocity) at $t = 1$ of our vehicle is given and is denoted by x_{init}. Consider

$$x_{\text{init}} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}.$$

To illustrate, the trajectory of our vehicle is shown in Figure 2 for a control signal $\{u(t)\colon t = 1, \ldots, T-1\}$ generated at random.

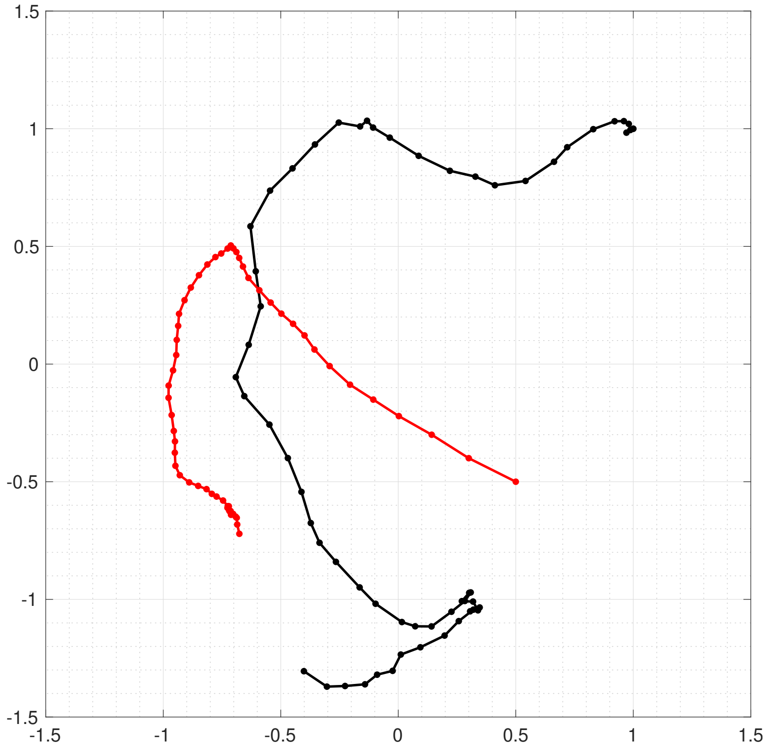

Figure 2: Trajectory of the target (red) and trajectory of our vehicle (black) corresponding to a control signal generated at random.

Our wishes for the control signal. As mentioned earlier, we wish a small tracking error. Thus, we want to design a control signal that makes our vehicle track the target as close as possible, which means that we want $p(t)$ (the position of our vehicle at time t) to be as close as possible to $q(t)$ (the position of the target at time t), for $1 \leq t \leq T$. In this project, we measure the tracking error (TE) as

$$\text{TE} = \sum_{t=1}^{T} \|Ex(t) - q(t)\|_\infty,$$

where $\|\cdot\|_\infty$ is the ℓ_∞ norm (that is, for a vector $z = (z_1, z_2, \ldots, z_d) \in \mathbf{R}^d$, we have $\|z\|_\infty = \max\{|z_1|, |z_2|, \ldots, |z_d|\}$) and

$$E = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}.$$

Note that, given the definition of the state $x(t)$ in (1), the term $Ex(t)$ is just $p(t)$, the position of our vehicle at time t.

We now add another wish: a small control signal. This is because acting on the vehicle consumes resources (ex: fuel). In this project, we measure the control effort (CE) as

$$\text{CE} = \sum_{t=1}^{T-1} \|u(t)\|_2^2,$$

where $\|\cdot\|_2$ is the usual Euclidean ℓ_2 norm (that is, for a vector $z = (z_1, z_2, \ldots, z_d) \in \mathbf{R}^d$, we have $\|z\|_2 = (z_1^2 + z_2^2 + \cdots + z_d^2)^{1/2}$).

The optimization problem. In sum, we have two wishes: a small tracking error TE and a small control effort CE. We now merge these two wishes in a single cost function, arriving at the optimization problem

$$\begin{array}{ll} \underset{x,u}{\text{minimize}} & \underbrace{\sum_{t=1}^{T} \|Ex(t) - q(t)\|_\infty}_{\text{TE}} + \lambda \underbrace{\sum_{t=1}^{T-1} \|u(t)\|_2^2}_{\text{CE}} \\ \text{subject to} & x(1) = x_{\text{initial}} \\ & x(t+1) = Ax(t) + Bu(t), \quad \text{for } 1 \leq t \leq T-1. \end{array} \tag{3}$$

Some comments about this optimization problem:

- the variables to optimize are x and u, where x stands for $\{x(1), x(2), \ldots, x(T)\}$, which is the sequence of states of our vehicle, and u stands for $\{u(1), u(2), \ldots, u(T-1)\}$, which is the control signal we apply to the vehicle. Thus, we are composing the state trajectory and the control signal, jointly. Of course, these variables are linked by the physics of the problem, which is expressed by the last constraint in (3);

- The two wishes, TE and CE, are merged additively in the cost function of (3) by a weight $\lambda > 0$. Increasing λ increases the importance of CE relative to TE, and vice-versa.

Task 1. [Numerical task] Use the software `CVX` from `http://cvxr.com/cvx` to solve problem (3) for various values of λ. Specifically, solve 9 instances of problem (3), defined as follows:

instance i	λ_i
1	10
2	5
3	1
4	0.5
5	0.1
6	0.05
7	0.01
8	0.005
9	0.001

After you solve instance i, plot the trajectory of the target against the trajectory of the vehicle. Also, take note of the tracking error and control effort, which we denote by $\mathrm{TE}(\lambda_i)$ and $\mathrm{CE}(\lambda_i)$. After you solve all instances, plot $\mathrm{TE}(\lambda_i)$ versus $\mathrm{CE}(\lambda_i)$, for $1 \leq i \leq 9$. Comment all your results, that is, explain why they make sense.

The trajectory of the target can be found in the Matlab file `target_1.mat`. Note: For squared-Euclidean norm terms such as $\|\cdot\|_2^2$, the function `square_pos` of CVX is useful.

Example. So that you can have some examples to check your code, we plot the trajectory of the target against the trajectory of the vehicle for instance $i = 5$, that is, for $\lambda = 0.1$) in Figure 3.

We also give the pairs $(\mathrm{TE}(\lambda_i), \mathrm{CE}(\lambda_i))$ for the first and last instance: $(\mathrm{TE}(\lambda_1), \mathrm{CE}(\lambda_1)) = (41.41, 2.98)$ and $(\mathrm{TE}(\lambda_9), \mathrm{CE}(\lambda_9)) = (3.55, 582.92)$, which are plotted in Figure 4.

Task 2. [Theoretical task] Let $\mathrm{TE}(\lambda)$ and $\mathrm{CE}(\lambda)$ denote the tracking error and control effort obtained after optimizing (3) for a given $\lambda > 0$. Consider now two values of λ, say, λ_a and λ_b, and suppose that $\mathrm{TE}(\lambda_a) \leq \mathrm{TE}(\lambda_b)$. Prove that $\mathrm{CE}(\lambda_a) \geq \mathrm{CE}(\lambda_b)$.

Task 3. [Theoretical task] Show that that the optimization problem (3) has a unique solution. Hint: rewrite (3) as an unconstrained optimization problem that depends only on the variable u.

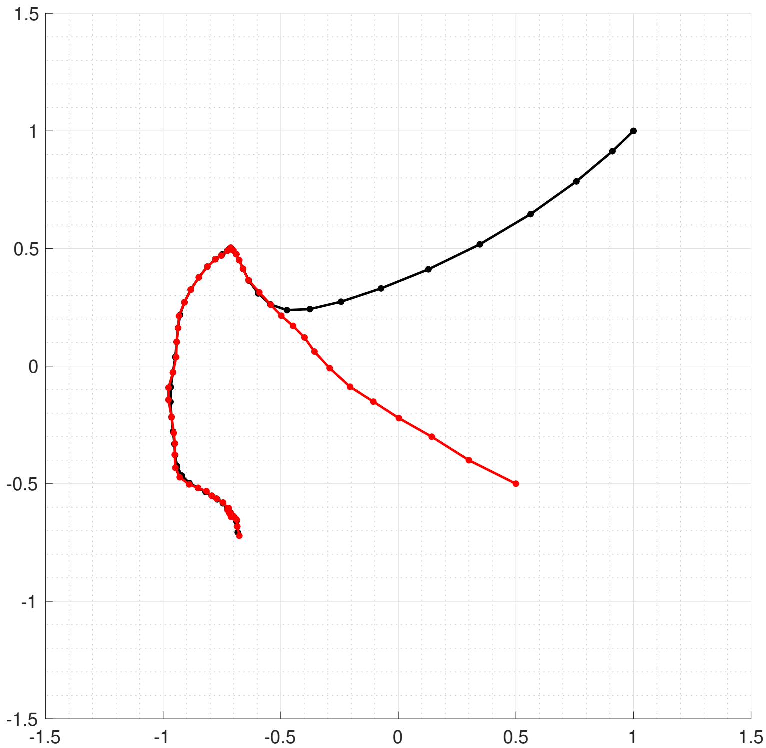

Figure 3: Trajectory of the target (red) and trajectory of our vehicle (black) corresponding to a control signal optimized as in (3) for $\lambda = 0.1$.

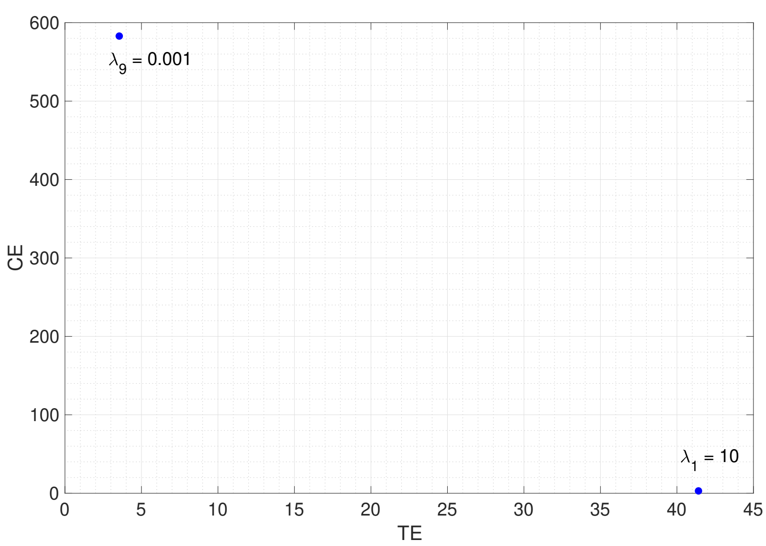

Figure 4: Pairs $(\mathrm{TE}(\lambda_i), \mathrm{CE}(\lambda_i))$ for $\lambda_1 = 10$ and $\lambda_9 = 0.001$. Note that Task 1 asks you to complete this figure by adding the remaining pairs for $i \in \{2, 3, \ldots, 8\}$.

1.2 Random target with no information

Up to now, the trajectory of the target was assumed deterministic. We consider now to a scenario that is more challenging, where the trajectory is random. Specifically, we know only that the target can execute two possible trajectories, but we ignore which one the target is going to execute in the time horizon $\{1, \ldots, T\}$. We know, however, the prior probabilities of the target executing each trajectory, that is, how likely the target is of executing each of the two possible trajectories.

We denote by p_1 the probability that the target executes trajectory 1 and by p_2 the probability that the target executes trajectory 2. Of course, being probabilities, the given constants p_1 and p_2 are non-negative and sum to one: $p_1 \geq 0$, $p_2 \geq 0$, and $p_1 + p_2 = 1$. As a curiosity, note that the special case $p_1 = 1$ and $p_2 = 0$ (respectively, the case $p_1 = 0$ and $p_2 = 1$) corresponds in fact to a deterministic target that executes always trajectory 1 (respectively, trajectory 2).

The two possible trajectories are denoted by $q_1(t) \in \mathbf{R}^2$ (trajectory 1) and by $q_2(t) \in \mathbf{R}^2$ (trajectory 2), for $1 \leq t \leq T$. These two trajectories, along with a trajectory of the vehicle for a randomly generated control signal, are shown in Figure 5.

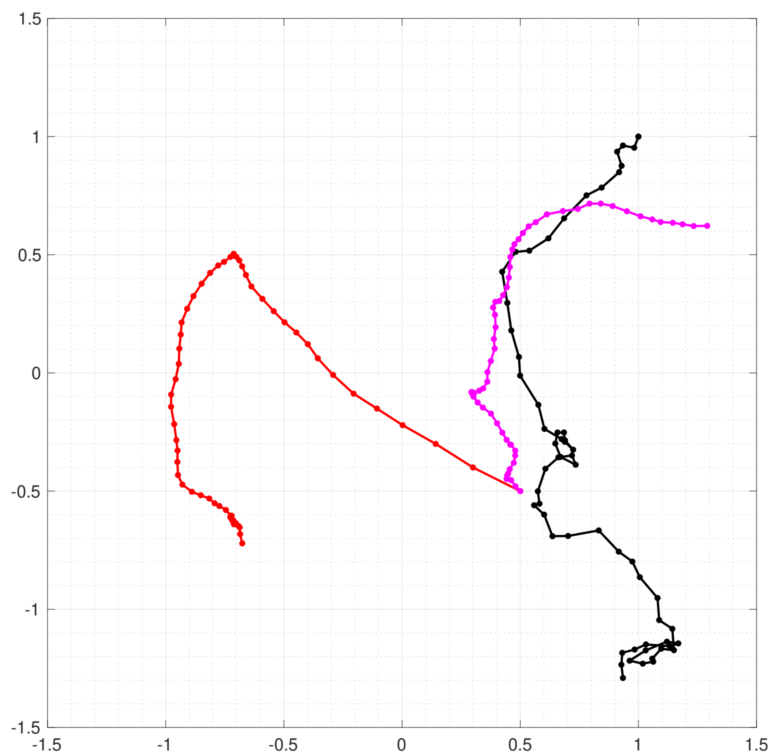

Figure 5: The two known possible trajectories that the target can execute, trajectory 1 (red) and trajectory 2 (magenta), and a trajectory of our vehicle (black) corresponding to a control signal generated at random.

The optimization problem. As mentioned, the probabilities p_1 and p_2 and the possible trajectories $q_1(t)$ and $q_2(t)$ are known to us. What we ignore is which one of the two possible trajectories the target is actually going to execute from $t = 1$ to $t = T$. To cope with this uncertainty, we choose to design a control signal that minimizes the average tracking error plus the control effort:

$$\begin{array}{ll} \underset{x,u}{\text{minimize}} & \underbrace{p_1 \underbrace{\sum_{t=1}^{T} \|Ex(t) - q_1(t)\|_\infty}_{\text{TE}_1} + p_2 \underbrace{\sum_{t=1}^{T} \|Ex(t) - q_2(t)\|_\infty}_{\text{TE}_2}}_{\text{average TE}} + \lambda \underbrace{\sum_{t=1}^{T-1} \|u(t)\|_2^2}_{\text{CE}} \\ \text{subject to} & x(1) = x_{\text{initial}} \\ & x(t+1) = Ax(t) + Bu(t). \quad \text{for } 1 \leq t \leq T-1. \end{array} \tag{4}$$

In (4), the terms TE_1 and TE_2 represent the mismatches between the trajectory of the vehicle and each of the possible trajectories of the target. Thus, the term $p_1\text{TE}_1 + p_2\text{TE}_2$ in the cost function represents the average tracking error.

Task 4. [Numerical task] Use the software `CVX` from `http://cvxr.com/cvx` to solve problem (4) for various values of p_1 and p_2, always taking $\lambda = 0.5$. Specifically, solve 6 instances of problem (3), defined as follows:

instance i	$(p_1)_i$	$(p_2)_i$
1	0	1
2	0.2	0.8
3	0.4	0.6
4	0.6	0.4
5	0.8	0.2
6	1	0

After you solve instance i, plot the two possible trajectories of the target against the trajectory of the vehicle. Comment your results.
Trajectory 1 of the target can be found in the Matlab file `target_1.mat`, and trajectory of the target can be found in the Matlab file `target_2.mat`.

Example. So you can check your code we plot the result of the instance $p_1 = 0.6$ and $p_2 = 0.4$ in Figure 6.

1.3 Random target with midway information

We keep the scenario described in the previous section 1.2 and add a twist: we assume that the trajectory—either 1 or 2—that the target executes from time $t = 1$ is revealed to us at

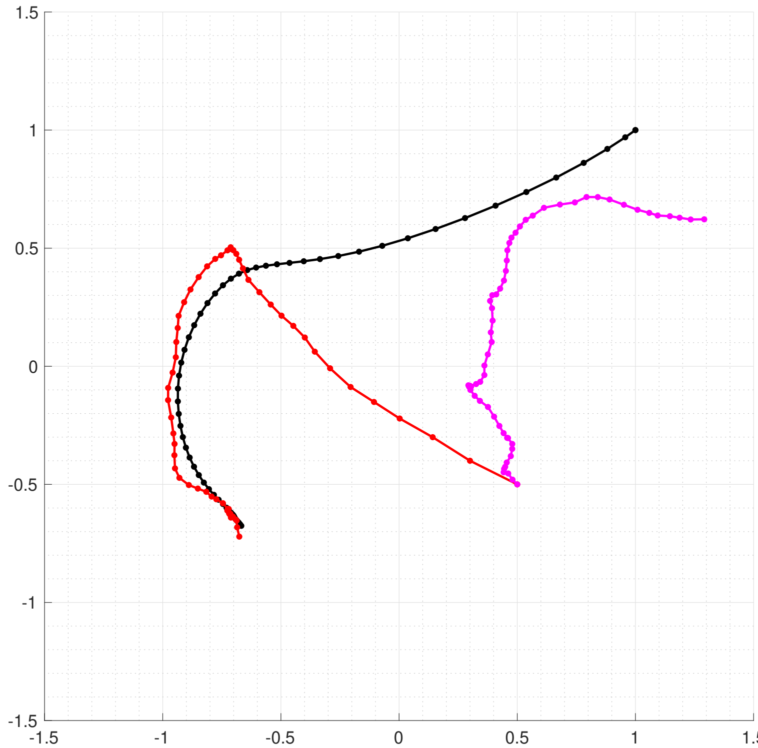

Figure 6: The two known possible trajectories that the target can execute, trajectory 1 (red) and trajectory 2 (magenta), and a trajectory of our vehicle (black) corresponding to a control signal optimized as in (4) for $p_1 = 0.6$ and $p_2 = 0.4$.

$t = 35$. That is, from $t = 1$ to $t = 34$ we ignore which trajectory the target is executing; we know only their prior probabilities. But, about midway in the time horizon, at $t = 35$, we are told which trajectory the target has actually chosen to execute since $t = 1$ (and will keep executing until the end of the time horizon).

The optimization problem. As a starting point to map this scenario into an optimization problem, consider the following incomplete formulation:

$$\begin{array}{ll} \underset{x_1,u_1,x_2,u_2}{\text{minimize}} & \sum_{k=1}^{K} p_k \left(\sum_{t=1}^{T} \|Ex_k(t) - q_k(t)\|_\infty + \lambda \sum_{t=1}^{T-1} \|u_k(t)\|_2^2 \right) \\ \text{subject to} & x_1(1) = x_{\text{initial}} \\ & x_1(t+1) = Ax_1(t) + Bu_1(t) \quad \text{for } 1 \leq t \leq T-1 \\ & x_2(1) = x_{\text{initial}} \\ & x_2(t+1) = Ax_2(t) + Bu_2(t) \quad \text{for } 1 \leq t \leq T-1 \\ & \boxed{\text{more constraints needed here}}. \end{array} \tag{5}$$

In problem (5) we have two kinds of variables: x_1, u_1 and x_2, u_2. The variable x_k stands for $\{x_k(1), x_k(2), \ldots, x_k(T)\}$, which is the sequence of states of the vehicle resulting from the application of the control sequence u_k, which stands for $\{u_k(1), u_k(2), \ldots, u_k(T-1)\}$.

Task 5. [Theoretical task] Explain why more constraints are indeed needed in problem (5). Or, put another way, what's wrong with the formulation in (5) if we don't add further constraints?

Task 6. [Theoretical task] State which extra constraints need to be added in the indicated slot in formulation (5). (You cannot add more variables; you cannot change the cost function.) Hint: note that $u_k(t)$ from $t = 35$ onwards is the control signal we apply after being told that the target is executing trajectory k.

Task 7. [Numerical task] Use the software `CVX` from `http://cvxr.com/cvx` to solve the optimization problem resulting from you answer to Task 6 for the case $\lambda = 0.5$, $p_1 = 0.6$ and $p_2 = 0.4$. After the problem is numerically solved, plot the two possible trajectories of the target against the two possible trajectories of the vehicle (which correspond to x_1 and x_2).

Examples. We give the solution for the case $\lambda = 5$, $p_1 = 0.6$ and $p_2 = 0.4$ in Figure 7. Additionally, the case $\lambda = 0.05$, $p_1 = 0.6$ and $p_2 = 0.4$ is shown in Figure 8.

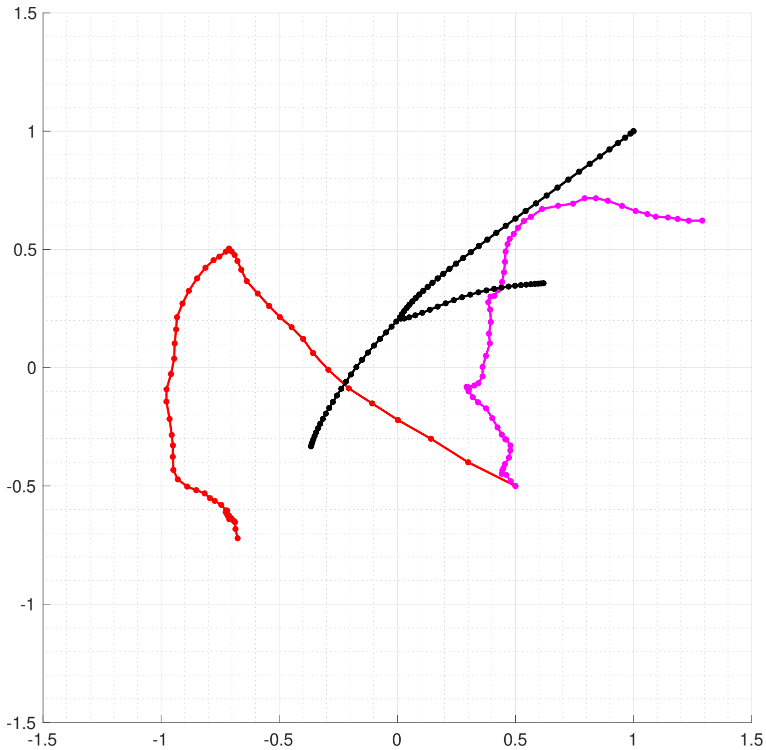

Figure 7: The two known possible trajectories that the target can execute, trajectory 1 (red) and trajectory 2 (magenta), and the two trajectories of our vehicle (black) corresponding to the control signals optimized as in (5) for $\lambda = 5$, $p_1 = 0.6$ and $p_2 = 0.4$. The trajectories x_1 and x_2 split at $t = 35$.

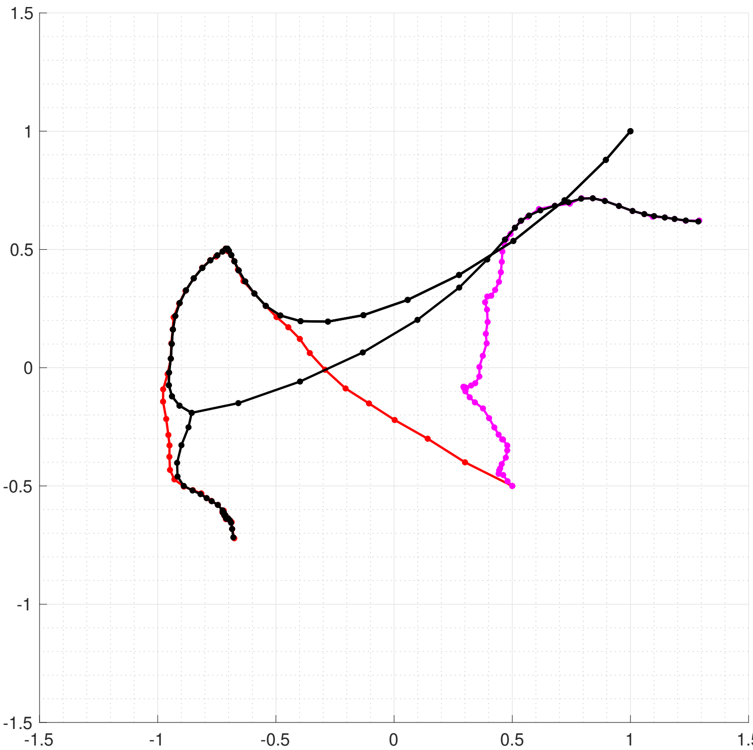

Figure 8: The two known possible trajectories that the target can execute, trajectory 1 (red) and trajectory 2 (magenta), and the two trajectories of our vehicle (black) corresponding to the control signals optimized as in (5) for $\lambda = 0.05$, $p_1 = 0.6$ and $p_2 = 0.4$. The trajectories x_1 and x_2 split at $t = 35$.

2 Estimating the trajectory of a target from noisy range measurements

Consider a target that moves with uniform velocity,

$$q(t) = p + tv,$$

where $q(t) \in \mathbf{R}^2$ is the position of the target at time $t \in \mathcal{T} = \{0, 0.1, 0.2, 0.3, \ldots, 4.8, 4.9, 5\}$. The trajectory of the target is parameterized by the initial position $p \in \mathbf{R}^2$ and the velocity $v \in \mathbf{R}^2$. Both q_0 and v are unknown to us and are the parameters we want to estimate.

2.1 The available data

To estimate these parameters, we exploit measurements collected from two sensors. Specifically, we have two sensors placed at $s_1 \in \mathbf{R}^2$ and $s_2 \in \mathbf{R}^2$, each of which measures its distance to the target, at each time $t \in \mathcal{T}$.

So, we have available measurements—denoted $r_1(t) \in \mathbf{R}$ and $r_2(t) \in \mathbf{R}$—for each $t \in \mathcal{T}$, with $r_1(t)$ reported by sensor 1 and $r_2(t)$ reported by sensor 2. These are called range measurements because they inform about the distance between each sensor and the target. Mathematically, they are modelled as

$$\begin{aligned} r_1(t) &= \| \underbrace{(p + tv)}_{q(t)} - s_1 \|_2 + \text{noise} && (6) \\ r_2(t) &= \| \underbrace{(p + tv)}_{q(t)} - s_2 \|_2 + \text{noise}, && (7) \end{aligned}$$

where "noise" represents a random fluctuation around zero, to account for the real-world imperfection of sensors.

2.2 The optimization problem

To estimate the unknown parameters p and v from the available measurements $r_1(t)$ and $r_2(t)$, we formulate the problem

$$\underset{p,v}{\text{minimize}} \quad \underbrace{\sum_{t \in \mathcal{T}} \left(\|(p + tv) - s_1\|_2 - r_1(t)\right)^2 + \left(\|(p + tv) - s_2\|_2 - r_2(t)\right)^2}_{f(p,v)}. \tag{8}$$

In problem (8), we search for the parameters p and v that best explain the measurements, in a least-squares sense: note that, for a given p and v, the cost function $f(p, v)$ returns the mismatch between the measurements predicted by the model (6) and (7) (in the absence of noise) and the actual measurements $r_1(t)$ and $r_2(t)$; thus, by minimizing f we find the choice of p and v that best matches the available data.

Task 8. [Numerical task] Use the Levenberg–Marquardt (LM) method, described in the slides of the course, to solve the optimization problem (8), starting your iterations from the initial point $p = (-1, 0)$ and $v = (0, 1)$ and stopping when the norm of the gradient of f falls below $\epsilon = 10^{-6}$. Also, start with the parameter $\lambda_0 = 1$. The sensors are placed at $s_1 = (0, -1)$ and $s_2 = (1, 5)$. Give the final estimate of p and v, and plot the cost function and the norm of the gradient across iterations.

The measurements r_1 and r_2 can be found in the Matlab file `measurements.mat`.

Example. So that you can check your code, we give the cost function and the norm of the gradient across iterations in Figures 9 and 10, for the initial point $p = (0, 0)$ and $v = (1, 0)$.

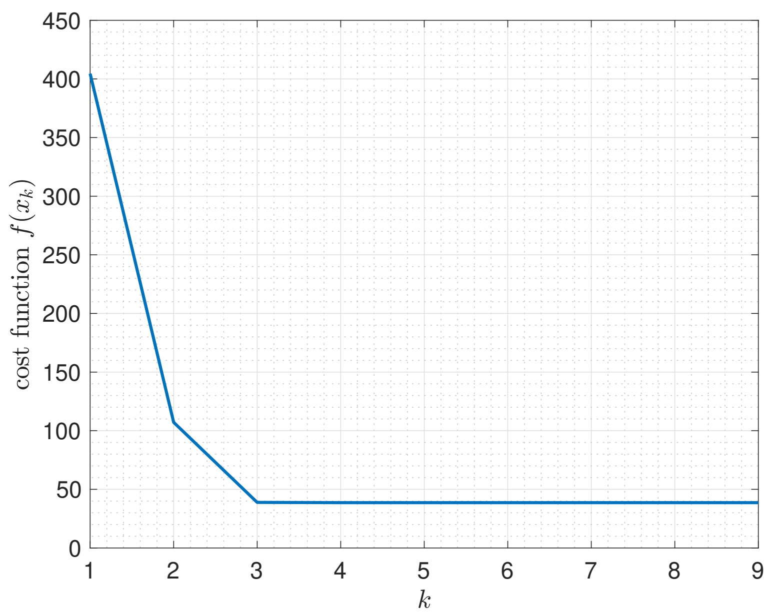

Figure 9: Cost function $f(x_k)$ across iterations k of the LM method, for the initial point $p = (0, 0)$ and $v = (1, 0)$. The symbol $x_k \in \mathbf{R}^4$ is defined as $x_k = (p_k, v_k)$.

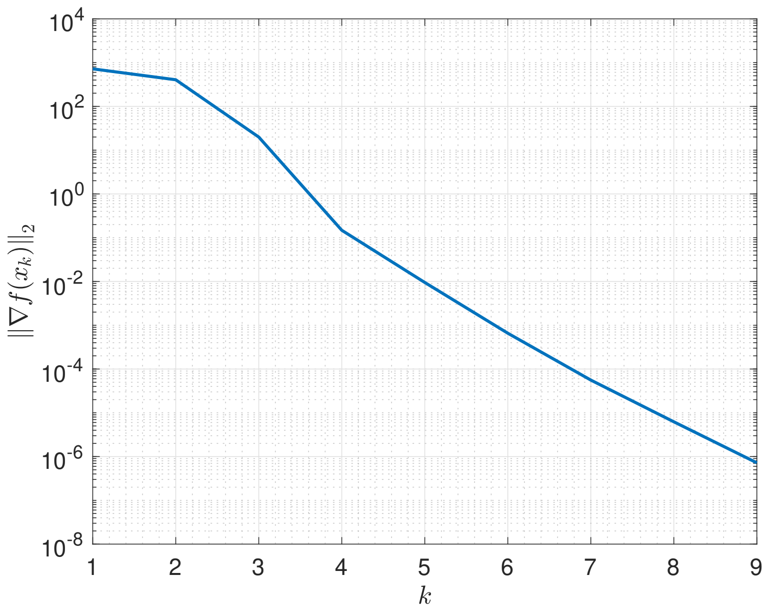

Figure 10: Norm of the gradient $\nabla f(x_k)$ across iterations k of the LM method, for the initial point $p = (0, 0)$ and $v = (1, 0)$. The symbol $x_k \in \mathbf{R}^4$ is defined as $x_k = (p_k, v_k)$.