Optimization and Algorithms
Part 1 of the Project

João Xavier

Instituto Superior Técnico

October 2019

Contents

1 Transferring a robot

In the first part of the project, we play with the following basic question: How should we act on a robot so that it transfers from a given position to another desired position?

We solve three separate variations of this basic question:

- (section 2) how should we act on the robot—using simple control signals—so that the robot also passes, as close as possible, to certain intermediate *points* along its journey?

- (section 3) how should we act on the robot—using simple control signals—so that the robot also passes, as close as possible, to certain intermediate *regions* along its journey?

- (section 4) how should we act on the robot so that it also passes *exactly* on as many intermediate points as possible?

2 Variation A: approximate intermediate points and simple control signals

Section 2.1 states the problem in words, and each section 2.2, 2.3, and 2.4 explores a different way to formulate it as an optimization problem.

2.1 Problem statement

We are interested in the behaviour of the robot over the finite time horizon $\{0, 1, 2, \ldots, T\}$.

Where is our robot? The state of the robot. The position of the robot at time t in $\{0, 1, \ldots, T\}$ is denoted by $p(t)$; its velocity, by $v(t)$. Because we consider a robot moving in a plane, the position and velocity vectors are two-dimensional, that is, $p(t) \in \mathbf{R}^2$ and $v(t) \in \mathbf{R}^2$ for $0 \leq t \leq T$. The *state* $x(t)$ of the robot at time t is

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

Note that $x(t)$ is a four-dimensional vector: $x(t) \in \mathbf{R}^4$, for $0 \leq t \leq T$.

How do we act on the robot? The control signal. We act on the robot by applying a force at each time t in $\{0, 1, \ldots, T-1\}$. (In practice, these forces are applied by the motor in the robot.) The force that we apply at time t, denoted by $u(t)$, is two-dimensional, that is, $u(t) \in \mathbf{R}^2$ for $0 \leq t \leq T-1$. The sequence of forces applied throughout time, $\{u(t) \colon t = 0, \ldots, T-1\}$, is the *control signal.*

How does the robot move? The dynamics of the robot. Each time we apply a force to the robot, thereby pushing it, we change its state (position and velocity). More precisely, if $x(t)$ is the state of the robot at time t and we apply the force $u(t)$ at time t, then the state of the robot changes to

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

where $A \in \mathbf{R}^{4 \times 4}$ and $B \in \mathbf{R}^{4 \times 2}$ are known matrices. These matrices depend on physical constants such as the mass of the robot and the drag coefficient of the environment (to know more details on how to obtain A and B, see slides 17—24 in Part 1, "The art of formulating optimization problems"). For this project, consider

$$A = \begin{bmatrix} 1 & 0 & 0.1 & 0 \\ 0 & 1 & 0 & 0.1 \\ 0 & 0 & 0.9 & 0 \\ 0 & 0 & 0 & 0.9 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}.$$

What do we want the robot to do? We have several wishes:

- **Wish 1: transfer.** We wish the robot, which is resting at $t = 0$ in a given initial position $p_{\text{initial}} \in \mathbf{R}^2$, to transfer to a desired position at time $t = T$, denoted $p_{\text{final}} \in \mathbf{R}^2$, and to rest there. That is, the initial state $x(0)$ of the robot is given by $x(0) = x_{\text{initial}}$, where $x_{\text{initial}} = (p_{\text{initial}}, 0) \in \mathbf{R}^4$, and we wish the final state $x(T)$ to be $x(T) = x_{\text{final}}$, where $x_{\text{final}} = (p_{\text{final}}, 0) \in \mathbf{R}^4$;

- **Wish 2: bounded control.** Because the motor of the robot is limited, we can apply forces only up to a certain magnitude, say U_{max}. That is, we wish the control signal to satisfy $\|u(t)\|_2 \leq U_{\text{max}}$ for $0 \leq t \leq T - 1$, where

$$\|x\|_2 = \left(x_1^2 + \cdots + x_n^2\right)^{1/2}$$

  is the usual Euclidean norm of a vector x in $\mathbf{R}^n$ with $x = (x_1, \ldots, x_n)$. This norm is also commonly called the ℓ_2 norm;

- **Wish 3: waypoints.** As the robot transfers from its initial position to its final position, we also would like it to pass as close as possible to certain intermediate locations—called *waypoints*—at given appointed times. For example, we could wish the robot to get data from information sources at those locations, via wireless; because wireless signals weaken with distance, the closer the robot is to each of the locations at the appointed times, the better.

  The set of waypoints is $\{w_1, \ldots, w_K\}$, and the associated set of appointed times is $\{\tau_1, \ldots, \tau_K\}$. Hence our wish is to have the position of the robot at each time τ_k to be close to w_k, that is, $p(\tau_k) \simeq w_k$ for $1 \leq k \leq K$;

- **Wish 4: simple control.** Finally, we wish to have a "simple" control signal, that is, a control signal that is constant, or, at least, a control signal that changes just a few times during the time horizon $\{0, 1, \ldots, T - 1\}$. This means that we wish to have $u(t) = u(t - 1)$ for most values of t in $\{1, \ldots, T - 1\}$.

2.2 The ℓ_2^2 regularizer

We now convert the wish list at the end of section 2.1 into an optimization problem. We encode the first two wishes (transfer and bounded control) as constraints of the optimization problem; to deal with the last two wishes (waypoints and simple control), we create a cost function that penalizes deviations from those wishes.

Optimization problem. We formulate the problem as follows:

$$
\begin{aligned}
\underset{x,u}{\text{minimize}} \quad & \sum_{k=1}^{K} \|Ex(\tau_k) - w_k\|_2^2 + \lambda \sum_{t=1}^{T-1} \|u(t) - u(t-1)\|_2^2 \quad &&(2)\\
\text{subject to} \quad & x(0) = x_{\text{initial}} \\
& x(T) = x_{\text{final}} \\
& \|u(t)\|_2 \leq U_{\text{max}}, \quad \text{for } 0 \leq t \leq T - 1 \\
& x(t+1) = Ax(t) + Bu(t), \quad \text{for } 0 \leq t \leq T - 1.
\end{aligned}
$$

The variables to optimize are x and u, where x stands for the sequence of states of the robot, $(x(0), x(1), \ldots, x(T))$, and u stands for the sequence of control signals we apply, $(u(0), u(1), \ldots, u(T-1))$.

Thus, in the optimization problem (2), we are composing the state trajectory and the control signal, jointly. Of course, these variables are linked by the physics of the problem, that is, the robot dynamics; this link is expressed by the last constraint in (2).

The first two wishes (transfer and bounded control). The first two wishes are enforced by the first three constraints, which force the initial and final state to be the given ones and the magnitude of the control signal to be upper bounded by the given maximum admissible value.

The last two wishes (waypoints and simple control). The matrix E in the cost function of (2) is

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

that is, $Ex(\tau_k)$ gives the position of the robot at time τ_k: $Ex(\tau_k) = p(\tau_k)$. Hence the first sum in the cost function,

$$\sum_{k=1}^{K} \|Ex(\tau_k) - w_k\|_2^2, \tag{3}$$

penalizes deviations of the position of the robot from the waypoints, at the given appointed times. This sum takes care of the third wish (waypoints).

The fourth wish (simple control) is dealt with by the last sum in the cost function,

$$\sum_{t=1}^{T-1} \|u(t) - u(t-1)\|_2^2, \tag{4}$$

which penalizes deviations of the control signal at time t from its previous value at time $t-1$, across the time horizon. The sum in (4) is the *regularizer*. We call it a ℓ_2^2 regularizer because it uses the *square* of the ℓ_2 norm of the control increments.

The two sums (3) and (4) are blended additively in the cost function of (2) by a weight $\lambda > 0$, the *regularization parameter*. Increasing λ gives more importance to the fourth wish, therefore playing down the third wish; decreasing λ does the opposite.

Constants. For problem (2), consider $T = 80$, $p_{\text{initial}} = (0, 5)$, and $p_{\text{final}} = (15, -15)$. Also, consider the $K = 6$ waypoints

$$w_1 = \begin{bmatrix} 10 \\ 10 \end{bmatrix}, w_2 = \begin{bmatrix} 20 \\ 10 \end{bmatrix}, w_3 = \begin{bmatrix} 30 \\ 10 \end{bmatrix}, w_4 = \begin{bmatrix} 30 \\ 0 \end{bmatrix}, w_5 = \begin{bmatrix} 20 \\ 0 \end{bmatrix} \text{ and } w_6 = \begin{bmatrix} 10 \\ -10 \end{bmatrix},$$

with appointed times

$$\tau_1 = 10, \tau_2 = 25, \tau_3 = 30, \tau_4 = 40, \tau_5 = 50, \text{ and } \tau_6 = 60.$$

4

The maximum control magnitude is $U_{\text{max}} = 100$.

Task 1. Use the software CVX from `http://cvxr.com/cvx` to solve problem (2) for $\lambda \in \{10^{-3}, 10^{-2}, 10^{-1}, 10^0, 10^1, 10^2, 10^3\}$. (Hint: For squared-Euclidean norm terms such as $\|\cdot\|_2^2$, check the function `quad_pos` of CVX.) So, you solve 7 problems (one per value of λ in the list). After each problem is solved, do the following:

(a) plot the optimal positions of the robot from $t = 0$ to $t = T$, marking its positions at the appointed times τ_k, for $1 \leq k \leq K$;

(b) plot the optimal control signal $u(t)$, where $u(t) = (u_1(t), u_2(t))$, from $t = 0$ to $t = T - 1$;

(c) report how many times the optimal control signal changes from $t = 1$ to $t = T - 1$: we say that the control signal changes at time t if $\|u(t) - u(t-1)\|_2 > 10^{-6}$; this tells us how much the fourth wish of a simple control is ignored;

(d) report the mean deviation from the waypoints, defined as

$$\frac{1}{K} \sum_{k=1}^{K} \|Ex(\tau_k) - w_k\|_2, \tag{5}$$

which tells us how much the third wish (waypoints) is ignored.

Example. So that you can have an example to check your code, we give the answer for the case $\lambda = 10^{-1}$: Figure 1 answers part (a), and figure 2 answers part (b). The answer to part (c) is 79, that is, the optimal control signal changes at each time instant $t \in \{0, 1, \ldots, T-1\}$ (recall that $T = 80$); you can also confirm this from figure 2. Finally, the answer to part (d), the mean waypoint deviation (5), is 2.1958.
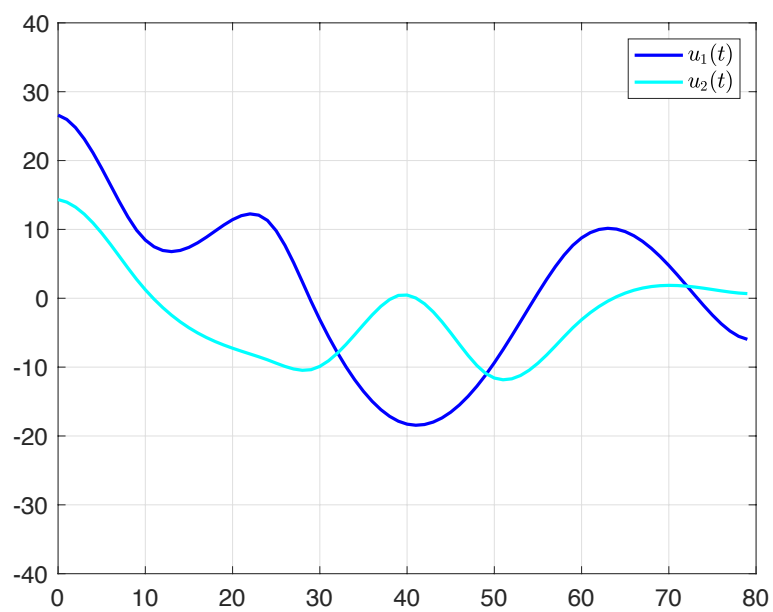
2.3 The ℓ_2 regularizer

We now remove the square of the ℓ_2 regularizer in (2) and formulate the problem as follows:

$$\begin{aligned}
\underset{x,u}{\text{minimize}} \quad & \sum_{k=1}^{K} \|Ex(\tau_k) - w_k\|_2^2 + \lambda \sum_{t=1}^{T-1} \|u(t) - u(t-1)\|_2 \tag{6} \\
\text{subject to} \quad & x(0) = x_{\text{initial}} \\
& x(T) = x_{\text{final}} \\
& \|u(t)\|_2 \leq U_{\text{max}}, \quad \text{for } 0 \leq t \leq T - 1 \\
& x(t+1) = Ax(t) + Bu(t), \quad \text{for } 0 \leq t \leq T - 1.
\end{aligned}$$
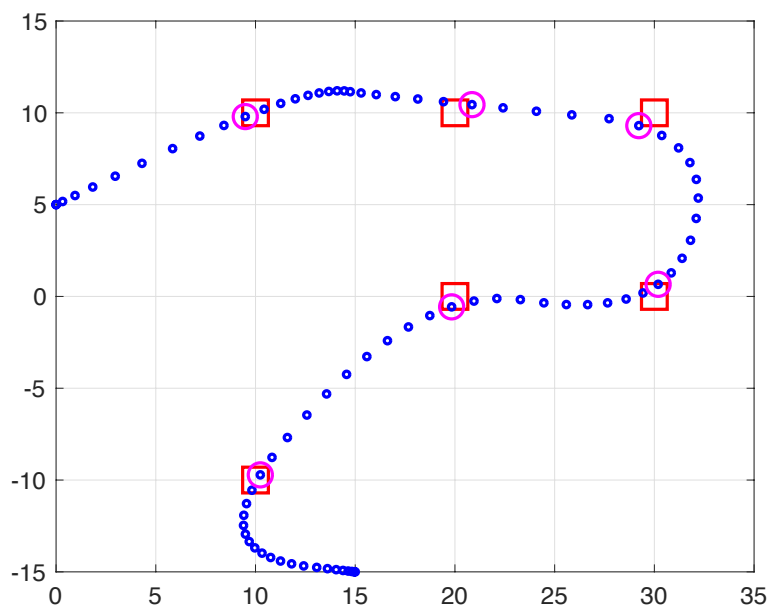
Task 2. Redo task 1 for the optimization problem (6).
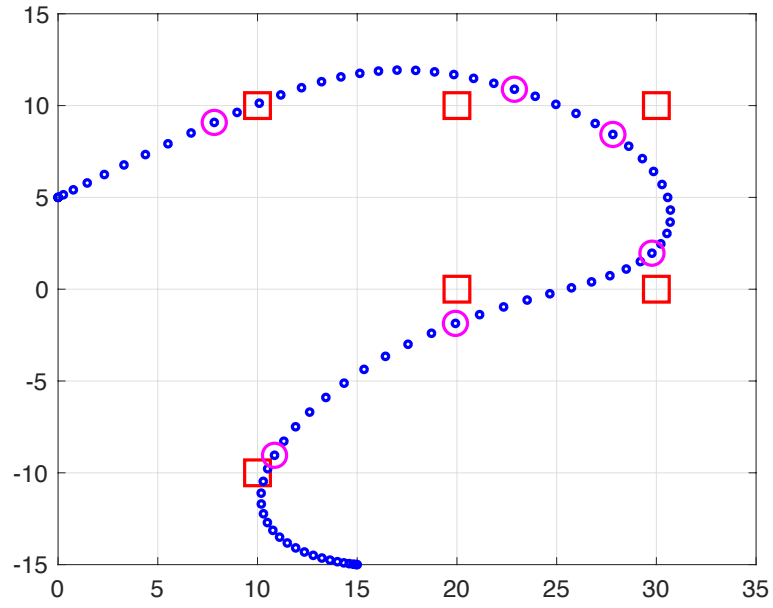
Figure 1: Positions of the robot from $t = 0$ to $t = T$ are the small blue circles; the positions at appointed times τ_k, for $1 \leq k \leq K$, are the large magenta circles. The waypoints are the red squares. Case $\lambda = 10^{-1}$ with ℓ_2^2 regularizer—see problem (2).

Figure 2: The two components, $u_1(t)$ and $u_2(t)$, of the control signal $u(t)$ from $t = 0$ to $t = T - 1$. Case $\lambda = 10^{-1}$ with ℓ_2^2 regularizer—see problem (2).

Example. We give the answer for $\lambda = 10^{-1}$, which you should reproduce. The answers to parts (a) and (b) are in figures 3 and 4, respectively. The answer to part (c) is 11, and the answer to part (d) is 0.7021.


Figure 3: Positions of the robot from $t = 0$ to $t = T$ are the small blue circles; the positions at appointed times τ_k, for $1 \le k \le K$, are the large magenta circles. The waypoints are the red squares. Case $\lambda = 10^{-1}$ with ℓ_2 regularizer—see problem (6).

2.4 The ℓ_1 regularizer

As a final way to tackle the fourth wish (simple control), we use a regularizer based on the ℓ_1 norm. For a vector $x \in \mathbf{R}^n$ with $x = (x_1, \ldots, x_n)$, the ℓ_1 norm is

$$\|x\|_1 = |x_1| + \cdots + |x_n|.$$
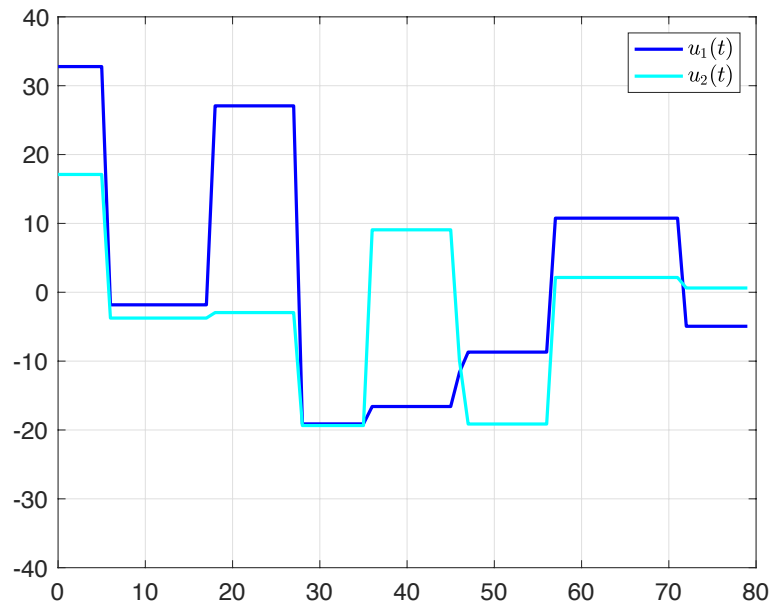
Figure 4: The two components, $u_1(t)$ and $u_2(t)$, of the control signal $u(t)$ from $t = 0$ to $t = T - 1$. Case $\lambda = 10^{-1}$ with ℓ_2 regularizer—see problem (6).

We formulate the problem as follows:

$$\begin{aligned}
\underset{x,u}{\text{minimize}} \quad & \textstyle\sum_{k=1}^{K} \|Ex(\tau_k) - w_k\|_2^2 + \lambda \sum_{t=1}^{T-1} \|u(t) - u(t-1)\|_1 \qquad (7) \\
\text{subject to} \quad & x(0) = x_{\text{initial}} \\
& x(T) = x_{\text{final}} \\
& \|u(t)\|_2 \leq U_{\text{max}}, \quad \text{for } 0 \leq t \leq T - 1 \\
& x(t+1) = Ax(t) + Bu(t), \quad \text{for } 0 \leq t \leq T - 1.
\end{aligned}$$

Task 3. Redo task 1 for the optimization problem (7).

Example. We give the answer for $\lambda = 10^{-1}$. Parts (a) and (b) are in answered in figures 5 and 6, respectively. The answer to part (c) is 14, and the answer to part (d) is 0.8863.

Task 4. Comment on what you have observed from Tasks 1 to 3 (for example, compare the impact of the three regularizers on the optimal control signal that they each induce).

3 Variation B: approximate intermediate regions and simple control signals

In section 2, we want the robot to pass near certain intermediate points, at appointed times.

Disks. Suppose now that we want the robot to pass near certain intermediate regions, namely, disks. A disk, denoted $D(c, r)$, is a set of the form $\{x \in \mathbf{R}^2 : \|x - c\|_2 \leq r\}$, where $c \in \mathbf{R}^2$ is the center of the disk and r its radius.

We consider K intermediate disks $\{D(c_k, r_k) : 1 \leq k \leq K\}$ along with appointed times $\{\tau_k : 1 \leq k \leq K\}$. In this variation, we want the robot to be as close as possible to disk $D(c_k, r_k)$ at time τ_k. We don't care whether the robot is close to the *center* of the disk; only whether it is close to the *whole* disk (for example, if the robot is inside the disk at time τ_k, we don't care whether the robot is near the center or near the boundary). Thus, we care only about $d(p(\tau_k), D(c_k, r_k))$, where $d(p, D(c, r))$ is the distance of the point $p \in \mathbf{R}^2$ to the disk $D(c, r)$:

$$d(p, D(c, r)) = \min \{\|p - y\|_2 : y \in D(c, r)\}.$$

Task 5. Using simple geometric arguments, give a closed-form expression for $d(p, D(c, r))$.
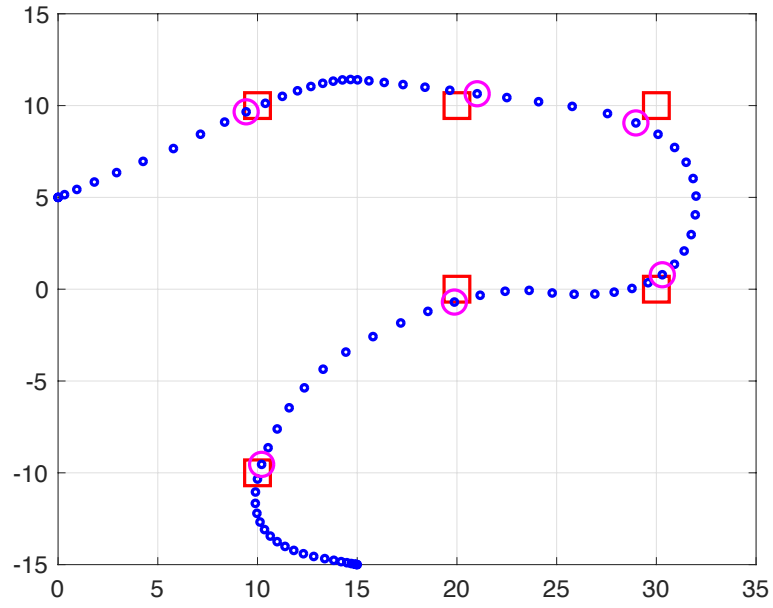
Figure 5: Positions of the robot from $t = 0$ to $t = T$ are the small blue circles; the positions at appointed times τ_k, for $1 \leq k \leq K$, are the large magenta circles. The waypoints are the red squares. Case $\lambda = 10^{-1}$ with ℓ_1 regularizer—see problem (7).
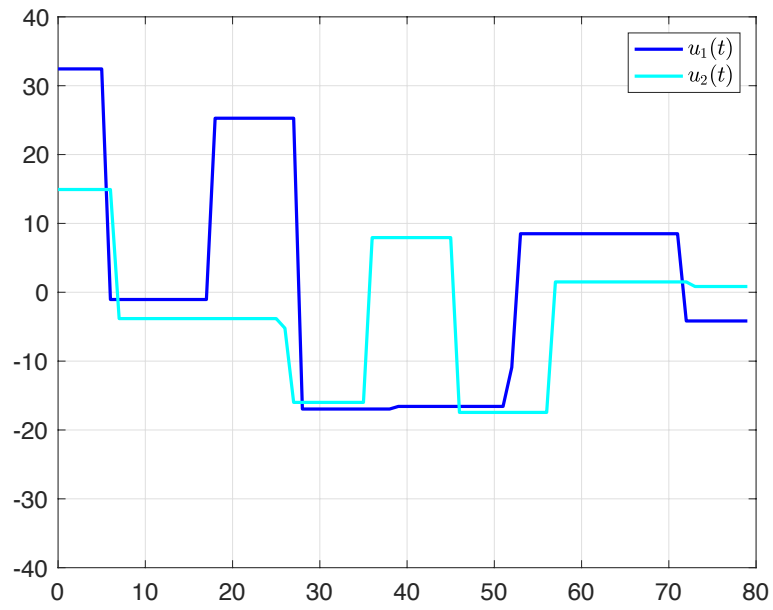
Figure 6: The two components, $u_1(t)$ and $u_2(t)$, of the control signal $u(t)$ from $t = 0$ to $t = T - 1$. Case $\lambda = 10^{-1}$ with ℓ_1 regularizer—see problem (7).

Problem formulation.   For this variation, we formulate the following optimization problem:

$$\underset{x,u}{\text{minimize}} \quad \sum_{k=1}^{K} d\left(Ex(\tau_k), D(c_k, r_k)\right)^2 + \lambda \sum_{t=1}^{T-1} \|u(t) - u(t-1)\|_2 \qquad (8)$$

$$\text{subject to} \quad x(0) = x_{\text{initial}}$$
$$x(T) = x_{\text{final}}$$
$$\|u(t)\|_2 \leq U_{\text{max}}, \quad \text{for } 0 \leq t \leq T-1$$
$$x(t+1) = Ax(t) + Bu(t), \quad \text{for } 0 \leq t \leq T-1.$$

Constants.   For problem (8), consider $\lambda = 10^{-1}$. The set of intermediate $K = 6$ disks is $\{D(c_k, r_k) : \text{ for } 1 \leq k \leq K\}$, where

$$c_1 = \begin{bmatrix} 10 \\ 10 \end{bmatrix}, c_2 = \begin{bmatrix} 20 \\ 10 \end{bmatrix}, c_3 = \begin{bmatrix} 30 \\ 10 \end{bmatrix}, c_4 = \begin{bmatrix} 30 \\ 0 \end{bmatrix}, c_5 = \begin{bmatrix} 20 \\ 0 \end{bmatrix} \text{ and } c_6 = \begin{bmatrix} 10 \\ -10 \end{bmatrix},$$

with appointed times

$$\tau_1 = 10, \tau_2 = 25, \tau_3 = 30, \tau_4 = 40, \tau_5 = 50, \text{ and } \tau_6 = 60.$$

Note that the centers of the disks are the previous waypoints. Take all the radiuses of the disks to be 2. All other constants are as in section 2.

Task 6.   Use the software `CVX` to solve problem (8). After you solve the problem, do the following:

(a) plot the optimal positions of the robot from $t = 0$ to $t = T$, marking its positions at the appointed times τ_k, for $1 \leq k \leq K$;

(b) plot the optimal control signal $u(t)$, where $u(t) = (u_1(t), u_2(t))$, from $t = 0$ to $t = T-1$;

(c) report how many times the optimal control signal changes from $t = 1$ to $t = T-1$;

(d) report the mean deviation from the waypoints (defined in (5));

(e) comment on the results from (a)-(d), as you compare them with those you obtained in Task 2. (Note that Task 2 considered waypoints, which can be seen as disks with radius zero.)

---

4   Variation C: exact intermediate points

For this variation, we drop the wish for a simple control signal. But we become more serious about the waypoints.

Exact waypoints. In section 2 we wish the robot to pass *approximately* on certain waypoints at appointed times, that is, we want the position $p(\tau_k)$ of the robot at an appointed time τ_k to be close to its associated waypoint w_k: $p(\tau_k) \simeq w_k$.

In this section, however, we want the robot to pass on waypoints *exactly*: We want to have

$$p(\tau_k) = w_k, \tag{9}$$

for as many waypoints $k \in \{1, \ldots, K\}$ as possible. When the robot passes exactly on a given waypoint, we say that the robot *captures* that waypoint. Hence we wish the robot to capture as many waypoints as possible along its transfer.

4.1 Two uninteresting formulations

A feasibility formulation. Why simply not enforce the equalities (9) for $1 \le k \le K$? That is, why not try to capture all the waypoints? This leads to the following formulation:

$$
\begin{aligned}
\underset{x,u}{\text{minimize}} \quad & 0 \\
\text{subject to} \quad & x(0) = x_{\text{initial}} \\
& x(T) = x_{\text{final}} \\
& \|u(t)\|_2 \le U_{\max}, \quad \text{for } 0 \le t \le T-1 \\
& x(t+1) = Ax(t) + Bu(t), \quad \text{for } 0 \le t \le T-1, \\
& Ex(\tau_k) = w_k, \quad \text{for } 1 \le k \le K.
\end{aligned}
\tag{10}
$$

In (10) the cost function is identically zero. That's because there is nothing to optimize. We just want a control signal that makes the robot pass exactly on all the waypoints. A problem in which the cost function is identically zero is called a *feasibility problem*: in such problems, we just want to find a point that is *feasible*—a point that satisfies all the constraints.

Constants. For problem (10) take $U_{\max} = 15$. Note that this constant has changed; previously, it was 100, which means that we are dealing now with a robot with a weaker motor. All other constants are as in section 2.

> **Task 7.** Use the software `CVX` to solve problem (10). Explain what happens.

A formulation too hard to solve. Another possible formulation for this problem is as follows:

$$
\begin{aligned}
\underset{x,u}{\text{minimize}} \quad & \sum_{k=1}^{K} \phi\left(Ex(\tau_k) - w_k\right) \\
\text{subject to} \quad & x(0) = x_{\text{initial}} \\
& x(T) = x_{\text{final}} \\
& \|u(t)\|_2 \le U_{\max}, \quad \text{for } 0 \le t \le T-1 \\
& x(t+1) = Ax(t) + Bu(t), \quad \text{for } 0 \le t \le T-1.
\end{aligned}
\tag{11}
$$

The function ϕ in (11) outputs a zero if its argument is zero and outputs a one, otherwise. That is, we have $\phi : \mathbf{R}^2 \to \mathbf{R}$, with

$$\phi(x) = \begin{cases} 0, & \text{if } x = 0, \\ 1, & \text{if } x \neq 0. \end{cases}$$

In words, the objective function in (11) counts the number of waypoints that the robot misses. So, in problem (11), we look for the control signal that makes the robot miss as few waypoints as possible.

It turns out that problem (11) is nonconvex and too hard to solve numerically.

> **Task 8.** Show that the function ϕ is nonconvex.

In sections 4.2, 4.3, and 4.4, we explore convex formulations that approximate problem (11) and are easy to solve numerically.

4.2 A ℓ_2^2 formulation

We start by replacing the function ϕ by a squared ℓ_2 norm:

$$\begin{aligned}
\underset{x,u}{\text{minimize}} \quad & \sum_{k=1}^{K} \|Ex(\tau_k) - w_k\|_2^2 & (12)\\
\text{subject to} \quad & x(0) = x_{\text{initial}} \\
& x(T) = x_{\text{final}} \\
& \|u(t)\|_2 \leq U_{\text{max}}, \quad \text{for } 0 \leq t \leq T - 1 \\
& x(t+1) = Ax(t) + Bu(t), \quad \text{for } 0 \leq t \leq T - 1.
\end{aligned}$$

Constants. For problem (12) consider $U_{\text{max}} = 15$. All other constants are as in section 2.

> **Task 9.** Use the software `CVX` to solve problem (12). After you solve the problem, do the following:
>
> (a) plot the optimal positions of the robot from $t = 0$ to $t = T$, marking its positions at the appointed times τ_k, for $1 \leq k \leq K$;
>
> (b) plot the optimal control signal $u(t)$, where $u(t) = (u_1(t), u_2(t))$, from $t = 0$ to $t = T - 1$;
>
> (c) report how many waypoints are captured by the robot; consider that a waypoint w_k is captured if the robot is sufficiently close to it at the appointed time τ_k, namely, if $\|p(\tau_k) - w_k\|_2 \leq 10^{-6}$.

4.3 A ℓ_2 formulation

Consider now that the function ϕ is replaced by a ℓ_2 norm:

$$
\begin{aligned}
\underset{x,u}{\text{minimize}} \quad & \sum_{k=1}^{K} \| Ex(\tau_k) - w_k \|_2 \\
\text{subject to} \quad & x(0) = x_{\text{initial}} \\
& x(T) = x_{\text{final}} \\
& \| u(t) \|_2 \leq U_{\text{max}}, \quad \text{for } 0 \leq t \leq T - 1 \\
& x(t+1) = Ax(t) + Bu(t), \quad \text{for } 0 \leq t \leq T - 1.
\end{aligned}
\tag{13}
$$

Constants. For problem (13) consider $U_{\text{max}} = 15$. All other constants are as in section 2.

> **Task 10.** Redo task 9 for problem (13).

4.4 The iterative reweighting technique

An iterative technique for improving on the results of problem (13) consists in solving reweighted versions of it. Specifically, each iteration of the technique takes a current guess $x^{(m)}$ and $u^{(m)}$ and produces a new one $x^{(m+1)}$ and $u^{(m+1)}$ by solving the following problem:

$$
\begin{aligned}
\underset{x,u}{\text{minimize}} \quad & \sum_{k=1}^{K} \frac{1}{\left\| Ex^{(m)}(\tau_k) - w_k \right\|_2 + \epsilon} \| Ex(\tau_k) - w_k \|_2 \\
\text{subject to} \quad & x(0) = x_{\text{initial}} \\
& x(T) = x_{\text{final}} \\
& \| u(t) \|_2 \leq U_{\text{max}}, \quad \text{for } 0 \leq t \leq T - 1 \\
& x(t+1) = Ax(t) + Bu(t), \quad \text{for } 0 \leq t \leq T - 1.
\end{aligned}
\tag{14}
$$

Note the weight $1 / \left(\left\| Ex^{(m)}(\tau_k) - w_k \right\|_2 + \epsilon \right)$ for each term k in the sum of the objective function of (14) (in (13), these weights are all 1). The constant ϵ is a very small positive number that prevents the denominator to be zero.

Solving (14) gives $x^{(m+1)}$ and $u^{(m+1)}$. The technique is initialized with $x^{(0)}$ and $u^{(0)}$, the solution of problem (13).

Constants. For problem (14) consider $U_{\max} = 15$ and $\epsilon = 10^{-6}$. All other constants are as in section 2.

Task 11. Apply the reweighting technique M times, with $M = 10$; that is, solve problem problem (14) for $m = 0, 1, \ldots, M - 1$. For each value of m, do the following:

(a) plot the positions of the robot in $x^{(m)}$ from $t = 0$ to $t = T$, marking its positions at the appointed times τ_k, for $1 \leq k \leq K$;

(b) plot the control signal $u^{(m)}(t)$, where $u^{(m)}(t) = \left(u_1^{(m)}(t), u_2^{(m)}(t) \right)$, from $t = 0$ to $t = T-1$;

(c) report how many waypoints are captured by the robot.

Task 12. What's the role of the weights in (14)?