Linear Programming

Case study

A Linear Programming Approach for Dynamical System Control

Marcello Sanguineti

marcello.sanguineti@unige.it

DIBRIS - University of Genova

LP with Absolute Value

Problem LP_{ABV}

Let $\mathbf{c} \in \mathbb{R}^n$ be a nonnegative vector. Find $\mathbf{x}^* \in \mathbb{R}^n$ such that

$$\mathbf{x}^* \in \arg\min \left\{ \mathbf{c}^\mathsf{T}|\mathbf{x}| : A\mathbf{x} = \mathbf{b} \right\}$$

LP_{ABV} is a LP problem with the two following differences with respect to the standard form:

1. the cost contains the absolute value: $\mathbf{c}^\mathsf{T}|\mathbf{x}| = \sum_{i=1}^n c_i|x_i|$;
2. the decision variables are unbounded: $-\infty < x_i < \infty$, $i = 1, \dots, n$.

Solution to LP_{ABV}

LP_{ABV} is equivalent to the following LP problem:

- Write the vector $\mathbf{x}$ as the difference of two vectors of auxiliary variables: $\mathbf{x} = \mathbf{x}_+ - \mathbf{x}_-$
- Find $\mathbf{x}_+^*, \mathbf{x}_-^*$ such that

$$\mathbf{x}_+^*, \mathbf{x}_-^* \in \arg\min \left\{ \mathbf{c}^\mathsf{T}(\mathbf{x}_+ + \mathbf{x}_-) : A(\mathbf{x}_+ - \mathbf{x}_-) = \mathbf{b};\ \mathbf{x}_+, \mathbf{x}_- \geqslant 0 \right\}$$

- Set $\mathbf{x}^* = \mathbf{x}_+^* - \mathbf{x}_-^*$

To see the equivalence, note that

- we are minimizing;
- $\mathbf{c} \in \mathbb{R}^n$ is a nonnegative vector;
- hence, an optimal solution must have $x_+ = 0$ or $x_- = 0$, otherwise we could reduce both x_+ and x_- by the same amount and obtain a better feasible solution;
- if $x_+ = 0$, then $x = -x_- \leqslant 0$, $|x| = -x$, and $\min \left\{ \mathbf{c}^\mathsf{T}(x_+ + x_-) : A(x_+ - x_-) = \mathbf{b};\ x_+, x_- \geqslant 0 \right\} \equiv \min \left\{ \mathbf{c}^\mathsf{T} x_- : A(-x_-) = \mathbf{b};\ x_- \geqslant 0 \right\} \equiv \min \left\{ -\mathbf{c}^\mathsf{T} x : Ax = \mathbf{b};\ x \leqslant 0 \right\} \equiv \min \left\{ \mathbf{c}^\mathsf{T}|x| : Ax = \mathbf{b};\ x \leqslant 0 \right\}$;
- if $x_- = 0$, then $x = x_+ \geqslant 0$, $|x| = x$, and $\min \left\{ \mathbf{c}^\mathsf{T}(x_+ + x_-) : A(x_+ - x_-) = \mathbf{b};\ x_+, x_- \geqslant 0 \right\} \equiv \min \left\{ \mathbf{c}^\mathsf{T} x_+ : Ax_+ = \mathbf{b};\ x_+ \geqslant 0 \right\} \equiv \min \left\{ \mathbf{c}^\mathsf{T} x : Ax = \mathbf{b};\ x \geqslant 0 \right\} \equiv \min \left\{ \mathbf{c}^\mathsf{T}|x| : Ax = \mathbf{b};\ x \geqslant 0 \right\}$.

Discretisation of a linear differential equation

Consider a continuous-time differential equation of the form

$$\dot{\mathbf{x}}(t) = A\,\mathbf{x}(t) + B\,\mathbf{u}(t) + G\,\mathbf{d}(t), \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^n$ (state vector), $\mathbf{u} \in \mathbb{R}^m$ (control), $\mathbf{d} \in \mathbb{R}^p$ (disturbances) and A, B, G are matrices of appropriate dimensions.

A forward Euler approximation of (1) is given by:

$$\mathbf{x}(k+1) = \Phi\,\mathbf{x}(k) + \Delta\,\mathbf{u}(k) + \Psi\,\mathbf{d}(k),$$

where $\Phi = I + A\,dt$, $\Delta = B\,dt$, $\Psi = G\,dt$ and dt is the sample time.

Mathematical structure of the model

Discrete-time linear system

$$x(k+1) = \Phi\, x(k) + \Delta\, u(k) + \Psi\, d(k), \quad k = 0, 1, \ldots$$

where

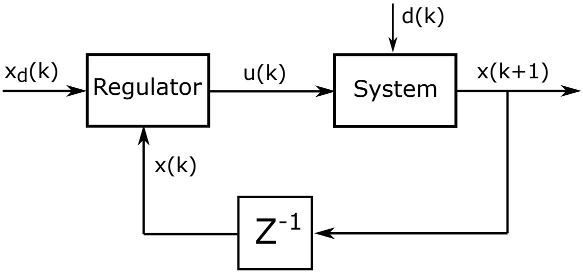
- $x \in \mathbb{R}^n$ is the state variable (measurable);
- $u \in \mathbb{R}^m$ is the control variable;
- $d \in \mathbb{R}^p$ is a disturbance acting on the system (not measurable);
- Φ, Δ, and Ψ are matrices of appropriate dimensions.

Dynamical System Control Problem (DSCP)

Design a control strategy such that $u(k)$, $k = 0, 1, \ldots,$ allows the system to follow a desired output $x_d(k)$ as close as possible (tracking), in such a way to fulfil the following constraints:

1. $\theta \leqslant x(k) \leqslant \lambda$ (constraint on the state)
2. $\alpha \leqslant u(k) \leqslant \beta$ (constraint on the control)

Block diagram of the problem



At stage k the information available to the regulator are:

- the state variable $x(k)$;
- the desired output $x_d(k)$;
- an estimation of the state variables at stage $k+1$ given by $\hat{x}(k+1) = \Phi\, x(k) + \Delta \mathbf{u}(k)$ (the disturbance vector is unknown).

Performance index

In order to apply optimisation theory, we need a performance index:

$$\mathcal{J} = \sum_{i=1}^{n} w_i |\hat{e}_i(k+1)| = \sum_{i=1}^{n} w_i |\hat{x}_i(k+1) - x_i^d(k)|,$$

where w_i are positive numbers to weight the various components of $\mathbf{x}$.

Remarks

1. At stage k we can influence $\mathbf{x}(k+1)$ through $\mathbf{u}(k)$, then, the best we can do is to make $\mathbf{x}(k+1)$ as close as possible to $\mathbf{x}^d(k)$.
2. Since $\mathbf{d}(k)$ is unknown, we can only guarantee that the estimation $\hat{\mathbf{x}}(k+1)$ satisfies the constraint on the state.
3. The optimisation routine to minimise $\mathcal{J}$ must be invoked at every stage k (this could be an issue).

Optimisation-based control policy

Optimisation based control policy

At every stage k, solve the following problem:

$$\min_{\mathbf{u}(k)} \mathcal{J} = \sum_{i=1}^{n} w_i |\hat{e}_i(k+1)|$$

such that

$$\boldsymbol{\alpha} \leqslant \mathbf{u}(k) \leqslant \boldsymbol{\beta} \quad \text{(constraint on the control)} \tag{2}$$

$$\boldsymbol{\theta} \leqslant \hat{\mathbf{x}}(k+1) \leqslant \boldsymbol{\lambda} \quad \text{(constraint on the state)} \tag{3}$$

The control problem is a LP problem with the absolute value in the cost

$$\Downarrow$$

This is an instance od the LP_{ABV} optimisation model

LP-based solution procedure

In order to convert this control problem into a standard LP, we ntroduce the following auxiliary variables:

- $\hat{e}_i(k+1) = x_i^+ - x_i^-$, where $x_i^+, x_i^- \geqslant 0$, for $i = 1, \ldots, n$.
- $\tilde{\mathbf{u}} = \mathbf{u}(k) - \boldsymbol{\alpha}$.

Since $\hat{e}_i(k+1) = \hat{x}_i(k+1) - x_i^d(k)$ and $\hat{\mathbf{x}}(k+1) = \Phi\,\mathbf{x}(k) + \Delta\mathbf{u}(k)$, we have the following equality constraint:

$$\mathbf{x}^+ - \mathbf{x}^- - \Delta\,\tilde{\mathbf{u}} = \Phi\,\mathbf{x}(k) - \mathbf{x}_d(k) + \Delta\,\alpha$$

Due to the constraint on the state, we have that $\boldsymbol{\lambda} \leqslant \mathbf{x}^+ - \mathbf{x}^- + \mathbf{x}_d(k) \leqslant \boldsymbol{\theta}$. Therefore, we need a slack and a surplus variable:

$$\mathbf{x}^+ - \mathbf{x}^- - \mathbf{x}_s^- = \boldsymbol{\theta} - \mathbf{x}_d(k)$$

$$\mathbf{x}^+ - \mathbf{x}^- + \mathbf{x}_s^+ = \boldsymbol{\lambda} - \mathbf{x}_d(k)$$

Due to the constraint on the control, we have that $0 \leqslant \tilde{\mathbf{u}} \leqslant \boldsymbol{\beta} - \boldsymbol{\alpha}$. Therefore, we need a slack variable $\mathbf{u}_s^+$:

$$\tilde{\mathbf{u}} + \mathbf{u}_s^+ = \boldsymbol{\beta} - \boldsymbol{\alpha}$$

LP-based solution procedure

By putting all the variables together, in the standard LP problem the unknown vector of variables is

$$z = \begin{bmatrix} x^+ & x^- & \tilde{u} & x_s^+ & x_s^- & u_s^+ \end{bmatrix}^\mathsf{T}$$

The dimension of z is $4n + 2m$.
The performance index can be written as

$$\mathcal{J} = \begin{bmatrix} w & w & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x^+ \\ x^- \\ \tilde{u} \\ x_s^+ \\ x_s^- \\ u_s^+ \end{bmatrix}$$

The linear equality constraint in the standard LP problem is

$$\begin{bmatrix} I & -I & -\Delta & 0 & 0 & 0 \\ I & -I & 0 & 0 & -I & 0 \\ I & -I & 0 & I & 0 & 0 \\ 0 & 0 & I & 0 & 0 & I \end{bmatrix} \begin{bmatrix} x^+ \\ x^- \\ \tilde{u} \\ x_s^+ \\ x_s^- \\ u_s^+ \end{bmatrix} = \begin{bmatrix} \Phi\, x(k) - x_d(k) + \Delta\, \alpha \\ \theta - x_d(k) \\ \lambda - x_d(k) \\ \beta - \alpha \end{bmatrix}$$

LP-based solution procedure

Summing up, the original problem can be formulated in the following compact form:

$$\min \mathcal{J} = \mathbf{c}^{\mathsf{T}} \boldsymbol{z} \ \text{ s. t. } \ \boldsymbol{A}\boldsymbol{z} = \mathbf{b}, \ \boldsymbol{z} \geqslant 0$$

At every iteration k, we get an optimal vector $\boldsymbol{z}^*$ containing the value $\tilde{\mathbf{u}}^*$. Then, by recalling the definition of $\tilde{\mathbf{u}}$, we set $\mathbf{u}(k) = \tilde{\mathbf{u}}^* + \boldsymbol{\alpha}$.

Pros and cons of the LP-based approach

Pros

- Software for LP is fast, efficient, and contained in almost all numerical packages.
- Ease of ensuring that the constraints on the state and the control are met.
- Possibility of extending the technique to nonlinear systems, via linearization.

Cons

- If the sampling time is short, one may require expensive hardware.
- This technique does not guarantee convergence of $x(k)$ to $x_d(k)$ for $k \to \infty$.

Example 1: n-th order integrator plant control

In order to show the scalability of the proposed control strategy, consider the following n-th order integrator system:

$$\frac{d^n}{dt^n} x(t) = u(t)$$

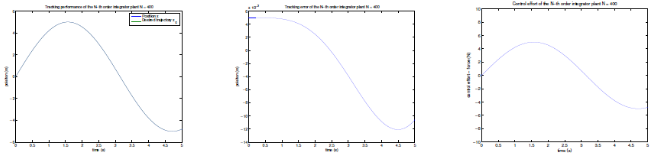
The system can be converted into a state-space form in the following way:

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} u$$

The system is discretized with a sample time of 1 ms.

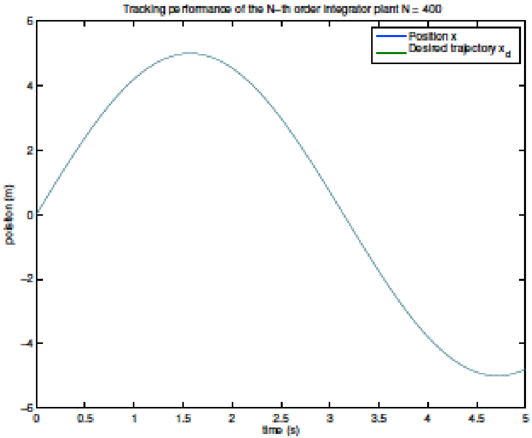
The goal of the control is to make the variable x_1 track the desired trajectory $x_d(t) = 5 \sin t$.

The first round of tests considers an instance of the problem with $n = 400$ and $u_{\text{max}=10}$. Therefore, the size of the LP problem is $4n + 2m = 1602$.

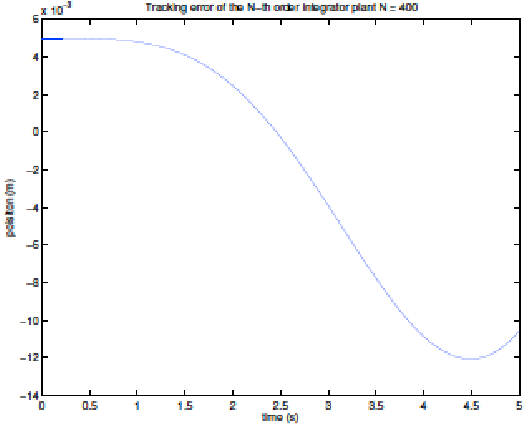


(a) Tracking performance (b) Tracking error (c) Control input

Tracking performance

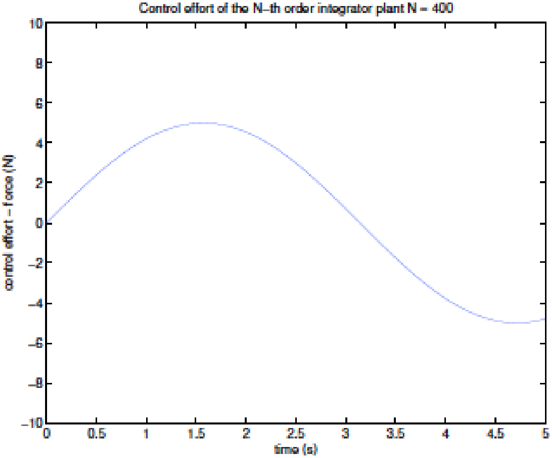


Tracking error



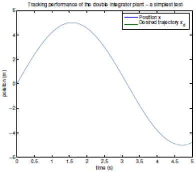
The maximum tracking error is 12×10^{-3}, which is good compared with a control design with a reasonable feedback gain.

Control input

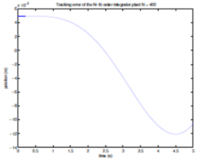


Simulation results - Second instance

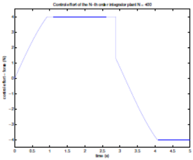
The second round of tests considers again $n = 400$, but the harder constraint $u_{\text{max}=4}$.



(a)Tracking performance

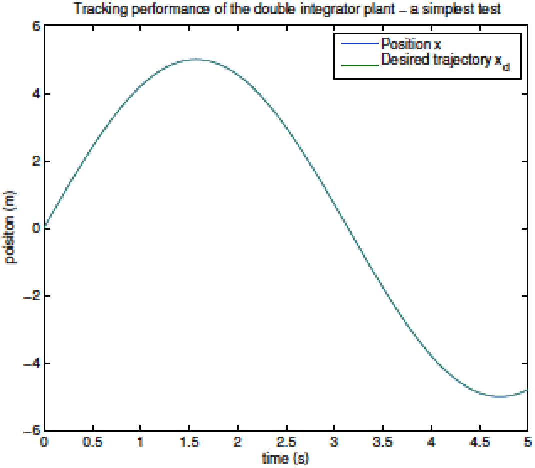


(b) Tracking error

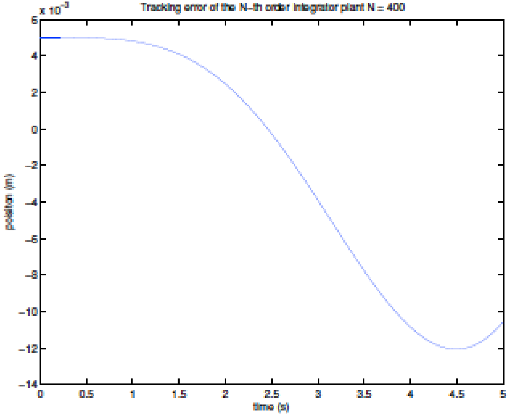


(c) Control input

Tracking performance

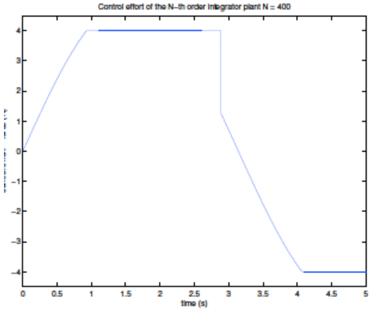


Tracking error



The tracking error has not changed much with respect to the first instance.

Control input



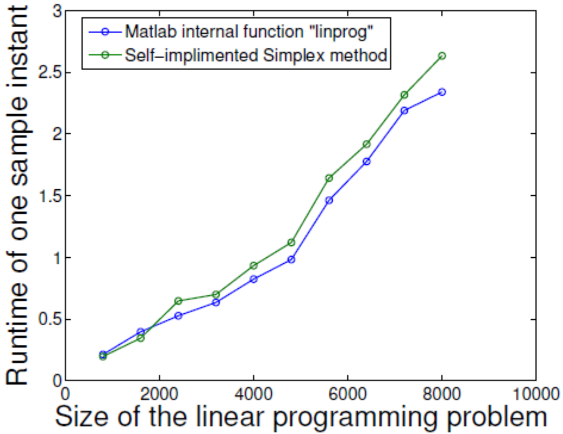
It can be seen that the LP-based control successfully limited the control variable.

Summing up, these simulations show the effectiveness of the LP-based control strategy.

Simulation results - Performance with increasing dimension of the PL problem

- Another test based on this system is the run time of the computation at a single sample instant.
- We measured the runtime of the simulation of 10 sample instants and then take the average to estimate the runtime of one LP problem.
- We used an increasing size of the linear dynamical system as $n = 50, 100, 150, ..., 1000$.
- The corresponding LP problem has a dimension $4n + 2r$ and for this test system $r = 1$.

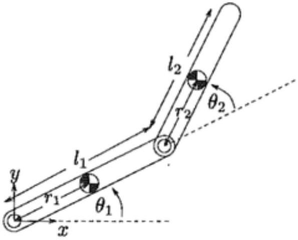
Runtime of one sample instant over the dimension of the LP problem. Blue: simulation time using the Matlab internal function ”linprog”. Green: simulation time using a self-implemented simplex function.



- The simulation shows that the average runtime of the LP problem is approximately linear with the problem dimension.
- However, when the LP-based control strategy is really used with an actual system, the worst case must be considered because the computation has to be done within one time step.
- Thus the sample rate of the system might be limited by the computation time.
- This is a major challenge for problems with a large size.

Example 2: Two-link robot arm system control

Problem: controlling the position of a two-link robot arm in the horizontal plane (hence, the gravity of the linkages can be ignored).



θ_1, θ_2: angles of the two joints.
l_1, l_2: lengths of the two linkages.
r_1, r_2: distances between the center of mass of the linkage to the corresponding joint.
Control inputs: torques acting at the two joints.

$$\begin{bmatrix} I_1 + I_2 m_2 l_1 r_2 \cos\theta_2 & I_2 + m_2 l_1 r_2 \cos\theta_2 \\ I_2 + m_2 l_1 r_2 \cos\theta_2 & I_2 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} +$$

$$\begin{bmatrix} -2m_2 l_1 r_2 \dot{\theta}_2 & -m_2 l_1 r_1 \sin\theta_2 \dot{\theta}_2 \\ -m_2 l_1 r_1 \sin\theta_2 \dot{\theta}_1 & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

By defining the vectors $\mathbf{q} = [\theta_1, \theta_2]^\mathsf{T}$ and $\mathbf{u} = [u_1, u_2]^\mathsf{T}$, the system can be written in compact form by means of analytical mechanics:

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} = \mathbf{u}.$$

By defining the state variables $x = [\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2]^\mathsf{T}$ and putting the system into a nonlinear state-space form, the equation can be write as:

$$\dot{x} = \begin{bmatrix} 0_{2\times 2} & \mathbf{I}_{2\times 2} \\ 0_{2\times 2} & \mathbf{H}^{-1}\mathbf{C} \end{bmatrix} x + \begin{bmatrix} 0_{2\times 2} \\ \mathbf{H}^{-1} \end{bmatrix} \mathbf{u}$$

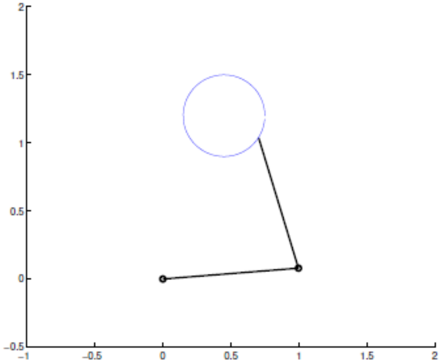
- The detailed derivation of this equation is shown in: M. I. of Technology. Project MAC. Engineering Robotics Group and M. Dertouzos, Control robotics: The procedural control of physical processes, 1973.
- It is a nonlinear dynamical system: the matrices H and C have dependency on the state variables
- This usually add great difficulty to the commonly used linear feedback control, since the feedback gains are fixed during the whole the execution process.
- Adaptive control algorithms update the feedback gains in real time depend on the robot configuration, and win great success in the field of Robotics.
- Being a control strategy that is doing on-line optimization, the LP-based control also has the potential to deal with this nonlinearity.
- We shall perform some simulations to verify the effectiveness of this LP-based control strategy on this nonlinear problem.

$$\dot{x} = \begin{bmatrix} 0_{2\times 2} & I_{2\times 2} \\ 0_{2\times 2} & H^{-1}C \end{bmatrix} x + \begin{bmatrix} 0_{2\times 2} \\ H^{-1} \end{bmatrix} u$$

- In order to apply the LP control strategy, we need to linearize the system through Taylor expansion. In such a way, the centrifugal and Coriolis term (matrix C) disappear.
- This is the major drawback of the linear feedback control.
- However in the LP-based control, we take the measurement of the state variable, and update the model based on our measurement every time step.
- That is, we make the linearization to this nonlinear system based on the state at the sample instant.

Control goal

Our control goal for this system is to make the end-effector of the robot arm track a given trajectory.



We want the end-effector of the arm to track this circle with a constant speed in the task space, which is the Cartesian coordinate in the figure.

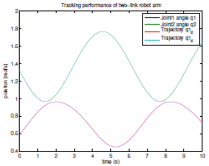
The control of this robot arm basically takes the following steps at one sample instant:

- Take the measurement of the state.
- Determine the $x - y$ space desired end-effector position (x_d, y_d) and the corresponding desired velocity.
- Use inverse kinematics transformation of the robot model to calculate the desired trajectory in the joint space $(\mathbf{q}_d, \dot{\mathbf{q}}_d)^T$.
- Formulate the linear system dynamics based on the current measurements.
- Change the system into a standard LP problem by means of the procedure previously described.
- Compute the control input $\mathbf{u}$ by means of LP.

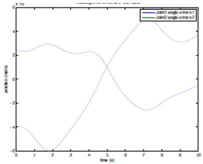
Simulation results

The simulations of this two-link robot are performed with a weighting factor $w = [101011]$ on the state tracking errors.

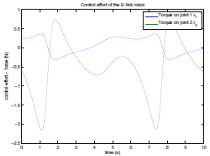
The figure presents the simulation result of the tracking performance, tracking error and the control input signals.



(a)Tracking performance.

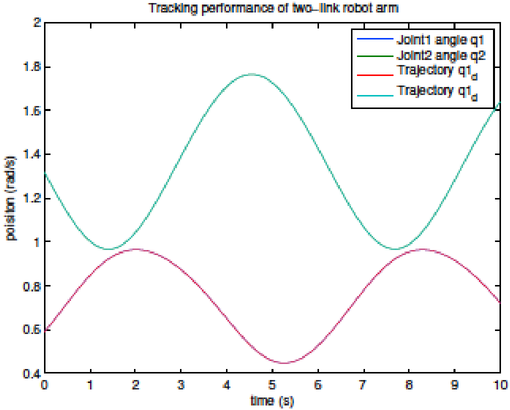


(b) Tracking error.

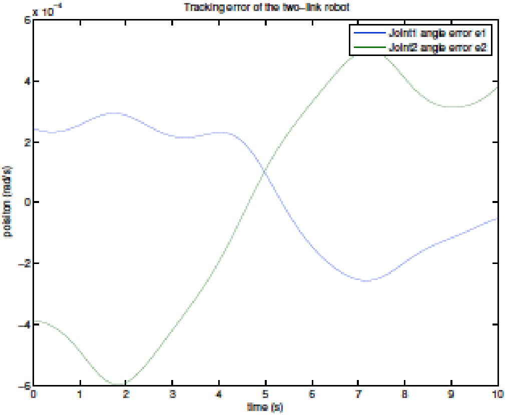


(c) Control input u.

Tracking performance

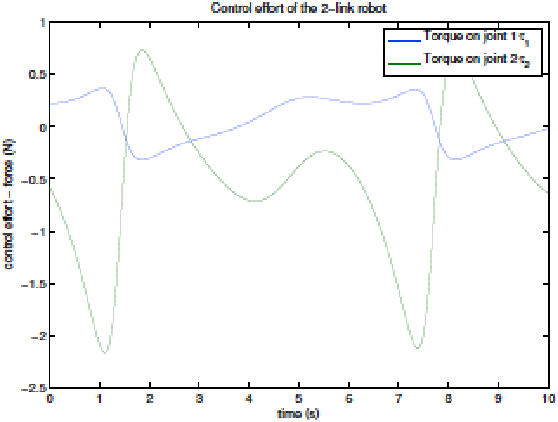


Tracking error



The maximum tracking error during the cycle is 6×10^{-4}.

Control input



Summing up, these simulations validated the effectiveness of the LP-based control for a smooth two-link robot arm system.