

---

# Coupling MPC and HJB for the Computation of POD-based Feedback Laws

Giulia Fabrini  
Maurizio Falcone  
Stefan Volkwein

---

Konstanzer Schriften in Mathematik

Nr. 371, Dezember 2017

ISSN 1430-3558

---

*Konstanzer Online-Publikations-System (KOPS)*  
URL: <http://nbn-resolving.de/urn:nbn:de:bsz:352-2-oamlm1pravmr2>



# Coupling MPC and HJB for the Computation of POD-based Feedback Laws

Giulia Fabrini<sup>1</sup>, Maurizio Falcone<sup>2</sup> and Stefan Volkwein<sup>1</sup>

<sup>1</sup> University of Konstanz, Department of Mathematics and Statistics,  
Universitätsstraße 10, D-78457 Konstanz, Germany

`Giulia.Fabrini@uni-konstanz.de`\*\*\*

<sup>2</sup> La Sapienza Università di Roma, Dipartimento di Matematica, Roma, Italy

**Abstract.** In this paper we use a reference trajectory computed by a model predictive method to shrink the computational domain where we set the Hamilton-Jacobi Bellman (HJB) equation. Via a reduced-order approach based on proper orthogonal decomposition (POD), this procedure allows for an efficient computation of feedback laws for systems driven by parabolic equations. Some numerical examples illustrate the successful realization of the proposed strategy.

## 1 Introduction

The numerical solution of nonlinear optimal control problems for system driven by partial differential equations is a challenging topic that has a great impact in many areas. By means of the Dynamic Programming Principle (DPP) one can obtain a Hamilton-Jacobi-Bellman equation which gives a characterization of the value function of a fully-nonlinear control problem. The value function is in fact the unique viscosity solution of a nonlinear Hamilton-Jacobi equation and solving this equation one can derive the approximation of an optimal feedback control. It is well known that the DP approach suffers from the so called *curse of dimensionality* and one of the main tasks is the choice of the domain where we want compute the value function. To solve this problem we apply the algorithm presented in [1] to a parabolic partial differential equation. We follow a reduced order modeling approach based on POD and we derive an *a-posteriori* error estimator for the optimal trajectory. The paper is organized as follows. Section 2 is devoted to the presentation of the optimal control problem. In Section 3 we introduce the reduced order modeling whereas in Section 4 we briefly explain the main features of HJB and MPC method, how we couple them and we provide an a posteriori error estimator which give us a criterium to choose the reduced domain. Finally, Section 5 presents a couple of numerical tests.

---

\*\*\* G. Fabrini gratefully acknowledges support by the German Science Fund DFG grant *Reduced-Order Methods for Nonlinear Model Predictive Control*.

## 2 The optimal control problem

Assume that  $\Omega = (0, 1)^2 \subset \mathbb{R}^2$  with boundary  $\Gamma = \partial\Omega$ . We set  $Q = (0, \infty) \times \Omega$  and  $\Sigma = (0, \infty) \times \Gamma$ . Moreover, let  $H = L^2(\Omega)$  and  $V = H^1(\Omega)$  with dual  $V'$ . Recall that the space

$$W(0, \infty) = \{\varphi \in L^2(0, T; V) \mid \varphi_t \in L^2(0, \infty; V')\}$$

is a Hilbert space endowed with the common inner product; cf. [3]. Let

$$U_{\text{ad}} = \{u \in \mathbb{R} \mid u_{\text{a}} \leq u \leq u_{\text{b}}\} \subset \mathbb{R},$$

where  $u_{\text{a}}, u_{\text{b}} \in \mathbb{R}^m$  are lower and upper bounds, respectively. For  $\mathbb{U} = L^2(0, \infty; \mathbb{R})$  the set of admissible control is given by

$$\mathbb{U}_{\text{ad}} = \{u = u \in \mathbb{U} \mid u(t) \in U_{\text{ad}} \text{ a.e. in } [0, \infty)\}.$$

where ‘a.e.’ stands for ‘almost everywhere’. For a given  $y_0 \in H$ ,  $b \in L^2(\Gamma)$ ,  $u \in \mathbb{U}_{\text{ad}}$  and  $\nu > 0$  the *state equation* is

$$y_t = \nu \Delta y \text{ in } Q \text{ a.e.}, \quad \nu \frac{\partial y}{\partial \mathbf{n}} = ub \text{ on } \Sigma \text{ a.e.}, \quad y(0) = y_0 \text{ in } \Omega \text{ a.e.} \quad (1)$$

Introducing the continuous bilinear form

$$a(\varphi, \psi) = \nu \int_{\Omega} \nabla \varphi \cdot \nabla \psi \, d\mathbf{x}, \quad \text{for } \varphi, \psi \in V$$

we can express (1) in the form

$$\begin{aligned} \frac{d}{dt} \langle y(t), \varphi \rangle_H + a(y(t), \varphi) &= u(t) \langle b, \varphi \rangle_{L^2(\Gamma)} \quad \forall \varphi \in V, \text{ f.a.a. } t > 0, \\ \langle y(0), \varphi \rangle_H &= \langle y_0, \varphi \rangle_H \quad \forall \varphi \in H, \end{aligned} \quad (2)$$

where ‘f.a.a.’ stands for ‘for almost all’. It is well-known that (2) admits a unique solution  $y \in \mathbb{Y}$ ; (see, e.g., [3]).

We introduce the Hilbert space  $\mathbb{X} = \mathbb{Y} \times \mathbb{U}$  endowed with common product topology. The set of admissible points is given as

$$\mathbb{X}_{\text{ad}} = \{x = (y, u) \in \mathbb{X} \mid y \text{ solves (2) for } u \in \mathbb{U}_{\text{ad}}\}.$$

For a given discount factor  $\lambda > 0$ , positive weight  $\sigma$  and desired state  $y_d \in H$  the quadratic cost is defined as

$$J(y, u) = \frac{1}{2} \int_0^\infty \left( \|y(t) - y_d\|_H^2 + \sigma |u(t)|^2 \right) e^{-\lambda t} dt, \quad \text{for } (y, u) \in \mathbb{X}.$$

Now, we consider the infinite horizon, linear-quadratic optimal control problem

$$\min J(x) \quad \text{subject to} \quad x \in \mathbb{X}_{\text{ad}}. \quad (\mathbf{P})$$

### 3 Reduced-order modeling (ROM)

We discretize  $(\mathbf{P})$  by using proper orthogonal decomposition (POD) (see [7,8] for details). For large  $T > 0$  and  $K \in \mathbb{N}$  let  $\{y^k(t)\}_{k=1}^K \subset L^2(0, T; V)$  be some given trajectories which will be specified later. Then, we consider the linear space of snapshots

$$\mathcal{V} = \text{span} \{y^k(t) \mid t \in [0, T] \text{ and } 1 \leq k \leq K\} \subset V \quad \text{with } d = \dim \mathcal{V} \leq \infty.$$

For any finite  $\ell \leq d$  we are interested in determining a POD basis of rank  $\ell$  which minimizes the mean square error between the trajectories  $\{y^k(t)\}_{k=1}^K$  and their corresponding  $\ell$ -th partial Fourier sums on average in  $[0, T]$ :

$$\begin{cases} \min \sum_{k=1}^K \int_0^T \left\| y^k(t) - \sum_{i=1}^{\ell} \langle y^k(t), \psi_i \rangle_V \psi_i \right\|_V^2 dt \\ \text{s.t. } \{\psi_i\}_{i=1}^{\ell} \subset V \text{ and } \langle \psi_i, \psi_j \rangle_V = \delta_{ij} \text{ for } 1 \leq i, j \leq \ell. \end{cases} \quad (3)$$

Assuming that we have computed a POD basis  $\{\psi_i\}_{i=1}^{\ell}$ , we define  $V^{\ell} = \text{span} \{\psi_1, \dots, \psi_{\ell}\} \subset V$ . Then, the POD state  $y^{\ell}(t) = \sum_{i=1}^{\ell} y_i(t) \psi_i \in V^{\ell}$  is determined by the POD Galerkin scheme for (2):

$$\begin{aligned} \frac{d}{dt} \langle y^{\ell}(t), \psi_j \rangle_H + a(y^{\ell}(t), \psi) &= u(t) \langle b, \psi \rangle_{L^2(\Gamma)} \quad \forall \psi \in V^{\ell}, \text{ f.a.a. } t > 0, \\ \langle y^{\ell}(0), \psi \rangle_V &= \langle y_0, \psi \rangle_H \quad \forall \psi \in V^{\ell}. \end{aligned} \quad (4)$$

We define the vectors

$$y(t) = (y_i^{\ell}(t)) \in \mathbb{R}^{\ell}, \quad y_0 = (\langle y_0, \psi_i \rangle_H) \in \mathbb{R}^{\ell}, \quad b = (\langle b, \psi_i \rangle_{L^2(\Gamma)}) \in \mathbb{R}^{\ell}$$

and the matrices  $M, A \in \mathbb{R}^{\ell \times \ell}$  by

$$M_{ij} = \langle \psi_j, \psi_i \rangle, \quad A_{ij} = -a(\psi_j, \psi_i).$$

Setting  $F(y, u) = M^{-1}(Ay + ubu) \in \mathbb{R}^{\ell}$ ,  $(y, u) \in \mathbb{R}^{\ell} \times \mathbb{R}$ , we can write (4) as

$$\dot{y}(t) = F(y(t), u(t)) \text{ f.a.a. } t > 0, \quad y(0) = y_0 \quad (5)$$

which is a  $\ell$ -dimensional system of differential equations. Let us also define

$$y_d = \langle y_d, \psi_i \rangle_H \in \mathbb{R}^{\ell}, \quad L(y, u) = (y^{\top} M y + y^{\top} y_d + \sigma u^2)/2$$

for  $(y, u) \in \mathbb{R}^{\ell} \times \mathbb{R}$ . Then, the POD Galerkin approximation for  $(\mathbf{P})$  reads

$$\min J^{\ell}(y, u) = \int_0^{\infty} L(y(t), u(t)) e^{-\lambda t} dt \quad \text{s.t. } (y, u) \in \mathbb{X}_{\text{ad}}^{\ell} \quad (\mathbf{P}^{\ell})$$

where the admissible set is

$$\mathbb{X}_{\text{ad}}^{\ell} = \{(y, u) \in \mathbb{X}^{\ell} \mid y \text{ solves (5) for } u \in \mathbb{U}_{\text{ad}}\}$$

and  $\mathbb{X}^{\ell} = H^1(0, \infty; \mathbb{R}^{\ell}) \times \mathbb{U}$ .

*Remark 1.* In our numerical realization of the POD Galerkin scheme we first have to introduce a high-fidelity discretization for (2). We use a discretization based on piecewise finite elements (FE) with  $n \gg \ell$  degrees of freedom.  $\diamond$

## 4 Coupling MPC and HJB

For the solution of  $(\mathbf{P}^\ell)$  we use the algorithm proposed in [1] and we provide a new a posteriori error-estimate.

**HJB equation.** Let us briefly recall the HJB equation (see e.g. [2,5]). For  $y_\circ \in \mathbb{R}^\ell$  let us introduce the value function

$$v^\ell(y_\circ) = \inf_{u \in \mathbb{U}_{\text{ad}}} \left\{ \min J^\ell(y, u) \mid \dot{y}(t) = F(y(t), u(t)) \text{ f.a.a. } t > 0, y(0) = y_\circ \right\}.$$

The value function can be characterized in terms of the *Bellman equation*

$$\lambda v^\ell(y_\circ) + \max_{u \in \mathbb{U}_{\text{ad}}} \left\{ -F(y_\circ, u)^\top \nabla v^\ell(y_\circ) - L(y_\circ, u) \right\} = 0, \quad \text{for all } y_\circ \in \mathbb{R}^\ell.$$

In the numerical realization we have to replace  $\mathbb{R}^\ell$  by a sufficiently large, but bounded subset  $\mathcal{D} \subset \mathbb{R}^\ell$ . The main advantage of this approach is that once the value function  $v^\ell$  has been computed, the optimal feedback control is

$$\bar{u}(y_\circ) = \arg \min_{u \in \mathbb{U}_{\text{ad}}} \left\{ -F(y_\circ, u)^\top \nabla v^\ell(y_\circ) - L(y_\circ, u) \right\} \quad \text{for all } y_\circ \in \mathcal{D}$$

which can be used as a closed-loop control. The optimal state solves (5) for the feedback control  $u = \bar{u}(y(\cdot))$ . For the numerical solution of the HJB equation we consider a fully-discrete semi-Lagrangian scheme which is based on the discretization of the system dynamics with time step  $h$  and a mesh parameter  $k$ , leading to a fully discrete approximation  $v_{hk}^\ell(y_\circ)$ :

$$v_{hk}^\ell(y_{\circ i}) = \min_{u \in \mathbb{U}_{\text{ad}}} \left\{ (1 - \lambda h) I[v_{hk}^\ell](y_{\circ i} + hF(y_{\circ i}, u)) + L(y_{\circ i}, u) \right\}$$

for every node  $y_{\circ i} \in \mathcal{D}$ ,  $i = 1, \dots, N_p$ , of the discretized state space. Note that in general, the arrival point  $y_{\circ i} + hF(y_{\circ i}, u)$  is not a node of the space grid. Therefore, this value is computed by means of a linear interpolation operator, denoted by  $I[v_{hk}^\ell]$ . The bottleneck of this approach is related to the so-called *curse of the dimensionality*, namely, the computational cost increases dramatically as soon as the dimension  $\ell$  does.

**Model predictive control (MPC).** MPC is an optimization based method for the computation of closed-loop controls for (non-)linear dynamical systems (see [6,9] for details). It consists in solving iteratively finite time horizon open-loop problems. Let  $N \in \mathbb{N}$ ,  $t_\circ \geq 0$  and  $t_\circ^N = t_\circ + N\Delta t$  for a chosen time step  $\Delta t > 0$ . We introduce the finite time horizon cost as

$$J^{\ell, N}(y_{[t_\circ, u]}, u) = \int_{t_\circ}^{t_\circ^N} L(y_{[t_\circ, u]}(t), u(t)) e^{-\lambda t} dt,$$

where the state  $y = y_{[t_\circ, u]}$  solves  $\dot{y}(t) = F(y(t), u(t))$  f.a.a.  $t \in (t_\circ, t_\circ^N]$  and  $y(t_\circ) = y_\circ$ .

The method works as follows: For  $t_o = 0$  and  $y_o = y_0$  we minimize  $J^{\ell, N}$  over  $[t_o, t_o^N)$  and store the optimal control  $\bar{u}^1$  on the subinterval  $(t_o, t_o + \Delta t]$  together with the associated optimal state  $y_{[t_o, \bar{u}^1]}$ . Then, we initialize the next finite time horizon control problem by setting  $y_o = y_{[t_o, \bar{u}^1]}(t_o + \Delta t)$ ,  $t_o = t_o + \Delta t$  and  $t_o^N = t_o + N\Delta t$ . This process is iterated.

**Coupling MPC and HJB.** The idea is to combine the advantages from both methods: HJB is global and gives the feedback law for every initial condition once the value function has been computed. On the other hand, MPC is faster, but gives an approximate feedback control just for a single initial condition. Let us assume that we are interested in the approximation of feedback controls for an optimal control problem given the initial condition  $y_0$ . We compute via MPC a reasonable suboptimal trajectory  $\bar{y}^N$  that we can use as reference trajectory for building a small domain, where we are going to compute the approximate value function. In our approach, we choose a rather *short* prediction horizon  $N\Delta t$  to obtain  $\bar{y}^N$  quickly. Then, the trajectory  $\bar{y}^N$  is used to build the smaller domain  $\mathcal{D}_\rho \subset \mathcal{D}$ , in which we solve the HJB equation (instead of solving the HJB on the whole domain  $\mathcal{D}$ ). We construct  $\mathcal{D}_\rho$  as a tube around  $\bar{y}^N$  defining

$$\mathcal{D}_\rho = \{y_o \in \mathbb{R}^\ell \mid \text{dist}(y_o, \bar{y}^N) \leq \rho\} \quad (6)$$

where  $\text{dist}(\cdot, \cdot)$  will be specified later. A larger  $\rho$  will allow for a better approximation of the value function, but at the same time enlarging  $\rho$  we will loose the localization around our trajectory, increase the number of nodes and the CPU time.

**Error Estimator.** Let  $\bar{x} = (\bar{y}, \bar{u})$  be the solution to **(P)** and  $T > 0$  sufficiently large. Due to the Bellman principle,  $\bar{u}$  is also optimal on  $[0, T]$ . Suppose that we have computed an approximate MPC control  $\bar{u}^N \in L^2(0, T)$  satisfying  $\bar{u}^N(t) \in U_{\text{ad}}$  in  $[0, T]$  a.e. Then, we usually have  $\bar{u} \neq \bar{u}^N$ . Let  $\bar{y}^N$  be the associated approximate MPC state solving (5) on  $[0, T]$  for  $u = \bar{u}^N$ . To evaluate the *a-posteriori* error estimator we have to compute the Lagrange multiplier  $p^N$  solving the dual equation

$$-M\dot{p}^N(t) = A^\top p^N(t) - e^{-\lambda t}(M\bar{y}^N(t) + y_d) \text{ for } t \in [0, T), \quad p^N(T) = 0.$$

Then, it follows from [10] that

$$\|\bar{u} - \bar{u}^N\|_{L^2(0, T)} \leq \frac{e^{\lambda T}}{\sigma} \|\zeta\|_{L^2(0, T)},$$

where the perturbation  $\zeta \in \mathbb{U}(T)$  is defined as:

$$\zeta(t) = \begin{cases} [(\sigma e^{-\lambda t} \bar{u}^N(t) - b^\top p^N(t))]_- & \text{in } \mathcal{A}_a = \{s \in [0, T] \mid \bar{u}^N(s) = u_a\}, \\ [(\sigma e^{-\lambda t} \bar{u}^N(t) - b^\top p^N(t))]_+ & \text{in } \mathcal{A}_b = \{s \in [0, T] \mid \bar{u}^N(s) = u_b\}, \\ -(\sigma e^{-\lambda t} \bar{u}^N(t) - b^\top p^N(t)) & \text{in } \mathcal{J} = [0, T] \setminus (\mathcal{A}_a \cup \mathcal{A}_b). \end{cases}$$

Here,  $[w]_-$  and  $[w]_+$  denote respectively the negative and positive part of  $w$ . Since  $\bar{y}$  solves (5) on  $[0, T]$  for  $u = \bar{u}$ , we have

$$\frac{1}{2} \frac{d}{dt} \|(\bar{y} - \bar{y}^N)(t)\|_M^2 + \|(\bar{y} - \bar{y}^N)(t)\|_A \leq \|b\|_2 \|(\bar{y} - \bar{y}^N)(t)\|_2 |(\bar{u} - \bar{u}^N)(t)|$$

f.a.a.  $t \in [0, T]$ , where, e.g.,  $\|\cdot\|_M$  is the weighted norm in  $\mathbb{R}^\ell$  induced respectively by the positive definite matrix  $M$  and  $\|\cdot\|_2$  stands for the Euclidean norm. By applying Young's inequality we derive the existence of a constant  $c_\ell > 0$ , which depends on  $b$ ,  $M$  and  $A$ , so that

$$\frac{1}{2} \frac{d}{dt} \|(\bar{y} - \bar{y}^N)(t)\|_M^2 \leq c_\ell |(\bar{u} - \bar{u}^N)(t)|^2 \quad \text{f.a.a. } t \in [0, T]$$

which implies

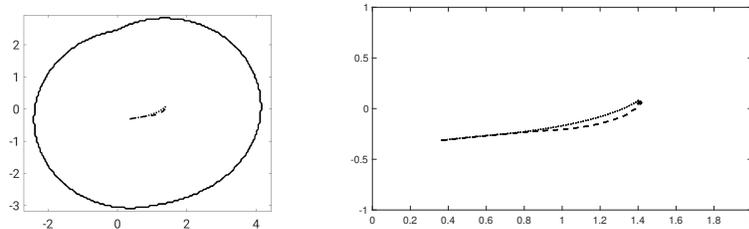
$$\max_{t \in [0, T]} \|(\bar{y} - \bar{y}^N)(t)\|_M \leq \frac{e^{\lambda T} \sqrt{c_\ell}}{\sigma} \|\zeta\|_{L^2(0, T)}. \quad (7)$$

Now, to define  $\mathcal{D}_\rho$  we set  $\text{dist}(y_\circ, \bar{y}^N) = \|y_\circ - \bar{y}^N\|_M$  and  $\rho = e^{\lambda T} \sqrt{c_\ell} \|\zeta\|_{L^2(0, T)} / \sigma$ .

## 5 Numerical Test

We consider equation (2) with  $\nu = 1$ ,  $b = \chi_\Gamma$  and  $y_0 = 0.5\chi_{[0.5, 1]}$ . For the cost functional we take  $\sigma = 0.01$ ,  $\lambda = 1$  and  $y_d = 1$ . To realize the reduced-approach numerically, we have to choose a high-fidelity spatial approximation. We apply a piecewise linear finite element (FE) model with  $n = 2673$  degrees of freedom. The snapshots for the POD method are chosen as the solution to the state and dual equation for the reference control  $u = 1$  and for  $T = 1$ . The POD basis rank chosen for the simulations is  $\ell = 2$ .

**Test 1.** The chosen discretization parameters are  $\Delta t = h = 0.02$ ,  $\Delta \tau = 0.01$  (the time step to integrate the trajectories). We discretize the set  $U_{\text{ad}} = [-4, 4]$  in 21 equidistant discrete values for the approximation of the value function and 81 for the computation of the state trajectories. In Figure 1



**Fig. 1.** Test 1: domain  $\mathcal{D}_\rho$  with optimal trajectories (left panel). The latter, optimal trajectory via MPC (dotted line) and via HJB (dashed line), are detailed in the right panel.

we can see on the left the first approximation of the MPC solver (dotted line) and the optimal solution corresponding to the feedback control (dashed line) in  $\mathcal{D}_\rho$ , while on the right we show the same trajectories in a smaller

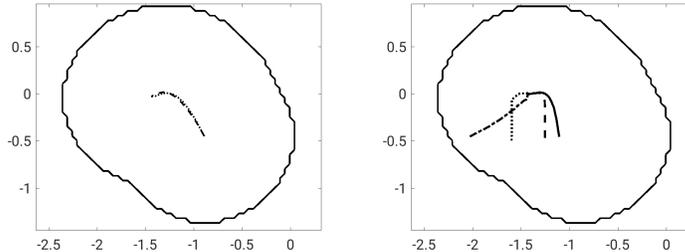
portion of  $\mathcal{D}_\rho$ , centered around them. In Table 1, we present the CPU times and distance of the controlled trajectory from the reference trajectory in  $L^2$ -norm at the final time. We can observe that the trajectory obtained via

|                          | MPC ( $N = 3$ ) | HJB in $\mathcal{D}_\rho$ | HJB in $\mathcal{D}$ |
|--------------------------|-----------------|---------------------------|----------------------|
| CPU time in [s]          | 11              | 25                        | 50                   |
| $\ \bar{y}(T) - y_d\ _M$ | 0.07            | 0.03                      | 0.03                 |

**Table 1.** Test 1: CPU times and distance of the controlled trajectory from the reference trajectory at the final time  $T = 1$ .

MPC is a rough approximation; we improve the result with a localized version of the DP and the trajectory obtained is closer to the target. Moreover if we compare the trajectories computed using the information given by the value function in  $\mathcal{D}_\rho$  and  $\mathcal{D}$  we have the same values for  $\|\bar{y}(T) - y_d\|_M$ . Concerning the CPU time, in the fourth column we show the global time needed to get the approximation of the value function in the whole domain and the time to obtain the optimal trajectory, whereas the third column shows the global time needed to compute all the steps of our algorithm. We can observe a speed up of factor 2.

**Test 2.** In this test we consider the smoother initial state  $y_o(\mathbf{x}) = \sin(\pi\mathbf{x})$ , the parameters chosen are the same of Test 1. Now we have a speed up of factor 3, as shown in Table 2. On the other hand we are able to steer the solution to the target  $y_d$  in  $\mathcal{D}_\rho$  and  $\mathcal{D}$ . One of the advantages of computing



**Fig. 2.** Test 1: optimal trajectory via MPC (dotted line) and via HJB (dashed line) in  $\mathcal{D}_\rho$ , optimal trajectories for different initial points (right)

the value function in  $\mathcal{D}_\rho$  is that we are able to reconstruct the feedback in all the points of the domain with a small computational effort. So we can change the initial state (with the constraint that the state projected in the reduced space has to be a point in  $\mathcal{D}_\rho$ ) and compute the optimal trajectory in a fast

|   | MPC ( $N = 3$ ) | HJB in $\mathcal{D}_\rho$ | HJB in $\mathcal{D}$ |
|---|-----------------|---------------------------|----------------------|
| CPU time in [s]                           | 10s             | 18s                       | 58 s                 |
| $\ \bar{y}(T) - y_d\ _{L^2(\mathcal{D})}$ | 0.04            | 0.01                      | 0.01                 |

**Table 2.** Test 2: CPU times and distance of the controlled trajectory from the reference trajectory at the final time  $T = 1$ .

way ( $\simeq 4s$ ). To this purpose we consider some perturbations of the initial state  $y_\circ$ , we project the system in the reduced space and we compute the optimal trajectories, in Figure 2 (right) we show the results for four different initial conditions. On the other hand if we want to apply MPC we have to start the whole procedure again, and the CPU time required is two times bigger. In the left plot of Figure 2 we show the trajectories in the reduced domain.

## References

1. A. Alla, G. Fabrini and M. Falcone, *Coupling MPC and DP methods for an efficient solution of optimal control problems*. Conference Proceedings of IFIP 2015.
2. M. Bardi, I. Capuzzo Dolcetta. *Optimal control and viscosity solutions of Hamilton-Jacobi-Bellman equations*. Birkhauser, Basel, 1997.
3. R. Dautray and J.-L. Lions. *Mathematical Analysis and Numerical Methods for Science and Technology. Volume 5: Evolution Problems I*. Springer-Verlag, Berlin, 2000.
4. L.C. Evans *Partial Differential Equations*, American Mathematical Society, Providence, Rhode Island (2008).
5. M. Falcone and R. Ferretti, *Semi-Lagrangian Approximation schemes for linear and Hamilton-Jacobi equations*, SIAM, 2013.
6. L. Grüne and J. Pannek, *Nonlinear Model Predictive Control*, Springer, London, 2011.
7. M. Gubisch and S. Volkwein: Proper orthogonal decomposition for linear-quadratic optimal control. P. Benner, A. Cohen, M. Ohlberger, and K. Willcox (eds.), *Model Reduction and Approximation: Theory and Algorithms*. Pages 5-66, *SIAM*, Philadelphia, PA, 2017.
8. P. Holmes, J.L. Lumley, G. Berkooz, and C.W. Rowley. *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*. Cambridge Monographs on Mechanics. Cambridge University Press, Cambridge, 2nd ed. edition, 2012.
9. J.B. Rawlings and D.Q. Mayne. *Model Predictive Control: Theory and Design*. Nob Hill Publishing, Madison, Wisconsin, 2009.
10. F. Tröltzsch and S. Volkwein: POD a-posteriori error estimates for linear-quadratic optimal control problems. *Computational Optimization and Applications*, 44:83-115, 2009.