Data-Centric Engineering (2022), 3: e6
doi:10.1017/dce.2022.6

CAMBRIDGE UNIVERSITY PRESS

RESEARCH ARTICLE

An end-to-end data-driven optimization framework for constrained trajectories

Florent Dewez[1], Benjamin Guedj[1,2,*], Arthur Talpaert[1] and Vincent Vandewalle[1,3]

[1]MODAL—MOdels for Data Analysis and Learning, Inria—Lille—Nord Europe Research Centre, Villeneuve d'Ascq, France
[2]Department of Computer Science, Centre for Artificial Intelligence, University College London, London, United Kingdom
[3]ULR 2694 Evaluations des Technologies de Santé et des Pratiques Médicales, CHU Lille, University of Lille, Lille, France
*Corresponding author. E-mail: benjamin.guedj@inria.fr

Received: 17 August 2021; **Revised:** 24 January 2022; **Accepted:** 11 February 2022

Keywords: Constrained optimization; functional data; statistical modeling

Abstract

Many real-world problems require to optimize trajectories under constraints. Classical approaches are often based on optimal control methods but require an exact knowledge of the underlying dynamics and constraints, which could be challenging or even out of reach. In view of this, we leverage data-driven approaches to design a new end-to-end framework which is dynamics-free for optimized and realistic trajectories. Trajectories are here decomposed on function basis, trading the initial infinite dimension problem on a multivariate functional space for a parameter optimization problem. Then a maximum a posteriori approach which incorporates information from data is used to obtain a new penalized optimization problem. The penalized term narrows the search on a region centered on data and includes estimated features of the problem. We apply our data-driven approach to two settings in aeronautics and sailing routes optimization. The developed approach is implemented in the Python library `PyRotor`.

Impact Statement

We present a generic method to optimize trajectories (in a broad sense) under constraints which leverages trajectory data stemming from many data-intensive industries. Through statistical modeling, we include information inferred from data, such as degrees of freedom, constraints, or even correlations between some covariates, into the optimization problem of interest. This restricts in a data-driven way the search space, hence drastically reducing the computational complexity and avoiding to resort to manual editing for dynamics or constraints. While generic, we show it to be of direct interest to two specific settings, namely aeronautics and sailing. The generic nature of the approach motivates further studies to many different data-centric engineering frameworks and industries.

1. Introduction

Optimizing trajectories under constraints appears in many real-world problems. The present paper stems from an initial work on aeronautics and the quest for designing fuel efficient aircraft trajectories based on available flight data. We have reached a generic data-driven methodology which falls in the much broader field of trajectory optimization under constraints. As such, it has potential applications to many more real

world problems, such as in robotics to minimize the work-based specific mechanical cost of transport (Srinivasan and Ruina, 2006) or in aerospace to reduce the total thermal flux when a space shuttle re-enters in the atmosphere (Trélat, 2012).

In aeronautics, optimization problems such as the minimization of the total travel time or the fuel reduction are often formulated in terms of optimal control problems (Codina and Menéndez, 2014; Girardet et al., 2014; Cots et al., 2018). This allows to take into account the dynamics of the system, leading to realistic solutions complying with additional constraints; we refer to Rao (2009) for an overview.

Nevertheless the differential equations describing the dynamics of the system of interest may be (partially) unknown. For instance, the differential system describing the motion of an aircraft moving in an air mass (Rommel et al., 2019) involves the lift and drag forces for which no analytic formulas exist. Aircraft manufacturers typically compute numerical models (not publicly released) by means of heavy simulations and wind tunnel tests. Another approach consists in reconstructing unknown forces based on physical formulas and available flight data; see for instance Rommel et al. (2017) and Dewez et al. (2020) for results in aeronautics and Ramsay et al. (2007) in the generic setting of parameter estimation for differential equations. While promising on paper, this reconstruction step requires restrictive assumptions and the statistical errors may impact strongly the solution of the optimal control problem. Moreover it does not tackle directly the optimization problem.

At the same time, additional safety and air control constraints should be taken into account and modelled so that the optimized trajectory is acceptable for both pilots and air controllers. However such constraints may be numerous and complex to model; we refer to Codina and Menéndez (2014) and Lim et al. (2019) for examples of such constraints. Due to the short time for the flight preparation on ground, the execution time to solve numerically such constrained optimization problems may be unacceptable, in particular in the case of nonlinear constraints.

In our work, we propose another kind of approach to provide efficiently realistic trajectories without involving noisy dynamical systems and numerous complex constraints. This is achieved by leveraging available trajectory data. Our approach lies mainly on the estimation of the trajectory distribution, which is assumed to contain intrinsically many information on realistic trajectories. This is then incorporated in the optimization problem of interest through a Bayesian approach, constraining then the problem by the data in a simple and natural way. The main benefit on this approach is that it directly uses the information contained in the data, requiring no explicit information on the dynamics or on additional constraints. This methodology is specific to the situation where the user has access to trajectory data but, at the same time, the approach is intended to be generic enough so that it can be exploited in a wide range of applications. In particular it is certainly not restricted to the aeronautic setting.

The idea of using data to improve optimization processes has been for instance validated by the paper Hewitt and Frejinger (2020) in the context of decision support systems. In this paper, the authors are interested in learning mathematical representations of business rules for mixed integer linear programs. Their work is motivated by the development of automatic processes for the implementations of rules given past decision data. Their numerical test cases have shown that such an approach can lead to high-quality decisions with respect to their objective function value while being able to model effectively rules contained in the data. We mention that, apart from the nature of the optimization problems and their applications, the main difference between the methodology in Hewitt and Frejinger (2020) and ours is the way we exploit the data: they learn a map sending theoretical optimized decisions to the associated past ones while we incorporate directly estimated features from the data into the optimization problem. In particular, a comparison between these two approaches falls out the scope of this paper. Let us also mention that another strategy could be to leverage purely data-driven reinforcement learning to provide a trajectory (see e.g., Berkenkamp et al., 2017; Mowbray et al., 2021), however, at a considerably higher computational cost.

Let us now give some details on our methodology from a technical point of view. We first assume that all the trajectories belong to a finite-dimensional space, which allows to reduce the complexity of the problem with low information loss for a well-chosen basis. In a Bayesian framework, we assume secondly

that the prior distribution of trajectories (through their related coefficients) is proportional to a decreasing exponential function of the cost, assuming that efficient trajectories are a priori more likely than inefficient ones. In our pipeline, the cost function can be especially learnt from the data if necessary. Thirdly we estimate the likelihood distribution which is expected to contain information on realistic trajectories. Here, the observed trajectories, that we call *reference* trajectories, are interpreted as noisy observations of an efficient one, the noise following a centered Gaussian multivariate distribution. In a Bayesian perspective, it is thus possible to deduce the posterior distribution of the efficient trajectory given the reference trajectories and we focus finally on the mode of this posterior for the sake of simplicity. Under our assumptions, the new objective function involves here the sum between the cost of a trajectory and its squared Mahalanobis distance to a weighted average of reference trajectories. In particular, the resulting optimization problem can be interpreted as a penalized one together with some affine constraints, modeling for instance initial and final conditions.

The role of the likelihood distribution, leading to the penalized term, is to force the solution to be close to real trajectories and so to be likely to comply with the constraints. The strength of the penalization is here controlled by a hyper-parameter and a tuning process is proposed to find an optimal balance between optimization and closeness to the trajectory distribution. Hence, the optimized trajectory may inherit a realistic behavior, even though the dynamics are not explicitly taken into account in our problem.

We mention that the present Gaussian assumption for the likelihood distribution has two advantages. First, it reduces the information on trajectories to the mean trajectory and the covariance matrix, making the results interpretable for experts. In particular, this matrix not only indicates the most unconstrained directions for the optimization, but also reveals linear relations between variables, some of them reflecting the dynamics or being unknown by the user. Second, the Gaussian assumption leads to a penalized term which is actually quadratic. So in certain cases, the problem is convex with affine constraints, allowing to make use of very efficient optimization algorithms.

In a nutshell, this data-driven approach restricts the search space to a region centered on the data in a metric space reflecting features estimated from the data. Further, it is flexible enough to cover not only Gaussian distributions, but also other families of distributions for other kinds of applications. Finally, it is noteworthy that, despite the above hyper-parameter tuning process, the optimized trajectory resulting from our approach may not comply with all the complex constraints of a given real problem, making it unacceptable in practice. To circumvent this issue, one could use for instance our not perfect trajectory as an initial guess in iterative (nonlinear) optimization solvers, which could at the end reduce drastically the number of steps while providing a trajectory fulfilling all the requirements.

1.1. Outline

We describe our approach in Section 2, and briefly discuss its Python implementation (the library `PyRotor`) in Section 3. Sections 4 and 5 are devoted to applications: the first one to the fuel reduction of aircraft during the climb and the second one to the maximization of the work of a force field along a path. We finish the paper by discussing on future works to improve and generalize our optimization methodology.

2. An End-to-End Optimization Workflow Based on Observed Trajectories

We are interested in finding a trajectory $y^\star$ which minimizes a certain cost function F, namely a solution of the following optimization problem:

$$\widetilde{y^\star} \in \underset{y \in \mathcal{A}_{\mathcal{G}}(y_0, y_T)}{\arg\min} \; F(y). \tag{1}$$

The set $\mathcal{A}_{\mathcal{G}}(y_0, y_T)$, which is defined in Section 2.1, models the constraints the trajectory has to comply with, such that the initial and final conditions or the dynamics. Note that a trajectory is typically a multivariate function defined on an interval and its components are given by states and controls (which are

not distinguished in this paper for the sake of presentation). In case of numerous constraints, which is often the case when dealing with real-world applications, the resulting optimization problem (1) may be computationally expensive. On the other hand, a partial knowledge of the constraints may lead to a solution which is by far unrealistic. Adding by hand user-defined constraints might circumvent this issue but may be time-consuming.

In view of this, we provide in this section our full workflow to obtain a new optimization problem which includes in a natural and simple way constraints coming from the data. This problem is actually designed to provide trajectories which have a realistic behavior.

We begin with elementary but necessary definitions for trajectories and constraints in Section 2.1. We aim at stating the optimization problem in a finite basis space so we define in Section 2.2 the mathematical formalization of how we decompose each trajectory as a projection on such a space. To extract information from the data for the optimization problem, a statistical modeling on the projected space of the available trajectory data is done in Section 2. In Section 2.4, we put everything together to obtain our new optimization problem Section 2.4 via a maximum a posteriori (MAP) approach. Section 2.5 presents a handy computation regarding the cost function in a quadratic case, for the sake of completeness. Additional details can be found in the Appendix A. Section 2.6 focuses on a hyper-parameter tuning for an optimal tradeoff between optimization and additional (nonlinear) constraints. Last but not least, Section 2.7 contains confidence intervals to assess the accuracy of the predicted optimized cost when the cost function is known up to a random noise term. We summarize our methodology in Figure 1. We also present an illustrative representation of our pipeline in Figure 2.

2.1. Admissible trajectories modeling

We start with definitions.

Definition 1 (Trajectory). *Let $T>0$ be a real number and let $D\geqslant 1$ be an integer. Any continuous $\mathbb{R}^D$-valued map y defined on $[0,T]$, that is $y\in C\big([0,T],\mathbb{R}^D\big)$,[1] is called a trajectory over the time interval $[0,T]$. The dth component of a trajectory y will be denoted by $y^{(d)}$. As such, a trajectory is at least a continuous map on a finite interval.*

When optimizing a trajectory with respect to a given criterion, the initial and final states are often constrained, that is to say the optimization is performed in an affine subspace modeling these endpoints conditions. This subspace is now introduced.

Definition 2 (Endpoints conditions). *Let $y_0,y_T\in\mathbb{R}^D$. We define the set $\mathcal{D}(y_0,y_T)\subset C\big([0,T],\mathbb{R}^D\big)$ as*

$$y\in\mathcal{D}(y_0,y_T)\quad\Longleftrightarrow\quad\begin{cases}y(0)=y_0,\\ y(T)=y_T.\end{cases}$$

In many applications, the trajectories have to satisfy some additional constraints defined by a set of (nonlinear) functions. For instance these functions may model physical or user-defined constraints. We define now the set of trajectories verifying such additional constraints.

Definition 3 *(Additional constraints). For $\ell=1,\dots,L$, let g_ℓ be a real-valued function defined on $\mathbb{R}^D$. We define the set $\mathcal{G}\subset C\big([0,T],\mathbb{R}^D\big)$ as the set of trajectories over $[0,T]$ satisfying the following L inequality constraints given by the functions g_ℓ, that is*

$$y\in\mathcal{G}\quad\Longleftrightarrow\quad\forall\ell=1,\dots,L\ \ \forall t\in[0,T]\quad g_\ell(y(t))\leqslant 0.$$

[1] The notation $C\big([0,T],\mathbb{R}^D\big)$ refers to the space of continuous functions over $[0,T]$.

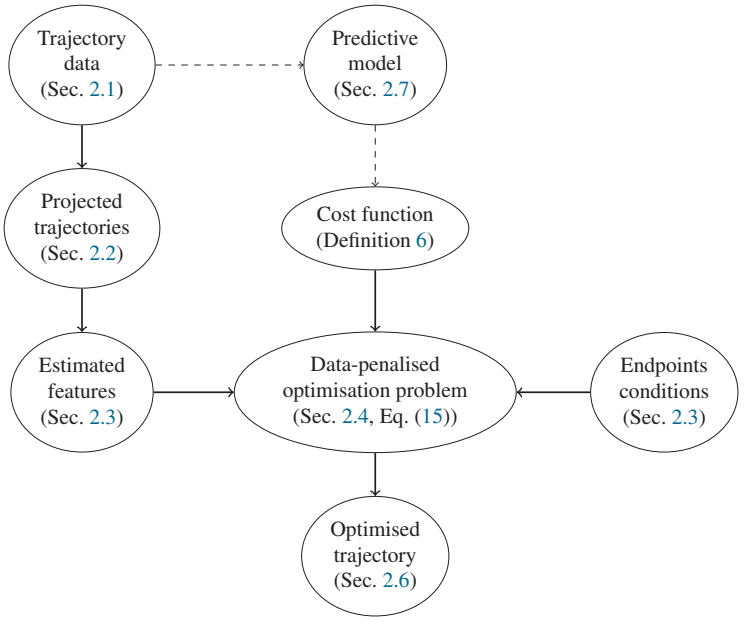

Figure 1. *Diagram of the global pipeline of our method (solid lines). Dashed lines denote optional components.*

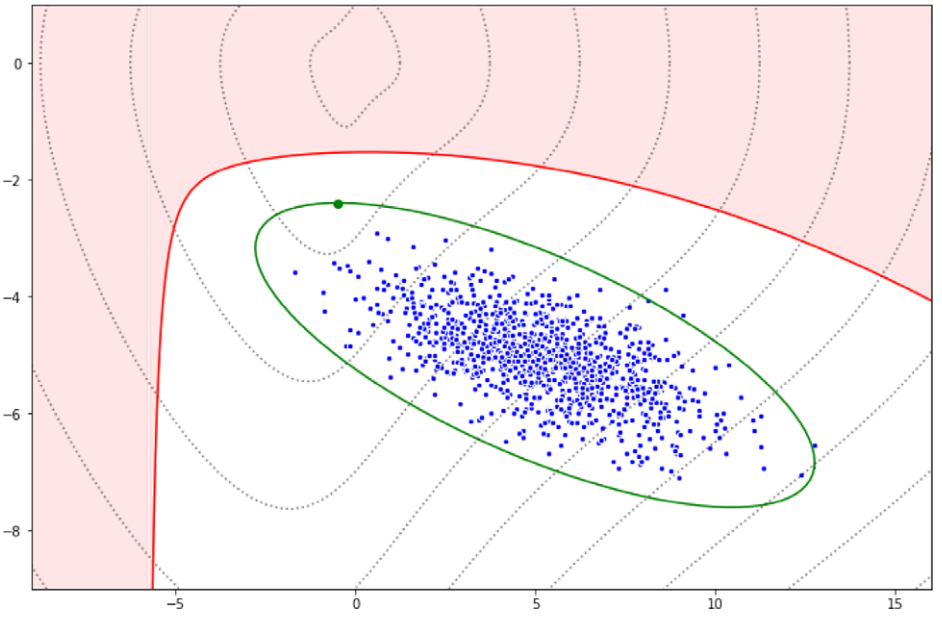

Figure 2. *Illustration of our approach. Blue points refer to reference trajectories, the green ellipse is the set of trajectories which is explored to find an optimized trajectory, the red portion is the set of nonadmissible trajectories (e.g., which do not comply with the set of constraints). Note that the size of the green ellipse is automatically adjusted in the process (as discussed in Section 2.6). Dotted lines are the level sets of the cost function (whose minimum is attained in (0,0)) and the optimized trajectory obtained from our method is given by the green point on the boundary of the ellipse.*

Lastly, we introduce the set of admissible trajectories which satisfy both the endpoints conditions and the additional constraints.

Definition 4 (Admissible trajectory). *We define the set* $\mathcal{A}_{\mathcal{G}}(y_0, y_T) \subset C\left([0,T],\mathbb{R}^D\right)$ *as follows:*

$$\mathcal{A}_{\mathcal{G}}(y_0, y_T) \coloneqq \mathcal{D}(y_0, y_T) \cap \mathcal{G}.$$

Any element of $\mathcal{A}_{\mathcal{G}}(y_0, y_T)$ *will be called an* admissible trajectory.

2.2. Projection for a finite-dimensional optimization problem

In our approach, a theoretical optimization problem in a finite-dimensional space is desired to reduce the inherent complexity of the problem. This can be achieved by decomposing the trajectories on a finite number of basis functions. While raw signals are unlikely to be described by a small number of parameters, this is not the case for smoothed versions of these signals which capture the important patterns. In particular, given a family of smoothed observed trajectories, one may suppose that there exists a basis such that the projection error on a certain number of basis functions of any trajectory is negligible (i.e., the set of projected trajectories in Figure 1).

From now on, the trajectories we consider are assumed to belong to a space spanned by a finite number of basis functions. For the sake of simplicity, we assume in addition that all the components of the trajectories can be decomposed on the same basis but with different dimensions. Extension to different bases is straightforward and does not change our findings but burdens the notation.

Definition 5. *Let* $\{\varphi_k\}_{k=1}^{+\infty}$ *be an orthonormal basis of* $L^2([0,T],\mathbb{R})$[2] *with respect to the inner product*

$$\langle f, g\rangle = \int_0^T f(t)g(t)dt,$$

such that each φ_k *is continuous on* $[0,T]$ *and let* $\mathcal{K} \coloneqq \{K_d\}_{d=1}^{D}$ *be a sequence of integers with* $K \coloneqq \sum_{d=1}^{D} K_d$. *We define the space of projected trajectories* $\mathcal{Y}_{\mathcal{K}}(0,T) \subset C\left([0,T],\mathbb{R}^D\right)$ *over* $[0,T]$ *as*

$$\mathcal{Y}_{\mathcal{K}}(0,T) \coloneqq \prod_{d=1}^{D} span\{\varphi_k\}_{k=1}^{K_d}.$$

If there is no risk of confusion, we write $\mathcal{Y}_{\mathcal{K}} \coloneqq \mathcal{Y}_{\mathcal{K}}(0,T)$ *for the sake of readability.*

Remark 1. *From the above definition, any projected trajectory* $y \in \mathcal{Y}_{\mathcal{K}}$ *is associated with a unique vector*

$$c = \left(c_1^{(1)}, \ldots, c_{K_1}^{(1)}, c_1^{(2)}, \ldots, c_{K_2}^{(2)}, \ldots, c_1^{(D)}, \ldots, c_{K_D}^{(D)}\right)^T \in \mathbb{R}^K$$

defined by

$$c_k^{(d)} \coloneqq \langle y^{(d)}, \varphi_k\rangle = \int_0^T y^{(d)}(t)\varphi_k(t)dt. \tag{2}$$

In other words, the vector c *is the image of the trajectory* y *by the projection operator* $\Phi : C\left([0,T],\mathbb{R}^D\right) \to \mathbb{R}^K$ *defined by* $\Phi y \coloneqq c$, *whose restriction* $\Phi|_{\mathcal{Y}_{\mathcal{K}}}$ *is bijective (as the Cartesian product of bijective operators). In particular, the spaces* $\mathcal{Y}_{\mathcal{K}}$ *and* $\mathbb{R}^K$ *are isomorphic, that is* $\mathcal{Y}_{\mathcal{K}} \simeq \mathbb{R}^K$.

Regarding the endpoints conditions introduced in Definition 2, we prove in the following result that satisfying these conditions is equivalent to satisfying a linear system for a projected trajectory.

[2] The notation $L^2([0,T],\mathbb{R})$ refers to the classical space of square-integrable functions over $[0,T]$.

Proposition 1. *A trajectory $y \in \mathcal{Y}_{\mathcal{K}}$ belongs to $\mathcal{D}(y_0, y_T)$ if and only if its associated vector $c := \Phi y \in \mathbb{R}^K$ satisfies the linear system*

$$A(0,T)c = \Gamma, \tag{3}$$

where the matrix $A(0,T) \in \mathbb{R}^{2D \times K}$ and the vector $\Gamma \in \mathbb{R}^{2D}$ are given by

$$A(0,T) := \begin{pmatrix} \varphi_1(0) & \dots & \varphi_{K_1}(0) & & & \\ & & & \ddots & & \\ & & & & \varphi_1(0) & \dots & \varphi_{K_D}(0) \\ \varphi_1(T) & \dots & \varphi_{K_1}(T) & & & \\ & & & \ddots & & \\ & & & & \varphi_1(T) & \dots & \varphi_{K_D}(T) \end{pmatrix}, \quad \Gamma := \begin{pmatrix} y_0 \\ y_T \end{pmatrix}.$$

Proof. Let $y \in \mathcal{Y}_{\mathcal{K}}$ and let $c := \Phi y \in \mathbb{R}^K$. By the definition of the matrix $A(0,T)$, we have

$$\begin{aligned} A(0,T)c &= A(0,T)\left(c_1^{(1)}, \dots, c_{K_1}^{(1)}, c_1^{(2)}, \dots, c_{K_2}^{(2)}, \dots, c_1^{(D)}, \dots, c_{K_D}^{(D)}\right)^T \\ &= \left(\sum_{k=1}^{K_1} c_k^{(1)} \varphi_k(0), \dots, \sum_{k=1}^{K_D} c_k^{(D)} \varphi_k(0), \dots, \sum_{k=1}^{K_1} c_k^{(1)} \varphi_k(T), \dots, \sum_{k=1}^{K_D} c_k^{(D)} \varphi_k(T)\right)^T \\ &= \begin{pmatrix} y(0) \\ y(T) \end{pmatrix}. \end{aligned}$$

The conclusion follows directly from the preceding relation.

2.3. Reference trajectories modeling

Let us now suppose that we have access to I recorded trajectories $y_{R_1}, \dots, y_{R_I}$, called *reference trajectories*, coming from some experiments. We propose here an example of a statistical modeling for these reference trajectories, permitting especially to exhibit some linear properties. This modeling will allow to take advantage of the information contained in these recorded trajectories when deriving optimization problems.

These trajectories being recorded, they are in particular admissible and we assume that they belong to the space $\mathcal{Y}_{\mathcal{K}}(0,T)$. As explained previously they may be interpreted as smoothed versions of recorded signals. In particular, each reference trajectory y_{R_i} is associated with a unique vector $c_{R_i} \in \mathbb{R}^K$. Moreover, we consider each reference trajectory as a noisy observation of a certain admissible and projected trajectory y_*. In other words, we suppose that there exists a trajectory $y_* \in \mathcal{Y}_{\mathcal{K}} \cap \mathcal{A}_{\mathcal{G}}(y_0, y_T)$ associated with a vector $c_* \in \mathbb{R}^K$ satisfying

$$\forall i = 1, \dots, I \quad c_{R_i} = c_* + \varepsilon_i.$$

The noise ε_i is here assumed to be a centered Gaussian[3] whose covariance matrix Σ_i is of the form

$$\Sigma_i = \frac{1}{2\omega_i}\Sigma,$$

[3] Note that this is a classical assumption and will ultimately lead to a tractable optimisation problem, with a quadratic penalty.

where $\Sigma \in \mathbb{R}^{K\times K}$. It is noteworthy that this matrix will not be known in most of the cases but an estimated covariance matrix can be computed on the basis of the reference vectors. The positive real numbers ω_i are here considered as weights so we require $\sum_{i=1}^{I}\omega_i = 1$; each ω_i plays actually the role of a noise intensity. Further from the hypothesis that the trajectory y_* and all the reference trajectories y_{R_i} verify the same endpoints conditions, we deduce

$$Ac_{R_i} = Ac_* + A\varepsilon_i \quad \Longleftrightarrow \quad A\varepsilon_i = 0_{\mathbb{R}^{2D}} \quad \Longleftrightarrow \quad \varepsilon_i \in \ker A,$$

for all $i = 1,\dots,I$ (we shorten $A(0,T)$ in A when the context is clear). Hence, the reference vector c_* satisfies the following I systems:

$$\begin{cases} c_{R_i} = c_* + \varepsilon_i, \\ \varepsilon_i \sim \mathcal{N}(0_{\mathbb{R}^K}, \Sigma_i), \\ \varepsilon_i \in \ker A. \end{cases} \tag{4}$$

To establish a more explicit system which is equivalent to the preceding one, we require the following preliminary proposition. Here, we diagonalize the matrices Σ and A^TA by exploiting the fact that the image of the first one is contained in the null space of the other one and vice versa; this is shown in the proof. This property is actually a consequence of the above modeling: the endpoints conditions modelled by A imply linear relations within the components of the vectors, which should be reflected by the covariance matrix Σ. The following result will be helpful to establish the upcoming proposition 3.

Proposition 2. *We define $\sigma := rank\,\Sigma$ and $a := rank\,A^TA$. In the setting of system (4), we have $\sigma + a \leqslant K$ and there exist an orthogonal matrix $V \in \mathbb{R}^{K\times K}$ and two matrices $\Lambda_\Sigma \in \mathbb{R}^{K\times K}$ and $\Lambda_A \in \mathbb{R}^{K\times K}$ of the following form:*

$$\Lambda_\Sigma = \begin{pmatrix} \Lambda_{\Sigma,1} & 0_{\mathbb{R}^{\sigma\times(K-\sigma)}} \\ 0_{\mathbb{R}^{(K-\sigma)\times\sigma}} & 0_{\mathbb{R}^{(K-\sigma)\times(K-\sigma)}} \end{pmatrix} \quad , \quad \Lambda_A = \begin{pmatrix} 0_{\mathbb{R}^{(K-a)\times(K-a)}} & 0_{\mathbb{R}^{(K-a)\times a}} \\ 0_{\mathbb{R}^{a\times(K-a)}} & \Lambda_{A,2} \end{pmatrix},$$

where $\Lambda_{\Sigma,1} \in \mathbb{R}^{\sigma\times\sigma}$ and $\Lambda_{A,2} \in \mathbb{R}^{a\times a}$ are diagonal matrices with positive elements, such that

$$\Sigma = V\Lambda_\Sigma V^T \quad , \quad A^TA = V\Lambda_A V^T.$$

Proof. The proof starts by noticing

$$\Sigma A^TA = A^TA\Sigma = 0_{\mathbb{R}^{K\times K}}. \tag{5}$$

Indeed using the hypothesis $\varepsilon_i \in \ker A$ for any $i = 1,\dots,I$ gives

$$\Sigma A^TA = 2\omega_i\Sigma_i A^TA = 2\omega_i\mathbb{E}\left(\varepsilon_i\varepsilon_i^T\right)A^TA = 2\omega_i\mathbb{E}\left(\varepsilon_i(A\varepsilon_i)^T\right)A = 0_{\mathbb{R}^{K\times K}};$$

similar arguments prove the second equality in (5). First, we can deduce

$$\operatorname{Im}\Sigma \subseteq \ker A^TA, \tag{6}$$

which leads to $\sigma \leqslant K - a$ by the rank-nullity theorem. Equalities (5) show also that Σ and A^TA are simultaneously diagonalizable (since they commute) so there exists an orthogonal matrix $V \in \mathbb{R}^{K\times K}$ such that

$$\Sigma = V\Lambda_\Sigma V^T \quad , \quad A^TA = V\Lambda_A V^T, \tag{7}$$

where $\Lambda_\Sigma \in \mathbb{R}^{K\times K}$ and $\Lambda_A \in \mathbb{R}^{K\times K}$ are diagonal matrices. Permuting if necessary columns of V, we can write the matrix Λ_Σ as follows:

$$\Lambda_\Sigma = \begin{pmatrix} \Lambda_{\Sigma,1} & 0_{\mathbb{R}^{\sigma\times(K-\sigma)}} \\ 0_{\mathbb{R}^{(K-\sigma)\times\sigma}} & 0_{\mathbb{R}^{(K-\sigma)\times(K-\sigma)}} \end{pmatrix}; \tag{8}$$

in other words, the σ first column vectors of V span the image of Σ. From the inclusion (6), we deduce that these vectors belong to the null space of A^TA. Hence, the σ first diagonal elements of Λ_A are equal to zero and, up to a permutation of the $K-\sigma$ last column vectors of V, we can write

$$\Lambda_A = \begin{pmatrix} 0_{\mathbb{R}^{(K-a)\times(K-a)}} & 0_{\mathbb{R}^{(K-a)\times a}} \\ 0_{\mathbb{R}^{a\times(K-a)}} & \Lambda_{A,2} \end{pmatrix},$$

which ends the proof. □

Remark 2. *From equalities (5), we can also deduce*

$$\operatorname{Im} A^TA \subseteq \ker \Sigma,$$

showing that Σ *is singular. Consequently the Gaussian noise* ε_i *involved in (4) is degenerate.*

A new formulation of system (4) which makes explicit the constrained and unconstrained parts of a vector satisfying this system is given in the following result. This is achieved using the preceding result which allows to decompose the space $\mathbb{R}^K$ into three orthogonal subspaces. We prove that the restriction of the noise ε_i to the first subspace is a non-degenerate Gaussian, showing that this first subspace corresponds to the unconstrained one. The two other subspaces describe affine relations coming from the endpoints conditions and from implicit relations within the vector components. These implicit relations, which may model for instance natural trends, are expected to be contained in the reference vectors c_{R_i} and reflected by the (estimated) covariance matrix Σ.

Prior to this, let us write the matrix $V \in \mathbb{R}^{K\times K}$ introduced in Proposition 2 as follows:

$$V = (V_1 \quad V_2 \quad V_3),$$

where $V_1 \in \mathbb{R}^{K\times\sigma}$, $V_2 \in \mathbb{R}^{K\times K-\sigma-a}$ and $V_3 \in \mathbb{R}^{K\times a}$. We emphasize that the column-vectors of the matrices V_1 and V_3 do not overlap according to the property $\sigma + a \leqslant K$ proved in proposition 2. In particular, the matrix V_2 has to be considered only in the case $\sigma + a < K$. Further for any $c \in \mathbb{R}^K$, we will use the notation

$$\widetilde{c} \coloneqq V^Tc \quad , \quad \widetilde{c}_\ell \coloneqq V_\ell^Tc,$$

for $\ell = 1,2,3$. Finally, we consider the singular value decomposition of A coming from the diagonalization of the symmetric matrix A^TA with V:

$$A = US_AV^T,$$

where $U \in \mathbb{R}^{2D\times 2D}$ is orthogonal and $S_A \in \mathbb{R}^{2D\times K}$ is a rectangular diagonal matrix of the following form:

$$S_A = (0_{\mathbb{R}^{2D\times K-2D}} \quad S_{A,2}), \tag{9}$$

with $S_{A,2} \coloneqq \sqrt{\Lambda_{A,2}} \in \mathbb{R}^{2D\times 2D}$.

Proposition 3. *Suppose that the matrix A is full rank, that is* $a = 2D$. *Then for any* $i = 1,\ldots,I$, *system (4) is equivalent to the following one*:

$$\begin{cases} \widetilde{c}_{R_i,1} = \widetilde{c}_{*,1} + \widetilde{\varepsilon}_{i,1}, \\ \widetilde{\varepsilon}_{i,1} \sim \mathcal{N}\left(0_{\mathbb{R}^\sigma}, \dfrac{1}{2\omega_i}\Lambda_{\Sigma,1}\right), \\ \widetilde{c}_{*,2} = V_2^T c_{R_i}, \\ \widetilde{c}_{*,3} = S_{A,2}^{-1} U^T \Gamma. \end{cases} \tag{10}$$

Proof. We first prove that system (4) is equivalent to

$$\begin{cases} \widetilde{c}_{R_i} = \widetilde{c}_* + \widetilde{\varepsilon}_i, \\ \widetilde{\varepsilon}_i \sim \mathcal{N}\left(0_{\mathbb{R}^K}, \dfrac{1}{2\omega_i}\Lambda_\Sigma\right), \\ S_A \widetilde{c}_* = U^T \Gamma. \end{cases} \tag{11}$$

The matrix V being orthogonal, it is nonsingular and so we have for all $i = 1, \ldots, I$,

$$c_{R_i} = c_* + \varepsilon_i \quad \Longleftrightarrow \quad \widetilde{c}_{R_i} = \widetilde{c}_* + \widetilde{\varepsilon}_i,$$

and, since $\Sigma_i = \frac{1}{2\omega_i}\Sigma = \frac{1}{2\omega_i} V \Lambda_\Sigma V^T$, we obtain

$$\varepsilon_i \sim \mathcal{N}(0_{\mathbb{R}^K}, \Sigma_i) \quad \Longleftrightarrow \quad \widetilde{\varepsilon}_i \sim \mathcal{N}\left(0_{\mathbb{R}^K}, \frac{1}{2\omega_i}\Lambda_\Sigma\right).$$

Finally, the property $\varepsilon_i \in \ker A$ is equivalent to

$$A c_* = \Gamma \quad \Longleftrightarrow \quad U S_A V^T c_* = \Gamma \quad \Longleftrightarrow \quad S_A \widetilde{c}_* = U^T \Gamma,$$

proving that the systems (4) and (11) are equivalent. Now the fact that the $K - \sigma$ last diagonal elements of Λ_Σ are zero implies that the components $\widetilde{c}_{*,2} \in \mathbb{R}^{K-\sigma-2D}$ and $\widetilde{c}_{*,3} \in \mathbb{R}^{2D}$ are constant. From the first equality of (11), we have on one side

$$\widetilde{c}_{R_i,2} = \widetilde{c}_{*,2} \quad \Longleftrightarrow \quad V_2^T c_{R_i} = \widetilde{c}_{*,2},$$

for any $i = 1, \ldots, I$. On the other side, combining the last relation of the system (11) with the form of the matrix S_A given in (9) yields

$$\begin{aligned} S_A \widetilde{c}_* = U^T \Gamma \quad &\Longleftrightarrow \quad S_{A,2} \widetilde{c}_{*,3} = U^T \Gamma \\ &\Longleftrightarrow \quad \widetilde{c}_{*,3} = S_{A,2}^{-1} U^T \Gamma, \end{aligned}$$

the last equivalence being justified by the hypothesis that the matrix A is full rank (which implies that the diagonal matrix $S_{A,2}$ is nonsingular). □

The above decomposition gives us access to nondegenerated density of $\widetilde{c}_{R_i,1}$ given $\widetilde{c}_{*,1}$ which is later denoted by $u(\widetilde{c}_{R_i,1} | \widetilde{c}_{*,1})$. In next section, we will assume a prior distribution on $\widetilde{c}_{*,1}$ with high density for low values of the cost function F.

2.4. A trajectory optimization problem via a MAP approach

Before introducing the Bayesian framework, let first recall that we are interested in minimizing a certain cost function $F : C([0,T], \mathbb{R}^D) \to \mathbb{R}$ over the set of projected and admissible trajectories $\mathcal{Y}_\mathcal{K} \cap \mathcal{A}_\mathcal{G}(y_0, y_T)$. As explained previously, we propose here a methodology leading to a constrained optimization problem based on the reference trajectories and designed to provide realistic trajectories (we refer again to Figure 1). Technically speaking, we seek for the mode of a posterior distribution which contains information from the reference trajectories. The aim of this subsection is then to obtain the posterior distribution via Bayes's rule,

using in particular the precise modeling of the reference trajectories given in Proposition 3 and defining an accurate prior distribution with high density for low values of the cost function F.

To do so, we recall firstly that all the trajectories considered here are assumed to belong to the space $\mathcal{Y}_{\mathcal{K}}$ which is isomorphic to $\mathbb{R}^K$. So each trajectory is here described by its associated vector in $\mathbb{R}^K$, permitting in particular to define distributions over finite-dimensional spaces. We also recall that the reference trajectories are interpreted as noisy observations of a certain y_* associated with a c_*. According to Proposition 3, this vector complies with some affine conditions which are described by the following subspace $\mathcal{V}_1$:

$$c \in \mathcal{V}_1 \iff \begin{cases} V_2^T c = V_2^T c_{R_i}, \\ V_3^T c = S_{A,2}^{-1} U^T \Gamma. \end{cases} \tag{12}$$

Hence, a vector c belonging to $\mathcal{V}_1$ is described only through its component $\widetilde{c}_1 := V_1^T c$. In addition, we note that the definition of $\mathcal{V}_1$ does not depend actually on the choice of i since $V_2^T c_{R_i}$ has been proved to be constant in Proposition 3. Further, we emphasize that the matrix A is supposed to be full rank in this case and we have $\mathcal{V}_1 \simeq \mathbb{R}^\sigma$; we recall that σ is the rank of the covariance matrix Σ.

Let us now define the cost function F over the spaces $\mathbb{R}^K$ and $\mathcal{V}_1$. This is necessary to define the prior distribution and to establish our optimization problem.

Definition 6 (Cost functions). *Let $\check{F} : \mathbb{R}^K \to \mathbb{R}$ and $\widetilde{F} : \mathbb{R}^\sigma \to \mathbb{R}$ be*

- $\check{F}(c) := F\left(\Phi\Big|_{\mathcal{Y}_{\mathcal{K}}}^{-1} c\right)$;
- $\widetilde{F}(\widetilde{c}_1) := F\left(\Phi|_{\mathcal{Y}_{\mathcal{K}}}^{-1} V \begin{pmatrix} \widetilde{c}_1^T & c_{R_i}^T V_2 & \Gamma^T U \left(S_{A,2}^{-1}\right)^T \end{pmatrix}^T\right)$.

Remark 3. *From the previous definition, we observe that for any $y \in \mathcal{Y}_{\mathcal{K}}$ and its associated vector $c \in \mathbb{R}^K$, we have*

$$\check{F}(c) = F\left(\Phi\Big|_{\mathcal{Y}_{\mathcal{K}}}^{-1} c\right) = F(y).$$

Further for any $c \in \mathcal{V}_1$, we have

$$\check{F}(c) = F\left(\Phi\Big|_{\mathcal{Y}_{\mathcal{K}}}^{-1} c\right) = F(\Phi|_{\mathcal{Y}_{\mathcal{K}}}^{-1} V\widetilde{c}) = F\left(\Phi|_{\mathcal{Y}_{\mathcal{K}}}^{-1} V \begin{pmatrix} \widetilde{c}_1^T & c_{R_i}^T V_2 & \Gamma^T U \left(S_{A,2}^{-1}\right)^T \end{pmatrix}^T\right) = \widetilde{F}(\widetilde{c}_1).$$

We deduce that $\widetilde{F}$ is actually the restriction of $\check{F}$ to the subspace $\mathcal{V}_1$.

From now on, the trajectory y_* and the associated vector c_* will be considered as random variables and will be denoted by y and c. We are interested in the posterior distribution

$$u(\widetilde{c}_1 \,|\, \widetilde{c}_{R_1,1}, \ldots, \widetilde{c}_{R_I,1}),$$

which depends only on the free component $\widetilde{c}_1$ of $c \in \mathcal{V}_1$, the two other ones $\widetilde{c}_2$ and $\widetilde{c}_3$ being fixed according to (12). We use Bayes's rule to model the posterior via the prior and likelihood distributions, leading to

$$u(\widetilde{c}_1 \,|\, \widetilde{c}_{R_1,1}, \ldots, \widetilde{c}_{R_I,1}) \propto u(\widetilde{c}_{R_1,1}, \ldots, \widetilde{c}_{R_I,1} \,|\, \widetilde{c}_1)\, u(\widetilde{c}_1).$$

Assuming now that the vectors $\widetilde{c}_{R_i,1}$ are independent gives

$$u(\widetilde{c}_{R_1,1}, \ldots, \widetilde{c}_{R_I,1} \,|\, \widetilde{c}_1)\, u(\widetilde{c}_1) = \prod_{i=1}^{I} u(\widetilde{c}_{R_i,1} \,|\, \widetilde{c}_1)\, u(\widetilde{c}_1).$$

The above likelihood is given by the modeling of the reference trajectories detailed in Proposition 3. In this case, we have

$$u(\widetilde{c}_{R_i,1} \,|\, \widetilde{c}_1) \propto \exp\left(-\omega_i (\widetilde{c}_1 - \widetilde{c}_{R_i,1})^T \Lambda_{\Sigma,1}^{-1} (\widetilde{c}_1 - \widetilde{c}_{R_i,1})\right).$$

The prior distribution is obtained by assuming that the most efficient trajectories (with respect to the cost function) are a priori the most likely ones:

$$u(\widetilde{c}_1) \propto \exp\left(-\kappa^{-1}\widetilde{F}(\widetilde{c}_1)\right), \tag{13}$$

where $\kappa > 0$. Putting everything together and taking the negative of the logarithm gives the following minimization problem, whose solution is the MAP estimator:

$$\begin{cases} \widetilde{c}_1^{\star} \in \underset{\widetilde{c}_1 \in \mathbb{R}^{\sigma}}{\arg\min}\, \widetilde{F}(\widetilde{c}_1) + \kappa \sum_{i=1}^{I} \omega_i (\widetilde{c}_1 - \widetilde{c}_{R_i,1})^T \Lambda_{\Sigma,1}^{-1} (\widetilde{c}_1 - \widetilde{c}_{R_i,1}), \\ \widetilde{c}_2 = V_2^T c_{R_i}, \\ \widetilde{c}_3 = S_{A,2}^{-1} U^T \Gamma. \end{cases} \tag{14}$$

where i is arbitrarily chosen in $\{1, \ldots, I\}$.

Let us now rewrite the above optimization problem with respect to the variable $c = V\widetilde{c} \in \mathbb{R}^K$ in order to make it more interpretable.

Proposition 4. *The optimization problem (14) is equivalent to the following one:*

$$c^{\star} \in \underset{c \in \mathcal{V}_1}{\arg\min}\, \check{F}(c) + \kappa \sum_{i=1}^{I} \omega_i (c - c_{R_i})^T \Sigma^{\dagger} (c - c_{R_i}), \tag{15}$$

where $\Sigma^{\dagger} \in \mathbb{R}^{K \times K}$ *denotes the pseudoinverse of the matrix* Σ.

Proof. From (8), we deduce

$$\begin{aligned} \sum_{i=1}^{I} \omega_i (\widetilde{c}_1 - \widetilde{c}_{R_i,1})^T \Lambda_{\Sigma,1}^{-1} (\widetilde{c}_1 - \widetilde{c}_{R_i,1}) &= \sum_{i=1}^{I} \omega_i (\widetilde{c} - \widetilde{c}_{R_i})^T \Lambda_{\Sigma}^{\dagger} (\widetilde{c} - \widetilde{c}_{R_i}) \\ &= \sum_{i=1}^{I} \omega_i (c - c_{R_i})^T V \Lambda_{\Sigma}^{\dagger} V^T (c - c_{R_i}) \\ &= \sum_{i=1}^{I} \omega_i (c - c_{R_i})^T \Sigma^{\dagger} (c - c_{R_i}). \end{aligned}$$

And from the proof of proposition 3, we have

$$Ac = \Gamma \quad \Longleftrightarrow \quad \widetilde{c}_3 = S_{A,2}^{-1} U^T \Gamma,$$

proving that $c \in \mathcal{V}_1$.

To conclude, let us comment on this optimization problem.

1. To interpret the optimization problem (15) (or equivalently (14)) from a geometric point of view, let us consider the following new problem:

$$\begin{aligned} &\min_{\widetilde{c}_1 \in \mathbb{R}^{\sigma}} \widetilde{F}(\widetilde{c}_1) \\ &\text{s.t.} \sum_{i=1}^{I} \omega_i (\widetilde{c}_1 - \widetilde{c}_{R_i,1})^T \Lambda_{\Sigma,1}^{-1} (\widetilde{c}_1 - \widetilde{c}_{R_i,1}) \leqslant \widetilde{\kappa} \end{aligned}, \tag{16}$$

where $\lambda \geqslant 0$. Here, we suppose that $\widetilde{F}$ is strictly convex and that the problem (16) has a solution (which is then unique). By Slater's theorem (Boyd and Vandenberghe, 2004, Section 5.2.3), the strong duality holds for the problem (16). It can then be proved that there exists a certain $\lambda^{\star} \geqslant 0$ such that the solution of (16) is the minimizer of the strictly convex function

$$\widetilde{c}_1 \mapsto \widetilde{F}(\widetilde{c}_1) + \lambda^\star \sum_{i=1}^{I} \omega_i (\widetilde{c}_1 - \widetilde{c}_{R_i,1})^T \Lambda_{\Sigma,1}^{-1} (\widetilde{c}_1 - \widetilde{c}_{R_i,1}),$$

which is actually the objective function of the optimization problem (14) for $\kappa = \lambda^\star$. Hence, the problem (14) minimizes the cost $\widetilde{F}$ in a ball centered on the weighted average of the reference trajectories. In particular, if the reference trajectories are close to an optimal one with respect to $\widetilde{F}$ then one could expect the solution of (14) to be equal to this optimal trajectory.

2. Further the optimization problem (15) takes into account the endpoints conditions through the subspace $\mathcal{V}_1$ but not the additional constraints. However, as explained in the preceding point, the solution is close to realistic trajectories and so is likely to comply with the additional constraints for a well-chosen parameter $\kappa > 0$. We refer to Section 2.6 for more details on an iterative method for the tuning of κ. In particular, a right choice for this parameter is expected to provide an optimized trajectory with a realistic behavior. This is for instance illustrated in Section 4.
3. Taking into account the linear information from the available data through the covariance matrix Σ allows to restrict the search to the subspace $\mathcal{V}_1$ describing these relations. This is of particular interest when implicit relations (modeled by the submatrix V_2) are revealed by the estimation of Σ on the basis of the reference trajectories; in this case, these implicit relations may not be known by the expert.
4. The optimization problem (15) has linear constraints and a quadratic penalized term. For instance, if the cost function $\check{F}$ is a convex function then we obtain a convex problem for which efficient algorithms exist.

2.5. Quadratic cost for a convex optimization problem

In this short subsection, we focus on a particular case where the cost function F is defined as the integral of an instantaneous quadratic cost function, that is

$$\forall y \in C\big([0,T],\mathbb{R}^D\big) \quad F(y) = \int_0^T f(y(t))\,dt, \tag{17}$$

where $f : \mathbb{R}^D \to \mathbb{R}$ is quadratic. Even though such a setting may appear to be restrictive, we emphasize that quadratic models may lead to highly accurate approximations of variables, as it is illustrated in Section 4. For a quadratic instantaneous cost, the associated function $\check{F} : \mathbb{R}^K \to \mathbb{R}$ can be proved to be quadratic as well and can be explicitly computed. In the following result, we provide a quadratic optimization problem equivalent to (15).

Proposition 5. *Suppose that the cost function F is of the form (17) with f quadratic. Then the optimization problem (15) is equivalent to the following one:*

$$c^\star \in \underset{c \in \mathcal{V}_1}{\arg\min}\, c^T\big(\check{Q} + \kappa\Sigma^\dagger\big)c + \left(\check{w} - 2\kappa \sum_{i=1}^{I} \omega_i \Sigma^\dagger c_{R_i}\right)^T c, \tag{18}$$

where $\check{Q} \in \mathbb{R}^{K\times K}$ and $\check{w} \in \mathbb{R}^K$ can be explicitly computed from f.

Proof. We defer the proof to Appendix A.

In particular, this allows to derive sufficient conditions on the parameter $\kappa > 0$, so that the optimization problem is proved to be equivalent to a quadratic program (Boyd and Vandenberghe, 2004, Section 4.4), namely the objective function is convex quadratic together with affine constraints. In practice, this allows to make use of efficient optimization libraries to solve numerically (18).

2.6. Iterative process to comply with additional constraints

As explained in Section 2.4, the trajectory optimization problem (15) is constrained by the endpoints conditions and by implicit linear relations revealed by the reference trajectories. Nevertheless the

additional constraints introduced in Definition 3 are not taken into account in this problem. In practice, such constraints assure that natural or user-defined features are verified and so a trajectory which does not comply with these constraints may be considered as unrealistic.

Our aim is then to assure that the trajectory $y^\star = \Phi|^{-1}_{\mathcal{Y}_\mathcal{K}} c^\star$, where $c^\star \in \mathcal{V}_1$ is the solution of the optimization problem (15), verifies the additional constraints, that is belongs to the set $\mathcal{G}$. A first solution would be to add the constraint $\Phi|^{-1}_{\mathcal{Y}_\mathcal{K}} c \in \mathcal{G}$ in the optimization problem (15). However, depending on the nature of the constraints functions g_ℓ, this may lead to nonlinear constraints which could be costly from a numerical point of view. The solution we propose consists rather in exploiting the degree of freedom coming from the parameter $\kappa > 0$ appearing in the problem (15).

First of all, let us factorize the problem (15) by κ to obtain the following new one for the sake of presentation:

$$c^\star \in \underset{c \in \mathcal{V}_1}{\arg\min}\, \nu \check{F}(c) + \sum_{i=1}^{I} \omega_i (c - c_{R_i})^T \Sigma^\dagger (c - c_{R_i}), \tag{19}$$

where $\nu := \kappa^{-1}$. On one hand, we observe that the solution of the optimization problem (19) for the limit case $\nu = 0$ is given by $\sum_{i=1}^{I} \omega_i c_{R_i}$ which is the average of the reference vectors. In this case, one may expect that the associated average trajectory complies with the constraints but is unlikely to optimize the cost function F. On the other hand, for very large $\nu > 0$, the second term of the objective function in (19) can be considered as negligible compared to the first one. In this case, the cost of the solution will surely be smaller than the costs of the reference trajectories but no guarantee regarding the additional constraints can be established in a general setting.

Given these observations, the task is then to find an appropriate value $\nu^\star > 0$ in order to reach a trade-off between optimizing and remaining close to the reference trajectories to comply with the additional constraints. Many methods can be developed to find such a $\nu^\star$ but here we propose a very simple one based on a binary search algorithm. We exploit the assumption that the solution for $\nu = 0$ (i.e., the reference trajectories average) is admissible. We proceed then as follows:

1. We set firstly a maximal value $\nu_{max} > 0$ so that the solution of (19) with ν_{max} does not satisfy the constraints. For instance, we choose a large value, solve the optimization problem and check the constraints. If they are still satisfied, we choose a larger ν_{max}.
2. Then apply a binary search between 0 and ν_{max}: solve the optimization problem for $\frac{\nu_{max}}{2}$; if the resulting solution verifies the constraints, solve the problem again for $\frac{3\nu_{max}}{4}$; else solve it for $\frac{\nu_{max}}{4}$.
3. Continue the process until a user-defined stopping criterion.

Other iterative processes can also be considered.

2.7. Confidence bounds on the integrated cost

In practice the cost function F considered is an estimation of the true cost $F^\star$, a random variable which cannot be fully predicted based on y. If the distribution $F(y)$ was known we could deduce a confidence bound on $F^\star$. This is for instance possible by considering multivariate functional regression Ramsay et al. (2007).

The simplest case from the estimation point of view is to consider that $F^\star$ is the integral of some instantaneous consumption function $f^\star$ as in Section 2.5, and to estimate the parameters of the standard multivariate regression

$$f^\star(y(t)) = f(y(t)) + \varepsilon(t),$$

where the random noise $\varepsilon(t)$ is assumed to follow a centered Gaussian distribution with variance σ. In this case, $F^\star$ can be expressed as the integral of a stochastic process

$$F^\star(y) := \int_0^T f^\star(y(t))\,dt = F(y) + \int_0^T \varepsilon(t)\,dt,$$

then assuming that $(\varepsilon(t))_{t\in[0,T]}$ independent, we obtain

$$\int_0^T \varepsilon(t)\,dt \sim \mathcal{N}\left(0, T\sigma^2\right).$$

Thus $F^\star(y)$ follows a Gaussian distribution centered on $F(y)$ and with variance equals to $T\sigma^2$. This makes it possible to compute confidence bounds on $F^\star(y)$. For a confidence level $1-u$, $u\in[0,1]$, a confidence interval for $F^\star(y)$ is obtained as

$$\mathtt{CI}^{1-u}(F^\star(y)) = F(y) \pm \zeta_{1-\frac{u}{2}}\sqrt{T}\sigma,$$

where $\zeta_{1-\frac{u}{2}}$ is the quantile of order $1-\frac{u}{2}$ of the standard Gaussian distribution.

The assumption that f and σ^2 are known is relevant since they are estimated based on a huge amount of training data. The assumption of white Gaussian noise can be seen as unrealistic, however, it appears to be the only route to explicit calculus. A more complex strategy could be derived using Gaussian processes, which is beyond the scope of this paper.

3. The Python Library Pyrotor

The above optimization methodology is aimed at being used in a wide range of applications, from path planning for industrial robots (Chettibi et al., 2004) to fuel-efficient aircraft trajectories (Rommel et al., 2019; Dewez et al., 2020). We therefore contribute a generic Python library PyRotor (standing for **Py**thon **Ro**ute **t**rajectory **o**ptimize**r**) which is intended to the largest audience. `PyRotor` is the backbone to numerical results given in this paper.

When using the `PyRotor` library, the practitioner has to define the endpoints conditions as a dictionary, the additional constraints in a list of functions, the name of the basis and the dimension for each variable. The current version of the library covers only the case of Section 2.5, that is to say the cost is given by a quadratic instantaneous function. This permits to make use of Proposition 5 in which a quadratic objective function is given. We mention that future releases of `PyRotor` are intended to cover more general cost functions. The value of the parameter v_{max} in (19) can also be manually set depending on the application. The Legendre basis and B-splines are currently the bases implemented in the first version of `PyRotor` (via the `legendre` module from `NumPy` package Harris et al., 2020 and the `interpolate` module from `Scipy` package Virtanen et al., 2020) but future developments including other general bases are planned. Further, the user indicates a path to a directory containing the data, each reference trajectory being contained in a `csv` file. The covariance matrix Σ is here estimated by using the `sklearn.covariance` package from the `Python` library `scikit-learn` (Pedregosa et al., 2011). Two optimization solvers are proposed: the generic solver `minimize(method='trust-constr')` (Conn et al., 2000) from `SciPy` and the quadratic programming solver from `CVXOPT` software (Andersen et al., 2020). The latter is intended to speed up the execution in case of convex quadratic objective function. Once the arguments are given by the user, a class is created and the optimization is performed by executing a method from this class. At the end, the optimized trajectory is provided in a dataframe: at each time, the position of the trajectory is given together with the value of f. The total cost is also computed and a quantitative comparison in terms of savings with the reference trajectories can be also displayed.

The open source `PyRotor` library is developed on GitHub and welcomes contributions from its users: we favor a community-based development to foster the diffusion of our work toward practitioners. `PyRotor` is intended to be PEP8 compliant and purposely rely on high standard coding practices. The continuous development platform Travis is used to certify the latest builds of the library. Finally, we provide Jupyter notebooks, for example, on how to use `PyRotor` along with online documentation. `PyRotor` is available at https://github.com/bguedj/pyrotor.

4. Application 1: Trajectory Optimization for Fuel-Efficient Aircrafts

In this section, we consider the aeronautic problem of reducing the total fuel consumption of an aircraft during the climb phase. This example illustrates the key role played by the reference trajectories since we are able to obtain optimized trajectories with realistic patterns thanks to a simple modeling involving few constraints.

4.1. Modeling

Here, the trajectories are supposed to be in a vertical plane and are defined by the altitude h, the Mach number M and the engines rotational speed N1 (expressed as a percentage of a maximal value). Hence, a trajectory y in this setting is a continuous $\mathbb{R}^3$-valued map defined on $[0,T]$, where T is a maximal climb duration fixed by the user. Hence, we have

$$\forall t \in [0,T] \quad y(t) \coloneqq (h(t), \mathrm{M}(t), \mathrm{N1}(t)).$$

The quantity to minimize is the total fuel consumption $\mathrm{TFC} : C\big([0,T], \mathbb{R}^3\big) \to \mathbb{R}_+$ which is defined via the fuel flow $\mathrm{FF} : \mathbb{R}^3 \to \mathbb{R}_+$ as follows[4]:

$$\mathrm{TFC}(y) \coloneqq \int_0^T \mathrm{FF}(y(t))\, dt.$$

Regarding the endpoints conditions, we require the trajectory to start at the altitude h_0 with Mach number M_0 and to end at the altitude h_T with Mach number M_T. In particular, the reference trajectories we use have to verify these conditions.

We consider also additional constraints which are conventional in the aeronautic setting:

- The rate of climb, that is the time-derivative of the altitude, has to be upper bounded by a given maximal value γ_{max} during the whole climb;
- The Mach number should not exceed a certain value called the maximum operational Mach (MMO).

The final time of the climb is given by $T^\star \in [0,T]$ which is the first time where the aircraft reaches h_T with Mach number M_T.

Finally, we mention that the fuel flow model FF is here estimated. To do so, we exploit the reference trajectories which contain recorded altitude, Mach number, engines power and fuel flow for each second of the flight. Having access to these data, we are in position to fit a statistical model. Following the numerical results in Dewez et al. (2020) which show that polynomials can accurately model aeronautic variables, we consider a polynomial model of degree 2 for the fuel flow. In particular, the requirements for the cost function in the current version of `PyRotor` are fulfilled. The prediction accuracy of the resulting estimated model is assessed in the following subsection.

4.2. Numerical results

We present now numerical results based on real flight data for the above aeronautic problem. Here, we have access to 2,162 recorded short and medium-haul flights performed by the same narrow-body airliner type, provided by a partner airline. In particular they cannot be publicly released for commercial reasons. The data is here recorded by the quick access recorder (QAR).

Before considering the optimization setting, we estimate a fuel flow model specific to the climb phase and to the considered airliner type. To do so, we extract the signals of the four variables of interest (altitude, Mach number, engines rotational speed, and fuel flow) and keep the observations from the take-off to the beginning of the cruise without level-off phases. Smoothing splines are then applied to the raw signals to remove the noise. We sample each 5 s to reduce the dataset size without impacting strongly the accuracy of

[4] In the notation of Section 2.5, FF and TFC play respectively the role of f and F.

the resulting models. At the end, we obtain 494,039 observations which are randomly split into training and test sets to fit a polynomial model of degree 2 using the `scikit-learn` library. The RMSE and MAPE values of this model on the test set are respectively equal to 3.64×10^{-2} kg/s and 1.73%.

Regarding the optimization, we are interested in climb phases from 3,000 to 38,000 ft. We mention that we remove lower altitudes because operational procedures constraint heavily the trajectory during the very beginning of the climb. Further the initial and final Mach numbers are required to be equal to 0.3 and 0.78. It is noteworthy that the optimization solvers used in `PyRotor` allow linear inequality conditions, permitting to slightly relax the endpoints conditions. Here we tolerate an error of 100 ft for the altitude and an error of 0.01 for the Mach number. The initial and final N1 values are let unconstrained. Finally the MMO and γ_{max} are respectively set to 0.82 and 3,600 ft/min.

The reference trajectories are given by 48 recorded flights which satisfy the above climb endpoints conditions among the 2,162 available ones. All these selected flights are used to estimate the covariance matrix involved in the optimization problem. On the other hand, we use only the five most fuel-efficient flights in the objective function to focus on a domain containing the most efficient recorded flights. Further the maximal duration T is here fixed to the duration of the longest climb among the five most fuel-efficient ones we use.

Legendre polynomials are used as the functional basis spanning the space in which lies the trajectories. Since we consider narrow-body airliners, polynomials are expected to be relevant to describe the slow variations of such aircrafts. Here the dimensions associated with the altitude, the Mach number and the engines power are given respectively by 4, 10, and 6. The reference vectors c_{R_i} are then computed using the formula (2). At the end, we amount to solving a constrained optimization problem in a space of dimension 20.

We are then in position to apply the optimization method developed in Section 2 using the PyRotor library. First of all a relevant value for $v_{max} > 0$ has to be fixed. In order to propose a realistic optimized climb, we choose a v_{max} relatively small so that the optimized climb remains close to the reference ones. In particular, the quadratic objective function in (19) turns out to be convex for all $v \in [0, v_{max}]$ permitting to use the quadratic programming solver from `CVXOPT` software imported in `PyRotor`. The preprocessing of the reference trajectories and the optimization steps have been executed 100 times using `PyRotor` on an Intel Core i7 6 cores running at 2.2 GHz. The mean of the execution time for both steps is equal to 3.76 s with standard deviation 0.11 s, illustrating that the library is time-efficient in this setting.

A plot of the optimized trajectory obtained using `PyRotor` is given in Figure 3. We observe that the optimized trajectory seeks to reach the maximum altitude in the minimum amount of time; this is in accordance with the existing literature (see, for instance, Codina and Menéndez, 2014 and references therein). In particular, the duration $T^\star$ is equal to 1,033 s which is actually slightly shorter than the reference durations. We note also that the optimized Mach number shares a very similar pattern with the references. On the other hand, the optimized engines rotational speed tends to slowly decrease until the cruise regime before reaching the top of climb. This is not the case for the reference engines speed which falls to the cruise regime just after reaching the final altitude. Most of the savings seem to be achieved in these last moments of the climb. At last but not least, the optimized trajectory presents a realistic pattern inherited from the reference trajectories.

For a quantitative comparison, we refer to Table 1 which provides statistical information on the fuel savings. The mean savings 16.54% together with the fact that the optimized trajectory verifies the additional constraints show that these first results are promising, motivating further studies. For instance one could model environmental conditions or take into account Air Traffic Control constraints for more realistic modelings. In particular, the minimal percentage indicates the savings compared with the best trajectory (i.e., with the lowest consumption).

5. Application 2: Trajectory Optimization to Maximize Work of a Force Field

Here, we consider the following generic example: given a moving point in a force field, find a trajectory starting and ending at two different given points which maximizes the work of the force along the

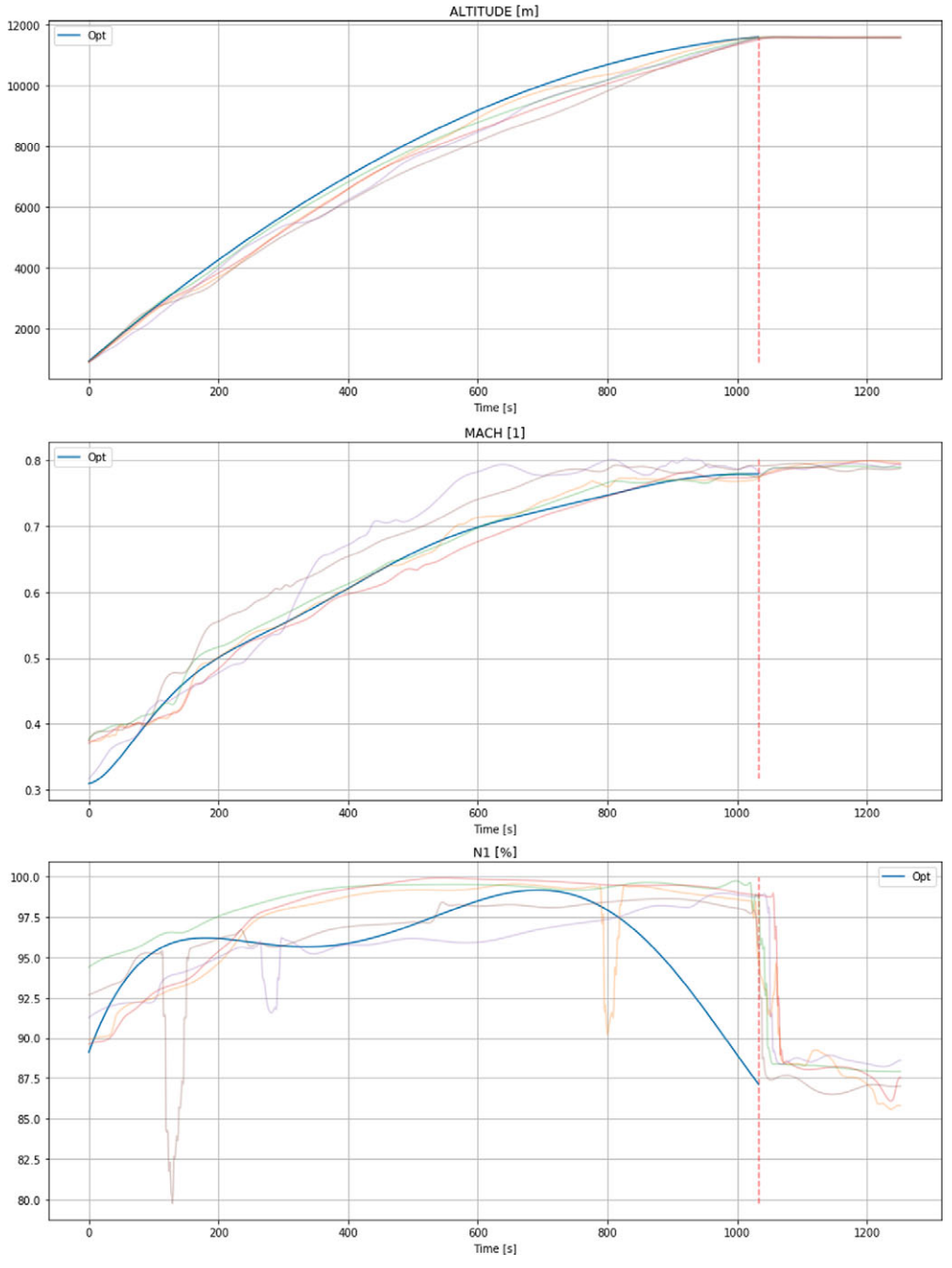

Figure 3. *Optimized and reference altitudes, Mach numbers and engines rotational speeds—the optimized trajectory is represented by the blue curves.*

trajectory while minimizing the travelled distance. For instance, this corresponds to a very simple modeling of a sailing boat which seeks to increase the power of the wind at each time, that is maximizing the wind work, without traveling a too large distance. This second example demonstrates that our generic optimization approach is flexible enough to take into account derivatives of trajectories and hence to cover dynamics settings.

Table 1. *Statistical description of the fuel savings of the optimized trajectory.*

	Mean	Standard deviation	Min	Q_1	Q_2	Q_3	Max
Fuel savings (kg)	260.38	86.21	71.79	202.40	261.87	330.32	393.73
Percentage (%)	16.54	4.73	5.27	13.56	16.88	20.39	23.39

The savings are compared with the 48 recorded flights satisfying the present endpoints and the total consumption of the optimized trajectory is estimated sing the statistical model for the fuel flow. Q_1, Q_2, and Q_3 refer to the first, second and third quartiles.

5.1. Modeling

To model this problem, we suppose without loss of generality that the trajectories are defined on the (time-) interval $[0,1]$ and we let $V:\mathbb{R}^D \to \mathbb{R}^D$ denote a vector field. Furthermore, the trajectories are assumed here to be continuously differentiable, that is they belong to $C^1\left([0,1],\mathbb{R}^D\right)$. The work of V along a trajectory $y \in C^1\left([0,1],\mathbb{R}^D\right)$ is

$$W(y,\dot{y}) \coloneqq \int_0^1 V(y(t))^T \dot{y}(t)\,dt;$$

here $\dot{y}$ denotes the derivative of y with respect to the independent variable t. Moreover, using Hamilton's principle in Lagrangian mechanics, it can be shown that the trajectory with constant velocity (i.e., a straight line travelled at constant speed) is the minimum of the following functional,

$$J(\dot{y}) = \int_0^1 \|\dot{y}(t)\|_2^2\,dt,$$

where the starting and ending points of y are fixed and different. This functional can be then used to control the travelled distance. It follows that minimizing the cost function

$$F_\alpha(y,\dot{y}) \coloneqq \alpha J(\dot{y}) - W(y,\dot{y}) = \int_0^1 \alpha\|\dot{y}(t)\|_2^2 - V(y(t))^T\dot{y}(t)\,dt,$$

where $\alpha \geqslant 0$ is arbitrarily chosen, is expected to lead to an optimized trajectory reflecting a trade-off between maximizing the work and minimizing the distance. Further we require the trajectory to stay in the hypercube $[0,1]^D$ and to start and to end respectively at $y_0 \in [0,1]^D$ and $y_1 \in [0,1]^D$.

Now, we remark that the above cost function involves the (time-)derivative $\dot{y}$. So one has to derive a formula permitting to compute the derivative of any trajectory $y = \Phi|_{\mathcal{Y}_\mathcal{K}}^{-1} c \in \mathcal{Y}_\mathcal{K}$ from its associated vector $c \in \mathbb{R}^K$, especially to compute $\check{F}(c)$. For instance, this can be easily achieved by assuming that each element of the functional basis is continuously differentiable. Indeed we can differentiate in this case any $y \in \mathcal{Y}_\mathcal{K}$:

$$\forall d = 1,\ldots,D \qquad \dot{y}^{(d)} = \sum_{k=1}^{K_d} c_k^{(d)}\dot{\varphi}_k = \left(\frac{d}{dt}\Phi\Big|_{\mathcal{Y}_\mathcal{K}}^{-1} c\right)^{(d)}.$$

We deduce then the following formula for $\check{F}(c)$ in the present setting:

$$\check{F}(c) \coloneqq F_\alpha\left(\Phi\Big|_{\mathcal{Y}_\mathcal{K}}^{-1} c, \frac{d}{dt}\Phi\Big|_{\mathcal{Y}_\mathcal{K}}^{-1} c\right).$$

Here, the vector c contains information on both position and velocity, permitting especially to keep the problem dimension unchanged. To finish, let us remark that it is possible to make the above formula for $\check{F}$ explicit with respect to c in certain settings. For instance it is possible to derive an explicit quadratic formula for $\check{F}(c)$ when the integrand defining F_α is quadratic with respect to $y(t)$ and $\dot{y}(t)$; this formula is implemented in `PyRotor` and the arguments to obtain it are similar to those proving Proposition 5.

5.2. Numerical results

Numerical results based on randomly generated data for the above physical application are presented in this section. We first consider trajectories with two components $y^{(1)}$ and $y^{(2)}$ lying in the square $[0,1]^2$ for the sake of simplicity. We set the starting and ending points as follows:

$$y^{(1)}(0)=0.111 \quad , \quad y^{(2)}(0)=0.926 \quad , \quad y^{(1)}(1)=0.912 \quad , \quad y^{(2)}(1)=0.211$$

with a tolerated error 1×10^{-4}, and the vector field $V:\mathbb{R}^2\to\mathbb{R}^2$ is here defined by

$$V\left(x^{(1)},x^{(2)}\right)=\left(0,x^{(1)}\right)^T.$$

Given the above endpoints and the vector field, we observe that the force modelled by V will be in average a resistance force to the motion. Indeed the force is oriented toward the top of the square while the moving point has to go downward. Further, let us note that the integrand of the cost function F_α in the present setting is actually quadratic with respect to $y(t)$ and $\dot{y}(t)$, so that an explicit quadratic formula for $\check{F}(c)$ implemented in `PyRotor` is available.

Here, the reference trajectories are obtained through a random generation process. To do so, we define an arbitrarily trajectory y_R verifying the endpoints conditions and we compute its associated vector c_R; Legendre polynomials are once again used and the dimensions of $y^{(1)}$ and $y^{(2)}$ are here set to 4 and 6. Let us note that y_R is designed in such a way that it has a relevant pattern but not the optimal one. Then we construct a set of reference trajectories by adding centered Gaussian noises to c_R. It is noteworthy that the noise is generated in such a way that it belongs to the null space of the matrix A describing the endpoints conditions; the resulting noised trajectories satisfy then these conditions. Further the trajectories which go out of the square $[0,1]^2$ are not kept. At the end, we get 122 generated reference trajectories assumed to be realistic in this setting, each of them containing 81 time observations. Among these reference trajectories, we use the 10 most efficient ones with respect to the cost F_α.

In the present example, we set a v_{max} relatively large to explore a large domain around the reference trajectories. In this case, the objective function of the optimization problem (19) may be not convex even if it is still quadratic. So we make use of the generic optimization solver `minimize(method='trust-constr')` imported in `PyRotor`. Regarding the execution time, we have randomly and uniformly generated 100 values in the interval $[0,10]$ for the parameter α and executed `PyRotor` for each of them. The mean of `PyRotor` execution time is 0.44 s with standard deviation 0.03 s on an Intel Core i7 6 cores running at 2.2 GHz.

In Figure 4, we plot four optimized trajectories associated with different values of α: 0, 0.35, 1, and 10. As expected the trajectory associated with the largest value of α gives the most straight trajectory while the most curvy one is associated with $\alpha=0$. In particular, the latter tends to move to the left at the beginning where the force V is the smallest before going to the ending point in a quasi-straightforward way so that the force is perpendicular to the motion. This example illustrates especially that our optimization approach may lead to optimized trajectories which slightly differ from the reference ones to reduce more the cost.

A quantitative comparison in terms of work gains for different values of α is provided in Table 2. The results confirm the above observations on the curves and show that a right value for α has to be fixed depending on the setting.

6. Perspectives

We have proposed an approach for data-driven optimized trajectories without involving dynamical system or numerous constraints. The approach can work based on a known cost function or a cost function learnt from the data. The modeling of the trajectories allows to take into account explicit and implicit linear constraints on the coefficients in the optimization problem. Contrary to full optimization approaches, our method finds a trade-off between high density constraints-compliant solutions and fully optimized solutions through the tuning of the regularization. In the aeronautic framework, our approach leads to promising fuel-efficient trajectories which can be considered by aeronautics experts. Our

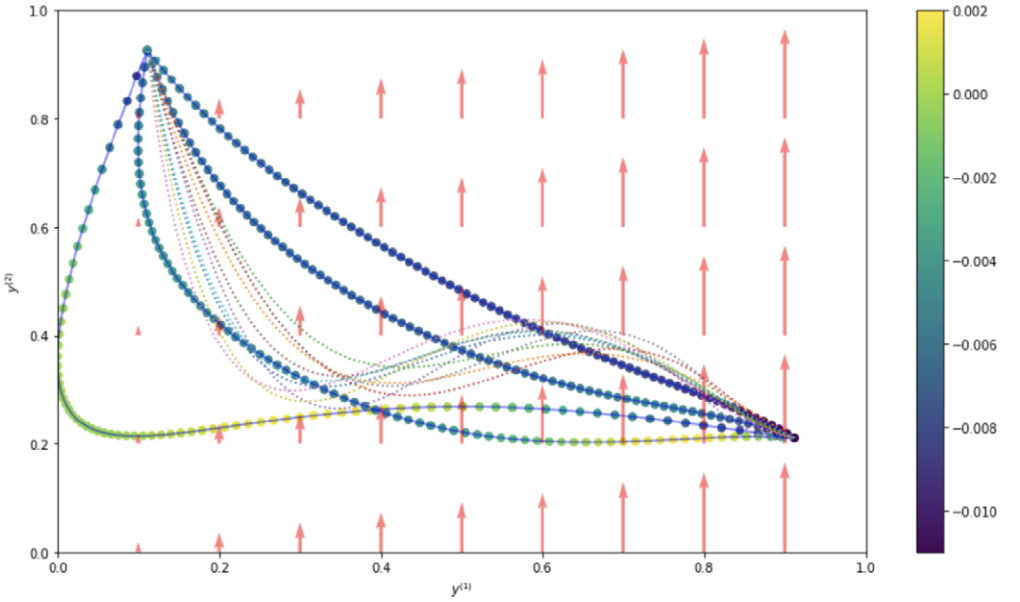

Figure 4. *Optimized trajectories in the square* $[0,1]^2$ *for* $\alpha \in \{0, 0.35, 1, 10\}$. *Optimized and reference trajectories are respectively given by plain and dotted curves. Coloured dots indicate the power value of the force at different points of the optimized trajectories and the bar shows the scale. Red arrows represent the pattern of the vector field* V.

Table 2. *Statistical description of the work gains in percentage for* $\alpha \in \{0, 0.35, 1, 10\}$

	Mean	Standard deviation	Min	Q_1	Q_2	Q_3	Max
$\alpha=0$	73.43	2.36	68.63	71.90	73.25	74.67	80.69
$\alpha=0.35$	45.88	4.81	36.09	42.75	45.49	48.39	60.66
$\alpha=1$	–6.12	9.43	–25.31	–12.26	–6.88	–1.20	22.87
$\alpha=10$	–34.54	11.96	–58.87	–42.32	–35.50	–28.30	2.22

The values have been computed using the 122 available reference trajectories. Negative percentages indicate that no work gains have been obtained. Q_1, Q_2, and Q_3 refer to the first, second, and third quartiles.

approach is generic enough to be applied to other physical settings such as the motion of a moving point in a force field (such as a sailing boat).

Some perspectives of this work are first to further exploit the flexibility of Bayesian setting, by not only searching for the mode of the posterior distribution but also sampling by means of MCMC algorithms. A second perspective would be to consider a clustering of reference trajectories, and apply our strategy on each cluster, then particularize the optimal trajectory depending on the cluster. Last but not least, we aim at adapting our approach where some component of the trajectory would be categorical variables: this would be particularly useful for decision making processes in various disciplines.

Acknowledgments. The authors are grateful to Baptiste Gregorutti and Pierre Jouniaux for fruitful discussions about the modeling of the problem and the validation of the results for the aeronautic application.

Data Availability Statement. Data that support the findings of this study have been gathered from the Quick Access Recorder (QAR) of aircrafts operated by a private airline, hence can not be released publicly for commercial reasons.

Author Contributions. Formal analysis: F.D., B.G., and V.V.; Methodology: F.D., B.G., A.T., and V.V.; Software: F.D., A.T.; Writing – original draft: F.D.; Writing – review & editing: B.G., A.T., and V.V. All authors approved the final submitted draft.

Funding Statement. This project has received funding from the Clean Sky 2 Joint Undertaking under the European Union's Horizon 2020 research and innovation programme under grant agreement No 815914 (PERF-AI). The funder had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing Interests. The authors declare no competing interests exist.

References

Andersen MS, **Dahl J and Vandenberghe L** (2020) Cvxopt: a python package for convex optimization. Available at cvxopt.org.

Berkenkamp F, **Turchetta M**, **Schoellig AP, and Krause A** (2017) Safe model-based reinforcement learning with stability guarantees. In *Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17*. Red Hook, NY: Curran Associates Inc., pp. 908–919.

Boyd S and Vandenberghe L (2004) *Convex Optimization*. Cambridge: Cambridge University Press.

Chettibi T, **Lehtihet H**, **Haddad M and Hanchi S** (2004) Minimum cost trajectory planning for industrial robots. *European Journal of Mechanics—A/Solids 23*(4), 703–715.

Codina RD and Menéndez XP (2014) *How much fuel and time can be saved in a perfect flight trajectory? Continuous cruise climbs vs. conventional operations. In Proceedings of the 6th International Congress on Research in Air Transportation (ICRAT).*

Conn AR, **Gould NIM and Toint PL** (2000) *Trust Region Methods*. Society for Industrial and Applied Mathematics.

Cots O, **Gergaud J and Goubinat D** (2018) Direct and indirect methods in optimal control with state constraints and the climbing trajectory of an aircraft. *Optimal Control Applications and Methods 39*(1), 281–301.

Dewez F, **Guedj B and Vandewalle V** (2020) From industry-wide parameters to aircraft-centric on-flight inference: improving aeronautics performance prediction with machine learning. *Data-Centric Engineering*.

Girardet B, **Lapasset L**, **Delahaye D and Rabut C** (2014) *Wind-optimal path planning: Application to aircraft trajectories. In 2014 13th International Conference on Control Automation Robotics Vision (ICARCV)*, pp. 1403–1408.

Harris CR, **Millman KJ**, **van der Walt SJ**, **Gommers R**, **Virtanen P**, **Cournapeau D**, **Wieser E**, **Taylor J**, **Berg S**, **Smith NJ**, **Kern R**, **Picus M**, **Hoyer S**, **van Kerkwijk MH**, **Brett M**, **Haldane A**, **Fernández del Río J**, **Wiebe M**, **Peterson P**, **Gérard-Marchant P**, **Sheppard K**, **Reddy T**, **Weckesser W**, **Abbasi H**, **Gohlke C and Oliphant TE** (2020) Array programming with NumPy. *Nature 585*, 357–362.

Hewitt M and Frejinger E (2020) Data-driven optimization model customization. *European Journal of Operational Research 287* (2), 438–451.

Lim Y, **Gardi A**, **Sabatini R**, **Ranasinghe K**, **Ezer N**, **Rodgers K and Salluce D** (2019) Optimal energy-based 4d guidance and control for terminal descent operations. *Aerospace Science and Technology 95*, 105436.

Mowbray M, **Petsagkourakis P**, **del Río Chanona EA and Zhang D** (2021) Safe chance constrained reinforcement learning for batch process control.

Pedregosa F, **Varoquaux G**, **Gramfort A**, **Michel V**, **Thirion B**, **Grisel O**, **Blondel M**, **Prettenhofer P**, **Weiss R**, **Dubourg V**, **Vanderplas J**, **Passos A**, **Cournapeau D**, **Brucher M**, **Perrot M and Duchesnay E** (2011) Scikit-learn: machine learning in Python. *Journal of Machine Learning Research 12*, 2825–2830.

Ramsay JO, **Hooker G**, **Campbell D and Cao J** (2007) Parameter estimation for differential equations: a generalized smoothing approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 69*(5), 741–796.

Rao AV (2009) A survey of numerical methods for optimal control. *Advances in the Astronautical Sciences 135*, 497–528.

Rommel C, **Bonnans F**, **Martinon P and Gregorutti B** (2017). Aircraft dynamics identification for optimal control. In *7th European Conference for Aeronautics and Aerospace Sciences (EUCASS)*.

Rommel C, **Bonnans F**, **Martinon P and Gregorutti B** (2019) Gaussian mixture penalty for trajectory optimization problems. *Journal of Guidance, Control, and Dynamics 42*(8), 1857–1862.

Srinivasan M and Ruina A (2006) Computer optimization of a minimal biped model discovers walking and running. *Nature 439* (7072), 72–75.

Trélat E (2012) Optimal control and applications to aerospace: Some results and challenges. *Journal of Optimization Theory and Applications 154*(3), 713–758.

Virtanen P, **Gommers R**, **Oliphant TE**, **Haberland M**, **Reddy T**, **Cournapeau D**, **Burovski E**, **Peterson P**, **Weckesser W**, **Bright J**, **van der Walt SJ**, **Brett M**, **Wilson J**, **Millman KJ**, **Mayorov N**, **Nelson ARJ**, **Jones E**, **Kern R**, **Larson E**, **Carey CJ**, **Polat, İ**, **Feng Y**, **Moore EW**, **VanderPlas J**, **Laxalde D**, **Perktold J**, **Cimrman R**, **Henriksen I**, **Quintero EA**, **Harris CR**, **Archibald AM**, **Ribeiro AH**, **Pedregosa F**, **van Mulbregt P** and **SciPy 1.0 Contributors** (2020) SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods 17*, 261–272.

Wang Y and Zheng S (2019) The converse of weyl's eigenvalue inequality. *Advances in Applied Mathematics 109*, 65–73.

A. Quadratic Cost Computations

Here, we focus on the case where the cost function $F : C\big([0,T],\mathbb{R}^D\big) \to \mathbb{R}^D$ is of the following form

$$F(y) = \int_0^T y(t)^T Q y(t) + w^T y(t) + r dt, \tag{20}$$

where $Q \in \mathbb{R}^{D\times D}$ is symmetric, $w \in \mathbb{R}^D$ and $r \in \mathbb{R}$. In this setting, we provide explicit formulas for the costs $\check{F} : \mathbb{R}^K \to \mathbb{R}$ and $\widetilde{F} : \mathbb{R}^\sigma \to \mathbb{R}$ defined in Section 2.4. A sufficient condition on the parameter $\kappa > 0$ so that the optimization problem

$$c^\star \in \underset{c\in\mathcal{V}_1}{\arg\min}\ \check{F}(c) + \kappa \sum_{i=1}^{I} \omega_i (c - c_{R_i})^T \Sigma^\dagger (c - c_{R_i}) \tag{21}$$

is a quadratic program in the present setting is then derived. From Section 2.4, we recall that the preceding optimization problem is equivalent to

$$\begin{cases} \widetilde{c}_1^\star \in \underset{\widetilde{c}_1\in\mathbb{R}^\sigma}{\arg\min}\ \widetilde{F}(\widetilde{c}_1) + \kappa \sum_{i=1}^{I} \omega_i (\widetilde{c}_1 - \widetilde{c}_{R_i,1})^T \Lambda_{\Sigma,1}^{-1} (\widetilde{c}_1 - \widetilde{c}_{R_i,1}), \\ \widetilde{c}_2 = V_2^T c_{R_i}, \\ \widetilde{c}_3 = S_{A,2}^{-1} U^T \Gamma. \end{cases} \tag{22}$$

Lemma 1. *Suppose that the cost function F is of the form (20). Then the costs $\check{F}$ and $\widetilde{F}$ are quadratic and explicit formulas are given in (24) and (26).*

Proof. Let $c \in \mathbb{R}^K$ and let $y := \Phi|_{\mathcal{Y}_\mathcal{K}}^{-1} c \in \mathcal{Y}_\mathcal{K}$ be its associated trajectory, which can be represented as follows:

$$\forall d \in \{1,\dots,D\} \quad y^{(d)} = \sum_{k=1}^{K_d} c_k^{(d)} \varphi_k.$$

We also remark that each component of the vector

$$c = \left(c_1^{(1)},\dots,c_{K_1}^{(1)},c_1^{(2)},\dots,c_{K_2}^{(2)},\dots,c_1^{(D)},\dots,c_{K_D}^{(D)}\right)^T$$

can be simply described by a single parameter so that we can write $c = (c_1, c_2, \dots, c_K)^T$.

- *Computation of $\check{F}$*:

We first insert the preceding representation of y into the above quadratic integrand to obtain:

$$\begin{aligned} &y(t)^T Q y(t) + w^T y(t) + r \\ &\quad = \sum_{d_1,d_2=1}^{D} \sum_{k_1=0}^{K_{d_1}} \sum_{k_2=0}^{K_{d_2}} Q_{d_1 d_2} c_{k_1}^{(d_1)} c_{k_2}^{(d_2)} \varphi_{k_1}(t)\varphi_{k_2}(t) + \sum_{d=1}^{D}\sum_{k=0}^{K_d} w_d c_k^{(d)} \varphi_k(t) + r, \end{aligned} \tag{23}$$

for all $t \in [0,T]$. The next step of the proof consists in changing the indices of the above sums. To do so, let us define the matrix $\overline{Q} \in \mathbb{R}^{K\times K}$ and the vector $\overline{w} \in \mathbb{R}^K$ as

$$\overline{Q} := \begin{pmatrix} Q_{11}J_{K_1,K_1} & \cdots & Q_{1D}J_{K_1,K_D} \\ \vdots & & \vdots \\ Q_{D1}J_{K_D,K_1} & \cdots & Q_{DD}J_{K_D,K_D} \end{pmatrix}, \quad \overline{w} := \begin{pmatrix} w_1 J_{1,K_1} & \cdots & w_D J_{1,K_D} \end{pmatrix}^T,$$

where $J_{m,n}$ is the all-ones matrix of size $m \times n$. We also introduce the map $\overline{\varphi} \in C\big([0,T],\mathbb{R}^K\big)$ as

$$\overline{\varphi}(t) := \left(\varphi_1(t),\dots,\varphi_{K_1}(t),\varphi_1(t),\dots,\varphi_{K_2}(t),\dots,\varphi_1(t),\dots,\varphi_{K_D}(t)\right)^T,$$

for all $t \in [0,T]$, where the φ_k are the functional basis elements. We are now in position to change the indices in the sums appearing in (23):

$$\begin{aligned} &\sum_{d_1,d_2=1}^{D} \sum_{k_1=0}^{K_{d_1}} \sum_{k_2=0}^{K_{d_2}} Q_{d_1 d_2} c_{k_1}^{(d_1)} c_{k_2}^{(d_2)} \varphi_{k_1}(t)\varphi_{k_2}(t) + \sum_{d=1}^{D}\sum_{k=0}^{K_d} w_d c_k^{(d)} \varphi_k(t) + r \\ &\quad = \sum_{k_1,k_2=1}^{K} \overline{Q}_{k_1 k_2} c_{k_1} c_{k_2} \overline{\varphi}_{k_1}(t)\overline{\varphi}_{k_2}(t) + \sum_{k=1}^{K} \overline{w}_k c_k \overline{\varphi}_k(t) + r, \end{aligned}$$

where we have used the above rewriting of the vector c. Integrating finally over $[0,T]$ gives

$$\begin{aligned}\check{F}(c) &= \int_0^T y(t)^T Q y(t) + w^T y(t) + r dt \\ &= \sum_{k_1,k_2=1}^{K} \overline{Q}_{k_1 k_2} \int_0^T \overline{\varphi}_{k_1}(t)\overline{\varphi}_{k_2}(t) dt c_{k_1} c_{k_2} + \sum_{k=1}^{K} \overline{w}_k \int_0^T \overline{\varphi}_k(t) dt c_k + rT \\ &= \sum_{k_1,k_2=1}^{K} \check{Q}_{k_1 k_2} c_{k_1} c_{k_2} + \sum_{k=1}^{K} \check{w}_k c_k + rT \\ &= c^T \check{Q} c + \check{w}^T c + rT,\end{aligned} \tag{24}$$

where we have defined

$$\check{Q}_{k_1 k_2} := \overline{Q}_{k_1 k_2} \int_0^T \overline{\varphi}_{k_1}(t)\overline{\varphi}_{k_2}(t) dt \quad , \quad \check{w}_k := \overline{w}_k \int_0^T \overline{\varphi}_k(t) dt. \tag{25}$$

- *Computation of* $\widetilde{F}$:

By the definition of $\widetilde{F}$ given in Section 2.4, we have

$$\widetilde{F}(\widetilde{c}_1) = \check{F}\left(V \begin{pmatrix} \widetilde{c}_1^T & \widetilde{c}_{2,3}^T \end{pmatrix}^T\right),$$

where V has been introduced in Section 2.3 and $\widetilde{c}_{2,3} \in \mathbb{R}^{K-\sigma}$ is defined as follows:

$$\widetilde{c}_{2,3} := \begin{pmatrix} V_2^T c_{R_i} \\ S_{A,2}^{-1} U^T \Gamma \end{pmatrix},$$

here the index i is arbitrarily chosen in $\{1,\dots,I\}$ since the vector $V_2^T c_{R_i}$ has been proved to be independent from i. We introduce now the matrix $\check{Q}_V := V^T \check{Q} V$ and the vector $\check{w}_V := V^T \check{w}$ which can be decomposed as follows:

$$\check{Q}_V = \begin{pmatrix} \check{Q}_{V,11} & \check{Q}_{V,12} \\ \check{Q}_{V,21} & \check{Q}_{V,22} \end{pmatrix} \quad , \quad \check{w}_V = \begin{pmatrix} \check{w}_{V,1} \\ \check{w}_{V,2} \end{pmatrix},$$

where $\check{Q}_{V,11} \in \mathbb{R}^{\sigma\times\sigma}$, $\check{Q}_{V,12} \in \mathbb{R}^{\sigma\times(K-\sigma)}$, $\check{Q}_{V,21} \in \mathbb{R}^{(K-\sigma)\times\sigma}$, $\check{Q}_{V,22} \in \mathbb{R}^{(K-\sigma)\times(K-\sigma)}$, $\check{w}_{V,1} \in \mathbb{R}^{\sigma}$, and $\check{w}_{V,2} \in \mathbb{R}^{K-\sigma}$. Given this and the preceding point, we obtain

$$\begin{aligned}\widetilde{F}(\widetilde{c}_1) &= \begin{pmatrix} \widetilde{c}_1^T & \widetilde{c}_{2,3}^T \end{pmatrix} \left(V^T \check{Q} V\right) \begin{pmatrix} \widetilde{c}_1^T & \widetilde{c}_{2,3}^T \end{pmatrix}^T + \left(V^T \check{w}\right)^T \begin{pmatrix} \widetilde{c}_1^T & \widetilde{c}_{2,3}^T \end{pmatrix}^T + rT \\ &= \begin{pmatrix} \widetilde{c}_1^T & \widetilde{c}_{2,3}^T \end{pmatrix} \check{Q}_V \begin{pmatrix} \widetilde{c}_1^T & \widetilde{c}_{2,3}^T \end{pmatrix}^T + \check{w}_V^T \begin{pmatrix} \widetilde{c}_1^T & \widetilde{c}_{2,3}^T \end{pmatrix}^T + rT \\ &= \widetilde{c}_1^T \check{Q}_{V,11} \widetilde{c}_1 + \widetilde{c}_1^T \check{Q}_{V,12} \widetilde{c}_{2,3} + \widetilde{c}_{2,3}^T \check{Q}_{V,21} \widetilde{c}_1 + \widetilde{c}_{2,3}^T \check{Q}_{V,22} \widetilde{c}_{2,3} \\ &\quad + \check{w}_{V,1}^T \widetilde{c}_1 + \check{w}_{V,2}^T \widetilde{c}_{2,3} + rT.\end{aligned}$$

Rearranging the preceding terms and using the fact that $\check{Q}_V$ is symmetric gives

$$\widetilde{F}(\widetilde{c}_1) = \widetilde{c}_1^T \widetilde{Q} \widetilde{c}_1 + \widetilde{w}^T \widetilde{c}_1 + \widetilde{r}, \tag{26}$$

where

- $\widetilde{Q} := \check{Q}_{V,11}$; (27)
- $\widetilde{w} := 2\check{Q}_{V,12}\widetilde{c}_{2,3} + \widetilde{w}_{V,1}$; (28)
- $\widetilde{r} := \widetilde{c}_{2,3}^T \check{Q}_{V,22} \widetilde{c}_{2,3} + \widetilde{w}_{V,2}^T \widetilde{c}_{2,3} + rT.$ (29)

The optimization problem (22) is then equivalent to the following one in the present quadratic setting:

$$\begin{cases} \widetilde{c}_1^\star \in \underset{\widetilde{c}_1 \in \mathbb{R}^\sigma}{\arg\min}\ \widetilde{c}_1^T \widetilde{Q} \widetilde{c}_1 + \widetilde{w}^T \widetilde{c}_1 + \kappa \sum_{i=1}^{I} \omega_i (\widetilde{c}_1 - \widetilde{c}_{R_i,1})^T \Lambda_{\Sigma,1}^{-1} (\widetilde{c}_1 - \widetilde{c}_{R_i,1}), \\ \widetilde{c}_2 = V_2^T c_{R_i}, \\ \widetilde{c}_3 = S_{A,2}^{-1} U^T \Gamma. \end{cases} \tag{30}$$

In the following result, we provide a sufficient condition on the parameter $\kappa > 0$ so that the problem (30) is a quadratic program. The proof uses the fact that the symmetric matrix associated with the quadratic objective function is now explicit and given by the sum of two matrices. A perturbation result for matrices is then applied to obtain a bound for κ assuring that the symmetric matrix is positive semidefinite.

Theorem 1. *Let $\rho_1 \geqslant \rho_2 \geqslant \cdots \geqslant \rho_\sigma$ and $\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_K$ be respectively the eigenvalues of the symmetric matrices $\widetilde{Q}$ and Σ. If $\kappa \geqslant -\rho_\sigma \lambda_1$ then the optimization problem (30) is a quadratic program.*

Proof. We first note that all the eigenvalues of the matrix Σ are non-negative (because Σ is a covariance matrix) and that $\lambda_{\sigma+1} = \ldots = \lambda_K = 0$ (because rank $\Sigma = \sigma$). In particular, the eigenvalue λ_1 is positive.

Standard calculations show that the symmetric matrix associated with the quadratic objective function of the problem (30) is given by

$$M(\kappa) \coloneqq \widetilde{Q} + \kappa \Lambda_{\Sigma,1}^{-1} \in \mathbb{R}^{\sigma \times \sigma}.$$

Let $\mu_1(\kappa) \geqslant \mu_2(\kappa) \geqslant \ldots \geqslant \mu_\sigma(\kappa)$ denote the eigenvalues of $M(\kappa)$. The goal is then to find a sufficient condition on $\kappa > 0$ so that $\mu_\sigma(\kappa)$ is non-negative to assure that M is positive semidefinite. Since $M(\kappa)$ can be interpreted as a perturbed version of $\widetilde{Q}$, we can apply Weyl's inequality (see for instance Wang and Zheng, 2019) which states

$$\mu_\sigma(\kappa) \geqslant \rho_\sigma + \frac{\kappa}{\lambda_1}.$$

Then choosing κ such that $\kappa \geqslant -\rho_\sigma \lambda_1$ implies that $\mu_\sigma(\kappa) \geqslant 0$, leading to the result.

For the sake of completeness, we finish by rewriting the problem (30) as a quadratic optimization problem in $\mathcal{V}_1 \subset \mathbb{R}^K$.

Proposition 6. *Suppose that the cost function F is of the form (20). Then the optimization problem (21) is equivalent to the following one:*

$$c^\star \in \underset{c \in \mathcal{V}_1}{\arg\min}\, c^T \left(\check{Q} + \kappa \Sigma^\dagger\right) c + \left(\check{w} - 2\kappa \sum_{i=1}^{I} \omega_i \Sigma^\dagger c_{R_i}\right)^T c.$$

Proof. It is sufficient to show that the two following objective functions $g_1, g_2 : \mathbb{R}^K \to \mathbb{R}$ have the same minima:

- $g_1(c) \coloneqq \check{F}(c) + \kappa \sum_{i=1}^{I} \omega_i (c - c_{R_i})^T \Sigma^\dagger (c - c_{R_i})$;
- $g_2(c) \coloneqq c^T \left(\check{Q} + \kappa \Sigma^\dagger\right) c + \left(\check{w} - 2\kappa \sum_{i=1}^{I} \omega_i \Sigma^\dagger c_{R_i}\right)^T c.$

First, we have by standard calculations,

$$\begin{aligned}\sum_{i=1}^{I} \omega_i (c - c_{R_i})^T \Sigma^\dagger (c - c_{R_i}) &= \sum_{i=1}^{I} \omega_i c^T \Sigma^\dagger c - 2\sum_{i=1}^{I} \omega_i c_{R_i}^T \Sigma^\dagger c + \sum_{i=1}^{I} \omega_i c_{R_i}^T \Sigma^\dagger c_{R_i} \\ &= c^T \Sigma^\dagger c - \left(2\sum_{i=1}^{I} \omega_i \Sigma^\dagger c_{R_i}\right)^T c + \sum_{i=1}^{I} \omega_i c_{R_i}^T \Sigma^\dagger c_{R_i},\end{aligned}$$

for any $c \in \mathbb{R}^K$, where we have used $\sum_{i=1}^{I} \omega_i = 1$. Combining now this equality with Lemma 1 implies

$$\begin{aligned}g_1(c) &= c^T \check{Q} c + \check{w}^T c + rT + \kappa \left(c^T \Sigma^\dagger c - \left(2\sum_{i=1}^{I} \omega_i \Sigma^\dagger c_{R_i}\right)^T c + \sum_{i=1}^{I} \omega_i c_{R_i}^T \Sigma^\dagger c_{R_i}\right) \\ &= c^T \left(\check{Q} + \kappa \Sigma^\dagger\right) c + \left(\check{w} - 2\kappa \sum_{i=1}^{I} \omega_i \Sigma^\dagger c_{R_i}\right)^T c + \kappa \sum_{i=1}^{I} \omega_i c_{R_i}^T \Sigma^\dagger c_{R_i} + rT \\ &= g_2(c) + \kappa \sum_{i=1}^{I} \omega_i c_{R_i}^T \Sigma^\dagger c_{R_i} + rT.\end{aligned}$$

Since the two last terms of the last right-hand side do not depend on c, we deduce that the objective functions g_1 and g_2 have the same minima.

Cite this article: Dewez F, Guedj B, Talpaert A and Vandewalle V (2022). An end-to-end data-driven optimization framework for constrained trajectories. *Data-Centric Engineering*, 3: e6. doi:10.1017/dce.2022.6