Polynomial Chaos Explicit Solution of the Optimal Control Problem in Model Predictive Control

T. Lefebvre[1,2], F. De Belie[1,2], and G. Crevecoeur[1,2]
[1]Department of Electrical Energy, Metals, Mechanical Constructions & Systems, Ghent University
Tech Lane Ghent Science Park 913, B-9052 Zwijnaarde, Belgium
[2]Flanders Make Strategic Research Centre for the Manufacturing Industry

Abstract— **A difficulty still hindering the widespread application of Model Predictive Control (MPC) methodologies, remains the computational burden that is related to solving the associated Optimal Control (OC) problem for every control period. In contrast to numerous approximation techniques that pursue acceleration of the online optimization procedure, relatively few work has been devoted towards shifting the optimization effort to a precomputational phase, especially for nonlinear system dynamics. Recently, interest revived in the theory of general Polynomial Chaos (gPC) in order to appraise the influence of variable parameters on dynamic system behaviour and proved to yield reliable results. This article establishes an explicit solution of the multi-parametric Nonlinear Problem (mp-NLP) based on the theoretical framework of gPC, which enabled a polynomial approximated nonlinear feedback law formulation. This resulted in real-time computations allowing for real-time MPC, with corresponding control frequencies up to $2\,\mathrm{kHz}$.**

I. Introduction

Practical implementation of Model Predictive Control (MPC) methods requires numerical solution of an Optimal Control (OC) problem at every control step [1], [2]. It is yet a recognized challenge to complete such optimization within the limits of a control period and therefore application usually remains limited to slowly varying systems. For some relatively simple problems, it is possible to obtain an explicit solution, allowing to replace the optimization by a simple function evaluation; however, the existence of such solution is not guaranteed for more complex systems. To tackle this problem, there exist two conceptions: on the one hand, the online computational effort can be simplified such that yet the solution can be obtained within a control period [3]. On the other hand, the explicit solution can be approximated by a piecewise linear solution.

The first approach makes a compromise on the optimality of the solution, since they are either required to introduce some sort of approximation measure, or have to interrupt the optimization after a single iteration [4]. The iLQR method [5] considers locally linearized system models along the optimized trajectory and allows relatively fast convergence to a (suboptimal) solution, by identification of recursive relations. In [6] the application of the iLQR method on an inverted pendulum is reported, capable of generating controls at a frequency of $4\,\mathrm{Hz}$ which is however still insufficient for highly dynamic systems.

The second idea aims to construct an approximation of the explicit solution of the OC problem in function of the initial state value, which requires to solve a multiparametric Nonlinear Program (mp-NLP). In [7] this is established by partitioning the feasible set of initial state values and by associating a local linear approximation of the feedback law to each partition. Evaluation of the feedback requires execution of a (possibly large) binary tree search through the partitioning structure and storage of an equal number of linear feedback laws. The presented method is able to deal with constraints, set both on input and state. When the required approximation quality is however increased, such will result in an expansion of the number of partitions, consequently increasing the computational effort spent on the tree search. Such is an acknowledged problem, which they solve by posteriorly reducing the total number of partitions by merging adjacent partitions whose policies are only slightly deviating [8].

Considering the latter methodology, one can argue that it is not necessary to construct a local linear approximation over the entire feasible set of initial state values. It might be beneficial to pursue an explicit solution by means of a smooth approximation for regions of the solution space that allow it. A smooth solution is however not suited to approximate the explicit solution when it is an element of the C^0 function class, locally. Typically, regions to which such applies, correspond to initial state values for which the optimal solution does adopt the supremum value(s) for one or more of its inequality constraints. This can be compensated for by reintroduction of partitions. However, the total number of partitions will still be significantly reduced.

In this work it is therefore proposed to establish an approximation of the explicit solution of the mp-NLP in function of the initial state value by expansion over a polynomial base. The polynomial expansion is rooted within the framework of general Polynomial Chaos (gPC) as originally introduced by Wiener [9], towards which interest has flared in the last decade [10]. Different studies already explored the application of the gPC framework to that of dynamic optimization and demonstrated its potential to decompose optimal trajectories over some variable parameters [11]–[13]. Additionally, polynomial expansions exhibit good convergence properties for relatively smooth surfaces even when using sparse grids, as was reported by [14], [15]. Moreover, the recursive generation of the one-dimensional polynomials allows for rapid numerical evaluation of the multivariate polynomial space. This paper presents a decomposition of the resulting large coupled NLP, using a collocation method, into a number of original OC problems in order to obtain the desired polynomial coefficients. This method results in a polynomial interpolation of the optimal feedback law. As a reference, a more general interpolator such as the Kriging method was applied to compare the resulting approximation qualities. It is expected that a Kriging interpolator will yield slightly superior results, certainly for constrained optimization problems. However, for an increasing number of datapoints, which the Kriging interpolator requires to assess all for each evaluation [16], it is expected to be somewhat slower, considering that the polynomial interpolator only requires to evaluate the polynomial base.

The presented framework is limited to unconstrained OC problems, as to guarantee sufficiently smooth control policies. The concept can be extended to constrained problems by reintroduction of partitions as in [8], which was however not considered here yet.

II. Conventional MPC

Within the framework of MPC one faces an optimization problem of the form (1). Conventionally, $\boldsymbol{x}(\tau) \in \mathbb{R}^n$, represents the system state vector, and, $\boldsymbol{u}(\tau) \in \mathbb{R}^m$, an arbitrary input signal. The function, $\boldsymbol{f} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$, describes the governing system dynamics and is assumed time-invariant. Functions, $l : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ and $h : \mathbb{R}^n \to \mathbb{R}$, characterize the running and final cost kernels, which summation ought to be minimized over control horizon N; that by proper choice of the function arguments $\boldsymbol{u} = (\bar{\boldsymbol{u}}(t) \ \cdots \ \bar{\boldsymbol{u}}(t+N-1))$. The control objective is assumed to be the asymptotic stabilization of $\boldsymbol{x}(\tau)$ to the origin.

$$\begin{aligned} V(\boldsymbol{x}(t)) = \min_{\boldsymbol{u}} \ & \sum_{k=0}^{N-1} l(\bar{\boldsymbol{x}}(t+k), \bar{\boldsymbol{u}}(t+k)) + h(\boldsymbol{x}(t+N)) \\ \text{subject to } & \bar{\boldsymbol{x}}(t+k+1) = \boldsymbol{f}(\bar{\boldsymbol{x}}(t+k), \bar{\boldsymbol{u}}(t+k)) \\ & \bar{\boldsymbol{x}}(t) = \boldsymbol{x}(t) \end{aligned} \tag{1}$$

The solution of this mp-NLP will be both function of the (discrete) time, $\tau \in \{t, \ldots, t+N-1\}$, and the initial state value, $\boldsymbol{x}(t)$, and will henceforth be referred to as the optimal control policy, $\boldsymbol{u}^\dagger(\tau, \boldsymbol{x}(t))$, corresponding optimal trajectory $\boldsymbol{x}^\dagger(\tau, \boldsymbol{x}(t))$. Here, $J(\cdot,\cdot)$ and $G(\cdot,\cdot)$ are compact representations of the cost and constraint functions of problem (1).

$$\boldsymbol{u}^\dagger(\cdot, \boldsymbol{x}) = \arg\min_{\boldsymbol{u}} \ J(\boldsymbol{u}, \boldsymbol{x}) \text{ subject to } G(\boldsymbol{u}, \boldsymbol{x}) = 0 \tag{2}$$

The principle idea in MPC is thus to solve the optimization problem for the current state (belief), $\boldsymbol{x}(t)$, at every time t and set the required input $\boldsymbol{u}(t)$ according the optimized policy $\boldsymbol{u}^\dagger(t, \boldsymbol{x}(t))$, yielding the state feedback function $\boldsymbol{\eta}(\cdot)$.

$$\boldsymbol{\eta}(\boldsymbol{x}(t)) \equiv \boldsymbol{u}^\dagger(t, \boldsymbol{x}(t)) \tag{3}$$

It is a known property of $V(\boldsymbol{x}) = J(\boldsymbol{u}^\dagger(\cdot, \boldsymbol{x}), \boldsymbol{x})$, that it serves as a Lyapunov function for the closed-loop system $\boldsymbol{x}(\tau+1) = \boldsymbol{f}(\boldsymbol{x}(\tau), \boldsymbol{\eta}(\boldsymbol{x}(\tau)))$, for well chosen l and h [2], and as such guarantees asymptotic stability to the origin.

Despite this strategy's stability assurance, it requires instant evaluation of the optimized control policy and thus instant solution of the optimization problem; which is unlikely, if not impossible. In the remaining we shall demonstrate that an explicit approximation of solution, $\boldsymbol{u}^\dagger(\tau, \boldsymbol{x}(t))$, can be established by expanding the optimized trajectory and policy over a polynomial base in function of the initial state, $\boldsymbol{x}(t)$. Recursive evaluation of the polynomial base and multiplication with the polynomials coefficients allow for rapid (almost instant) evaluation of the optimized feedback law, as is required to apply the MPC method.

III. Polynomial Chaos expansion

A. Wiener-Askey Polynomial Chaos

According to the gPC framework, the dependency of a sufficiently smooth variable $\boldsymbol{z}(\boldsymbol{\xi}) \in \mathbb{R}^k$ on the variable $\boldsymbol{\xi} \in \Omega = \otimes_{i=1}^n \Gamma_i \subset \mathbb{R}^n$, can be modelled by decomposition over a family of polynomials, ϕ_i, as illustrated by the equation below, with, $\boldsymbol{\xi}$, being a random variable associated to probability function $\rho(\boldsymbol{\xi}) : \Omega \to \mathbb{R}^+$.

$$\boldsymbol{z}(\boldsymbol{\xi}) = \sum_{i=0}^{\infty} \boldsymbol{z}_i \phi_i(\boldsymbol{\xi}) \tag{4}$$

A relation is established between various distributions and their corresponding polynomial basis by the Wiener-Askey scheme [17]. As we intent to decompose the optimal trajectories over a set of feasible initial states, and considering that all are equally important, a uniform distribution is reckoned most appropriate to model the dependency. According the scheme, the uniform distribution is related to the one-dimensional Legendre polynomials, $P_k(\xi)$, which are defined on the interval $\Gamma = [-1, 1]$. The base polynomials satisfy the orthonormality relation $\langle P_i, P_j \rangle = \delta_{ij}$, with the inner product defined as $\langle \psi, \phi \rangle = \int_\Gamma \psi(\xi)\phi(\xi)\rho(\xi)\mathrm{d}\xi$. It is of interest to note that these polynomials satisfy the Bonnet recursion formula, allowing for rapid numerical evaluation.

$$\begin{aligned} (k+1)P_{k+1}(\xi) = (2k+1)\xi P_k(\xi) - kP_{k-1}(\xi) \\ P_0(\xi) = 1, \ P_1(\xi) = \xi \end{aligned} \tag{5}$$

The one-dimensional orthogonal polynomial space Λ^p, spanning polynomials up to degree p, can formally be defined as $\Lambda^p = \{v : \Gamma \to \mathbb{R} : v \in \operatorname{span}\{P_k(\xi)\}_{k=0}^p\}$. Correspondingly we can construct an n-dimensional polynomial space on Ω as $\Lambda_n^p = \bigotimes_{|\boldsymbol{i}| \leq p} \Lambda^{i_k}$. With multi-index $\boldsymbol{i} = (i_1 \ \cdots \ i_n)$ and the total degree $\sum_k i_k \leq p$. As such the multivariate polynomials $\phi_i(\boldsymbol{\xi})$ are evaluated as below. The possibility to exploit the recursion formula during numerical evaluation is not affected by the multivariate construction.

$$\phi_i(\boldsymbol{\xi}) = P_{i_1}(\xi_1) \cdots P_{i_n}(\xi_n) \tag{6}$$

Given that the polynomial degree is restricted to p, the series expansion as introduced in (4) is truncated after M polynomial contributions, where $M = \binom{n+p}{p}$. We obtain expansion $\boldsymbol{z}^p(\boldsymbol{\xi})$, which is an approximation of $\boldsymbol{z}(\boldsymbol{\xi})$.

$$\boldsymbol{z}^p(\boldsymbol{\xi}) = \sum_{i=1}^{M} \boldsymbol{z}_i^p \phi_i(\boldsymbol{\xi}) \tag{7}$$

This approximate series can be written more compactly as

$$\begin{aligned}\boldsymbol{z}^p(\boldsymbol{\xi}) &= \begin{pmatrix}\phi_1(\boldsymbol{\xi}) & \cdots & \phi_M(\boldsymbol{\xi})\end{pmatrix} \otimes \mathbb{I}_n \cdot \begin{pmatrix} \boldsymbol{z}_1^p \\ \vdots \\ \boldsymbol{z}_M^p \end{pmatrix} \\ &= \boldsymbol{\phi}^\top(\boldsymbol{\xi}) \otimes \mathbb{I}_n(\boldsymbol{\xi}) \cdot \mathrm{Z}^p \end{aligned} \tag{8}$$

Where $\otimes$ represents the Kronecker product, and $\mathbb{I}_n$ the identity matrix of dimension n.

B. Determination of Coefficients

As we wish to appraise the effect of arbitrary $\boldsymbol{\xi}$ on the variable $\boldsymbol{z}(\boldsymbol{\xi})$, we require to dispose over the polynomial coefficients Z^p. In order to determine this set, one would usually try to exploit the orthonormality property and project the variable $\boldsymbol{z}(\boldsymbol{\xi})$ on the multivariate polynomial space, Λ_n^p. Those are referred to as Galerkin procedures; this and others are documented in [10]. This would however require to evaluate the inner product integral which is unfortunately nontrivial, if not impossible in the case of nonlinear relations [18]. One is thus obligated to use an approximate integration rule to compute the exact integral, based on a number of actual function evaluations, $\boldsymbol{z}(\boldsymbol{\xi}^j)$, at Q arbitrary - so called - collocation nodes, $\boldsymbol{\xi}^j$. In case of a one dimensional variable, we obtain for every, $j = 1, \ldots, Q$.

$$z_j^\xi = z(\xi^j) \approx z^p(\xi^j) = \boldsymbol{\phi}^\top(\xi^j) \cdot \mathrm{Z}^p \tag{9}$$

Introducing the collocation output vector, Z^ξ, and the system matrix, $\Phi \in \mathbb{R}^{Q \times M}$, consisting of the elements $\Phi_{ji} = \phi_i(\xi^j)$, this can be formulated as a linear system.

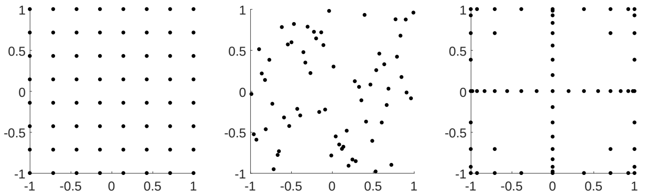
$$\mathrm{Z}^\xi = \begin{pmatrix} z_1^\xi \\ \vdots \\ z_Q^\xi \end{pmatrix} \approx \begin{pmatrix} \boldsymbol{\phi}^\top(\xi^1) \\ \vdots \\ \boldsymbol{\phi}^\top(\xi^Q) \end{pmatrix} \cdot \begin{pmatrix} z_1^p \\ \vdots \\ z_M^p \end{pmatrix} = \boldsymbol{\Phi} \cdot Z^p \tag{10}$$

In [18], it is suggested to approximate the integral that determines the projection $\langle z, \phi_i \rangle$ by an approximate integration rule, resulting in the estimate, $\tilde{\mathrm{Z}}^p = \boldsymbol{\Phi}^\top \mathrm{W} \cdot \mathrm{Z}^\xi$ with W a diagonal matrix containing the associated integration weights. The authors in [19] suggest however to minimize the cost defined by $\mathbb{E}([z^p - z]^2)$ over the coefficients Z^p, with $\mathbb{E}(z)$ defined as $\int_\Omega z(\xi)\rho(\xi)\mathrm{d}\xi$ and approximate the associated integral. For a uniform distribution this results in the familiar Least Squares (LS) approximation [10].

$$\tilde{\mathrm{Z}}^p = \left(\boldsymbol{\Phi}^\top \mathrm{W} \boldsymbol{\Phi}\right)^{-1} \boldsymbol{\Phi}^\top \mathrm{W} \cdot \mathrm{Z}^\xi \tag{11}$$

The matrix representations of both collocation approaches holds a certain similarity; in that the estimated coefficients $\tilde{\mathrm{Z}}^p$ are spanned by the weighted collocation output, $\mathrm{W} \cdot \mathrm{Z}^\xi$, projected on the coefficient space by the transposed system matrix $\boldsymbol{\Phi}$. Whether or not the estimate is multiplied with $(\boldsymbol{\Phi}^\top \mathrm{W} \boldsymbol{\Phi})^{-1}$ determines the method. One can observe that the first method directly estimates the N first coefficients, whilst the LS method additionally rescales the same estimate to account for the truncated tail of the polynomial expansion and is therefore preferred in this study. The corresponding approximate polynomial decomposition is then given by

$$\tilde{z}^p(\xi) = \boldsymbol{\phi}^\top(\xi) \cdot \tilde{\mathrm{Z}}^p \tag{12}$$



(a) Uniform, $Q = 64$ (b) Random, $Q = 64$ (c) Smolyak, $Q = 65$

Fig. 1: Nodal set distributions

C. Nodal Set Distribution

The sole restriction to which the choice of Q collocation nodes is subject, is that the corresponding system matrix, $\boldsymbol{\Phi}$, has to be well-conditioned. Additionally, a lower limit is set by the total truncation number, M, which the number of nodes, Q, must either match or exceed. Regarding these requirements, two sampling techniques are convenient: either a full tensor-product grid is spanned over the feasible space or an (orthogonal) random sampling distribution is practiced.

Both are however flawed by practical objections; the number of nodes corresponding the tensor product grows exponentially with respect to the problems dimensionality, whilst the random sampling cannot guarantee solution consistency over each run. An answer to both casualties is possibly given by the Smolyak method [18] which only considers a subset of the full tensor-product, ruling out all the nodes with negligible importance to the quality of the approximation. In case the grid is constructed over the one-dimensional Chebyshev points, the grid is referred to as the Clenshaw-Curtis abscissas. The approximation quality corresponding each distribution schemes is compared in section V, a visual illustration of the set themselves can be found in figure 1.

D. Polynomial Approximation Error

In order to provide an estimate for the accumulated approximation error, introduced by the truncation and the coefficient estimation, two metrics are defined. They indicate the relative approximation errors between: the real function $\boldsymbol{z}(\boldsymbol{\xi})$ and the truncated expansion, $\boldsymbol{z}^p(\boldsymbol{\xi})$, and, between the latter and the estimated coefficient expansion, $\tilde{\boldsymbol{z}}^p(\boldsymbol{\xi})$, respectively given by $\epsilon_M = \mathbb{E}([\boldsymbol{z} - \boldsymbol{z}^p]^2)^{\frac{1}{2}}$ and $\epsilon_Q = \mathbb{E}([\boldsymbol{z}^p - \tilde{\boldsymbol{z}}^p]^2)^{\frac{1}{2}}$.

As shown in [18], the accumulated error, $\epsilon = \mathbb{E}([\boldsymbol{z} - \tilde{\boldsymbol{z}}^p]^2)^{\frac{1}{2}}$, then satisfies the inequality below, with both ϵ_M and ϵ_Q converging to 0 for increasing M or Q, respectively.

$$\epsilon \leq (\epsilon_M^2 + \epsilon_Q^2)^{\frac{1}{2}} \tag{13}$$

IV. Polynomial Chaos Expansion of the OCP

Here, we introduce the theory of gPC into the framework of optimal control, to establish an eplixit solution of the mp-NLP problem (1) over the variable initial state value, $\boldsymbol{x}(t)$. We shortly discuss how the desired polynomial coefficients, Z^p, can be obtained by merit of the collocation approach. Once the coefficients are available, an approximation of the explicit optimal feedback law, $\boldsymbol{\eta}(\boldsymbol{x}(t))$, for arbitrary initial state values can be easily obtained. For notational clarity, the derivation is carried out on a scalar variable.

A. Explicit Solution of the mp-NLP

We wish to solve problem (1) explicitly in function of $x(t) = \xi \in \Omega$, yielding the explicit trajectories $x^\dagger(\tau, \xi)$ and $u^\dagger(\tau, \xi)$. This thus requires to solve the mp-NLP

$$\arg\min_u \; J(u(\tau,\xi),\xi) \text{ subject to } G(u(\tau,\xi),\xi) = 0 \quad (14)$$

According to the gPC framework, the variable, $z(\xi)$, can be decomposed over a polynomial space, Λ_n^p. The expansion is characterised by the coefficients, Z^p, or an appropriate estimate. In case of a time dependent variable, $z(\tau, \xi)$, the coefficients will be time dependent as well, $\mathrm{Z}^p(\tau)$. This allows to decompose the explicit trajectories over the polynomial base: $x^p(\tau,\xi) = \sum_{i=1}^M x_i^p(\tau)\phi_i(\xi)$ and $u^p(\tau,\xi) = \sum_{i=1}^M u_i^p(\tau)\phi_i(\xi)$, for $\xi \in \Omega$ and Ω a feasible hypercube of initial states. With appropriate scaling this then coincides with the feasible set of the Legendre polynomials.

We now need to to find the time dependent coefficients $x_i^p(\tau)$ and $u_i^p(\tau)$, which dynamics are still governed by the original dynamic relation. Substitution yields

$$\sum_{i=1}^M x_i^p(\tau+1)\phi_i(\xi) = f\left(\sum_{i=1}^M x_i^p(\tau)\phi_i(\xi), \sum_{i=1}^M u_i^p(\tau)\phi_i(\xi)\right) \quad (15)$$

This large system of coupled time dependent coefficients is not straightforward to solve. However, following the collocation technique, we invoke Q arbitrary nodes, ξ^j, that are assigned to the variable initial state. Such allows to consider the variables $x_j^\xi(\tau)$ and $u_j^\xi(t)$ similar to the approach in [20].

$$x_j^\xi(\tau) = \sum_{i=1}^M x_i^p(\tau)\phi_{ji} \qquad u_j^\xi(\tau) = \sum_{i=1}^M u_i^p(\tau)\phi_{ji} \quad (16)$$

By merit of this collocation technique, solution of problem (14) simplifies to solving Q problems of the form (1), for the variables, x_j^ξ and u_j^ξ, and dynamic relation, $x_j^\xi(\tau+1) = f(x_j^\xi(\tau), u_j^\xi(\tau))$, and, initial state values equal to collocation node, ξ^j. The solution yields the optimal collocation trajectories, $x_j^{\xi,\dagger}(\tau)$ and $u_j^{\xi,\dagger}(\tau)$, $j = 1, \ldots, Q$.

$$u_j^{\xi,\dagger}\left(\cdot, \xi^j\right) = \arg\min_u J\left(u, \xi^j\right) \text{ subject to } G\left(u, \xi^j\right) \quad (17)$$

According to (10) these collocation trajectories can on their turn be related to the optimal polynomial coefficients, $x_i^{p,\dagger}(t)$, and, $u_i^{p,\dagger}(t)$, as illustrated by the expression below.

$$\begin{pmatrix} u_1^{\xi,\dagger}(\tau) \\ \vdots \\ u_Q^{\xi,\dagger}(\tau) \end{pmatrix} = \begin{pmatrix} \phi_{11} & \cdots & \phi_{1M} \\ \vdots & \ddots & \vdots \\ \phi_{Q1} & \cdots & \phi_{QM} \end{pmatrix} \cdot \begin{pmatrix} u_1^{p,\dagger}(\tau) \\ \vdots \\ u_M^{p,\dagger}(\tau) \end{pmatrix} \quad (18a)$$

$$\mathrm{U}^{\xi,\dagger}(\tau) = \mathbf{\Phi} \cdot \mathrm{U}^{p,\dagger}(\tau) \quad (18b)$$

This system of equations can now be solved to yield the optimal time dependent polynomial coefficients. In case the number of collocation nodes Q equals the number of coefficients M this requires inversion of matrix $\mathbf{\Phi}$; if the number of collocation nodes is however larger than the number of coefficients, one can adopt the LS approach.

$$\tilde{\mathrm{U}}^{p,\dagger}(\tau) = \left(\mathbf{\Phi}^\top \mathrm{W} \mathbf{\Phi}\right)^{-1} \mathbf{\Phi}^\top \mathrm{W} \cdot \mathrm{U}^{\xi,\dagger}(\tau) \quad (19)$$

B. Smooth Interpolated MPC

Now that we dispose over the optimal control coefficient vector, $\tilde{\mathbf{U}}^{p,\dagger}(\tau)$, the polynomial expansion is applied to approximate the explicit optimal control policy, $\boldsymbol{u}^\dagger(t, \boldsymbol{x}(t))$.

$$\boldsymbol{u}^\dagger(t,\boldsymbol{x}(t)) \approx \tilde{\boldsymbol{u}}^{p,\dagger}(t,\boldsymbol{x}(t)) = \boldsymbol{\phi}^\top(\boldsymbol{x}(t)) \otimes \mathbb{I}_n \cdot \tilde{\mathbf{U}}^{p,\dagger}(t) \quad (20)$$

Recalling the definition of the MPC feedback (3), substitution of the approximate expression (20) for the accurate solution yields the interpolated feedback law

$$\tilde{\boldsymbol{\eta}}(\boldsymbol{x}(t)) \equiv \boldsymbol{\phi}^\top(\boldsymbol{x}(t)) \otimes \mathbb{I}_m \cdot \tilde{\mathbf{U}}^{p,\dagger}(t) \quad (21)$$

The online computations are as such reduced to evaluation of the polynomial expansion, which will require only a fraction of the computational resources and storage that is required to numerically solve (1) online, nor does it require to execute a possibly extensive tree search and storage of an equal amount of linear state feedbacks.

C. Stability Criterion

By merit of the MPC framework, it holds that $V(\boldsymbol{x})$ is a Lyapunov function for the closed-loop system $\boldsymbol{x}(\tau+1) = \boldsymbol{f}(\boldsymbol{x}(\tau), \boldsymbol{\eta}(\boldsymbol{x}(\tau))) = \boldsymbol{f}_\eta(\boldsymbol{x}(t))$ and assures asymptotic stability. It can also be shown that $V(\boldsymbol{x}) - V(\boldsymbol{f}_\eta(\boldsymbol{x})) > l(\boldsymbol{x}, \boldsymbol{\eta}(\boldsymbol{x})), \forall \boldsymbol{x} \in \Omega$ [2]. Now, consider the closed-loop system dynamics $\boldsymbol{f}_{\tilde{\eta}}(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x}, \tilde{\boldsymbol{\eta}}(\boldsymbol{x}))$. Assessing the inequality and the property of V, $V(\boldsymbol{x}) > 0, \forall \boldsymbol{x} \in \Omega \backslash \{0\}$, it holds that stability of the system can still be assured if $V(\boldsymbol{f}_{\tilde{\eta}}(\boldsymbol{x})) - V(\boldsymbol{f}_\eta(\boldsymbol{x})) < V(\boldsymbol{x}) - V(\boldsymbol{f}_\eta(\boldsymbol{x})), \forall x \in \Omega \backslash \{0\}$. This condition allows us to establish a tolerance on the error, $\|\tilde{\boldsymbol{\eta}}(\boldsymbol{x}) - \boldsymbol{\eta}(\boldsymbol{x})\|$, which the approximation must satisfy. Assume that V and $\boldsymbol{f}$ are both C^1 functions; therefor there must exist local Lipschitz constants $\kappa_v(\boldsymbol{x}) > 0$ and $\kappa_f(\boldsymbol{x}) > 0$ such that $V(\boldsymbol{f}_{\tilde{\eta}}(\boldsymbol{x})) - V(\boldsymbol{f}_\eta(\boldsymbol{x})) \leq \kappa_v(\boldsymbol{x})\kappa_f(\boldsymbol{x})\|\tilde{\boldsymbol{\eta}}(\boldsymbol{x}) - \boldsymbol{\eta}(\boldsymbol{x})\|$. Substitution in the stability criterion $V(\boldsymbol{f}_{\tilde{\eta}}(\boldsymbol{x})) - V(\boldsymbol{f}_\eta(\boldsymbol{x})) < V(\boldsymbol{x}) - V(\boldsymbol{f}_\eta(\boldsymbol{x}))$ and some basic manipulation yield then a limit for the maximal polynomial error ϵ.

$$\epsilon \simeq \|\tilde{\boldsymbol{\eta}}(\boldsymbol{x}) - \boldsymbol{\eta}(\boldsymbol{x})\| < \frac{l(\boldsymbol{x}, \boldsymbol{\eta}(\boldsymbol{x}))}{\kappa_v(\boldsymbol{x})\kappa_f(\boldsymbol{x})} \quad (22)$$

From III-D we know, that the above criterion can be enforced if M and Q are taken large enough.

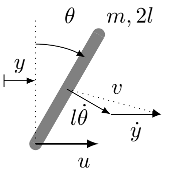
D. Ordinary Kriging Interpolator

Alternative to the polynomial expansion in (21), motivated by the explicit solution of (14), one could argue to fit an interpolator directly on the data $\mathrm{U}^{\xi,\dagger}(t)$ as to approximate $\boldsymbol{\eta}(\boldsymbol{x})$. The Ordinary Kriging Interpolator is given by ($m = 1$, for $m > 1$ the procedure is simply repeated m times) [21].

$$\tilde{\eta}(\boldsymbol{x}) = \hat{\eta} + \boldsymbol{r}(x)\mathbf{R}^{-1}\left(\mathrm{U}^{\xi,\dagger}(t) - \mathbf{1}\hat{\eta}\right) \quad (23)$$

Where $\mathrm{R}_{ij} = \psi(\boldsymbol{\xi}^i - \boldsymbol{\xi}^j, \boldsymbol{\theta})$, $\hat{\eta} = \mathbf{1}^\top\mathbf{R}^{-1}\mathrm{U}^{\xi,\dagger}(t)/\mathbf{1}^\top\mathbf{R}^{-1}\mathbf{1}$ and $\boldsymbol{r}(\boldsymbol{x}) = (\psi(\boldsymbol{x} - \boldsymbol{\xi}^1, \boldsymbol{\theta}) \; \cdots \; \psi(\boldsymbol{x} - \boldsymbol{\xi}^Q, \boldsymbol{\theta}))$. The correlation parameter $\boldsymbol{\theta}$ is determined by maximizing $-\log(|\mathbf{R}(\boldsymbol{\theta})|)$, which is a measure for the Likelihood of observation $\mathrm{U}^{\xi,\dagger}(t)$. The function $\psi : \mathbb{R}^n \to \mathbb{R}$ is usually chosen to be a radial base function, e.g. $\psi(\boldsymbol{d}) = \prod_{i=1}^n \exp(-\theta_j d_j^2)$. Note that computation of the interpolator requires to evaluate Q distances $|\boldsymbol{x} - \boldsymbol{\xi}_j|$. Implementational details are found in [16].

V. Balancing of the inverted pendulum



Consider the model of a nonlinear inverted pendulum which base is free to move along the horizontal direction.

Expressions for the kinetic and potential energy, denoted as, T, and, V, are presented in equation (24) in terms of the coordinates θ and y, with $m = 0.2\,\mathrm{kg}$ and $2l = 0.6\,\mathrm{m}$. The velocity of the centre of mass satisfies the relation, $v^2 = l^2\dot{\theta}^2 + \dot{y}^2 + 2l\dot{\theta}\dot{y}\cos(\theta)$, following the law of cosines.

$$T = \tfrac{1}{2}\tfrac{1}{3}ml^2\dot{\theta}^2 + \tfrac{1}{2}mv^2 \quad V = mgl\cos(\theta) \tag{24}$$

The Lagrange formalism yields the governing equation (25) for the state variable θ with arbitrary force term u applied to the base. An additional damping term, $\lambda\dot{\theta}$, with λ equal to $0.1\,\mathrm{Nms}$ is added to model friction at the suspension point. We introduce the state vector, $\boldsymbol{x} = (\theta\ \dot{\theta})^\top$, and the arbitrary force term, u, is associated with the system input.

$$\ddot{\theta} = -\frac{\sin(\theta)\cos(\theta)\dot{\theta}^2 + \frac{\lambda}{ml^2}\dot{\theta} - \frac{g}{l}\sin(\theta)}{\frac{4}{3} - \cos^2(\theta)} - \frac{\cos(\theta)}{\frac{4}{3} - \cos^2(\theta)}\frac{u}{ml} \tag{25}$$

An OC problem is defined with a quadratic running cost kernel, presented in equation (26). The control horizon is set to $500\,\mathrm{ms}$ with a control period of $20\,\mathrm{ms}$, resulting in $N = 25$. The feasible subspace, Ω, is defined as $\left\{\frac{\pi}{3}\,[-1,1] \times 2\pi\,[-1,1]\right\}$. The NLP (17) are solved using the commercial software package MATLAB. Solution of the OC problem, with a $2.7\,\mathrm{GHz}$ processor and $4\,\mathrm{GB}$ RAM, lasts $0.8\,\mathrm{s}$ up to $1.7\,\mathrm{s}$, ruling out real-time MPC regardlessly.

$$\boldsymbol{x}(\tau)^\top \mathrm{diag}(100, 10)\boldsymbol{x}(\tau) + u^2(\tau) \tag{26}$$

Figure 2 portrays a contour plot of the exact MPC feedback, $\eta(\boldsymbol{x}(t))$, over the considered subspace. The approximation was performed for several nodal set distributions and varying polynomial degree. The number of collocation points, Q, corresponding the Smolyak distribution is restricted to the set: $\{1, 5, 13, 29, 65, 145, 321, \ldots\}$ of which the first two were not considered to secure well-conditioning of the system matrix for 2nd degree approximations. In order to achieve comparable results, the number of collocation points of the uniform grid were selected as proximate to those of the Smolyak grid, yielding the set: $\{16, 25, 64, 144, 324\}$. For the random grid the same numbers were adopted. Recalling the average solution time of $1\,\mathrm{s}$, the highest precomputational load, corresponding the largest number of collocation nodes, took about $10\,\mathrm{min}$. In order to compare results to a more general interpolator, also an Ordinary Kriging (OK) approximation was considered.

The approximation quality was estimated by means of the error, $\epsilon = \mathbb{E}([\tilde{\eta} - \eta]^2)^{\frac{1}{2}}$, which was approximated over a uniform 30×30 grid, $\epsilon \simeq \frac{1}{N^2}\sqrt{\sum_{i,j=1}^{N}(\tilde{\eta}(\boldsymbol{\xi}^{ij}) - \eta(\boldsymbol{\xi}^{ij}))^2}$. The values corresponding each combination are reported in table I. Considering the values observed for the random grid, the mean value over 10 tries is depicted. In case a value is lacking the associated systems matrix is ill-conditioned.

Uniform $[1e-3]$

$Q\backslash p$	2	3	4	5	6	7	8	9	OK
16	211	211							989
25	194	197							695
64	178	178	56	56	40	40			35
144	167	170	54	54	23	23	2	2	4
324	166	166	53	53	21	21	2	2	2

Random $[1e-3]$

$Q\backslash p$	2	3	4	5	6	7	8	9	OK
16	356								507
25	223	420	289						94
64	188	224	87	143	105	248	61		30
144	177	191	66	93	47	69	10	21	9
324	173	176	59	63	35	41	4	3	1

Smolyak $[1e-3]$

$Q\backslash p$	2	3	4	5	6	7	8	9	OK
13	299	299							1210
29	275	275	77	77					44
65	223	223	73	73	34	34			38
145	208	208	65	65	29	29	2	2	43
321	200	200	63	63	26	26	2	2	1

TABLE I: $\mathbb{E}([\tilde{\eta} - \eta]^2)^{\frac{1}{2}}$ values corresponding different nodal set distributions strategies for varying m and s.

The uniform and Smolyak nodal sets exhibit close to equal approximation qualities and similar convergence properties. The random grid yields less reliable approximations levels, yet results in an improved conditioning of the system matrix for higher approximation levels with low collocation number. As expected the OK interpolator slightly outperforms the polynomial approximation for large collocation numbers, especially in case of the random grid. Note that in the case of the uniform and the Smolyak grid, the quality of the odd degree approximation is equivalent to that of the lower even degree, which is explained considering both the symmetry exhibited by the surface as the nodal set and the presence of the same set of even polynomials in the expansion.

The average calculation time required to obtain the interpolated feedback value is less than $400\,\mu\mathrm{s}$, corresponding the 9th degree polynomial approximation and is negligible compared to the control period, as such enabling for real-time MPC implementations. This is somewhat faster than the OK

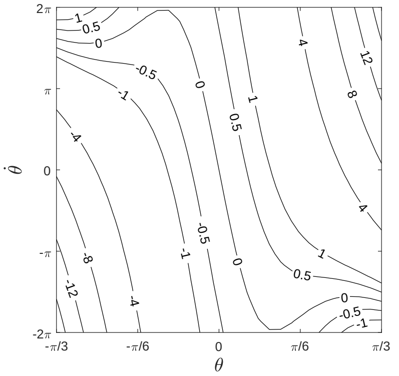


Fig. 2: Contour plot of the exact MPC feedback, $\eta(\boldsymbol{x}(t))$.

approximation which computation time was about 700 µs for the same nodal set. This is explained by observing that the polynomial approximation only requires to evaluate 55 polynomials compared to 324 radial base function evaluations that the OK interpolator requires to execute. Additionally, it is observed that overfitting the nodal set with respect to the polynomial degree seems to be a necessary practice as to secure well-conditioning of the system matrix.

Lastly, a phase plot of varying optimal trajectories, simulated over the period of 1 s, is portrayed in figure 3. The pendulum is successfully stabilized to the upward position, starting from 13 different initial state values; both for the exact MPC as for the polynomial MPC strategy. However indiscernible visually, the largest deviation occurred close to the origin given that the difference, $l(\boldsymbol{x}, \boldsymbol{\eta}(\boldsymbol{x}))$, vanishes when approaching the origin. By consequence of the stability criterion (22), the effect of the approximation error will thus be more pronounced in proximity of the origin.

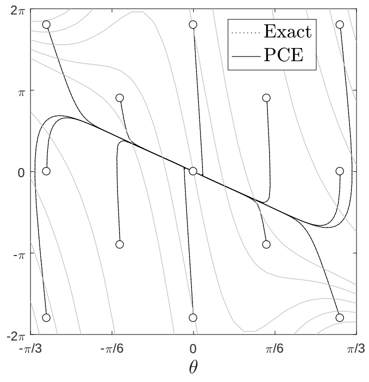


Fig. 3: Simulated trajectories for the MPC feedback, both exact and gPC expanded with polynomial degree $p = 9$ over a uniform nodal set of 324 collocation points. The trajectories coincide almost perfectly and can't be distinguished visually.

VI. Conclusion and future perspectives

This discussion established an explicit solution of the mp-NLP, associated to the OC problem in MPC methodologies, that by merit of the gPC framework and the collocation approach. As a consequence of this explicit solution, the computation time associated with solution of the exact online OC problem ($\sim$1 s) drastically diminished ($\sim$400 µs), lowering the implementational threshold of MPC strategies, even for low-cost embedded applications. A criterion to maintain the stabilizing property was obtained, in the sense of a bound on the approximation error. It is noted that once the collocation output vector is available, the polynomial degree can still be varied yielding a required approximation level. Nonetheless, the authors wish to emphasize that further research is required in terms of stability and scalability.

In conclusion, it is brought to attention that the polynomial coefficients are determined by solving Q deterministic optimal control problems. In the future we intent to consider a stochastic optimal control problem at each of the collocation nodes in terms of some variable systems parameters.

Acknowledgements

The authors wish to express their gratitude to five anonymous reviewers for their constructive feedback that resulted into significant improvement of the manuscript. The authors also acknowledge support of the Research Foundation Flanders project G0D9316N, and the strategic basic research project EMODO of the Flanders Make Strategic Research Centre for the Manufacturing Industry.

References

[1] M. Morari and J. Lee. Model predictive control: past, present and future. *Computers & Chemical Engineering*, 23(4):667–682, 1999.

[2] S.S. Keerthi and E. G. Gilbert. Optimal infinite-horizon feedback laws for a general class of constrained discrete-time systems: Stability and moving-horizon approximations. *Journal of optimization theory and applications*, 57(2):265–293, 1988.

[3] Y. Wang and S. Boyd. Fast model predictive control using online optimization. *IEEE Transactions on Control Systems Technology*, 18(2):267–278, 2010.

[4] M. Diehl, H. Bock, H. Diedam, and P. Wieber. Fast direct multiple shooting algorithms for optimal robot control. *Fast Motions in Biomechanics and Robotics*, 2005.

[5] A. Bemporad, M. Morari, V. Dua, and E. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1):3–20, 2002.

[6] W. Li and E. Todorov. Iterative linear quadratic regulator design for nonlinear biological movement systems. *Proceedings of the International Conference on Informatics in Control, Automation and Robotics*, 2004.

[7] T. A. Johansen. Approximate explicit receding horizon control of constrained nonlinear systems. *Automatica*, 40(2):293–300, 2004.

[8] T. A Johansen. On multi-parametric nonlinear programming and explicit nonlinear model predictive control. *Proceedings of the 41st IEEE Conference on Decision and Control*, 2002.

[9] N. Wiener. The homogeneous chaos. *American Journal of Mathematics*, 60(4):897–936, 1938.

[10] K. K. Kim, D. E. Shen, Z. K. Nagy, and R. D. Braatz. Wiener's polynomial chaos for the analysis and control of nonlinear dynamical systems with probabilistic uncertainties. *IEEE Control Systems*, 33(5):58–67, 2013.

[11] P. Nimmegeers, D. Telen, F. Logist, and J. Van Impe. Dynamic optimization of biological networks under parametric uncertainty. *BMC Systems Biology*, 10(1):86, 2016.

[12] F. S. Hover and M. S. Triantafyllou. Dynamic optimization using hermite chaos. *Proceedings of the IASTED conference on control applications*, 2005.

[13] F. S. Hover. Gradient dynamic optimization with legendre chaos. *Automatica*, 44(1):135–140, 2008.

[14] Bruno Sudret. Global sensitivity analysis using polynomial chaos expansions. *Reliability Engineering & System Safety*, 93(7):964–979, 2008.

[15] G. T. Buzzard. Global sensitivity analysis using sparse grid interpolation and polynomial chaos. *Reliability Engineering & System Safety*, 107:82–89, 2012.

[16] S. N. Lophaven, H. B. Nielsen, and J. Søndergaard. Dace-a matlab kriging toolbox, version 2.0. Technical report, 2002.

[17] D. Xiu and G. Karniadakis. The wiener-askey polynomial chaos for stochastic differential equations. *SIAM journal on scientific computing*, 24(2):619–644, 2002.

[18] D. Xiu. Efficient collocational approach for parametric uncertainty analysis. *Communications in Computational Physics*, 2(2):293–309, 2007.

[19] H. Cheng and A. Sandu. Collocation least-squares polynomial chaos method. *Proceedings of the 2010 Spring Simulation Multiconference*, 2010.

[20] P. Manfredi, D. Van de Ginste, D. De Zutter, and F. G. Canavero. Generalized decoupled polynomial chaos for nonlinear circuits with many random parameters. *IEEE Microwave and Wireless Components Letters*, 25(8):505–507, 2015.

[21] Donald R Jones. A taxonomy of global optimization methods based on response surfaces. *Journal of global optimization*, 21(4):345–383, 2001.