2019 6th International Conference on Control, Decision and
Information Technologies (CoDIT'19) | Paris, France / April 23-26, 2019

Gradient-Descent based Nonlinear Model Predictive Control for Input-Affine Systems

Carlos Andrés Devia[1], Julian Colorado[2] and Diego Patino[3]

Abstract— **This paper addresses the Nonlinear Model Predictive Control of Input-Affine Systems. The Two Point Boundary Value Problem resulting from the associated Optimal Control Problem is reformulated as an optimization problem, which is locally convex under assumptions coherent with the application. This optimization problem is solved on-line using the gradient descent method, where the gradients are approximated based on geometrical information of the dynamic system differential equations. The resulting control method is summarized in three algorithms. The proposed controller is easy to implement and requires no iterations. As a consequence, the suboptimal control input can be computed in a short time interval, making it ideal for fast highly nonlinear systems. As an example the attitude control of a quadrotor is presented. Simulation results show excellent performance in a wide range of state values, well beyond linear regimes.**

I. Introduction

Optimal control is a control technique that computes and applies an optimal control signal to a dynamic system. The optimal control signal is the solution of an Optimal Control Problem (OCP) which is an optimization problem, where a cost function depending on the states of the system and the control input is minimized having as constraint the dynamic of the system represented in the corresponding set of differential equations. Model Predictive Control (MPC) is the application of an optimal controller in closed loop. During a fix time interval an optimal control signal is computed and applied. At the start of the next time interval the states are measured and a new optimal control signal applied.

The need to solve a new OCP in each time interval limits the applicability of MPC to slow or linear plants. This is due to the fact that being an optimization problem the solution of an OCP for a nonlinear plant requires the solution of a nonconvex optimal problem, which demands large computation efforts and time. That is the reason why Nonlinear MPC (NMPC) is mainly used in slow plants.

There are two types of approaches to solve a NMPC. One way aims to solve the OCP using some optimization technique that reduces the required computation time. The second type of solution relies on circumventing the need of solving the OCP using alternative techniques.

On one hand, in the first type of approach the main difference is the algorithm used to solve the associated nonlinear optimization problem. In [1] Neighboring External Updates were used in the solution of a dynamic optimization problem applied to MPC. Optimistic Optimization was applied in [2] to continuous piecewise affine system. Interior Point methods and Sequential Quadratic Programming applied to NMPC were studied in [3]. A NMPC for a robotic arm was considered in [4], the optimization problem was solved using a Differential Evolution algorithm. In [5] a Recurrent Neural Network was used to solve the optimization problem.

On the other hand, the second type of solution relies on necessary and sufficient conditions for optimality or numerical methods that are proven to converge to near optimal solutions. In [6] the projected gradient method [7] is used to develop a gradient based nonlinear model predictive control software *GRAMCP*. If the nonlinear system is affine with respect to the input it is possible to use a State-Dependent Riccati Equation (SDRE) Control. This method relies on the extended linearization of the nonlinear system and then, in each sample instant solve an Algebraic Riccati Equation (ARE) [8]. Although this method has proven efficient for the control of input affine systems, it still requires the online solution of an ARE, which depending on the sample frequency and available hardware may not be possible.

A continuation/GMRES method is proposed in [9] to solve the NMPC problem. In this approach the continuation method is coupled with a fast algorithm for linear equations in order to compute the optimal control law. Dynamic programming, and numerical methods without NNA for approximately solving the HJB equation [10] are also contained in this second class of solutions.

The technique proposed in this paper falls into the second type of solution, overcoming the need of online optimization by solving the OCP analytically. This approach is known as Pontryagin's Maximum Principle (PMP) and leads a system of ordinary differential equations (ODEs) whose solutions are the optimal states and costates, which then are used to compute the optimal control signal. Depending on the boundary conditions of the states, PMP produces different systems of ODEs. In order to compute the optimal control signal a solution of the system of ODEs that satisfies the required boundary conditions needs to be found. This is known as a *Two Point Boundary Value Problem* (TPBVP).

Common methods to solve the TPBVP include Simple Shooting Method, Multiple Shooting Method, Shooting to a Fitting Point and Relaxation Methods [11]. Generally these techniques rely of numerical analysis and require multiple iterations as well as the numerical integration of the dynamic system ODEs. More advanced methods such as the Modified Simple Shooting Method [12] improve on convergence speed

*This work was supported by Pontificia Universidad Javeriana

[1]Faculty of Electrical Engineering, Pontificia Universidad Javeriana, Bogotá, Colombia devia.c@javeriana.edu.co
[2]Faculty of Electrical Engineering, Pontificia Universidad Javeriana, Bogotá, Colombia coloradoj@javeriana.edu.co
[3]Faculty of Electrical Engineering, Pontificia Universidad Javeriana, Bogotá, Colombia patino-d@javeriana.edu.co

and reduction on computational burden, while still requiring the dynamic system numerical integration.

In this paper a completely different approach is explored. The TPBVP is reformulated as an optimization problem, that in the context of NMPC becomes the tracking of an optimal point corresponding to the initial costates of the system. This optimization problem is then solved using a gradient-descent like method. The approximation is valid only in a short time interval, which is consistent with the application to NMPC.

The proposed technique provides several important advantages over conventional methods to solve TPBVPs. First of all it requires no iterations and the costates are approximated in a single step. There is no need to numerically solve the dynamic system since only the first input is applied and a significant portion of the computations can be performed off-line. As a result the proposed control algorithm is both simple to implement and fast. The required computational burden is minimal and can be used in both linear and nonlinear systems.

This paper is organized as follows: in section II the Two Point Boundary Value Problem is introduced and reformulated as an optimization problem. The proposed controller is described in section III, which ends with three algorithms that summarize the control method. An application example is presented in section IV together with results analysis. Finally section V discusses conclusions and future work.

II. Two Point Boundary Value Problem

A. Optimal Control Problem

Consider the following optimal control problem for an input-affine dynamic system:

$$\min_{u(\cdot)} \left\{ J(x,u) = \frac{1}{2}\int_{t_o}^{t_f} \left(x^T Q x + u^T R u \right) dt \right\} \tag{1}$$

$$s.t. \quad \dot{x} = F(x) + G(x)u \quad x(t_o) = \widehat{x_o} \quad x(t_f) = \widehat{x_f} \tag{2}$$

Where $F : \mathbb{R}^n \to \mathbb{R}^n$ and $G : \mathbb{R}^n \to \mathbb{R}^{n\times m}$ are analytic vector functions of the state $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$ is the system input. Q is a positive definite matrix, R is a positive semi definite matrix and $u(t)$ is the unconstrained system input.

The necessary condition for optimality is given by PMP in the form of the following system of ODEs:

$$\dot{x} = F - GR^{-1}G^T\lambda \tag{3}$$

$$\dot{\lambda} = -Qx - \partial_x F^T \lambda + \begin{bmatrix} \lambda^T \partial_{x_1} G R^{-1} G^T \lambda \\ \vdots \\ \lambda^T \partial_{x_n} G R^{-1} G^T \lambda \end{bmatrix} \tag{4}$$

Subject to the boundary conditions $x(t_o) = \widehat{x_o}$ and $x(t_f) = \widehat{x_f}$.Where $\lambda = \lambda(t) \in \mathbb{R}^n$ are known as the costates, and the optimal control input is given by:

$$u^* = -R^{-1}G^T(x)\lambda \tag{5}$$

The solution to this system of nonlinear differential equations gives the optimal control as well as the state trajectory.

Although there are no constraints in the optimal control problem the use of this technique is justified for systems with a high degree of nonlinearities. In such instances linear-based methods fail to control the system even if the linearization is done for several operating points. For such cases the complete system nonlinearities must be taken into account.

B. Power Series Solution of PMP Equations

The dynamic system (2) can be expanded using Taylor series around the operation states $\bar{x}$, yielding an equivalent input-affine dynamic system $\dot{x} = \tilde{F}(x) + \tilde{G}(x)u$. Where the coefficients of the vector functions $\tilde{F}$ and $\tilde{G}$ are power series corresponding to the multivariable Taylor expansion of the original coefficients of F and G (which exist, given that the vector functions are analytical). As a consequence, equations (3) and (4) transform accordingly to

$$\dot{x} = \tilde{F} - \tilde{G}R^{-1}\tilde{G}^T\lambda \tag{6}$$

$$\dot{\lambda} = -Qx - \partial_x \tilde{F}^T \lambda + \begin{bmatrix} \lambda^T \partial_{x_1} \tilde{G} R^{-1} \tilde{G}^T \lambda \\ \vdots \\ \lambda^T \partial_{x_n} \tilde{G} R^{-1} \tilde{G}^T \lambda \end{bmatrix} \tag{7}$$

The resulting system of differential equations although nonlinear involves only polynomial terms of the states. Thus, the solution of equations (6) and (7) can be approximated by a power series of the form (8) for a short time interval.

$$x^i(t) = \sum_{k=0}^{\infty} a_k^i (t-t_o)^k \qquad \lambda^i(t) = \sum_{k=0}^{\infty} b_k^i (t-t_o)^k \quad i = 1,\ldots,n \tag{8}$$

The coefficients a_k^i and b_k^i can be computed via a recurrence equation obtained by replacing the assumed power series and corresponding derivatives into equations (3), (4) and then matching the corresponding coefficients in each side of the equations.

Given that the original vector functions F and G were assumed analytic, in a neighborhood of the operation states $\bar{x}$ and in a sufficiently small time interval the suboptimal control signal obtained by replacing solutions (8) in (5) approaches arbitrarily the optimal solution depending on the number of terms used in the expansion.

C. Problem Statement

In order to numerically compute the coefficients a_k^i and b_k^i of equations (8) not only the recurrence equations are required but also the initial values for x and λ.

In a practical implementation the initial values of the states x can be measured. However, the initial values of the costates λ have no physical interpretation and there is no direct relation between the initial costates and the final states besides integration of the vector fields (3) and (4). Which in general can only be done numerically, demanding significant computational effort. Limiting the applicability of this approach to slow dynamical systems.

The problem then becomes: *compute the initial costates $\widehat{\lambda_o}$ that together with the initial states $\widehat{x_o}$ produce the desired final states $\widehat{x_f}$ in a fix time period $t_o - t_f$ satisfying the dynamic system $\dot{x} = F(x) + G(x)u$.*

III. Gradient-Descent based NMPC

A. Proposed Solution of the TPBVP

The TPBVP can be formally stated as an optimization problem in the following way:

$$\min_{\lambda_o \in \Lambda_o} \left\{ J(\lambda_o) = (x_f - \widehat{x_f})^T \widehat{Q} (x_f - \widehat{x_f}) \right\} \tag{9}$$
$$\text{s.t.} \quad x_f = \varphi_{\widehat{x_o}}(\lambda_o)$$

Where $\widehat{Q}$ is a positive definite matrix and $\varphi_{\widehat{x_o}}(\lambda_o)$ is a vector function that returns the final states x_f obtained by integrating the vector fields (3) and (4) with the initial conditions $x(t_o) = \widehat{x_o}$ and $\lambda(t_o) = \lambda_o$. The corresponding coefficients of the vector function $\varphi_{\widehat{x_o}}$ are denoted $\varphi^i_{\widehat{x_o}}$.

The functions $\varphi^i_{\widehat{x_o}}$ are smooth given that the vector fields (3) and (4) are smooth. However, there is no guarantee of convexity that could aid in the solution of the optimization problem (9), thus two further assumptions are made:

1) Previous states and coestates $\overline{x_o}$ and $\overline{\lambda_o}$ are known to produce final states $\overline{x_f}$ close to the desired final states $\widehat{x_f}$. Moreover these previous states and costates are in a neighborhood of the current states $\widehat{x_o}$ and optimal costates $\widehat{\lambda_o}$. This assumption makes sense in the context of model predictive control, since in each sample time the initial and final states vary with a bounded rate of change.
2) The problem is feasible. That is, there exists at least one solution $\widehat{\lambda_o}$ such that $\varphi_{\widehat{x_o}}(\widehat{\lambda_o}) = \widehat{x_f}$. Furthermore in a neighborhood of $\widehat{\lambda_o}$ that contains $\overline{\lambda_o}$ the function $\varphi_{\widehat{x_o}}(\cdot)$ is convex. This requirement relies on the sample time and the intrinsic geometry of the vector fields.

Returning to (9), if the matrix $\widehat{Q}$ is diagonal the cost function becomes

$$J = \sum_{i=1}^{n} q_i (\varphi^i_{\widehat{x_o}}(\lambda_o) - \widehat{x_f}^i)^2$$

Thus

$$\frac{\partial}{\partial \lambda_o^j} J = 2 \sum_{i=1}^{n} q_i (\varphi^i_{\widehat{x_o}}(\lambda_o) - \widehat{x_f}^i) \frac{\partial}{\partial \lambda_o^j} \varphi^i_{\widehat{x_o}}(\lambda_o)$$

Using the two assumptions, the function $\varphi^i_{\widehat{x_o}}$ can be locally approximated by

$$\varphi^i_{\widehat{x_o}}(\lambda_o) \approx \varphi^i_{\overline{x_o}}(\overline{\lambda_o}) + \nabla \varphi^i_{\widehat{x_o}}(\overline{\lambda_o}) \cdot (\lambda_o - \overline{\lambda_o})$$

Hence

$$\frac{\partial}{\partial \lambda_o^j} \varphi^i_{\widehat{x_o}}(\lambda_o) \approx \left(\varphi^i_{\widehat{x_o}}(\overline{\lambda_o}) \right)^j$$

Finally, the gradient of the cost function J can be approximated locally by:

$$\nabla J \Big|_{\lambda_o = \overline{\lambda_o}} = 2 \sum_{i=1}^{n} q_i (\overline{x_f}^i - \widehat{x_f}^i) \cdot \nabla \varphi^i_{\overline{x_o}}(\overline{\lambda_o}) \tag{10}$$

The problem becomes the computation of $\nabla \varphi^i_{\overline{x_o}}(\overline{\lambda_o})$. Assuming sufficiently small sample periods where the states do not change significantly it is possible to make the following approximation:

$$\nabla \varphi^i_{\overline{x_o}}(\overline{\lambda_o}) \approx \nabla_\lambda (\dot{x}^i)|_{\overline{x_o}, \overline{\lambda_o}} \tag{11}$$

Where ∇_λ denotes the gradient with respect to λ only. The great advantage of equation (11) is that it can be computed analytically off-line, thus ∇J reduces to a linear combination of known gradients whose weights are the differences between the measured and desired final states $\overline{x_f}^i - \widehat{x_f}^i$. Furthermore, assuming an input affine system the divergence $\nabla_\lambda(\dot{x}^i)$ is simply the i-th row of the matrix $G(x)R^{-1}G(x)^T$ and only depends on the states x. As a consequence equation (10) can be written in matrix form resulting in equation (12).

$$\nabla J = G(x_o)^T R^{-1} G(x_o)(\overline{x_f} - \widehat{x_f}) \tag{12}$$

Finally the optimization problem (9) is solved using the classical gradient descent method. Approximating the gradient with equation (12).

B. Model Predictive Control Algorithm

The aforementioned procedure can applied to nonlinear model predictive control in two algorithms: first an off-line algorithm were the divergences are computed and the initial costate found and then the iterative on-line algorithm used to compute the suboptimal control law in each sample interval.

Both algorithms have the following inputs:

- The sample time interval T_s.
- A vector of references for each state $x^i_{\text{ref}}[k]$, $1 \leq k \leq K$.
- An initial state x_o.
- The dynamic model $\dot{x} = F(x) + G(x)u$.
- A gradient descent stop criteria ε.
- The step size for the gradient descent method μ.

The objective of the off-line algorithm (algorithm 1) is to compute off-line the necessary functions and the initial optimal costates.

Algorithm 1 Off-line computations

Fix the order of approximation N and compute the Taylor polynomial of order N of the dynamic system

$$\dot{x} = \tilde{F}^{(N)}(x) + \tilde{G}^{(N)}(x)u \tag{13}$$

Using the approximation functions $\tilde{F}^{(N)}$ and $\tilde{G}^{(N)}$ compute the associated PMP equations (6) and (7).

Assume a power series solution for the states and the costates.

$$x^i(t) = \sum_{n=0}^{\infty} a^i_n t^n \qquad \lambda^i(t) = \sum_{n=0}^{\infty} b^i_n t^n$$

Replace in the differential equations obtained in the previous step and match coefficients of the same power to obtain the recurrence equation for the coefficients a^i_n and b^i_n.

Initialize a random seed for λ_o and ∇J.

while $\varepsilon < |\nabla J|$ **do**

  Compute the final states $x_f = \varphi_{x_o}(\lambda_o)$.

  Compute the approximation of the cost function gradient ∇J

$$\nabla J = G(x_o)^T R^{-1} G(x_o)(x_f - x_{\text{ref}}[k])$$

  Update the initial optimal $\lambda_o = \lambda_o - \mu \nabla J$

The on-line algorithm (algorithm 2) computes the suboptimal input signal for each sample period using the approximation of the gradient and the state errors as weights.

Algorithm 2 On-line computations

Fix a limit M for the approximate solutions.
while $1 \leq k \leq K$ **do**
  Using x_o, λ_o and the recurrence equations compute the power series approximate solution to PMP equations.

$$x^i(t) = \sum_{n=0}^{M} a_n^i t^n \qquad \lambda^i(t) = \sum_{n=0}^{M} b_n^i t^n \qquad i = 1,2,\ldots,n$$

  Compute the suboptimal control law

$$u(t) = -R^{-1}G^T(x(t))\lambda(t)$$

  For $t < T_s$ apply the suboptimal control law (note that in each sample instant the initial time t_o is equal to 0, therefore the final time t_f becomes T_s).
  Measure the final states x_f.
  Compute the approximation of the cost function gradient ∇J

$$\nabla J = G(x_o)^T R^{-1} G(x_o)(x_f - x_{\text{ref}}[k])$$

  Update the new initial costates

$$\lambda_o = \lambda_o - \mu \nabla J$$

  Update the new initial states $x_o = x_f$.

Note that if only the first control input is applied $x(t) = a_0$ and $\lambda(t) = b_0$. Moreover the initial conditions imply that $a_0 = x_o$ and $b_0 = \lambda_o$, hence algorithm 2 simplifies to algorithm 3.

Algorithm 3 Simplified on-line computations

while $1 \leq k \leq K$ **do**
  For $t < T_s$ apply the constant suboptimal control law

$$u = -R^{-1}G^T(x_o)\lambda_o$$

  Measure the final states x_f.
  Compute the approximation of the cost function gradient ∇J

$$\nabla J = G(x_o)^T R^{-1} G(x_o)(x_f - x_{\text{ref}}[k])$$

  Update the new initial costates

$$\lambda_o = \lambda_o - \mu \nabla J$$

  Update the new initial states $x_o = x_f$.

IV. Application Example

As an application example the attitude control problem of a quadrotor is considered.

A. Quaternion-Based Attitude Model

The attitude model is divided into two parts:

1) *Angular rates of change* with state variables p, q and r that represent the rates of change in roll, pitch and yaw. The inputs of this subsystem are torques τ_ψ, τ_θ and τ_ϕ and the parameters are the moments of inertia I_x, I_y and I_z in the x, y and z axes respectively.

$$\dot{p} = \left(\frac{I_y - I_z}{I_x}\right) qr + \frac{\tau_\phi}{I_x}$$
$$\dot{q} = \left(\frac{I_z - I_x}{I_y}\right) pr + \frac{\tau_\theta}{I_y}$$
$$\dot{r} = \left(\frac{I_x - I_y}{I_z}\right) qp + \frac{\tau_\psi}{I_z}$$

2) *Angular position* with state variables ψ, θ and ϕ the Euler angles roll, pitch and yaw respectively. The inputs are the angular rates of change p, q and r.

$$\dot{\phi} = p + q\sin\phi\tan\theta + r\cos\phi\tan\theta$$
$$\dot{\theta} = q\cos\phi - r\sin\phi$$
$$\dot{\psi} = \frac{\sin\phi}{\cos\theta}q + \frac{\cos\phi}{\cos\theta}r$$

The angular position subsystem is highly nonlinear and although it is possible to find a Taylor series expansion, the alternative representation using quaternions yields an already expanded form:

$$\dot{q}_0 = \frac{1}{2}(-q_1 p - q_2 q - q_3 r)$$
$$\dot{q}_1 = \frac{1}{2}(q_0 p + q_2 r - q_3 q)$$
$$\dot{q}_2 = \frac{1}{2}(q_0 q - q_1 r + q_3 p)$$
$$\dot{q}_3 = \frac{1}{2}(q_0 r + q_1 q - q_2 p)$$

In the quaternion-based attitude model a single unit quaternion $\mathbf{q} = q_o + q_1\hat{i} + q_2\hat{j} + q_3\hat{k}$ is used to represent the rotation in space of the quadrotor, thus relating its angular position. The angular state variables ϕ, θ and ψ become the components of the quaternion that describes the rotation of the quadrotor in space q_0, q_1, q_2 and q_3.

B. Control Architecture

As with the model, the control architecture is divided into two parts: the angular rate of change control and the quaternion-based angular position control:

1) *Angular rate of change control*: given the symmetry of the dynamic system it is straight forward to control p, q and r using feedback linearization simply by making the torques

$$\tau_\psi = u_p I_x - (I_y - I_z)qr$$
$$\tau_\theta = u_q I_y - (I_z - I_x)pr$$
$$\tau_\phi = u_r I_z - (I_x - I_y)pq$$

The overall dynamic decouples into three separate systems

$$\dot{p} = u_p \quad \dot{q} = u_q \quad \dot{r} = u_r$$

2) *Quaternion-based angular position control*: writing the system in standard input affine system (2) (renaming

the state variables $x_1 = q_0, x_2 = q_1, x_3 = q_2$ and $x_4 = q_3$) yields:

$$F(x) = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \qquad G(x) = \frac{1}{2}\begin{pmatrix} -x_2 & -x_3 & -x_4 \\ x_1 & -x_4 & x_3 \\ x_4 & x_1 & -x_2 \\ -x_3 & x_2 & -x_1 \end{pmatrix}$$

This controller is implemented using algorithms 1 and 3.

C. Simulation and Results

For the simulation the following parameters were used:

- Total time $T_f = 3$s.
- The sample time interval $T_s = 1$ms.
- An initial state $x_o = x_{\text{ref}}[0]$.
- A gradient descent stop criteria $\varepsilon = 10^{-3}$.
- The step size for the gradient descent method $\mu = 200$.
- Control input cost matrix $R = \text{diag}(10, 15, 20)$.

Assuming that the actuators frequency update is $f = 1$kHz in each sample interval the input is constant, thus the on-line simplified algorithm is used.

The angular references ψ_{ref}, θ_{ref} and ϕ_{ref} were obtained from the xy trajectory shown in figure 1.

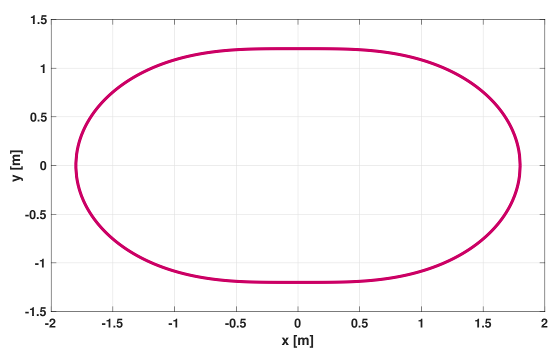

Fig. 1. xy trajectory used to create the attitude references.

The ψ_{ref}, θ_{ref} and ϕ_{ref} references were then translated into quaternion references with the following transformations [13]:

$$\begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{pmatrix} = \begin{pmatrix} \cos(\phi/2)\cos(\theta/2)\cos(\psi/2) + \sin(\phi/2)\sin(\theta/2)\sin(\psi/2) \\ \sin(\phi/2)\cos(\theta/2)\cos(\psi/2) - \cos(\phi/2)\sin(\theta/2)\sin(\psi/2) \\ \cos(\phi/2)\sin(\theta/2)\cos(\psi/2) + \sin(\phi/2)\cos(\theta/2)\sin(\psi/2) \\ \cos(\phi/2)\cos(\theta/2)\sin(\psi/2) - \sin(\phi/2)\sin(\theta/2)\cos(\psi/2) \end{pmatrix}$$

Figure 2 shows the reference and real quaternion components, while figure 3 presents the corresponding angular position references and states.

Table I summarizes important information from figure 3. It presents the minimum and maximum values for each angle (roll, pitch and yaw) as well as the Mean Absolute Error (MAE) given by:

$$\text{MAE} = \frac{1}{n}\sum_{k=1}^{n} |x_{\text{real}}[k] - x_{\text{ref}}[k]|$$

Note that because of the short time interval in which the maneuver is performed it requires a displacement across a wide range of values for the angular states. As a result, the

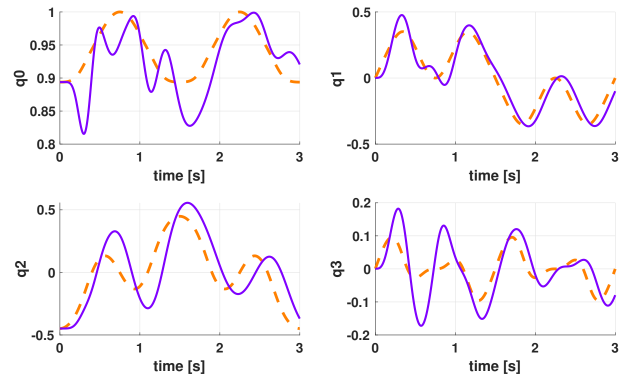

Fig. 2. Quaternion component references (dotted orange line) and real states (solid purple line).

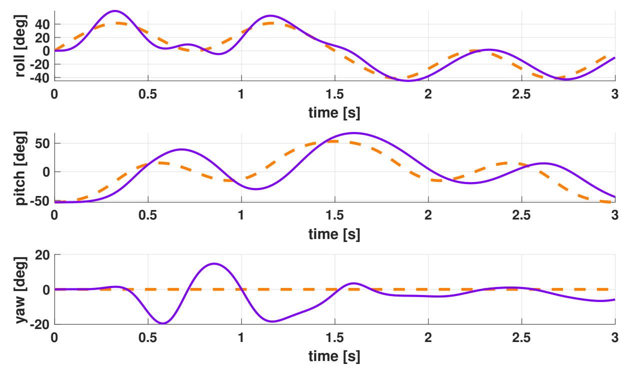

Fig. 3. Angular position results. The reference in the dotted orange line while the real states are in purple solid line.

roll and pitch angles range from $-45.03°$ to $59.6°$ and $53.2°$ to $67.7°$ respectively, well beyond the linear regime.

Angle	Min		Max		MAE	
	Rad	Deg	Rad	Deg	Rad	Deg
Roll	-0.78	-45.03	1.04	59.6	0.13	7.8
Pitch	-0.93	-53.2	1.18	67.7	0.32	18.6
Yaw	-0.34	-19.5	0.25	14.6	0.09	5.6

TABLE I

RESULTS OF THE SIMULATION.

As a consequence of the wide dynamic range of roll and pitch angles linearization-based controllers can not be applied, thus the proposed technique could be compared only with advanced nonlinear controllers such as gain-scheduling or sliding modes. These controllers however, would require a much more complex analysis and control algorithm than algorithm 3.

Table I also shows a significant variation in yaw although the reference is always zero. This is due to the fact that the angles are deeply coupled and by applying only the first input the control signal is suboptimal.

Figure 4 presents the corresponding inputs to the attitude subsystem. With respect to the angular rate of change subsystem a simple P controller was implemented coupled with the aforementioned feedback linearization.

In figure 5 the response of the control system is depicted for a perturbation of $20°$ in the roll angle at time $t = 1.5$s. Note that although it is a significant perturbation the

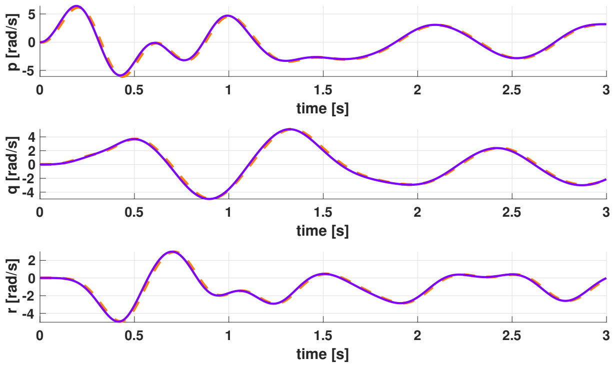

Fig. 4. p, q and r inputs

controller is able to overcome it. This is expected given that the system already operates in a wider range of angles.

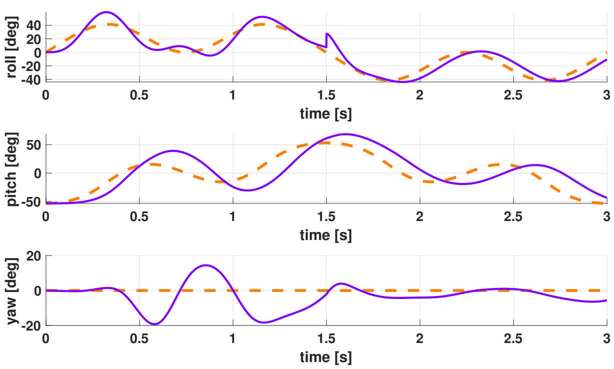

Fig. 5. Angular position results with perturbation of 20° in the roll angle at time $t = 1.5$s . The reference in the dotted orange line while the real states are in purple solid line.

V. Conclusions and Future Work

In this paper a new approach to Nonlinear Model Predictive Control for input-affine systems is presented. The control is based on a power series solution of Pontryagin's Maximum Principle differential equations and a modified gradient-descent method for the computation of the initial optimal costates and the solution of the Two Point Boundary Value Problem.

In section II the considered Optimal Control Problem is introduced, the solution using power series explained and the Two Point Boundary Value Problem is reformulated as an optimization problem. The proposed controller is presented and explained in section III, which ends in subsection III-B with three algorithms that summarize the implementation of the proposed controller to NMPC.

An application example is presented in section IV where the attitude control of a quadrotor is used to test the proposed algorithm. Subsection IV-A introduces the angle-based and quaternion-based attitude models and the control architecture explained in IV-B. Finally the simulation results are shown and analyzed in subsection IV-C.

In general, despite being a very simple algorithm it exhibits very good performance. Simulation results show that the controller performs well in a state space beyond linearization regimes. The proposed suboptimal control not only reduces the required on-line computations to a minimum but also the necessary off-line calculations. The independence of linearity, ease of implementation and low demand of computational power makes this controller an ideal option for fast nonlinear systems, such as the presented quadrotor.

Regarding future work it divides into two. In first instance future theoretical development. In terms of error bounds, prove of convergence and comparison with existing nonlinear optimal control as well as advanced control techniques. Further analysis of parametric uncertainty, external perturbations and noise. Secondly experimental validation of the proposed controller in any nonlinear plant and comparison with similar methods such as the continuation/GMRES method, *GRAMPC* and (SDRE) Control.

References

[1] L. Würth, R. Hannemann, and W. Marquardt, "Neighboring-extremal updates for nonlinear model-predictive control and dynamic real-time optimization," *Journal of Process Control*, vol. 19, no. 8, pp. 1277 – 1288, 2009. Special Section on Hybrid Systems: Modeling, Simulation and Optimization.

[2] J. Xu, T. Van Den Boom, L. Busoniu, and B. De Schutter, "Model predictive control for continuous piecewise affine systems using optimistic optimization," in *Proceedings of the American Control Conference*, vol. 2016-July, pp. 4482–4487, 2016.

[3] M. Diehl, H. Ferreau, and N. Haverbeke, "Efficient numerical methods for nonlinear mpc and moving horizon estimation," *Lecture Notes in Control and Information Sciences*, vol. 384, pp. 391–417, 2009. cited By 210.

[4] G. H. Negri, V. H. B. Preuss, M. S. M. Cavalca, and J. De Oliveira, "Differential evolution optimization applied in multivariate nonlinear model-based predictive control," in *2015 Latin-America Congress on Computational Intelligence, LA-CCI 2015*, 2016.

[5] Z. Yan and J. Wang, "Model predictive control of nonlinear systems with unmodeled dynamics based on feedforward and recurrent neural networks," *IEEE Transactions on Industrial Informatics*, vol. 8, no. 4, pp. 746–756, 2012.

[6] B. Kapernick and K. Graichen, "The gradient based nonlinear model predictive control software grampc," pp. 1170–1175, 2014. cited By 25.

[7] K. Graichen and B. Kapernick, "A real-time gradient method for nonlinear model predictive control," in *Frontiers of Model Predictive Control*, 2012.

[8] T. Çimen, "State-dependent riccati equation (sdre) control: A survey," vol. 17, 2008. cited By 202.

[9] T. Ohtsuka, "A continuation/gmres method for fast computation of nonlinear receding horizon control," *Automatica*, vol. 40, no. 4, pp. 563–574, 2004. cited By 400.

[10] Y. Zhang, S. Li, and L. Liao, "Near-optimal control of nonlinear dynamical systems: A brief survey," *Annual Reviews in Control*, 2019. cited By 0.

[11] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C*. Cambridge University Press, 1997.

[12] R. Holsapple and R. Venkataraman, "New, fast numerical method for solving two-point boundary-value problems," *Journal of guidance, control and dynamics*, vol. 27, no. 2, pp. 301–303, 2004.

[13] E. Fresk and G. Nikolakopoulos, "Full quaternion based attitude control for a quadrotor," in *2013 European Control Conference, ECC 2013*, pp. 3864–3869, 2013.