The Polynomial Approximation of the Explicit Solution of Model-Based Predictive Controller for Drive Applications

N. AL_Sheakh Ameen, B. S. Galal
Institute of electrical machines and drives
University of Wuppertal
Wuppertal, Germany
nael_nas@uni-wuppertal.de
bahaa_samy@hotmail.com

R. M. Kennel
Institute of electrical drives and power electronics
Technische Universitaet Muenchen
Munich, Germany
ralph.kennel@tum.de

R. S. Kanchan
ABB AB, Corporate Research
Vasteras, Sweden
rahul.kanchan@se.abb.com

Abstract—The complexity of implementing the Explicit solution of Model-based Predictive Control (Exp-MPC) in drive applications results in a higher number of generated regions. This increases the required memory space to store these regions and the coefficients of the associated optimal control laws, even when the solution is implemented using an efficient algorithm like Binary Search Tree (BST). The proposed method aims to replace the optimal linear/affine control laws, defined over several regions of the feasible state space, with one polynomial. The polynomial will be multivariate, with higher order, and defined over a combination of feasible regions. Using the polynomial approximation, the necessary memory space, to store region coordinates and the coefficients of optimal control laws, is reduced significantly. Although the accuracy of the optimal controller is reduced with use of polynomial approximation, it makes the implementation more realistic; provided the stability and the feasibility of the problem is still guaranteed.

Keywords—Predictive control, Explicit solution of MPC, Polynomial approximation, Electrical drives

I. INTRODUCTION

In last five years, a significant and increasing attention has been paid to use the explicit solution of model-based predictive control (Exp-MPC) in a wide range of electric drive applications [1]. A lot of literature was published to reduce the online computation complexity of the MPC to make it more feasible for this field of applications [5] [6] [7]. For example, the use of the binary search tree (BST) strategy [2] to reduce the evaluation time plays a significant role in that direction. Nevertheless, there are still some limitations like memory size and the computation capability of the control set-up, because of which the Exp-MPC has not been yet well accepted as an alternative to traditional controllers in drive applications [9]. This work proposes the polynomial approximation of the Exp-MPC as a new contribution to reduce the implementation complexity of Exp-MPC. Solving the MPC optimization problem using multi-parametric programming approach [1] divides the feasible state space of the control problem into a number of regions. Inside each region, there is one valid optimal control law, which is defined as an affine function of the state variables. The final solution is a look-up table of the region coordinates and the optimal control laws. The explicit solution of model-based predictive control becomes more complex when detailed motor models are considered while building Exp-MPC controller. This complexity is reflected in the number of generated regions. Increase in the number of regions requires more memory space to store the explicit solution of MPC as well as the evaluation time of the proper affine control laws. These drawbacks limit the implementation of the Exp-MPC in fast sampling rate applications [9]. The polynomial approximation intends to reformulate a combination of the optimal linear/affine control laws in several regions with only one multivariate polynomial of higher degree and covers those regions. This approximation will definitely reduce the accuracy of the optimal controller, but on the other hand, it will make the implementation of the Exp-MPC more simple and feasible provided that the stability and the feasibility of the problem are still guaranteed. Complexity reduction of the Exp-MPC using the polynomial approximation, without considering the cross-product terms\(^1\), was introduced in the literature in [13], [14], [15], and [16]. The authors in [13]-[16] use the sum-of-squares and the Polya’s theorem\(^2\) to find an approximate polynomial inside a stabilization set defined as Polytope. Building the stabilization set depends on the principle of existence of more than one Lyapunov function assure the stability of the optimization problem. On the other hand and because of the expensive symbolic computation of the extended Polya’s polynomial, this formulation is suitable for small problems. It works well for small problem size with guarantee for stability and feasibility of the problem.

The novelty of the proposed work is in introducing the multivariate polynomial approximation using the cross-product

\(^1\) The terminology “Cross products” is used in this paper to refer to multiplications between the polynomial variables of various powers.

\(^2\) Polya’s theorem [15]: If a homogenous Polynomial \((C,x)\) is positive for all possible values of the polynomial variables \(x\) belong to the \(N\)-dimensional polytope, all the polynomial coefficients \((C)\) are positive for a sufficiently large polynomial degree \(P_0\).
terms for drives applications. The simulation results carried on vector controlled motor drive case prove the feasibility and the stability of the machine problem with the approximated control laws.

II. POLYNOMIAL FORMULATION

Curve fitting methods are introduced in this work to approximate the affine control laws (1) defined over several regions with one approximate polynomial (2), (4) of higher degree and cover those regions.

\[ U^*_t = F^t x_t + G^t \]  

where \( U^*_t \) is the optimal control law at time instance \( t \), the matrices \( F^t, G^t \) contain coefficients of the affine control laws, associated with region \( i \), and \( x_t \) are the measured/estimated state variables, which define the region \( i \).

The used multivariate polynomial expression takes the following form [17] with two possible formulations (2), (4) as given below:

- Considering the cross-product terms:
  \[ \hat{U}(x) = \sum_{i=1}^{n_x} C_{i0} \prod_{j=1}^{n_x} x_j^i \]  

\[ \sum_{i=1}^{n_x} i_j \leq P_0 \]  

where \( \hat{U}(x) \) is the approximated control law, \( C \) are the polynomial coefficients, \( n_x \) is number of the polynomial variables, \( x_j \) is the degree of the polynomial variables, and \( P_0 \) is the polynomial degree. The multivariate polynomial in this expression can be introduced as a sum of terms, where each term is a multiplication of real coefficients with the polynomial expression can be introduced as a sum of terms, where each term is a multiplication of real coefficients with the polynomial variables of various degrees; e.g.:

\[ P(x_1, x_2, x_3) = C_{000} + C_{100} x_1 + C_{010} x_2 + C_{001} x_3 + \]  

\[ C_{011} x_1 x_2 + C_{101} x_1 x_2 + C_{110} x_1 x_3 + \ldots \]  

- Avoiding the cross-product terms [15]:
  \[ \hat{U}(x) = \sum_{i,j=1}^{n_x} [C_{ij}] x_j^i \]  

where \( \hat{U}(x) \) is the approximated control law, \( C \) are the polynomial coefficients, and \( P_0 \) is the polynomial degree. Avoiding the cross-product terms means avoiding the cross multiplication between the polynomial variables, and each term contains only one of the polynomial variables of different degree and multiplied with a real coefficient; e.g.:

\[ P(x_1, x_2) = C_{00} + C_{10} x_1 + C_{01} x_2 + C_{20} x_1^2 + C_{02} x_2^2 + \]  

\[ C_{11} x_1 x_2 + C_{12} x_1 x_2 + C_{21} x_1 x_2 + \ldots \]  

Having the cross products in the polynomial expression (as in (2)) increases the accuracy of the approximation and delivers much better results than without, but it will increase the total number of polynomial coefficients (\( N_i \)). More coefficients mean more memory space is required to store these coefficients. In some cases this number could exceed the number of coefficients using BST of the Exp-MPC. However a compromise should be settled.

1) Requirements and the pre-processing procedure

All steps introduced in [3], [8] to get the explicit solution of the MPC optimization problem are to be considered here before going further with the polynomial approximation. The preparation stage starts by generating the explicit solution from the MPC controller. The Exp-MPC represented as a structure should be calculated first and provided. To fulfill this requirement, the freely available Multi-Parametric Toolbox (MPT) from ETH-Zurich [11], [12] should be already installed with Matlab program from The MathworksTM. The Exp-MPC structure contains the coordinates of all regions and the associated control laws in its fields. The curve fitting procedure starts by selecting some fitting points of each region and evaluating the proper control law. Degree of computational complexity and the off-line preparation time depend basically on the number of fitting points and their locations in each region. For example, taking only the vertices and the Cheby-Ball center' of each region reduces the required time to extract these points significantly, but they will not be enough for a fine fitting. Whereas taking a fixed number of mesh points from each region will improve the fitting but on contrary will increase the complexity of the preparation procedure. As it is desired to obtain a good approximation of the optimal control laws over the regions, both vertices and mesh points are used in the proposed method in this paper. The combination procedure of the regions associated with one polynomial are not covered in this work, where it is mathematically not possible to combine the regions in a way to get a convex one, which assures the continuity of the resulting polynomial. As an elementary solution for this problem, it can be tried to divide the state space with respect to the main state variables before solving the optimization problem. An initial solution for the combination procedure associated with some remarks for future research in this direction is introduced in [10].

2) Least-square curve fitting

General, the goal of least-square curve fitting method is to keep a minimum distance (fitting error (E)) between the original curve and the approximated one. This problem is usually solved as a linear optimization problem aimed at finding the polynomial coefficients 'C' which minimize the approximation error E.

\[ \text{min } E = \text{min } C \sum_{n=1}^{n_p} [U_n(x) - \hat{U}_n(x)]^2 \]  

where \( C \) are the optimal polynomial coefficients, \( n_p \) is the number of the selected points for the fitting procedure in each region, \( n \) denotes the total number of the regions, resulted by the explicit solution of the MPC problem, \( U_n(x) \) are the calculated control inputs according to Exp-MPC, and \( \hat{U}(x) \) is the approximated control law.

Cheby-Ball center is the center of the largest ball inscribed inside the considered region.
3) Complexity analysis

This section introduces the complexity reduction of the online evaluation procedure for both the Exp-MPC using binary search tree (BST) method and the polynomial one:

1. The online evaluation procedure of Exp-MPC using BST requires storing different matrices:
   - **Branching Matrix (BM):** This matrix stores the coefficients of all linear inequality-equations to be evaluated later in the branching procedure over the tree-levels. Its size can be roughly approximated as [2], [4]:
     \[
     \text{size}(BM) = n_r \cdot (n_x + 3) + n_c \cdot n_u \cdot (n_x + 1)
     \]
     \[
     : \ n_c \approx n_r / 4 \tag{7}
     \]
     where \( n_r \) is the number of the Exp-MPC regions, \( n_c \) is number of the unique control laws, and \( n_x, n_u \) denote the number of state and the input variables, respectively.
   - **Control law matrices (F, G):** Two more matrices \((F, G)\) are to be stored, which contain the coefficients of the affine control laws. Their sizes can be given by the following equation:
     \[
     \text{size}(F) + \text{size}(G) = n_x \cdot n_u \cdot (n_x + 1) \tag{8}
     \]
     Thus the total number of coefficients in the Exp-MPC case using the BST is:
     \[
     U_L = \text{size}(BM) + \text{size}(F) + \text{size}(G) \tag{9}
     \]
     This number could be considered as upper limit \((U_L)\) for the polynomial strategy and should not be violated to say that the proposed strategy is effective enough and delivers a reduction in number of coefficients over the Exp-MPC solution.

2. On the other hand, the number of the coefficients \((N_c)\) to be stored in polynomial case is:
   - **Without cross-product terms:**
     \[
     N_c = n_p \cdot n_u \cdot (P_0 \cdot n_x + 1) \tag{10}
     \]
   - **With cross-product terms:**
     \[
     N_c = n_p \cdot n_u \cdot \left\{ 1 + \sum_{i=1}^{P_0} \frac{(i + (n_x - 1))!}{i!(n_x - 1)!} \right\} \tag{11}
     \]
     where \( n_u \) is the number of the control inputs, \( n_p \) is the number of the polynomials, \( P_0 \) is the polynomial’s order, \( n_r \) is the total number of regions, and \( n_c, n_u \) denote the number of state and the input variables, respectively.

For both cases (with or without cross products) the coefficients of \( n_p \) regions’ inequalities have to be stored as well if \( n_p < n_r \). Number of coefficients \((N_c)\) is increased proportionally with number of polynomials \((n_p)\) and number of control inputs \((n_u)\). To cover more regions with less number of polynomials, the polynomials’ orders have to be increased \((P_0 > 1)\) gradually. But an increase in the polynomial order \((P_0)\) will increase the evaluation time especially when cross-product terms are considered with different power multiplications, which makes the evaluation in some cases prohibitive. Hence a compromise should be found here between the above two approaches. The first approximation way without cross products was introduced in [15]. It easy to implement and delivers less number of coefficients to be stored, but it does not lead to a good approximation. Employing the cross products in the approximation delivers much better results, but on contrary, it will increase the number of coefficients as well as the complexity of the evaluation procedure. In this paper the polynomial expression (2) is used up to some degree, where the upper limit of Exp-MPC using the BST (9) was not violated.

B. Examples with different complexity

To demonstrate the concept of polynomial approximation, the simulation results are presented in this section consequently. It is applied to a simple example first and then it is extended to much complicated examples, having a large number of states and regions.

1) Example-1

The following equation represents a system model with two dynamics, one state and one input variable [15]:

\[
\begin{align*}
4 \leq x(k) + 2u(k), & \quad \text{if} \quad x > 0 \\
-6 \leq x(k) + u(k), & \quad \text{if} \quad x \leq 0
\end{align*}
\]

With the associated system constraints:

\[
x_k \in [-4, 4], u_k \in [-1, 1].
\]

The explicit solution of MPC for this problem with a linear norm and an infinite prediction horizon delivers five regions \(R_1, \ldots, R_5\) in one dimension. For each region, one optimal control law \((u)\) is valid; as shown in Fig.1.

![Figure 1. The controller regions \(R_i\), the associated control laws \(u\) the fitted polynomial curve \(P_f\), and the tested polynomial with more points \(P_t\) (The green curve).](image_url)
By considering the extreme points only of each region (vertexes) to do the curve fitting, the resulting curve will oscillate far away from the line segments and then come back to match at the extreme points. This means that the approximated curve will deliver good results just for the extreme points but not in between. Thus, an increase in number of fitting points in each region will improve the fitting results. In this example, 10 points (marked with ‘x’ in Fig.1) of each region were considered to do the fitting. However, it is advisable to test the resulting polynomial with much more test points, as the green-continuous line shows in Fig.1. For a univariate polynomial of 7th order, it is necessary to store eight coefficients instead of storing five regions plus ten coefficients for the Exp-MPC control laws \( \approx 33 \).

2) Example-2
To introduce to the multidimensional control structure of Exp-MPC, consider the following system model (13) with one dynamic, two state variables, and one control input.

\[
\begin{bmatrix}
    x_1 \\
    x_{2,1:1}
\end{bmatrix} = \begin{bmatrix}
    0.7326 & -0.0861 \\
    0.1722 & 0.9909
\end{bmatrix} \begin{bmatrix}
    x_1 \\
    x_{2,1:1}
\end{bmatrix} + \begin{bmatrix}
    0.0609 \\
    0.0064
\end{bmatrix} u_k
\]

(13)

System variables subject to the following constraints:

\[ x_k \in [-1.5, 1.5], u_k \in [-2, 2]. \]

The explicit solution of the MPC controller for this problem with a quadratic norm and 2 steps prediction horizon delivers five regions \( R_1...R_5 \) in two dimensions, Fig.2, where there is one valid optimal control law \( (u) \) in each region. Plotting the control input for two state variables in one figure, as in Fig.2, will not help to observe the fitting procedure; instead of that the evaluation procedure was done in Simulink. The evaluation results of both Exp-MPC and polynomial controllers are shown in Fig.3, where the controllers have to find the optimal control law \( (u) \) to steer the state signals \( x_1, x_2 \) to the origin. By assigning one bivariate polynomial for each region, and if the fitting procedure works correctly, then both methods should deliver exactly the same results. This means, both controllers deliver same control input and hence the state and control signals will match, respectively.

Figure 3 shows the states and the control input for both Exp-MPC and polynomial methods by using one bivariate polynomial for all the regions of 1st order; here the approximated control inputs are distinguished very clearly from the Exp-MPC.

3) Example-3
This is more complex and complete example to control the currents of an induction machine in synchronous coordinates, where the approximation method still works effectively. The induction machine currents are controlled using one MIMO-MPC current controller instead of 2 SISO-PI controllers in the Filed-Oriented Control scheme, Fig.4. The compensation procedure of the induced-voltages was done outside the MPC controller in a passive way as explained in [10]. To consider the delay compensation and the free-offset tracking of the machine currents [8], the original state variables \( (i_{sd}, i_{sq}) \) in (14) are to be augmented to include the control inputs \( (u_{sd}, u_{sq}) \) from previous time step and the reference signals \( (i_{sd}*, i_{sq}*) \). It results in 6 state variables (15).

![Figure 2. The regions with the associated control laws.](image)

![Figure 3. EXP-MPC and polynomial controllers’ response; 1st order Polynomial for all the regions.](image)

![Figure 4. Field oriented control structure using MPC controller.](image)
\[
\begin{bmatrix}
    i_{ad, k+1} \\
    i_{aq, k+1} \\
    u_{ad, k} \\
    u_{aq, k} \\
    i^*_a_{ad, k+1} \\
    i^*_a_{aq, k+1}
\end{bmatrix} =
\begin{bmatrix}
    1 & 0 \\
    0 & 1 \\
    T_o & 0 \\
    0 & T_o \\
    0 & 1 \\
    1 & 0
\end{bmatrix}\begin{bmatrix}
    i_{ad, k} \\
    i_{aq, k} \\
    u_{ad, k-1} \\
    u_{aq, k-1} \\
    i^*_a_{ad, k} \\
    i^*_a_{aq, k}
\end{bmatrix} +
\begin{bmatrix}
    \Delta u_{ad} \\
    \Delta u_{aq}
\end{bmatrix}
\]  

(14)

All machine quantities are normalized according to the normalization procedure in [10]. \(T_o\) is the normalized sampling time.

The explicit solution of MPC for this problem with a quadratic norm and 4 steps prediction horizon delivers a 6-dimensional control structure defined over 410 regions. Hence the coordinates of the six-dimensional 410 regions (the matrix \(BM\)) plus the coefficients of the associated control laws (the matrices \(F, G\)) are to be stored here.

\[
U_L = size(BM) + size(F) + size(G)
\]

(16)

This number has to be considered as an upper limit for the polynomial strategy and should not be violated to say that this strategy is effective enough and delivers a reduction in the Exp-MPC solution. The pre-processing procedure for this case is much complex than the former ones, where the fitting points are taken from a 6-dimensional mesh grid constructed for each region plus the extreme points of that region. In Fig.5 the machine currents and their references in synchronous reference frame are observed, by applying a nominal speed step.

Using only one multivariate polynomial (with cross products) of 3rd order and covers all the regions requires storing only 168 coefficients, instead of 10865 coefficients \((U_L)\) required for the Exp-MPC. Furthermore, with the approximate polynomial there is no need to store the coordinates of any region. The evaluation procedure of the resulting polynomial exhibits a steady-state error, which is not acceptable in practice. Figure 6 shows the machine currents and their references in synchronous reference frame, when the machine accelerates from zero to the nominal speed and under different load conditions. Noting that the tracking error stays constant for different speed levels even by keeping applying the nominal torque, Fig.6; therefore it could be compensated later at the output of the controller. In Fig.6 the machine was driven under different speed conditions with and without applying the nominal torque. The lower part of the figure shows the speed steps, while the upper part shows the machine currents \((i_{ad}, i_{aq})\) with their references. The resulting constant error was tuned and compensated in the simulation manually.

Figure 5. Induction machine currents response using polynomial controller of the 3rd order for each group of 41 regions; by applying a current step in \((i_{ad})\) with full magnetization of the machine.

Figure 6. Induction machine currents using polynomial controller of the 3rd order for all the 410 regions, in synchronous coordinates system.

Figure 7. Current step response in \((i_{ad})\) using EXP-MPC and the polynomial controller of the 3rd order for all the 410 regions.
In Fig.7 a fully current step was applied in the torque-produced current ($i_{sq} = 0 \rightarrow 1$), while keeping the machine fully magnetized. It shows the dynamic behavior of the machine currents using the polynomial controller against the MPC. Figure 7 proves that the polynomial approximation method of the Exp-MPC solution exhibits a “very good” and an “acceptable” dynamic response as compared to the original Exp-MPC; having the stability and feasibility of the problem are still assured.

III. CONCLUSIONS

The multivariate polynomial approximation method using the cross-product terms has been presented in this paper as an effective way to accelerate the implementation procedure of the explicit solution of model-based predictive control with a very close dynamic performance to the Exp-MPC. The number of the coefficients to be stored during the evaluation procedure and consequently the memory footprints is reduced significantly. A comparison with binary search tree algorithm (BST) was also introduced. BST could be considered as fast and simplest method of implementing the Exp-MPC. This work reformulates the explicit solution of model-based predictive control using least-square curve fitting strategy, and does include neither experimental results nor the combination procedure of the regions which are covered with the resulting polynomials. The combination procedure could be found in [10], with some suggestions and improvements on the presented procedure.

ACKNOWLEDGMENT

The authors are pleased to acknowledge the financial support of the Syrian Ministry of Higher Education.

REFERENCES


