یازدهمین کنگره ملی سراسری
فناوریهای نوین در حوزه توسعه پایدار ایران
11th National Congress of
the New Technologies in Sustainable Development of Iran
senaconf.ir

Design and Analysis of Explicit Predictive Model Control to Reduce the Number of Convex Areas based on Parametric Optimization using Multi-Parameter Toolbox

Sara Mahmoudi Rashid, Ph.D. student in Electrical Engineering Department
University of Tabriz, Tabriz, Iran
s.mahmoudirashid@tabrizu.ac.ir

Abstract—**In this paper, one of the challenges of predicting control based on the model is the implementation of explicit predicting model control (EMPC) method in real time that EMPC provides an explicit and real time algorithm to predicting designed for implement explicit processes. When the predictive control is designed, the resulting cost function is minimized. This cost function is a quadratic programming (QP) problem that has several forms. The general structure is that the cost function should be solved in relation to the optimization variable which is the control input and minimize the optimal control signal of the cost function which is conditional on the initial state and system model. In this case, a convergent and optimal answer is obtained, in which case the control input implicitly depends on the mode, i.e. the effect of the mode on the control input is undeniable, but it can't be explicitly stated. The greater the number of system constraints, the longer the optimization problem will be solved, and if it varies over time, the execution speed may be high or low. Therefore, numerical solution can't be a real time method, i.e. it varies with the time of the solution, this optimization problem is difficult to converge and optimize the answer. As a result, the complexity of the problem and the slowness of the solution are two problems that are encountered in online problem solving. Especially to apply to explicit processes where control signals must be updated at all times. These two factors gave rise to the EMPC, the practice of having system responses calculated offline in advance. According to the independent variables that exist, a space is created that EMPC converts this space into convex regions and solves a Multi Parametric Toolbox (MPT) programming problem for each region and takes controllers and reduces the number of convex areas.**

Keywords: Explicit Model Predictive Controller, Parametric Optimization, Convex Areas, Multi-Parameter Toolbox.

1. Main text

One of the challenges that has arisen since the beginning of predictive control was that the optimization algorithm, which is applied as a method of recurring horizon control (Receding Horizon Control), must be solved by the optimization algorithm. Due to the time they had very slow computers as well as the optimization algorithms that existed and due to the systematic limitations, the design and implementation of predictive control took about 10^3 seconds. It was necessary to implement the control signal. Over time, with a gentle slope and with the development of computers and the development of methods to optimize computing time, the processing power of computers has decreased [1]. In recent years, there has been a leap that has led to a predictive control algorithm for fast processes such as car and engine control that need to quickly update control signals for a problem (Receding Horizon Control) that needs to be addressed. They were solved to be implemented very quickly, i.e. the time to apply and calculate the control signal was close to 1 millisecond. Therefore, fast processors did not require large computers and supercomputers to control a system, and the predictive control algorithm was implemented with only one processor having high processing power [2]. At this time, the EMPC algorithm became popular and was introduced by Professor Bemporad with the aim of speeding up the implementation of the predictive control algorithm [3]. Then a method called Sub-Optimal Explicit MPC was introduced, which is a combination of Online Optimization EMPC and Offline Optimization EMPC, which reduced the processing speed and updated the control signal to one nanosecond, so it can be said that almost control Predictors can be implemented on any fast process. In this case, conventional microcontrollers and microprocessors can be used to implement the predictive control algorithm [4].

The purpose of this method is to control linear processes, so the control of the Matlab control toolbox developed for EMPC by Professor Bemporad has been considered. The explicit linear quadratic regulator for constrained systems, which is one of the most referenced articles, has provided the horizon for working on offline predictive control algorithms and the basic principles of predictive control. This article describes [5]. Reference to the article on how the MPC problem becomes Multi parametric Quadratic Programming, how Polyhedrals are formed. Polyhedrals are the same areas of convex polyhedral. A toolbox called MPT has also been developed that converts optimization problems to Multi parametric and solves the problem [6].

Real-time control is one of the cases in which the MPC controller requires time, in which system states are usually not available, and the estimation of system states is considered, that is, using system outputs and other system model values. Attempts are made to estimate system states. According to the estimated states and linear values and other system variables, the general state of the system is defined, which can include the general state of the system, process states, perturbation and noise model. The next step, which is the basic step for the values of the states and the predictive control is designed based on this, is in which area the identified system is [7].

Compared to existing studies, this is the main contribution of this article. In the second part, the prediction control design is expressed by multi-parameter programming method. In the third step, a multi-parameter method to solve the quadratic optimization problem is obtained. The fourth section will review the EMPC signal design method. In the fifth part, and in fact the simulation of the article, the simulation results are presented. Finally, in the sixth and final part of the article, the general results of the article are stated.

2. Design of predictive control by multi-parameter programming method

When the predictive control is designed, the resulting cost function is minimized. This cost function is a quadratic programming (QP) problem that has various forms. The general structure is that the cost function should be solved relative to the optimization variable which is the control input and minimize the optimal control signal of the cost function which is conditional on the initial state and system model which we assume is the state space and constraint. The system is based on modes and outputs and operating limits (R, Q), according to which the optimization problem is solved and the control signal is obtained. The first member of this control signal is applied to the process, which results in receding horizon control and process outputs [8]. This description is shown diagrammatically in Figure 1.

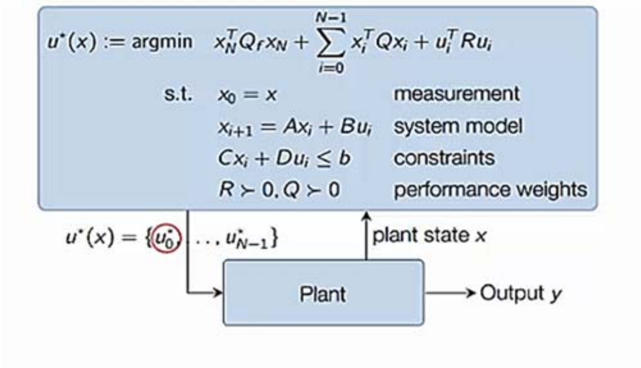

Figure 1: Block diagram of the optimization problem

If the states are visible, i.e. they can be measured, they can be re-entered into the optimization system. This loop is repeated, but if the output is measured, the gain is necessary to estimate these states and the estimated states enter the closed loop.

3. Multi-parameter method to solve the quadratic optimization problem

If the relationship between states and inputs is generally considered to be $u = Fx + G$, F and G are fixed matrices with respect to these matrices each time the states are updated. The control signal can be obtained and applied to the process. If the QP problem does not involve system constraints, and with the availability of the system model, the explicit response is known by the input relationship and the F and G states, but when there are constraints on the input and states, then the Obtained the explicit form of control input. In this case, this optimization problem must be solved from iterative and optimization algorithms for which there are different methods. In this case, the convergent and optimal answer is obtained, in which the control input implicitly depends on the mode, i.e. the effect of the mode on the control input is undeniable, but can not be explicitly stated. The greater the number of system constraints, the longer the optimization problem will be solved, and if it varies over time, the execution speed may be high or low. Therefore, numerical solution can not be a real time method, i.e. it varies with the time of the solution, this optimization problem is difficult to converge and optimize the answer. When there is no limit, F and G are already obtained and applied online. In this case, the optimization problem is not solved, but when the iterative problem is solved to converge the answer. As a result, the complexity of the problem and the slowness of the solution are two problems that are encountered in online problem solving. Especially to apply to explicit processes where control signals must be updated at all times. These two factors led to the introduction of EMPC, the practice of having system responses calculated in advance offline.

Now the optimization problem becomes Multi Parametric Programming (MPT) and in this case it is possible to obtain control inputs offline for different areas $u = F_i x + G_i, (i = 1, 2, \ldots n_v)$ that have limitations which n_v is the number of areas. These controllers explicitly depend on the state of the system, i.e. first the workspace is determined to be the parameters of the system and according to the existing restrictions are divided into different areas, each of which Of these areas F_i and G_i are convex regions, which is actually the MPT method. In each of these areas, one of these conditions is active and the others are inactive, which for each of these areas is obtained by solving the MPT problem. Once the states are found, it is determined which area they belong to and the control input is obtained. Control input is applied to the process and a new state is obtained or estimated. The relevant area is detected again and the control input is obtained. This description is shown diagrammatically in Figure ۲.

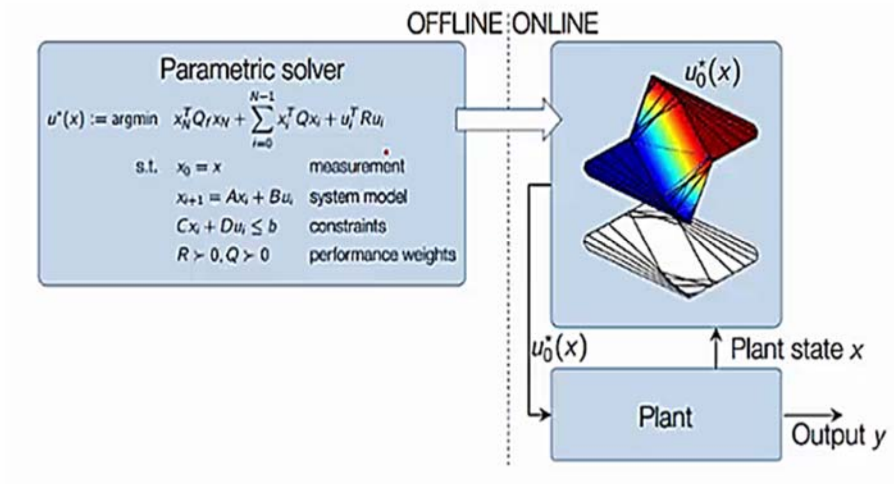

Figure 2: Offline and online optimization problem method

In fact, EMPC transforms online optimization calculations into offline optimization calculations, causing the computational volume to be significantly reduced. The advantage of this method is that it can be implemented for fast Real Time processes. The higher the number of parameters and the number of constraints, the greater the number of areas that must be specified according to the parameters. So this is the main part that takes time because there may be a large number of states and areas and as a result a lot of time is spent to determine which area it is in. It is then searched according to the area that has been identified. In the final part, according to the states and obtained, the control signal is calculated.

Given the above explanations, it seems that the EMPC itself may not be Real Time, because when the number of zones increases, many zones must be examined to determine whether this condition is in this zone or not. It then searches and calculates the control signal, all of which are time-consuming. So the first problem is the search time, which depends on the number of areas, and the second is the memory problem. When the number of areas and parameters is large, in order to be able to cover the QP problem with the desired constraints, the number of areas increases, so when the number of constraints and parameters is high, the QP problem is multiplied by the function. EMPC is resolved. In this case, the number of areas should be increased, which will increase both the search time and the memory required to store the information areas. As a result, EMPC itself faces the challenge of running in a short time, in which case various methods have been proposed to solve this problem [9] and in most cases MPC classic online optimization itself gives a better answer or recently Sub Optimal Explicit MPC has been introduced that combines EMPC Online Optimization and EMPC Offline Optimization to be able to increase speeds of up to 1 nanosecond [10].

4. EMPC signal design method

The first step in designing an EMPC controller in the Matlab Predictor Control Toolbox is to design a common predictor controller according to the process model, predictor horizon, and predictor control design parameters. Be. Then, in order to turn this predictive controller offline, you must specify system modes and independent variables, which can include process modes themselves, confusion modes, inputs, and so on. Turbulence outputs and signals are measurable. In the next step, the boundary of these states must be determined, which is in fact one of the challenges of MPC. That is, according to the independent variables that exist, a space is created that EMPC converts this space into convex regions and solves a Multi Parametric Programming problem for each region and the controllers to Achieves. If the range of independent variables in MPC design and simulation is different, then if during implementation the independent variables and variables are in the region where the difference is made, because these values are defined for EMPC the controller does not cause an error because an overrun is allowed.

The challenge here is that because many system parameters and states are not exactly the physical parameters of the system and there is no information about the exact change in their value and range. Now if this large boundary is considered, the number of zones increases and the EMPC problem becomes more complex. Therefore, proper border estimation, which is important, reduces space and improves MPC performance. Once the borders are identified, Matlab software designs the EMPC controller, that is, it identifies an area that, $|H_i x| < K_i$ that is, it actually determines which area of the system it is in and what controller there is for that area $u = F_i x + G_i$

Which is a piecewise affine controller that has a linear part and an affine part. This explicit predictor controller results in 4 parameters that $H_i\ K_i\ F_i\ G_i$

Creates areas and creates a controller according to the conditions specified for the system boundaries and despite the upper and lower boundaries $u_L < u < u_U$ defined using the generate Explicit MPC command. Recommendations in EMPC design have been made by Matlab software so that the number of areas is not too much. One of these recommendations is that if there is an entry restriction, then a small control horizon should be considered, which means that the number of areas should not be too much. The second recommendation in this predictive control toolbox is not to take into account the output constraints in the control design and to try to output the output by adjusting the predictive control parameters and weighting the cost function and determining the predictive horizon. Control horizon to be done. However, if the output limit is taken into account, there are methods that can reduce the number of areas and prevent them from increasing too much. The EMPC design process was then determined by Matlab software.

The main challenge is the complexity and large number of EMPC areas. In many simple cases, they are present up to the area that covers the QP problem, which if the number of areas is too large, the implementation problem will be difficult. For this reason, EMPC has provided a series of tools in the Matlab Toolbox control [11] that can be used to reduce the number of areas. The first tool is to be able to view images of these areas with the plotsector command. In many systems, the number of independent parameters is very large and it is not possible to draw all the multidimensional polyhedral space. The second thing that can be done is to reduce the number of areas. That is, areas that are next to each other and have the same guinea, and especially and have the same, if the union of these two areas can be a convex part, the community of these areas can be obtained together and the work of simplification can be done. And reduce the number of areas, which Matlab software itself does with the simplify command. The third task is to eliminate small areas and merge the problem of this smaller area with the large area to avoid increasing computations and complexity, but in this case the optimization problem will no longer be an optimization problem, and It will actually be a Sub-Optimal, that is, it will be imaged by the weights and gains of larger areas.

The last step is the implementation step, which explains the implementation process. To implement EMPC, if the system output is measured, the system states must be estimated, ie the system states must be specified, and the values of the general system states, including the process state itself, the perturbation state, the noise state, and other independent variables. Is also determined and finally the state vector is formed. The second step is to answer the question of whether the state vector is in the functional range. There is an upper bound and a lower bound for each of the states, and it must now be checked whether the state vector is also in this region. If it is not in this area, the EMPC will make an error and set the instantaneous control signal applied to the process equal to the instantaneous control signal. That is, the control signal of the previous moment extends to the control signal of this moment, and the sampling time of the system increases somewhat. It is now assumed that this input is applied to the system and the resulting system states return to the permissible area, otherwise it will run into problems. In the third step, it is placed and it is checked whether it is in the area, in which case it is obtained by using and control signal and is applied to the process. In the next neighborhood, if it is not in the first area, we put it in this state, and we calculate and name it, and we put it in the minimum state, which is the amount of violation of the area, that is, this How transgressive is the state we are in? If this is smaller than the other violations, then we consider it equal. In the fifth step, one unit is incremented and it is checked again if it is in my area and then we get and and apply it to the process and this loop is repeated again and the areas They are examined and as long as they are not in these areas, the degree of violation of the states from that area is calculated using that which is for each area. If this amount of violation is less than the minimum that already exists, in this case it is considered equal. This is repeated until all areas are examined. In some cases, the answer may be immediate and sometimes it may not be in any area. If this area is not in any case, in this case, and is considered and the control signal is calculated. In this case, due to numerical problems and simplification that may arise, in this case, the closest area is selected. In this case, the problem becomes a sub-optimal sub-optimal problem and this control input is applied to the process.

5. Simulation

In this section, to demonstrate the accuracy of the proposed EMPC controller, the proposed cruise control system is as shown in Figure 3 [12].

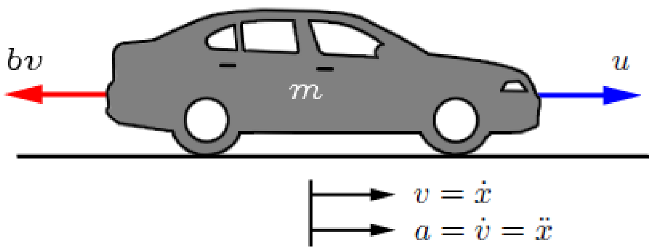

Figure 3: Cruise Control Physical setup

Automatic cruise control is an excellent example of a feedback control system found in many modern vehicles. The purpose of the cruise control system is to maintain a constant vehicle speed despite external disturbances, such as changes in wind or road grade. This is accomplished by measuring the vehicle speed, comparing it to the desired or reference speed, and automatically adjusting the throttle according to a control law.
We consider here a simple model of the vehicle dynamics, shown in the free-body diagram (FBD) above. The vehicle, of mass m, is acted on by a control force, u. The force u represents the force generated at the road/tire interface. For this simplified model we will assume that we can control this force directly and will neglect the dynamics of the powertrain, tires, etc., that go into generating the force. The resistive forces, bv, due to rolling resistance and wind drag, are assumed to vary linearly with the vehicle velocity, v, and act in the direction opposite the vehicle's motion. With these assumptions we are left with a first-order mass-damper system. Summing forces in the x-direction and applying Newton's 2nd law, we arrive at the following system equation $m\dot{v}+bv=u$ and y=v
Since we are interested in controlling the speed of the vehicle, the output equation is chosen as follows
For this example, let's assume that the parameters of the system are:
vehicle mass 1000 kg
damping coefficient 50 N.s/m

First-order systems have only a single energy storage mode, in this case the kinetic energy of the car, and therefore only one state variable is needed, the velocity. The state-space representation is therefore $\dot{x}=[\dot{v}]$ so
$[\dot{v}]=[(-b)/m][v]+[1/m][u]$ and y=[1][v].

Sampling time is 0.1 and forecast horizon is 10 and control horizon is 3. The number of polyhedrons is reduced to 19 convex numbers and the number of parameters is 4. Kalman's interest rate estimate. Due to the fact that the system has two variables, it is possible to graphically display the proportional areas of the predictive control problem in this particular case. There is using the result of the paper for the discrete time model of the system in the presence of constraints, the number of areas obtained for the control law, after simplifying and unifying the areas with the same law, the number of areas reaches 19, which is shown in Figure 4. In this form, the area in the middle, which is larger than the other areas, is the critical area associated with a state in which there are no active constraints.

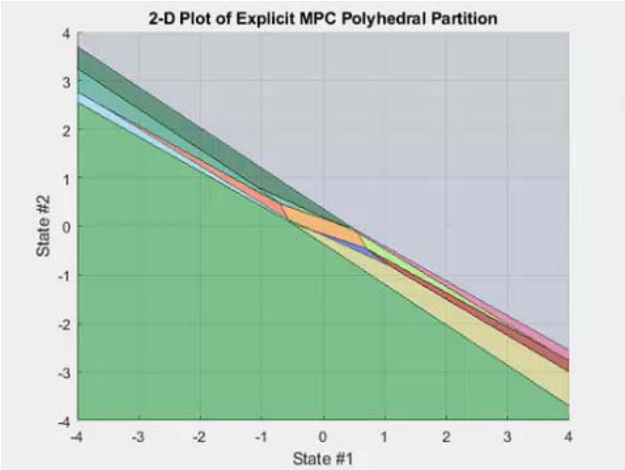

Figure 4: Segmentation of the boiler system state space by multi-parameter programming method

Given that the simulation time of the control system performance is mainly related to the execution of the control algorithm, to compare the execution time of the control algorithms, it is sufficient to compare the total execution time of the simulation for different controllers. The simulation is performed in MATLAB software environment in the form of code-m and it can be assumed that the execution time in other software environments is proportional to the times obtained in MATLAB environment. Because MATLAB software uses an internal function (quadprog which is not in code-m) to achieve higher speeds for executing square programming. Source used as code-m is also checked so that the control algorithm is based only on code-m. The considered open-source function is a function known as minq.

In MPT-MPC implementation method, there is no need to perform optimization (square programming) during the execution of the control algorithm. In this method, it must first be determined by checking an unequal number in which area of the state space. Then it is enough to use the feedback gain corresponding to the relevant area to calculate the control signal. Therefore, it is clear that the implementation of MPT-MPC is much faster than conventional predictive control. To simulate the performance of the MPT-MPC control, the MPT toolbox functions in the MATLAB environment have been used to design the MPC-MPC controller. Conventional predictive control and MPT-MPC differ only in the method of calculation and execution time and have the same performance, so there is no need to compare the performance of the two controllers. It is observed that the execution time of the predictor control

MPT-MPC is very fast. But conventional predictive control is much slower than MPT-MPC due to the need for heavy constraints on constrained optimization at runtime. Figure 5 shows the control signal and output changes in the script environment and Figure 6 and 7 in the Simulink environment.

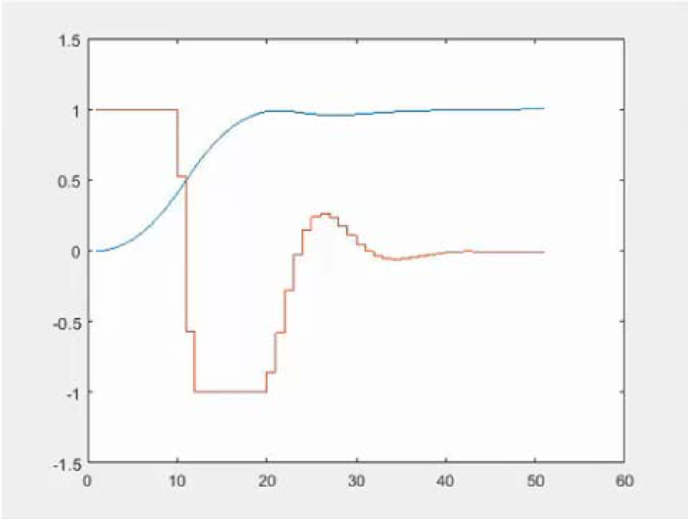

Figure 5: Changes in the control signal and output in the script environment

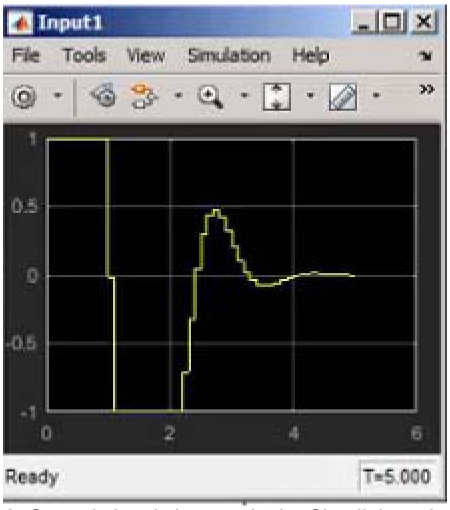

Figure 6: Control signal changes in the Simulink environment

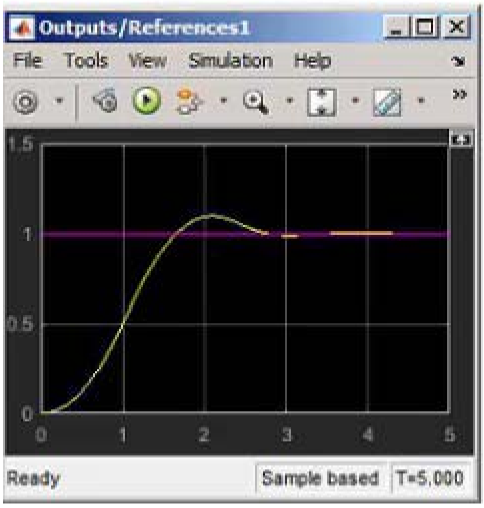

Figure 7: Control output in Simulink environment

6. Conclusion

In this paper, the controller designed by predictive control method based on multi-parameter optimization has a better performance and speed than conventional predictive control. In addition, it is possible to use this controller to meet the constraints of the system, which is one of the strengths of this controller. As observed in the simulations, the computational volume for the implementation of this controller is much smaller than the conventional constrained predictive control and comparable to the execution time of other controllers. Therefore, this control method is a very good option for implementing predictive control in practice. The complexity of the problem and the slowness of solving the problem that they face in online problem solving. Especially to apply to explicit processes where control signals must be updated at all times. According to the independent variables, the space has been converted to convex areas by the EMPC controller and a Multi Parametric Programming problem has been solved for each area. As a result, controllers are obtained and the number of convex areas is reduced.

References

[۱] J. Drgoňa *et al.*, "All you need to know about model predictive control for buildings," 2020.

[۲] A. Krishnan Prakash *et al.*, "Solar+ Optimizer: A Model Predictive Control Optimization Platform for Grid Responsive Building Microgrids," vol. 13, no. 12 ,p. 3093, 2020.

[۳] A. Bemporad, J. Baillieul, and T. Samad, "Explicit Model Predictive Control," ed, 2015.

[۴] A. Gersnoviez, M. Brox, and I. Baturone, "Hierarchical Fuzzy Controllers for Explicit MPC Control Laws: Adaptive Cruise Control Example," in *2 ۰۲۰ IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2020: IEEE, pp. 1-7 .

[۵] J. R. Salvador, D. M. de la Peña, T. Alamo, and A. J. I.-P. Bemporad, "Data-based predictive control via direct weight optimization," vol. 51, no. 20, pp. 356-361, 2.۰۱۸

[۶] D. Arnström, A. Bemporad, and D. J. I. C. S. L. Axehill, "A Linear Programming Method Based on Proximal-Point Iterations with Applications to Multi-Parametric Programming," 2021.

[۷] Y. Xie, W. Guo, Q. Wu, K. J. I. J. o. E. P. Wang, and E. Systems, "Robust MPC-based bidding strategy for wind storage systems in real-time energy and regulation markets," vol. 124, p. 106361, 2021.

[۸] S. Coskun, C. Huang, F. J. T. o. t. I. o. M. Zhang, and Control, "Quadratic programming-based cooperative adaptive cruise control under uncertainty via receding horizon strategy," p. 0142331221992741, 2021.

[۹] Y. Li, G. Qu, and N. Li, "Using predictions in online optimization with switching costs: A fast algorithm and a fundamental limit," in *2018 Annual American Control Conference (ACC)*, 2018: IEEE, pp. 3008-3013 .

[۱۰] M. J. a. p. a. Molinaro, "Curvature of feasible sets in offline and online optimization," 2020.

[۱۱] F. Almeida, "Management of Electrical Power Generation on Gas Turbine with Model Predictive Control," in *AIAA SCITECH 2022 Forum*, 2022, p. 0444 .

[۱۲] H. M. Y. Naeem, A. I. Bhatti, and A. Mahmood, "Longitudinal Cruise Control of a Car using Sliding Mode Approach," in *2019 International Conference on Electrical, Communication, and Computer Engineering (ICECCE)*, 2019: IEEE, pp. 1-5 .