

Article

Fast Model Predictive Control of PEM Fuel Cell System Using the L_1 Norm

Robert Nebeluk *  and Maciej Ławryńczuk 

Institute of Control and Computation Engineering, Faculty of Electronics and Information Technology, Warsaw University of Technology, ul. Nowowiejska 15/19, 00-665 Warsaw, Poland; maciej.lawrynczuk@pw.edu.pl
* Correspondence: robert.nebeluk@pw.edu.pl

Abstract: This work describes the development of a fast Model Predictive Control (MPC) algorithm for a Proton Exchange Membrane (PEM) fuel cell. The MPC cost-function used considers the sum of absolute values of predicted control errors (the L_1 norm). Unlike previous approaches to nonlinear MPC- L_1 , in which quite complicated neural approximators have been used, two analytical approximators of the absolute value function are utilised. An advanced trajectory linearisation is performed on-line. As a result, an easy-to-solve quadratic optimisation task is derived. All implementation details of the discussed algorithm are detailed for two considered approximators. Furthermore, the algorithm is thoroughly compared with the classical MPC- L_2 method in which the sum of squared predicted control errors is minimised. A multi-criteria control quality assessment is performed as the MPC- L_1 and MPC- L_2 algorithms are compared using four control quality indicators. It is shown that the presented MPC- L_1 scheme gives better results for the PEM.

Keywords: proton exchange membrane fuel cell; model predictive control; optimisation; L_1 cost function



Citation: Nebeluk, R.; Ławryńczuk, M. Fast Model Predictive Control of PEM Fuel Cell System Using the L_1 Cost-Function. *Energies* **2022**, *15*, 5157. <https://doi.org/10.3390/en15145157>

Academic Editors: Pierpaolo Polverino and Cesare Pianese

Received: 13 June 2022

Accepted: 14 July 2022

Published: 15 July 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Proton Exchange Membrane (PEM) fuel cells were first developed by General Electric in the United States in the 1960s and used in a spacecraft developed by the National Aeronautics and Space Administration (NASA) in the Gemini program [1]. They are a type of acid fuel cell in which a solid or quasi-solid “membrane” material is used instead of the proton-conducting liquid electrolyte necessary in earlier acid cells. They can be used in cars, scooters, bicycles, boats, underwater vessels [2,3] and also aircrafts [4]. These fuel cells have several advantages. One of them is low operation temperature which gives a fast start-up. Furthermore, a solid electrolyte is used so no electrolyte leakage can happen. PEM fuel cell design is also very simple and compact. Finally, they are very reliable in operation. On the other hand, there are minor drawbacks, such as the formation of a significant amount of water that may not have time to evaporate from the cell and thus damage it and the platinum catalyst's high susceptibility to carbon monoxide. However, all of the positive features of the PEM fuel cells make them great alternatives to conventional power generation methods that rely on fossil fuels. Technological details related to different aspects of PEM fuel cells can be found in numerous publications, e.g., [5–7].

Since PEM fuel cells are nonlinear processes, their efficient control is quite difficult. Different control strategies are possible for optimising the power split between the battery and PEM fuel cell to maximise system efficiency and reduce fuel consumption [8]. Simple controllers, including a Sliding-Mode Controller (SMC) and a fuzzy logic controller, can supply a convenient voltage [9]. A more advanced approach is to use a fuzzy Proportional-Integral-Derivative (PID) controller [10] for temperature control of the fuel cell cooling system or an adaptive SMC algorithm which is integrated with the radial basis function neural network to control the current or voltage of the fuel cell power supply [11]. Recent works show that PEM output power quality can be improved using classical and fractional-order PID controllers [12]. An example of a fuzzy control method to control the

temperature and humidity of the stack has been presented in [13]. It is also possible to use a fuzzy adaptive PI controller with an improved advanced genetic algorithm [14]. Hierarchical techniques can also be applied to develop a comprehensive energy management strategy consisting of high-level control, which is based on fuzzy logic technique as well as adaptive control loop and low-level control that generates a pulse-width-modulation signal so that appropriate power distribution is achieved and the operating voltage of the powertrain is stabilised [15]. In recent years, applications of MPC algorithms [16–18] to PEM fuel cells have been reported. A linear model can be used for MPC prediction [19]. However, typically, a nonlinear model is used, meaning that a nonlinear optimisation task must be solved at each sampling instant on-line [20–23]. A practical method leading to simplification of the nonlinear task is successive on-line linearisation which leads to a quadratic optimisation problem [24,25].

In all MPC algorithms cited above, the MPC cost-function used considers the sum of absolute values of the predicted control errors (the L_2 norm). However, in a few publications concerned with other applications, it is suggested that better control is possible when the sum of absolute predicted control errors is utilised as the MPC cost function (the L_1 norm) [26–30]. This paper aims to present a computationally simple MPC- L_1 control strategy and its application to the PEM process. This work extends previous research concerned with computationally efficient nonlinear MPC- L_1 in which a neural approximator of the absolute value function is utilised [30]. Unfortunately, such a control strategy's effectiveness heavily depends on the approximator's accuracy. A more straightforward approach is recommended in this work. The neural network is replaced by two analytical approximators that are defined explicitly. The predicted process trajectory is linearised on-line to find a quadratic optimisation task. All implementation details of the discussed algorithm are detailed. Furthermore, the algorithm is thoroughly compared with the classical MPC- L_2 approach. A multi-criteria control quality assessment is performed as the MPC- L_1 and MPC- L_2 algorithms are compared using four control quality indicators. It is shown that the presented MPC- L_1 scheme gives better results for the PEM process.

The paper is organised as follows. Section 2 presents the continuous-time PEM fuel system model used as the simulated process and a description of the Wiener model that is utilised in all MPC algorithms is next presented. Section 3 recalls the general MPC optimisation task with both L_2 and L_1 norms. The main contribution is detailed in Section 4 where a computationally efficient MPC- L_1 algorithm with two analytical approximators is described. The simulation results are presented and discussed in Section 5. Finally, the research findings are summarised in Section 6.

2. Proton Exchange Membrane Fuel Cell

Fuel cells are devices that allow electricity production, typically through the chemical conversion of hydrogen and oxygen. Fuel (hydrogen) is supplied continuously to the anode; oxygen is supplied to the cathode. An electrolyte differs depending on the fuel cell type. Energy and heat are generated during the electrochemical reaction; water is the only byproduct. In cars, the entire process begins with the supply of hydrogen from a high-pressure tank to the cell. In parallel, compressed air is also supplied. The result of the reaction in the cell is a generated current, which is converted into alternating current and supplied to the electric motor responsible for traction.

In this work, a PEM fuel cell is considered, which has the polymer membrane as the electrolyte that conducts protons. A catalyst covers the anode, facilitating the splitting of hydrogen atoms into protons and electrons. Ions pass through a polymer membrane conducting protons, which is an isolator for the electrons. Electrons flow through an external circuit, creating an external current in the cell. These fuel cells have been incredibly popular lately as alternative power sources. It is beneficial to use them due to their low operating temperature, great efficiency and minimal emission of harmful greenhouse gases.

This section describes a fundamental model of the PEM process. It is used in simulations only as the process. Next, a neural Wiener model is detailed. It is used in MPC algorithms only.

2.1. Description of the PEM Fuel Cell System

The PEM fuel cell model considered in this paper was first introduced in [31] and further discussed in [32–34]. The manipulated variable is the methane flow rate denoted as q (mol s^{-1}) and the output is the stack output voltage denoted as V (V). The system is influenced by one measured disturbance which is the external current load, denoted as I (A). The partial pressures of hydrogen, oxygen, and water are represented by the variables p_{H_2} , p_{O_2} and $p_{\text{H}_2\text{O}}$, respectively (atm). The input hydrogen flow, hydrogen reacted flow, and oxygen input flow are denoted by $q_{\text{H}_2}^{\text{in}}$, $q_{\text{H}_2}^{\text{r}}$ and $q_{\text{O}_2}^{\text{in}}$, respectively (mol s^{-1}).

2.2. PEM Fuel Cell Continuous-Time Fundamental Model

The fundamental continuous-time model of the PEM system is described by the following continuous-time equations

$$q_{\text{H}_2}^{\text{in}} = \frac{CV}{(\tau_1 s + 1)(\tau_2 s + 1)} q \quad (1)$$

$$p_{\text{H}_2} = \frac{1/K_{\text{H}_2}}{\tau_{\text{H}_2} s + 1} \left(\frac{CV}{(\tau_1 s + 1)(\tau_2 s + 1)} q - 2K_{\text{r}} \right) \quad (2)$$

$$p_{\text{O}_2} = \frac{1/K_{\text{O}_2}}{\tau_{\text{O}_2} s + 1} \left(\frac{CV/\tau_{\text{H-O}}}{(\tau_1 s + 1)(\tau_2 s + 1)} q - K_{\text{r}} I_{\text{r}} \right) \quad (3)$$

$$q_{\text{H}_2}^{\text{r}} = 2K_{\text{r}} I_{\text{r}} \quad (4)$$

$$p_{\text{H}_2\text{O}} = \frac{2K_{\text{r}}/K_{\text{H}_2\text{O}}}{\tau_{\text{H}_2\text{O}} s + 1} I \quad (5)$$

$$V = E - \eta_{\text{act}} - \eta_{\text{ohmic}} \quad (6)$$

$$E = N_0 \left[E_0 + \frac{RT}{2F} \ln \frac{p_{\text{H}_2} \sqrt{p_{\text{O}_2}}}{p_{\text{H}_2\text{O}}} \right] \quad (7)$$

$$\eta_{\text{act}} = B \log(CI) \quad (8)$$

$$\eta_{\text{ohmic}} = R^{\text{int}} I \quad (9)$$

Equation (1) represents the hydrogen flow which can be obtained from the reformer, where q is the methane flow rate while CV , τ_1 and τ_2 are constants. The pressure of hydrogen is calculated from Equation (2), where K_{H_2} and τ_{H_2} denote the valve molar constant for hydrogen and the response time of hydrogen flow, respectively. The current pressure of oxygen is found using Equation (3), where K_{O_2} and τ_{O_2} denote the valve molar constant for oxygen and the response time of oxygen flow, respectively. Equations (4) and (5) describe the hydrogen flow that reacts and the pressure of water, respectively, where K_{r} is a constant. The stack voltage is found from Equation (6), where E is calculated from the Nernst's formula in Equation (7). Additionally, the following symbols are used: N_0 is the number of cells in series in the stack, E_0 is the ideal standard potential, R_0 is the universal gas constant, T_0 is the absolute temperature and F_0 is the Faraday's constant. Finally, activation losses and ohmic losses are described by Equations (8) and (9), respectively, where R^{int} is the internal resistance while B and C are activation voltage constants. The values of model parameters are given in Table 1.

The values of process input and disturbance signals are constrained as follows

$$0.1 \text{ mol s}^{-1} \leq q \leq 2 \text{ mol s}^{-1} \quad (10)$$

$$50 \text{ A} \leq I \leq 150 \text{ A} \quad (11)$$

The values of process variables for the initial operating point are the following: $I = 100$ A, $q = 0.5$ mol s⁻¹, $p_{\text{H}_2} = 5.9102$ atm, $p_{\text{O}_2} = 39.4959$ atm, $p_{\text{H}_2\text{O}} = 22.6160$ atm and $V = 56.6179$ V.

Table 1. Parameters of the fundamental continuous-time model of the PEM system.

| Parameter | Value | Unit | Description |
|-----------------------------|-------------------------|---------------------------------------|-----------------------------|
| B | 0.04777 | A ⁻¹ | Activation voltage constant |
| C | 0.0136 | V | Activation voltage constant |
| CV | 2 | — | Conversion factor |
| E_0 | 0.6 | V | No load voltage |
| F | 96,485 | C mol ⁻¹ | Faraday's constant |
| K_{H_2} | 4.22×10^{-3} | mol s ⁻¹ atm ⁻¹ | Hydrogen valve constant |
| $K_{\text{H}_2\text{O}}$ | 7.716×10^{-3} | mol s ⁻¹ atm ⁻¹ | Water time constant |
| $K_r = N_0/4F_0$ | 2.2802×10^{-3} | mol s ⁻¹ A ⁻¹ | Constant |
| K_{O_2} | 2.11×10^{-2} | mol s ⁻¹ atm ⁻¹ | Oxygen time constant |
| N_0 | 88 | — | Number of cells |
| R_0 | 8.314 | J mol ⁻¹ K ⁻¹ | Universal gas constant |
| R^{int} | 0.00303 | Ω | Internal resistance |
| T_0 | 343 | K | Absolute temperature |
| $\tau_1 = \tau_2$ | 2 | s | Reformer time constants |
| τ_{H_2} | 3.37 | s | Hydrogen time constant |
| $\tau_{\text{H-O}}$ | 1.168 | — | Hydrogen-oxygen flow ratio |
| $\tau_{\text{H}_2\text{O}}$ | 18.418 | s | Water time constant |
| τ_{O_2} | 6.74 | s | Oxygen time constant |

2.3. Wiener Model of PEM Fuel Cell

The discussed PEM fuel cell's continuous-time fundamental model has linear dynamic transfer functions in Equations (2), (3) and (5), but the stack voltage is defined by the nonlinear steady-state Nernst's equation (7). This means that the outputs of the linear dynamic part of the model are inputs of the nonlinear steady-state one. As a result, it is evident that a Wiener structure [35] as an empirical model of the PEM fuel cell under consideration can be used. For the Wiener model, the process variables q , I , and V are scaled as follows

$$u = q - \bar{q}, h = 0.01(I - \bar{I}), y = V - \bar{V} \quad (12)$$

where \bar{q} , \bar{I} and \bar{V} denote values of the signals at the initial operating point.

Identification of a few variants of the Wiener model and selection of the best one is thoroughly discussed in [24,35]. The structure of the chosen model is shown graphically in Figure 1. It has three linear dynamic blocks and one nonlinear steady-state block. The outputs of each linear block, at every sampling instant k , are found with the use of the following formulas

$$v_1(k) = \sum_{i=1}^{n_B^{11}} b_i^{11} u(k-i) + \sum_{i=1}^{n_B^{12}} b_i^{12} h(k-i) - \sum_{i=1}^{n_A^1} a_i^1 v_1(k-i) \quad (13)$$

$$v_2(k) = \sum_{i=1}^{n_B^{21}} b_i^{21} u(k-i) + \sum_{i=1}^{n_B^{22}} b_i^{22} h(k-i) - \sum_{i=1}^{n_A^2} a_i^2 v_2(k-i) \quad (14)$$

$$v_3(k) = \sum_{i=1}^{n_B^3} b_i^3 h(k-i) - \sum_{i=1}^{n_A^3} a_i^3 v_3(k-i) \quad (15)$$

The integers n_A^j for $j = 1, 2, 3$, n_B^{ij} for $i = 1, 2$, $j = 1, 2$ and n_B^3 define the order of the model dynamics. The constant parameters of the linear dynamic blocks are denoted by the real numbers a_i^j ($i = 1, \dots, n_A^j$, $j = 1, 2, 3$), b_i^{jl} ($i = 1, \dots, n_B^{jl}$, $j = 1, 2$, $l = 1, 2$) and b_i^3 ($i = 1, \dots, n_B^3$). It is to be noted that the steady-state block of the model does not use just

the measured outputs of the linear blocks, v_1, v_2, v_3 , but also the measured disturbance signal, h . The output of the model, y , is calculated from the nonlinear steady-state block by using the following general equation

$$y(k) = g(v_1(k), v_2(k), v_3(k), h(k)) \quad (16)$$

A neural network with four inputs, one hidden layer with K units and one output is used as the nonlinear part of the model. The model output is

$$y(k) = w_0^2 + \sum_{l=1}^K w_l^2 \varphi \left(w_{l,0}^1 + \sum_{j=1}^3 w_{l,j}^1 v_j(k) + w_{l,4}^1 h(k) \right) \quad (17)$$

Weights of the network are denoted by $w_{l,j}^1$, $l = 1, \dots, K$, $j = 0, \dots, 4$ and w_l^2 , $l = 0, \dots, K$, for the first and the second layers, respectively.

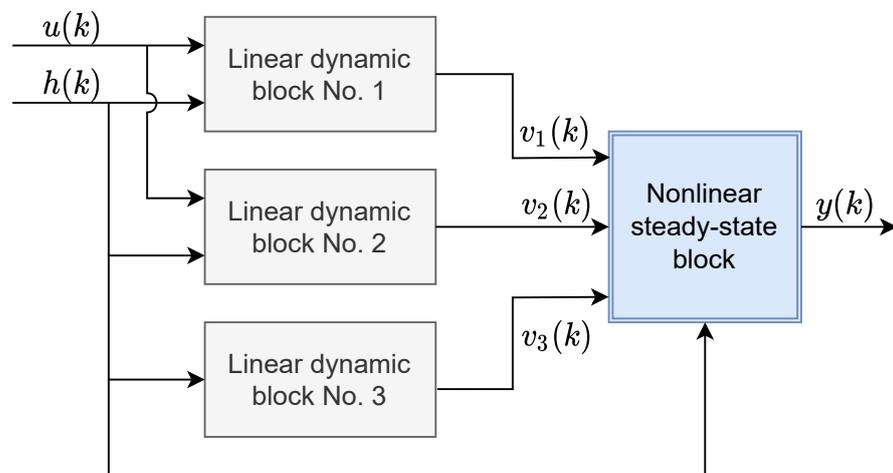


Figure 1. The structure of the Wiener model.

3. Problem Formulation

Although this work is concerned with a specific process (the PEM fuel cell), let us use universal notation in derivation of the MPC algorithm. The manipulated variable (the input of the process) is denoted as u and the controlled variable (the output of the process) is denoted as y . For scaling, Equation (12) is used. Then, the classical rudimentary MPC optimisation task can be presented in the following form

$$\begin{aligned} & \min_{\Delta u(k)} \{J(k)\} \\ & \text{subject to} \\ & u^{\min} \leq u(k+p|k) \leq u^{\max}, \quad p = 0, \dots, N_u - 1 \\ & \Delta u^{\min} \leq \Delta u(k+p|k) \leq \Delta u^{\max}, \quad p = 0, \dots, N_u - 1 \end{aligned} \quad (18)$$

The task is solved at each sampling instant and the solution is the vector of decision variables, of length equal to the control horizon denoted as N_u , containing the increments of the manipulated variable

$$\Delta u(k) = [\Delta u(k|k) \dots \Delta u(k+N_u-1|k)]^T \quad (19)$$

The solution is found with respect to the constraints imposed on the magnitude of the input, defined by u^{\min} and u^{\max} , and the constraints imposed on the increments of that variable, defined by Δu^{\min} and Δu^{\max} . Only the first element of the decision variables vector is applied to the process, i.e., PEM fuel cell system. In general form, the minimised cost-function $J(k)$ can be written in the task as a sum of two parts as follows

$$\begin{aligned}
& \min_{\Delta u(k)} \{J_e(k) + J_{\Delta u}(k)\} \\
& \text{subject to} \\
& u^{\min} \leq u(k+p|k) \leq u^{\max}, p = 0, \dots, N_u - 1 \\
& \Delta u^{\min} \leq \Delta u(k+p|k) \leq \Delta u^{\max}, p = 0, \dots, N_u - 1
\end{aligned} \tag{20}$$

The first part of the MPC cost-function is of the following form

$$J_e(k) = \sum_{p=1}^N F_e(e(k+p|k)) \tag{21}$$

The predicted control error for the future sampling instant $k+p$ computed at the current time step k is denoted by $e(k+p|k)$. In the cost-function, they are taken into account over the prediction horizon N as arguments of a function F_e . The predicted control errors are calculated as follows

$$e(k+p|k) = y^{\text{SP}}(k+p|k) - \hat{y}(k+p|k) \tag{22}$$

where $y^{\text{SP}}(k+p|k)$ is the set-point for the future sampling instant $k+p$ available at the current time step k and $\hat{y}(k+p|k)$ is the predicted output of the process the sampling instant $k+p$ found from a process model at the time step k . The second part of the MPC cost-function is of the following form

$$J_{\Delta u}(k) = \lambda \sum_{p=0}^{N_u-1} (\Delta u(k+p|k))^2 \tag{23}$$

All changes of the manipulated variable (over the control horizon) are minimised when this part is included into the cost-function $J(k)$. In this part, there is also a weighting parameter λ (the larger its value, the larger the penalty term and the slower the control). The function F_e that weighs the predicted control errors (Equation (21)) can be a squared function

$$J_e(k) = (e(k+p|k))^2 \tag{24}$$

or it can be an absolute value function

$$J_e(k) = |e(k+p|k)| \tag{25}$$

Hence, if the square function (24) is used, we obtain the classical MPC cost-function in which squared values of the predicted control errors are considered

$$J_2(k) = \sum_{p=1}^N (y^{\text{SP}}(k+p|k) - \hat{y}(k+p|k))^2 + J_{\Delta u}(k) \tag{26}$$

It is called the L_2 cost-function and the resulting MPC algorithm is named MPC- L_2 . In this work, the L_1 cost-function is studied, which uses the part given by the Equation (25). As a result, we obtain the following L_1 cost-function in which absolute values of the predicted control errors are considered

$$J_1(k) = \sum_{p=1}^N |y^{\text{SP}}(k+p|k) - \hat{y}(k+p|k)| + J_{\Delta u}(k) \tag{27}$$

Let us note that both MPC cost-functions have the same second part defined by Equation (23).

4. Computationally Efficient Nonlinear MPC Using the L₁ Cost-Function

The nonlinear MPC-L₁ optimisation task is formulated with the use of the general form of the optimisation task in (18) and the MPC-L₁ cost function (27). It takes the following form

$$\begin{aligned} \min_{\Delta u(k)} & \left\{ J_1(k) = \sum_{p=1}^N |y^{sp}(k+p|k) - \hat{y}(k+p|k)| + \lambda \sum_{p=0}^{N_u-1} (\Delta u(k+p|k))^2 \right\} \\ \text{subject to} & \\ u^{\min} & \leq u(k+p|k) \leq u^{\max}, \quad p = 0, \dots, N_u - 1 \\ \Delta u^{\min} & \leq \Delta u(k+p|k) \leq \Delta u^{\max}, \quad p = 0, \dots, N_u - 1 \end{aligned} \tag{28}$$

It is important to note that the PEM fuel cell system model is nonlinear, so the predictions of the process output, $\hat{y}(k+p|k)$, are nonlinear functions of the calculated moves $\Delta u(k)$ as well. In such a case, we have to distinguish two problems to solve.

Problem 1: The absolute value function used in the cost-function is not differentiable. As a result, the MPC-L₁ cost-function is also not differentiable. It makes it impossible to use the classical gradient-based optimisation method. For the L₂ norm, this problem does not exist because the MPC-L₂ cost-function is quadratic and differentiable.

Solution to problem 1: The first part of the non-differentiable cost function (27) is replaced by its differentiable representation. This work uses two analytical approximations of the absolute value function. It is also possible to solve this problem using neural approximators [30].

Problem 2: Assuming that a nonlinear model is used in MPC, predictions $\hat{y}(k+p|k)$ are nonlinear functions of the calculated decision vector (19), which means that the MPC-L₁ or MPC-L₂ cost function is nonlinear. As a result, a nonlinear optimisation task is obtained and it must be repeatedly solved at each sampling instant.

Solution to problem 2: The $J_1(k)$ cost function after applying an analytical approximator is still nonlinear in terms of the computed control moves (19). An advanced trajectory linearisation method adopted from [35] is used to improve the derivation of a simple quadratic optimisation task from the nonlinear MPC-L₁ optimisation task.

4.1. Analytical Approximation of the MPC-L₁ Cost-Function

In this work, it is postulated that the absolute value of the predicted error is replaced by a square of an auxiliary function α

$$(\alpha(e(k+p|k)))^2 = |e(k+p|k)| \tag{29}$$

over the whole prediction horizon, i.e., for all $p = 1, \dots, N$. Two versions of analytical approximators of the absolute value function are considered. The first one (v. 1) is described by the following equation [36]

$$(\alpha(e(k+p|k)))^2 = \sqrt{(e(k+p|k))^2 + c} - \sqrt{c}, \quad c > 0 \tag{30}$$

The second one (v. 2) is characterised by Equation [37]

$$(\alpha(e(k+p|k)))^2 = c \log_{10} \left(\cosh \left(\frac{e(k+p|k)}{c} \right) \right), \quad c > 0 \tag{31}$$

A test of both analytical approximators is necessary to find the appropriate value of parameter c which is used in Equations (30) and (31). Figure 2 depicts the comparison between the real absolute function and two approximators for different example values of the parameter c . For the first approximator, two values are chosen for further comparisons of MPC algorithms: $c = 0.01$ and $c = 0.0001$. The value $c = 0.01$ gives quite a rough approximation; it will be interesting to analyse how this inaccuracy influences the resulting

control quality. The influence of this imprecise approximation on control quality possible in MPC is discussed in Section 5. The value $c = 0.0001$ gives very good accuracy. The value $c = 0.01$ is chosen for the second approximator as it gives good accuracy.

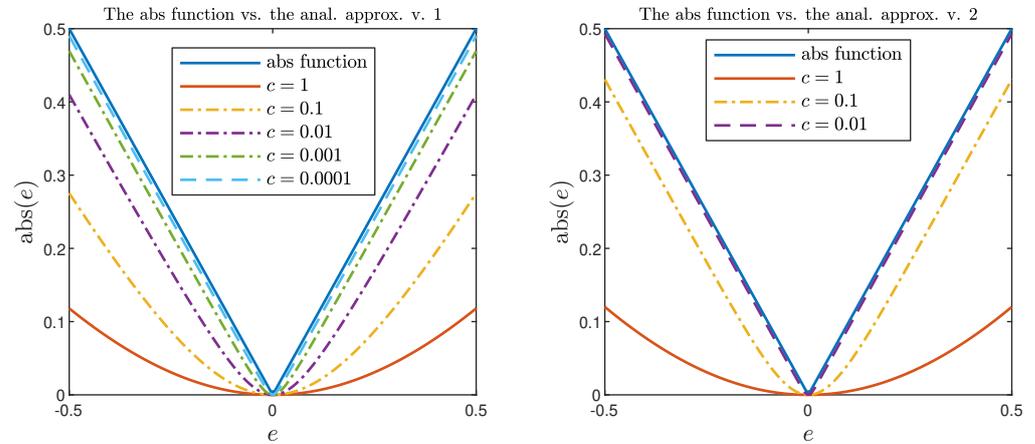


Figure 2. Comparison between the absolute value function and its analytical approximations for different values of parameter c .

For the first version of the approximator, the MPC-L₁ cost function (MPC-L₁ v. 1) is given by

$$J_1(k) = \sum_{p=1}^N \sqrt{(e(k+p|k))^2 + c} - \sqrt{c} + \lambda \sum_{p=0}^{N_u-1} (\Delta u(k+p|k))^2 \quad (32)$$

and for the second version, the MPC-L₁ cost function (MPC-L₁ v. 2) is

$$J_1(k) = \sum_{p=1}^N c \log_{10} \left(\cosh \left(\frac{e(k+p|k)}{c} \right) \right) + \lambda \sum_{p=0}^{N_u-1} (\Delta u(k+p|k))^2 \quad (33)$$

4.2. Advanced Trajectory Linearisation of the MPC-L₁ Cost-Function

As a result of using differentiable approximators of the absolute value function, both $J_1(k)$ cost-functions with the analytical approximators, as defined by Equations (32) and (33), are differentiable now, but still nonlinear. The source of nonlinearity is the model of the controlled process. Namely, the predictions of the future values of the controlled variable and hence the predicted control errors are nonlinear in terms of the calculated future control increments (19). In order to solve this problem, an advanced on-line trajectory linearisation is carried out. It may be proved elsewhere [30] that the linear approximation of the predicted trajectory of the control error over the prediction horizon embedded in the nonlinear function α can be written in general form

$$\alpha(k) = \frac{d\alpha(e^{\text{traj}}(k))}{du^{\text{traj}}(k)} J \Delta u(k) + \alpha(e^{\text{traj}}(k)) + \frac{d\alpha(e^{\text{traj}}(k))}{du^{\text{traj}}(k)} (u(k-1) - u^{\text{traj}}(k)) \quad (34)$$

The future trajectory of the control error

$$\alpha(k) = [\alpha(e(k+1|k)) \dots \alpha(e(k+N|k))]^T \quad (35)$$

and the trajectory of controller errors embedded in the function α

$$\alpha(e^{\text{traj}}(k)) = [\alpha(e^{\text{traj}}(k+1|k)) \dots \alpha(e^{\text{traj}}(k+N|k))]^T \quad (36)$$

are both vectors of length N . The control errors used in the trajectory (35) are calculated from Equation (22) while the following equation

$$e^{\text{traj}}(k+p|k) = y^{\text{sp}}(k+p|k) - \hat{y}^{\text{traj}}(k+p|k) \tag{37}$$

is utilised to determine the control errors in the trajectory (36). Linearisation is carried out at each discrete time sampling instant k about the following trajectory of future control scenario

$$u^{\text{traj}}(k) = [u^{\text{traj}}(k|k) \dots u^{\text{traj}}(k+N_u-1|k)]^T \tag{38}$$

which is a vector of length N_u . Matrix

$$J = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 1 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & \dots & 1 \end{bmatrix} \tag{39}$$

is of size $N_u \times N_u$ and the vector

$$u(k-1) = [u(k-1) \dots u(k-1)]^T \tag{40}$$

is of length N_u . The $N \times N_u$ matrix $\frac{d\alpha(e^{\text{traj}}(k))}{du^{\text{traj}}(k)}$ is required for trajectory approximation (Equation (34)). Its entries, i.e., the partial derivatives $\frac{\partial \alpha(e^{\text{traj}}(k+p|k))}{\partial u^{\text{traj}}(k+r|k)}$ are calculated for all $p = 1, \dots, N$ and $r = 0, \dots, N_u$ at each sampling instant. Differentiation depends on the type of the absolute value function approximator used. For the first approximation method (Equation (30)), we have

$$\frac{\partial \alpha(e^{\text{traj}}(k+p|k))}{\partial u^{\text{traj}}(k+r|k)} = \frac{-e^{\text{traj}}(k+p|k)}{2\sqrt{((e^{\text{traj}}(k+p|k))^2 + c) \left(\sqrt{(e^{\text{traj}}(k+p|k))^2 + c} - \sqrt{c}\right)}} \frac{\partial \hat{y}^{\text{traj}}(k+p|k)}{\partial u^{\text{traj}}(k+r|k)} \tag{41}$$

and for the second one (Equation (31)), we derive

$$\frac{\partial \alpha(e^{\text{traj}}(k+p|k))}{\partial u^{\text{traj}}(k+r|k)} = \frac{-\tanh\left(\frac{e^{\text{traj}}(k+p|k)}{c}\right)}{2\sqrt{c \log_{10}\left(\cosh\left(\frac{e^{\text{traj}}(k+p|k)}{c}\right)\right)}} \frac{\partial \hat{y}^{\text{traj}}(k+p|k)}{\partial u^{\text{traj}}(k+r|k)} \tag{42}$$

Finally, the partial derivatives $\frac{\partial \hat{y}^{\text{traj}}(k+p|k)}{\partial u^{\text{traj}}(k+r|k)}$ are calculated for all $p = 1, \dots, N$ and $r = 0, \dots, N_u$ at each sampling instant for a given dynamical model of the process used for prediction in MPC. As far as the neural Wiener model of the PEM fuel cell system described in Section 2 is concerned, all implementation details are given in [24].

4.3. Formulation of the Computationally Simple MPC-L1 Quadratic Optimisation Task

We consider the general nonlinear MPC-L1 optimisation task defined by Equation (28), where the first part of the minimised cost function is approximated by Equation (34). The same approximation of the trajectory $\alpha(k)$ is used for both analytical approximators of the absolute value function described by Equations (30) and (31). Thanks to the fact that the trajectory (34) is linear in terms of the calculated decision variable vector of MPC, $\Delta u(k)$, we derive the following quadratic programming MPC-L1 optimisation problem

$$\min_{\Delta u(k)} \left\{ J_1(k) = \left\| \frac{d\alpha(e^{\text{traj}}(k))}{du^{\text{traj}}(k)} J \Delta u(k) + \alpha(e^{\text{traj}}(k)) + \frac{d\alpha(e^{\text{traj}}(k))}{du^{\text{traj}}(k)} (u(k-1) - u^{\text{traj}}(k)) \right\|^2 + \|\Delta u(k)\|_{\Lambda}^2 \right\}$$

subject to

$$u^{\min} \leq J \Delta u(k) + u(k-1) \leq u^{\max}$$

$$\Delta u^{\min} \leq \Delta u(k) \leq \Delta u^{\max}$$
(43)

The matrix $\Lambda = \text{diag}(\lambda, \dots, \lambda)$ is of size $N_u \times N_u$. Due to linearisation, the minimised cost function is quadratic in terms of the decision vector, $\Delta u(k)$, and all constraints are linear with respect to the vector $\Delta u(k)$. The vector constraints are defined by the following vectors of length N_u

$$u^{\min} = [u^{\min} \dots u^{\min}]^T, \quad u^{\max} = [u^{\max} \dots u^{\max}]^T, \quad (44)$$

$$\Delta u^{\min} = [\Delta u^{\min} \dots \Delta u^{\min}]^T, \quad \Delta u^{\max} = [\Delta u^{\max} \dots \Delta u^{\max}]^T \quad (45)$$

We name the obtained control scheme the MPC algorithm with Nonlinear Prediction and Linearisation along the Trajectory (MPC-NPLT).

The presented MPC-NPLT- L_1 algorithm is very universal since different kinds of dynamical models may be used for prediction. A specific form of a neural Wiener model (as shown in Figure 1) is used in this work. Nevertheless, due to a universal formulation of the MPC-NPLT- L_1 algorithm, neural networks and fuzzy models of different structures may be used. The general calculation scheme, including the formulation of the quadratic optimisation task (43), is unchanged. The actual model structure must be taken into account when the following two components are calculated:

- (a) The predicted trajectory $\hat{y}^{\text{traj}}(k+p|k)$ for $p = 1, \dots, N$ in Equation (37);
- (b) The derivatives of the predicted trajectory $\frac{\partial \hat{y}^{\text{traj}}(k+p|k)}{\partial u^{\text{traj}}(k+r|k)}$ $p = 1, \dots, N$ and $r = 0, \dots, N_u - 1$ in Equation (41) or (42).

5. Simulations

5.1. Compared Algorithms

The effectiveness of several MPC algorithms for the PEM fuel cell system has been verified using MATLAB software. All experiments have been conducted for $N = 10$, $N_u = 3$ and $\lambda = 1$; the values of MPC tuning parameters are the same as those used in previous research [24]. Let us remind that the first version of the approximation is relatively imprecise while the second one is excellent, as depicted in Figure 2. The MPC algorithms have been assigned to two groups.

The first group of compared control schemes consists of MPC algorithms with Nonlinear Optimisation (MPC-NO) repeated at each sampling instant. The following acronyms are used:

- MPC-NO- L_2 refers to the MPC algorithm with the classical L_2 cost-function that measures the sum of squared predicted control errors.
- MPC-NO- L_1 refers to the MPC algorithm with the sum of absolute values of predicted control errors. It is important to stress that the real absolute function is used, without any approximation. The absolute value function is not differentiable at 0. It may create problems for a nonlinear solver used to minimise the MPC-NO- L_1 optimisation task.
- MPC-NO- L_1 v. 1 refers to the MPC algorithm in which the possibly non-differentiable absolute value function is replaced by its approximation defined by Equation (30).
- Finally, MPC-NO- L_1 v. 2 refers to the MPC algorithm in which the absolute value function is replaced by its approximation defined by Equation (31).

The second group of compared control schemes consists of different configurations of the MPC-NPLT algorithm in which quadratic optimisation is used at each sampling instant in place of nonlinear programming necessary in the MPC-NO approach. The following acronyms are used:

- MPC-NPLT-L₂ stands for the algorithm in which the classical cost-function is used with the quadratic norm.
- MPC-NPLT-L₁ v. 1 stands for the algorithm in which the possibly non-differentiable absolute value function is replaced by its approximation defined by Equation (30).
- MPC-NPLT-L₁ v. 2 refers to the MPC algorithm in which the absolute value function is replaced by its approximation defined by Equation (31).

As recommended in [24], linearisation is performed around the trajectory $u^{\text{traj}}(k)$ (Equation (38)) defined by the most recent value of the manipulated variable computed and applied to the process at the previous sampling step, i.e., $u^{\text{traj}}(k) = [1 \dots 1]^T$.

Three experiments have been conducted for the same control scenario. In the scenario, the set-point trajectory is constant, but there are five step-changes in the disturbance signal. The disturbance trajectory is presented in Figure 3.

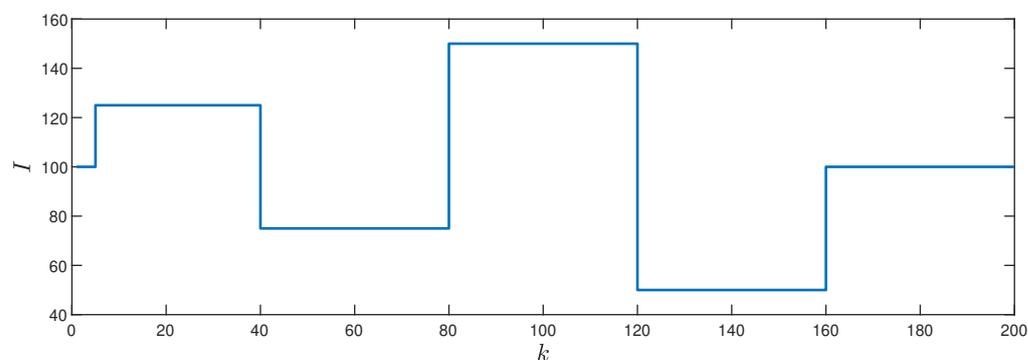


Figure 3. Disturbance step changes.

All simulation results of MPC algorithms which are presented in the following figures, use two analytical approximations of the absolute value function, defined by Equations (30) and (30), respectively, are developed for the parameter $c = 0.01$. Additionally, a different value is considered in Section 5.5 that compares MPC algorithms in terms of different control quality criteria.

5.2. Experiment 1

The first experiment compares the algorithms MPC-NO-L₁, MPC-NO-L₁ v. 1 and MPC-NO-L₁ v. 2. This experiment aims to check if the algorithms with approximations of the absolute value function give similar control quality to those possible when the algorithm with the classical form of the absolute value function is used. All considered MPC algorithms use nonlinear optimisation. For this purpose, the `fmincon` function available in MATLAB is used.

All obtained simulation results are presented in Figure 4. The top panel presents the constant output set-point, denoted as V^{sp} , and the process output signals and the middle panel depicts the trajectories of the manipulated variable. Two bottom panels depict enlarged fragments of the first panel. It is clearly shown that the recorded trajectories are nearly identical. It is evident taking into account two bottom plots, which show parts of the trajectories for two selected step changes of the disturbance. Although the difference between them is minimal, we may note that the second version of the absolute value function approximator used in the MPC-NO-L₁ v. 2 algorithm gives the same trajectories as those possible in the rudimentary MPC-NO-L₁ algorithm. The first version of the absolute value function approximator used in the MPC-NO-L₁ v. 1 algorithm gives slightly different trajectories. It is because the first version of the approximator gives worse results

than the second one, as depicted in Figure 2. The conclusion is that the approximators of the absolute value function give good results. Of course, nonlinear optimisation is used. This difficulty is removed in the following experiments.

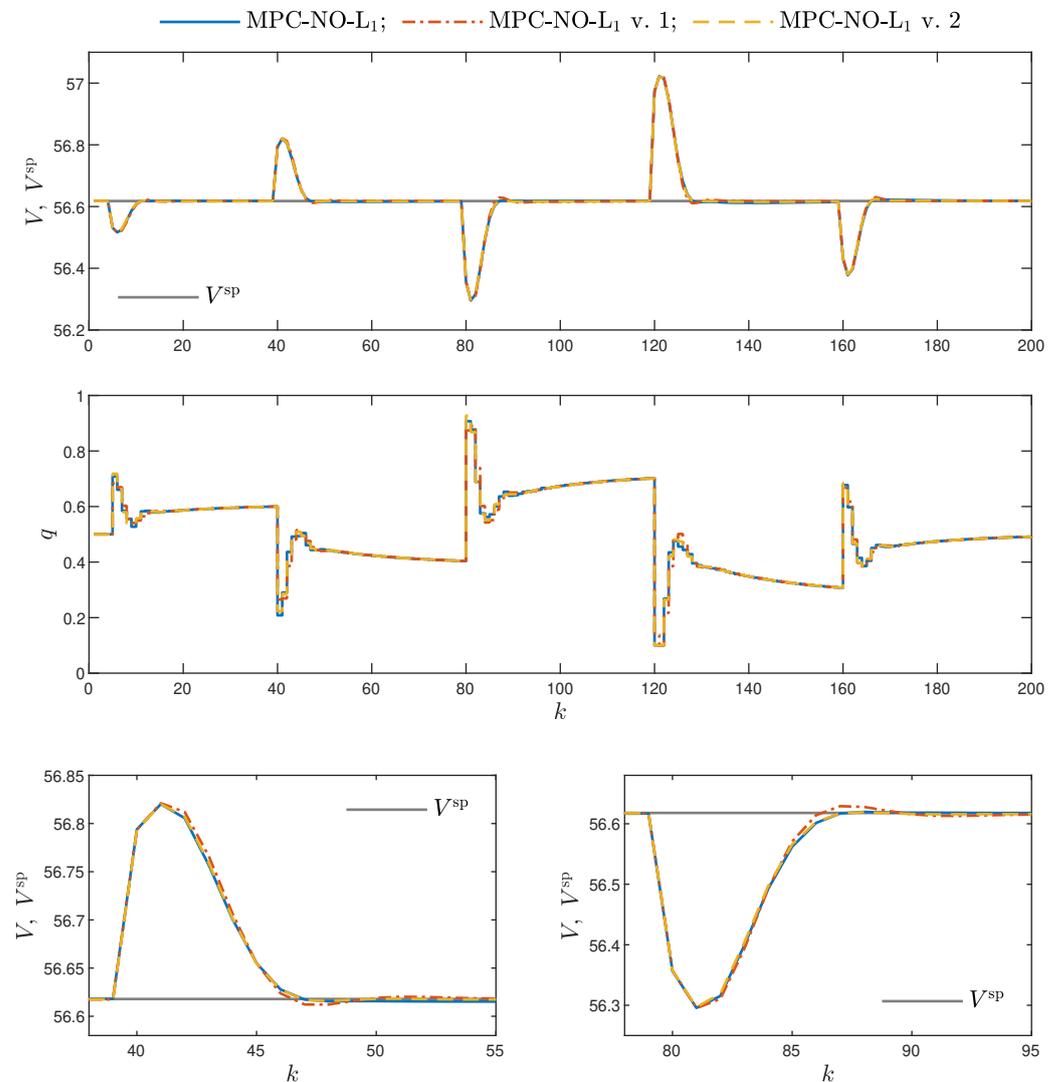


Figure 4. Simulation results: the MPC-NO- L_1 , MPC-NO- L_1 v. 1 and MPC-NO- L_1 v. 2 algorithms.

5.3. Experiment 2

The following algorithms are compared in the second experiment: MPC-NO- L_1 , MPC-NPLT- L_1 v. 1 and MPC-NPLT- L_1 v. 2. This experiment aims to check if the algorithms with advanced linearisation of the predicted output trajectory perform similarly to the MPC-NO- L_1 control scheme. It is to be noted that this is the first test of the algorithms with quadratic optimisation tasks against the algorithm with the nonlinear optimisation task. Both versions of the MPC-NPLT- L_1 algorithms use quadratic optimisation. For this purpose, the quadprog function available in MATLAB is used.

The obtained trajectories are shown in Figure 5. The presented trajectories have minor differences between them. We can draw two conclusions. Firstly, the trajectories of the MPC-NPLT- L_1 v. 2 algorithm are practically the same as those generated by the MPC-NO- L_1 method. It means that advanced trajectory linearisation and quadratic programming may be used to eliminate nonlinear optimisation used at each sampling instant. Secondly, the trajectories of the MPC-NPLT- L_1 v. 1 algorithm are slightly different from two other tested algorithms. As in the first experiment, it is because the first version of the approximator gives worse results than the second one, as depicted in Figure 2.

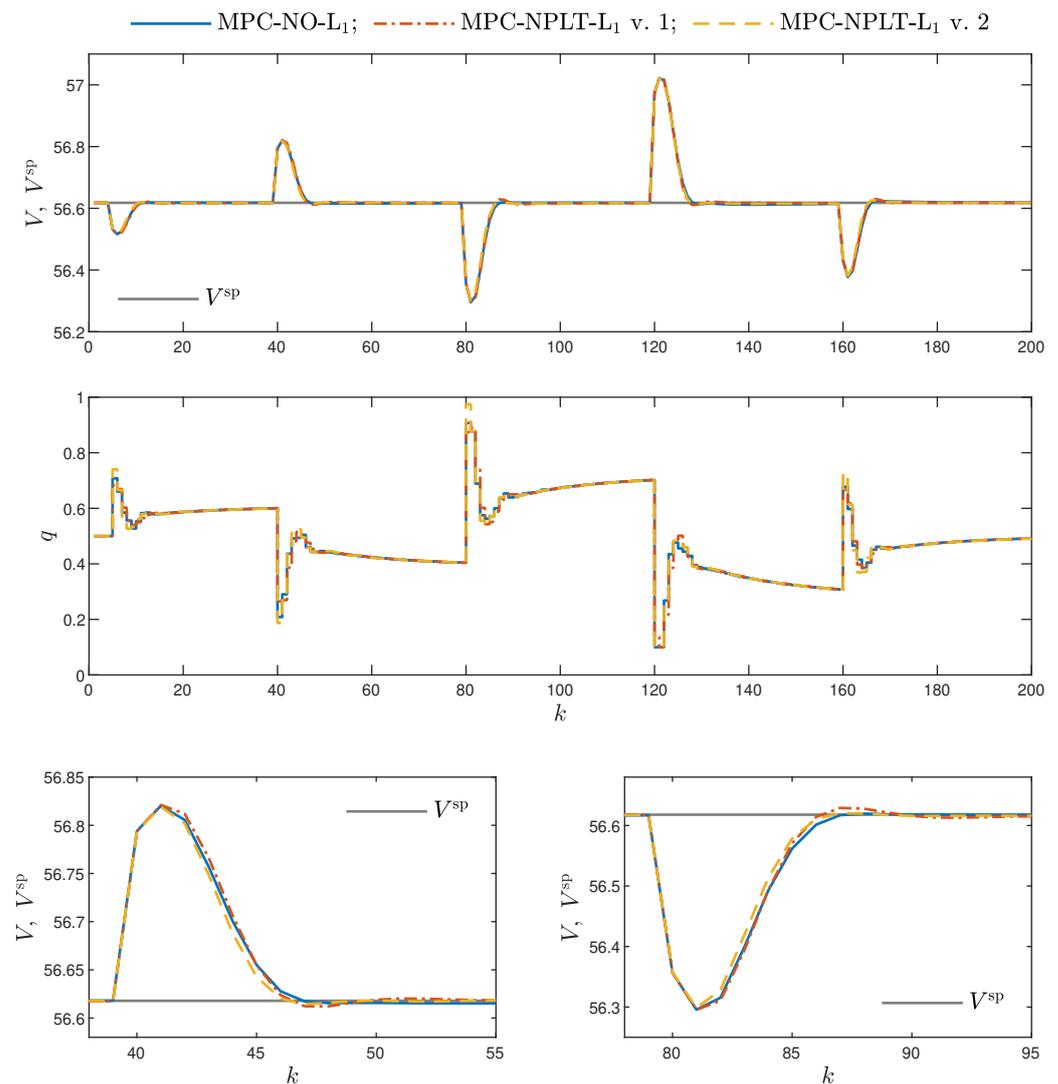


Figure 5. Simulation results: the MPC-NO- L_1 , MPC-NPLT- L_1 v. 1 and MPC-NPLT- L_1 v. 2 algorithms.

5.4. Experiment 3

The third experiment compares process trajectories for MPC-NPLT- L_2 , MPC-NPLT- L_1 v. 1 and MPC-NPLT- L_1 v. 2 control methods. The focus is to check the performance of computationally efficient algorithms with the classical cost-function (the MPC- L_2 cost-function) vs. two versions of approximations of the absolute value functions. All algorithms use quadratic optimisation. For this purpose, the quadprog function is used.

The trajectories of all MPC algorithms are presented in Figure 6. We can make two observations. Firstly, in general, the algorithms with the L_1 cost-function give better results than the algorithm with the L_2 cost-function. The difference in control quality is quite significant. Secondly, as far as the algorithms with the L_1 norm are concerned, better results are possible when the more precise second version of the approximator is used.

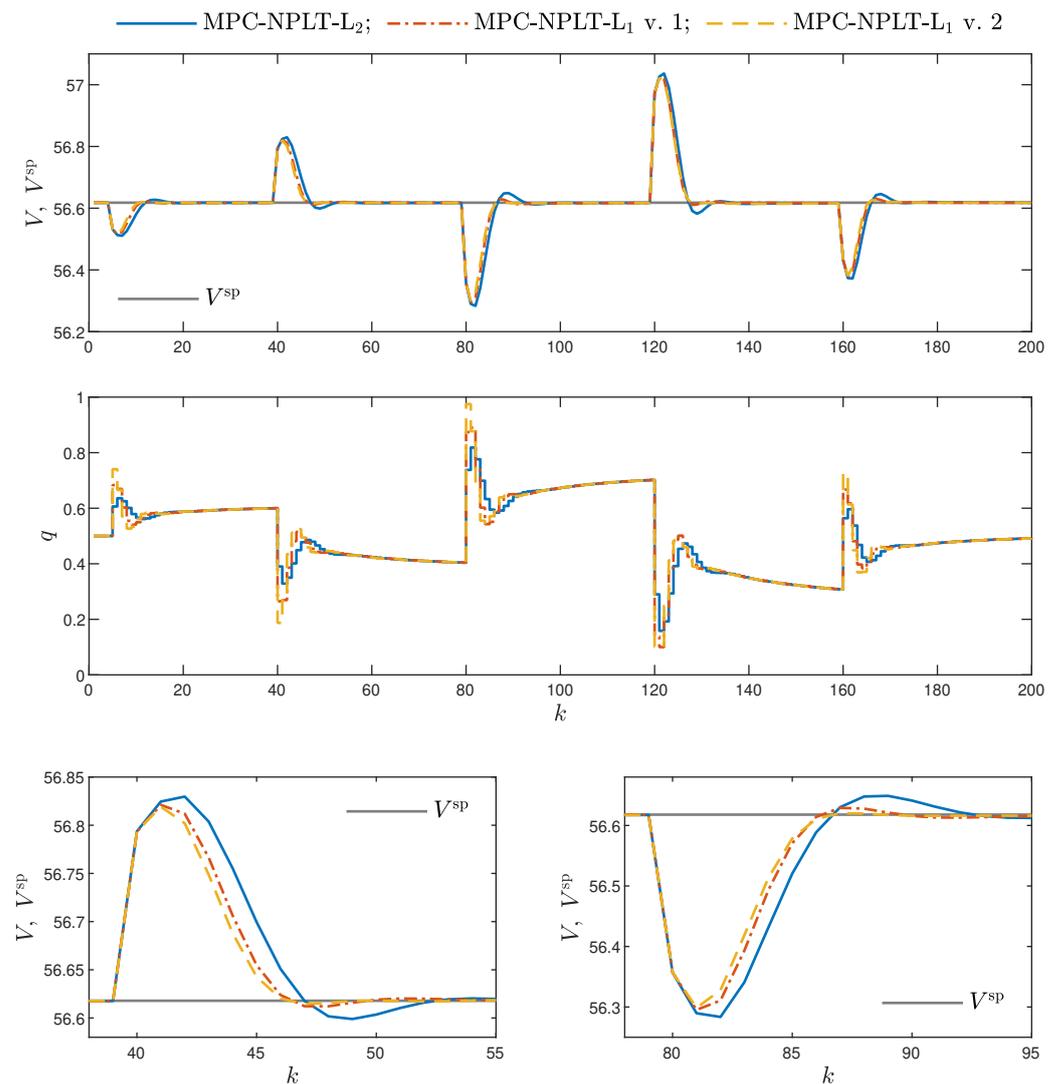


Figure 6. Simulation results: the MPC-NPLT-L₂, MPC-NPLT-L₁ v. 1 and MPC-NPLT-L₁ v. 2 algorithms.

5.5. Control Quality Evaluation

In this work, a multi-criteria control quality assessment is considered [38]. As many as four control quality indicators are calculated after simulation and their values are compared. Two of these indices directly correspond with the L₁ and L₂ cost-functions. Firstly, we consider the sum of absolute values of control errors over the whole simulation horizon

$$E_1 = \sum_{k=0}^{200} |y^{\text{sp}}(k) - y(k)| \quad (46)$$

and the sum of squared values of control errors

$$E_2 = \sum_{k=0}^{200} (y^{\text{sp}}(k) - y(k))^2 \quad (47)$$

The third index is the Gauss standard deviation of the control error denoted as σ_{Gauss} . Finally, the last one is the rational entropy of the control error denoted as H . As emphasised in [38], control quality assessment should be performed using various indicators as only one index, e.g., the commonly used sum of squared control errors, may lead to wrong conclusions about the effectiveness of the analysed control algorithm.

Tables 2 and 3 compare all discussed MPC algorithms in terms of four considered control quality indicators. The first version of the absolute value function approximator uses two values of the parameter c , i.e., 0.01 and 0.0001. Let us remind that the $c = 0.01$ gives a pretty rough approximation. The second version of the approximator uses the parameter $c = 0.01$, which has excellent accuracy. We may make the following observations:

- In general, minimisation of the L_1 cost function gives better all control quality indicators in comparison with the L_2 cost function, i.e., the MPC-NO- L_2 and MPC-NPLT- L_2 algorithms lead to bigger values of quality indices than the algorithm that the L_1 cost function.
- Advanced trajectory linearisation and quadratic optimisation in the MPC-NPLT- L_1 algorithm lead to very similar control quality indicators as the computationally difficult nonlinear optimisation used in the MPC-NO- L_1 scheme.
- The first version of the approximator of the absolute value function with the parameter $c = 0.01$ gives worse results than the same approximator with the parameter $c = 0.0001$ and the second one with $c = 0.01$. It is straightforward since the first approximator is quite imprecise for $c = 0.01$, as depicted in Figure 2.
- As a result of excellent accuracy, the first version of the approximator of the absolute value function with the parameter $c = 0.0001$ and the second version of the approximator with $c = 0.01$ give very good results, practically the same.

The first version of the approximator seems to be computationally simpler. Hence, it is recommended to use the first approximator.

Table 2. Simulation results: E_1 and E_2 indices; the coloured control quality indicators correspond to the minimised cost-function used in MPC.

| Algorithm | E_1 | E_2 |
|------------------------------------|--------|--------|
| MPC-NO- L_1 | 5.6615 | 1.2995 |
| MPC-NO- L_1 v. 1, $c = 0.01$ | 5.5617 | 1.3114 |
| MPC-NO- L_1 v. 1, $c = 0.0001$ | 5.4832 | 1.2826 |
| MPC-NO- L_1 v. 2, $c = 0.01$ | 5.4842 | 1.2827 |
| MPC-NPLT- L_1 v. 1, $c = 0.01$ | 5.5517 | 1.3113 |
| MPC-NPLT- L_1 v. 1, $c = 0.0001$ | 5.3068 | 1.2588 |
| MPC-NPLT- L_1 v. 2, $c = 0.01$ | 5.2920 | 1.2398 |
| MPC-NO- L_2 | 6.7640 | 1.5779 |
| MPC-NPLT- L_2 | 6.7640 | 1.5779 |

Table 3. Simulation results: σ_{Gauss} and H indices.

| Algorithm | σ_{Gauss} | H |
|------------------------------------|-------------------------|-------------------------|
| MPC-NO- L_1 | 8.0808×10^{-2} | 4.3298×10^{-2} |
| MPC-NO- L_1 v. 1, $c = 0.01$ | 8.1175×10^{-2} | 4.3480×10^{-2} |
| MPC-NO- L_1 v. 1, $c = 0.0001$ | 8.0281×10^{-2} | 4.2735×10^{-2} |
| MPC-NO- L_1 v. 2, $c = 0.01$ | 8.0282×10^{-2} | 4.2735×10^{-2} |
| MPC-NPLT- L_1 v. 1, $c = 0.01$ | 8.1170×10^{-2} | 4.3480×10^{-2} |
| MPC-NPLT- L_1 v. 1, $c = 0.0001$ | 7.9527×10^{-2} | 3.9145×10^{-2} |
| MPC-NPLT- L_1 v. 2, $c = 0.01$ | 7.8926×10^{-2} | 3.8998×10^{-2} |
| MPC-NO- L_2 | 8.9044×10^{-2} | 5.4130×10^{-2} |
| MPC-NPLT- L_2 | 8.9044×10^{-2} | 5.4130×10^{-2} |

6. Conclusions

This paper describes a fast MPC algorithm with the L_1 norm and its application to control the PEM fuel cell system. The classical non-differentiable absolute value function is replaced with its analytical approximation. Moreover, linearisation of the predicted process trajectory is performed at every sampling time instant to improve computational efficiency further. Two approximation methods are described and implementation details are derived. The obtained trajectories and four control quality indices have been compared to analyse the algorithm's performance thoroughly.

The presented MPC-NPLT- L_1 algorithm allows to control the PEM fuel cell effectively. Firstly, it gives excellent control quality, the same as the MPC-NO- L_1 scheme with nonlinear optimisation. Secondly, minimisation of the L_1 cost function gives better results than the use of the classical L_2 cost function. Thirdly, the use of the analytical approximation of the absolute value function is much easier than the application of a neural approximation considered in the literature [30]. Two versions of the analytical approximators are considered in this work. When properly tuned, both give excellent results, but the first one is computationally simpler and recommended.

It is necessary to emphasise a universal nature of the presented MPC-NPLT- L_1 algorithm since different kinds of dynamical models may be used for prediction. Although a specific form of a neural Wiener model is used in this work, neural networks and fuzzy models of different structures may also be used. The general calculation scheme, including the formulation of the quadratic optimisation task, is unchanged. The actual model structure must be taken into account when the predicted output trajectory and the derivatives of that trajectory are calculated. The only limitation is differentiability of the model.

In the future, it is planned to study the choice of alternative cost-functions to control PEM fuel cell systems and their influence on control quality.

Author Contributions: Conceptualization, M.Ł. and R.N.; methodology, M.Ł. and R.N.; software, M.Ł. and R.N.; validation, R.N.; formal analysis, M.Ł. and R.N.; investigation, M.Ł. and R.N.; resources, M.Ł. and R.N.; data curation, M.Ł. and R.N.; writing—original draft preparation, M.Ł. and R.N.; writing—review and editing, M.Ł. and R.N.; visualization, M.Ł. and R.N.; supervision, M.Ł.; project administration, M.Ł.; funding acquisition, M.Ł. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Warsaw University of Technology.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Larminie, J.; Dicks, A. *Fuel Cell Systems Explained*; John Wiley & Sons: Hoboken, NJ, USA, 2000.
2. Barbir, F. *PEM Fuel Cells: Theory and Practice*; Academic Press: London, UK, 2013.
3. Özbek, M. *Modeling, Simulation, and Concept Studies of a Fuel Cell Hybrid Electric Vehicle Powertrain*; University of Duisburg-Essen: Duisburg, Germany, 2010.
4. Schröter, J.; Frank, D.; Radke, V.; Bauer, C.; Kallo, J.; Willich, C. Influence of Low Inlet Pressure and Temperature on the Compressor Map Limits of Electrical Turbo Chargers for Airborne Fuel Cell Applications. *Energies* **2022**, *15*, 2896. [[CrossRef](#)]
5. Cao, Y.; Ayed, H.; Jafarmadar, S.; Abdollahi, M.A.A.; Farag, A.; Wae-hayee, M.; Hashemian, M. PEM fuel cell cathode-side flow field design optimization based on multi-criteria analysis of liquid-slug dynamics. *J. Ind. Eng. Chem.* **2021**, *98*, 397–412. [[CrossRef](#)]
6. Yin, C.; Gao, Y.; Li, K.; Wu, D.; Song, Y.; Tang, H. Design and numerical analysis of air-cooled proton exchange membrane fuel cell stack for performance optimization. *Energy Convers. Manag.* **2021**, *245*, 114604. [[CrossRef](#)]
7. Yin, C.; Song, Y.; Liu, M.; Gao, Y.; Li, K.; Qiao, Z.; Tang, H. Investigation of proton exchange membrane fuel cell stack with inversely phased wavy flow field design. *Appl. Energy* **2022**, *305*, 117893. [[CrossRef](#)]
8. Luciani, S.; Tonoli, A. Control Strategy Assessment for Improving PEM Fuel Cell System Efficiency in Fuel Cell Hybrid Vehicles. *Energies* **2022**, *15*, 2004. [[CrossRef](#)]
9. Napole, C.; Derbeli, M.; Barambones, O. Experimental analysis of a fuzzy scheme against a robust controller for a proton exchange membrane fuel cell system. *Symmetry* **2022**, *14*, 139. [[CrossRef](#)]
10. Jia, Y.; Zhang, R.; Lv, X.; Zhang, T.; Fan, Z. Research on temperature control of fuel-cell cooling system based on variable domain fuzzy PID. *Processes* **2022**, *10*, 534. [[CrossRef](#)]
11. Xiao, X.; Lv, J.; Chang, Y.; Chen, J.; He, H. Adaptive sliding mode control integrating with RBFNN for proton exchange membrane fuel cell power conditioning. *Appl. Sci.* **2022**, *12*, 3132. [[CrossRef](#)]
12. Silaa, M.Y.; Barambones, O.; Derbeli, M.; Napole, C.; Bencherif, A. Fractional order PID design for a proton exchange membrane fuel cell system using an extended grey wolf optimizer. *Processes* **2022**, *10*, 450. [[CrossRef](#)]
13. Xiong, S.; Wu, Z.; Li, W.; Li, D.; Zhang, T.; Lan, Y.; Zhang, X.; Ye, S.; Peng, S.; Han, Z.; et al. Improvement of temperature and humidity control of proton exchange membrane fuel cells. *Sustainability* **2021**, *13*, 10578. [[CrossRef](#)]

14. Chen, J.; Zhang, C.; Li, K.; Zhan, Y.; Sun, B. Hybrid adaptive control for PEMFC gas pressure. *Energies* **2020**, *13*, 5334. [[CrossRef](#)]
15. Trinh, H.; Truong, H.; Ahn, K. Development of fuzzy-adaptive control based energy management strategy for PEM fuel cell hybrid tramway system. *Appl. Sci.* **2022**, *12*, 3880. [[CrossRef](#)]
16. Derbeli, M.; Charaabi, A.; Barambones, O.; Napole, C. High-performance tracking for proton exchange membrane fuel cell system PEMFC using model predictive control. *Mathematics* **2021**, *9*, 1158. [[CrossRef](#)]
17. Chen, X.; Fang, Y.; Liu, Q.; He, L.; Zhao, Y.; Huang, T.; Wan, Z.; Wang, X. Temperature and voltage dynamic control of PEMFC Stack using MPC method. *Energy Rep.* **2022**, *8*, 798–808. [[CrossRef](#)]
18. Kanouni, B.; Badoud, A.E.; Mekhilef, S. A multi-objective model predictive current control with two-step horizon for double-stage grid-connected inverter PEMFC system. *Int. J. Hydrogen Energy* **2022**, *47*, 2685–2707. [[CrossRef](#)]
19. Wang, Y.; Li, H.; Feng, H.; Han, K.; He, S.; Gao, M. Simulation study on the PEMFC oxygen starvation based on the coupling algorithm of model predictive control and PID. *Energy Convers. Manag.* **2021**, *249*, 114851. [[CrossRef](#)]
20. Hähnel, C.; Aul, V.; Horn, J. Power control for efficient operation of a PEM fuel cell system by nonlinear model predictive control. *IFAC-PapersOnLine* **2015**, *48*, 174–179. [[CrossRef](#)]
21. Rosanas-Boeta, N.; Ocampo-Martinez, C.; Kunusch, C. On the anode pressure and humidity regulation in PEM fuel cells: A nonlinear predictive control approach. *IFAC-PapersOnLine* **2015**, *48*, 434–439. [[CrossRef](#)]
22. Schultze, M.; Horn, J. Modeling, state estimation and nonlinear model predictive control of cathode exhaust gas mass flow for PEM fuel cells. *Control Eng. Pract.* **2016**, *43*, 76–86. [[CrossRef](#)]
23. Ziogou, C.; Papadopoulou, S.; Georgiadis, M.; Voutetakis, S. On-line nonlinear model predictive control of a PEM fuel cell system. *J. Process. Control* **2013**, *23*, 483–492. [[CrossRef](#)]
24. Ławryńczuk, M.; Söffker, D. Wiener Structures for Modeling and Nonlinear Predictive Control of Proton Exchange Membrane Fuel Cell. *Nonlinear Dyn.* **2019**, *95*, 1639–1660. [[CrossRef](#)]
25. Vrlić, M.; Ritzberger, D.; Jakubek, S. Safe and efficient polymer electrolyte membrane fuel cell control using successive linearization based model predictive control validated on real vehicle data. *Energies* **2020**, *13*, 5353. [[CrossRef](#)]
26. Dötlinger, A.; Kennel, R.M. Near time-optimal model predictive control using an L1-norm based cost functional. In Proceedings of the 2014 IEEE Energy Conversion Congress and Exposition, Pittsburgh, PA, USA, 14–18 September 2014; pp. 3504–3511.
27. Domański, P.; Ławryńczuk, M. Impact of MPC embedded performance index on control quality. *IEEE Access* **2021**, *9*, 24787–24795. [[CrossRef](#)]
28. Fehér, M.; Straka, O.; Šmídl, V. Model predictive control of electric drive system with L1-norm. *Eur. J. Control* **2020**, *56*, 242–253. [[CrossRef](#)]
29. Karamanakos, P.; Geyer, T.; Kennel, R. On the choice of norm in finite control set model predictive control. *IEEE Trans. Power Electron.* **2018**, *33*, 7105–7117. [[CrossRef](#)]
30. Ławryńczuk, M.; Nebeluk, R. Computationally Efficient Nonlinear Model Predictive Control Using the L₁ Cost-Function. *Sensors* **2021**, *21*, 270–289.
31. Uzunoglu, M.; Alam, M.S. Dynamic modeling, design and simulation of a combined PEM fuel cell and ultracapacitor system for stand-alone residential applications. *IEEE Trans. Energy Conv.* **2006**, *21*, 767–775. [[CrossRef](#)]
32. Uzunoglu, M.; Alam, M.S. Dynamic modeling, design and simulation of a PEM fuel cell/ultra-capacitor hybrid system for vehicular applications. *Energy Conv. Manag.* **2009**, *48*, 1544–1553. [[CrossRef](#)]
33. Erdinc, O.; Vural, B.; Uzunoglu, M.; Ates, Y. Modeling and analysis of an FC/UC hybrid vehicular power system using a wavelet-fuzzy logic based load sharing and control algorithm. *Int. J. Hydrogen Energy* **2009**, *34*, 5223–5233. [[CrossRef](#)]
34. Kisacikoglu, M.C.; Uzunoglu, M.; Alam, M. Load sharing using fuzzy logic control in a fuel cell/ultracapacitor hybrid vehicle. *Int. J. Hydrogen Energy* **2009**, *34*, 1497–1507. [[CrossRef](#)]
35. Ławryńczuk, M. *Nonlinear Predictive Control Using Wiener Models: Computationally Efficient Approaches for Polynomial and Neural Structures*; Studies in Systems, Decision and Control Series; Springer: Cham, Switzerland, 2022; Volume 389.
36. Stack Exchange. Smooth Approximation of Absolute Value Inequalities. Available online: <https://math.stackexchange.com/questions/172439/smooth-approximation-of-absolute-value-inequalities> (accessed on 6 June 2022).
37. Zhang, H.; Wang, J.; Zeng, D.; Tao, X.; Ma, J. Regularization strategies in statistical image reconstruction of low-dose X-ray CT: A review. *Med. Phys.* **2018**, *45*, e886–e907. [[CrossRef](#)] [[PubMed](#)]
38. Domański, P. *Control Performance Assessment: Theoretical Analyses and Industrial Practice*; Studies in Systems, Decision and Control Series; Springer: Cham, Switzerland, 2020; Volume 245.