



Article Data-Driven Predictive Control of Exoskeleton for Hand Rehabilitation with Subspace Identification

Erkan Kaplanoglu ^{1,*} and Gazi Akgun ²

- ¹ Department of Engineering Management & Technology, University of Tennessee at Chattanooga, Chattanooga, TN 37403, USA
- ² Department of Mechatronics Engineering, Marmara University, Istanbul 34744, Turkey
- * Correspondence: erkan-kaplanoglu@utc.edu

Abstract: This study proposed a control method, a data-driven predictive control (DDPC), for the hand exoskeleton used for active, passive, and resistive rehabilitation. DDPC is a model-free approach based on past system data. One of the strengths of DDPC is that constraints of states can be added to the controller while performing the controller design. These features of the control algorithm eliminate an essential problem for rehabilitation robots in terms of easy customization and safe repetitive rehabilitation tasks that can be planned within certain constraints. Experiments were carried out with a designed hand rehabilitation system under repetitive and various therapy tasks. Real-time experiment results demonstrate the feasibility and efficiency of the proposed control approach to rehabilitation systems.

Keywords: DDPC; hand rehabilitation; subspace identification



Citation: Kaplanoglu, E.; Akgun, G. Data-Driven Predictive Control of Exoskeleton for Hand Rehabilitation with Subspace Identification. *Sensors* 2022, 22, 7645. https://doi.org/ 10.3390/s22197645

Academic Editors: Emilio Trigili, Francesca Cordella and Jan Babič

Received: 30 August 2022 Accepted: 7 October 2022 Published: 9 October 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

To regain the limb movement ability lost because of any disease or accident, a repetitive and intense rehabilitation process is required. Physiotherapists treat patients during this process. They frequently use rehabilitation robots to help patients perform the right movements with the right intensity while under control inside or outside of the clinic [1]. In recent years, academic or R&D studies were conducted frequently for design and implementation of rehabilitation robotics. Exoskeletons, which allow for direct limb manipulation, or end-effect robots, that support therapy by manipulating the limb's distal point, are two mechanical structures that are used in rehabilitation robot design. Specially designed hand rehabilitation robots are used to treat post-stroke hand movement limitations. These robots need to be made in accordance with the complex structure of the hand, which has about 20 degrees of freedom; it should also be supported by a robust and adaptive control algorithm so that it can function consistently for each patient [2-5]. Devices that support active rehabilitation can be used during the patient's completely lost movement during the passive phase of the repetitive rehabilitation process. The rehabilitation robot should perform fewer movement tasks as the patient recovers more and give them more responsibility. It should assist in this instance with active-assistive rehabilitation procedures. Resistance exercises for muscle strengthening can be done once the patient regains his mobility [6,7]. Robot control design must have an adaptive structure to perform the rehabilitation processes. Active power control is not possible for the direct trajectory control robots described in the literature. Rehabilitation robots are controlled to carry out active, active-assistive, and passive rehabilitation tasks using control algorithms, such as impedance or admittance control [8–10].

Systems with nonlinear structured uncertainties can also be controlled using variable structure controllers. These controllers allow for the use of parametric perturbations between lower and upper limits to deal with complexity and noise from the external world [11]. The control of a lower extremity exoskeleton was accomplished in [12] using the anti-disturbance sliding mode controller. Additionally, a robust adaptive sliding mode controller is proposed in [13] to deal with unknown and bounded dynamic uncertainties of the upper limb exoskeleton for passive rehabilitation tasks. A fuzzy approximation-based backstepping control is another approach to the control of an exoskeleton for rehabilitation [14,15].

System identification is crucial for establishing the controller design. The kinematic and dynamic parameters must be accurately determined, and the robot must generate the force required by the patient for it to adapt to the kinematic structure of each patient [16,17].

The limb that the robot works on simultaneously during rehabilitation is breakable and sensitive. While manipulating this limb, the robot must avoid harming it. Both the patient and the robot must cooperate, and the robot can only move the patient's hand within limits. In this instance, the control algorithm should find the best solution—not just any solution—for the targeted task while taking the constraints into account. Algorithms for model predictive control can be employed to achieve this [2,18].

Model-based control algorithms, such as model predictive control, are frequently used in process control applications because they produce the optimal solutions possible, given the constraints. These algorithms can also be used for rehabilitation procedures because they use past data collected over a specific horizon to determine which estimation model offers the best solution [19,20]. For every patient, a unique model will be developed with unique system parameters and dynamics. A universal model cannot be used in this case. Since MPC algorithms are model-based algorithms, continuous identification data-driven predictive control algorithms can be applied in this situation [21].

The subspace-based parameter estimation approach is used in this study to estimate system parameters. The parameters appropriate for the model defined along a horizon are obtained using the state data and control signal gathered during open loop operation along this horizon [22]. A predictive control algorithm and an optimal control rule are built using the model and parameters acquired here. The control rule is handled independently of the constraints, and experiments are carried out to determine how much each model variable contributes to the control rule's success.

The rest of the paper is organized as follows: Section 2.1 introduces the suggested control strategy, followed by Sections 2.2 and 2.3, which describe the subspace identification method and data-driven predictive controller, respectively. The experimental setup is also described in the Section 2. The Section 3 of the paper includes the experimental results. This section examines and presents the impact of all parameters on the success of the DDPC. Finally, every result is discussed and concluded.

2. Materials and Methods

2.1. Exoskeleton for Hand Rehabilitation

The mechanical structure of the exoskeleton that was previously proposed is effective with the simple design. In this structure, the middle and proximal phalanxes serve as links for two 4-bar mechanisms that are sequentially coupled. Additionally, the metacarpal phalangeal (MCP) joint has a range of motion of 55 degrees of flexion, whereas the proximal interphalangeal (PIP) joint has a range of motion of 65 degrees of flexion from a fully extended posture. The linear actuator may move at a maximum speed of 12 mm per second and a maximum force of 45 N. A full hand opening or closing action requires roughly 5 s for a stroke of 50 mm. The mechanical structure and biodynamic fit of the hand make the designed system more practical in terms of usage and productivity than similar ones [2,23,24].

The system consists of *L*0, *L*1, *L*2 linkages. The MCP joint angle (ϕ_1) and PIP joint angle (ϕ_2) rely on the length of the linear actuator (l_0), as shown in Figure 1. Therefore, simultaneous actuation of both joints occurs. Whenever a new user (subject or patient) puts on the exoskeleton on their hand, the exoskeleton and the finger unite to form a single, distinctive 1-DOF system.



Figure 1. Designed exoskeleton for hand rehabilitation.

2.2. Proposed Control Method

In this study, position tracking was performed with a DDPC designed using subspace identification and model predictive control (MPC) techniques. MPC is a model-based technique that was successfully applied over the years. The difficulties (cost, time, and effort) associated with the identification of a predictive model of the system are major barriers that prevent the widespread adoption of MPC for complex systems.

It is challenging to describe the system model when using rehabilitation robots, orthotics, and prosthetic applications, because the controlled system's parameters vary depending on the patient or the healing process. Therefore, data-driven algorithms are chosen over model-based algorithms. Model-based control is often a time-consuming and expensive process. Additionally, it is ineffective for a system that will be applied to multi-user processes with various mechanical characteristics. The benefit of the DDPC method is that it contains historical data continuously, making the system sensitive to any changes in mechanical parameters. The need for memory is made clear by the necessity of storing historical data for modeling. This may also be viewed as a drawback.

The input and output signals obtained from open loop data are used as input and output variables in the DDPC. The obtained sub-space matrices can be used to calculate the DDPC algorithm weight parameters. During the specified horizon, the system can keep track of the reference using a control rule based on past input and output data as well as tracking errors. By closely monitoring all activity along the horizon, the rule based on previous data will be able to react quickly. Particularly, the fast and accurate identification of new circumstances will enable quick adaptation. The structure of the proposed control method is shown in Figure 2.



Figure 2. Structure of proposed data-driven predictive controller block diagram.

2.3. System Identification Methods

2.3.1. Output Error Method for Identification

The purpose of the output error method is to find the best parametric model according to the given specific criteria. The criteria is the error between the measured noisy output and the simulated model output, as shown in Equation (1).

$$\epsilon = x_2 - \hat{x}_2 \tag{1}$$

Here, \hat{x}_2 is the estimated state variable of the system calculated using estimated values of unknown dynamic parameters. If the parallel model configuration of the model is similar to 2, we can find the parameter with equations in 3.

$$\hat{x}_2 = -\hat{a}\hat{x}_2 - \hat{f}_s sgn(\hat{x}_2) + \hat{b}u, \quad \hat{x}_2(0) = (\hat{x}_2)_0$$
(2)

where a, b, and f_s are the unknown parameters to be identified. In this equation, x_2 is the state variable, and u is the input of the system, and both are measured and/or observed.

where γ_1 , γ_2 and γ_3 are the gain of the error effected to parameter. The parameters *a*, *b*, and *f*_s, estimated by the output error method, were studied and compared in [2] previous studies.

2.3.2. Subspace Identification Method

Background information on the subspace identification matrices from open-loop data is provided in this subsection. The following section will use these matrices to design a data-driven predictive controller. We will begin by defining the system's state-space model. As a result, the equations below can be written in state-space form for a linear discrete time-invariant system:

$$x_{k+1} = Ax_k + Bu_k + Ke_k \tag{4}$$

$$y_k = Cx_k + Du_k + e_k \tag{5}$$

where $u_k \in \mathbb{R}^m$, $y_k \in \mathbb{R}^l$, and $x_k \in \mathbb{R}^m$ are the input variables, the output variables, and the state vector variables of the system, respectively; $e_k \in \mathbb{R}^l$ is white noise disturbance. The system matrices $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{l \times n}$, $D \in \mathbb{R}^{l \times m}$, and $K \in \mathbb{R}^{l \times l}$ are the state, input, output, feed-through, and Kalman filter gain matrices of the system, respectively.

We assume that the measurements of the inputs u_k and the outputs y_k for $k \in \{1, 2, ..., N\}$ are available for identification. The input Hankel matrices for u_k are represented as U_p and U_f .

$$U_{p} \triangleq \begin{bmatrix} u_{1} & u_{2} & \cdots & u_{N-2M+1} \\ u_{2} & u_{3} & \cdots & u_{N-2M+2} \\ \vdots & \vdots & \ddots & \vdots \\ u_{M} & u_{M+1} & \cdots & u_{N-M+1} \end{bmatrix}$$
(6)
$$U_{f} \triangleq \begin{bmatrix} u_{M+1} & u_{M+2} & \cdots & u_{N-M+1} \\ u_{M+2} & u_{M+3} & \cdots & u_{N-M+2} \\ \vdots & \vdots & \ddots & \vdots \\ u_{2M} & u_{2M+1} & \cdots & u_{N} \end{bmatrix}$$
(7)

where the subscripts 'p' and 'f' represent the 'past' and 'future' matrices of the variables. Similarly, Hankel matrices for y_k , represented as Y_p and Y_f defined as (8) and

meth

Equa

past and future state vectors are written as:

(9), respectively. The dimensions of the matrices are $\{Y_p, Y_f\} \in \mathbb{R}^{Ml \times N-2M+1}$ and $\left\{ U_p, U_f \right\} \in \mathbb{R}^{Mm \times N - 2M + 1}$, respectively.

$$Y_{p} \triangleq \begin{bmatrix} y_{1} & y_{2} & \cdots & y_{N-2M+1} \\ y & y_{3} & \cdots & y_{N-2M+2} \\ \vdots & \vdots & \ddots & \vdots \\ y_{M} & y_{M+1} & \cdots & y_{N-M+1} \end{bmatrix}$$

$$Y_{f} \triangleq \begin{bmatrix} y_{M+1} & y_{M+2} & \cdots & y_{N-M+1} \\ y_{M+2} & y_{M+3} & \cdots & y_{N-M+2} \\ \vdots & \vdots & \ddots & \vdots \end{bmatrix}$$
(8)
$$(9)$$

$$\begin{bmatrix} y_{2M} & y_{2M+1} & \cdots & y_N \end{bmatrix}$$

These data block Hankel matrices are made rectangular in the subspace identification method to reduce the undesirable effects of noise on the identification system. This situation can be achieved via having a large set of data, denoted by the variable *N*. Moreover, *M* in Equations (6)–(9) can be understood as the order of the predictor equation. For a successful identification of the system behavior, the order *M* must be bigger or at least equal to the real system order *n* as manifested in the dimension of the state matrix A [25]. The system's

$$X_p \triangleq \begin{bmatrix} x_1 & x_2 & \cdots & x_{N-2M+1} \end{bmatrix}$$
(10)

$$X_f \triangleq \begin{bmatrix} x_{M+1} & x_{M+2} & \cdots & x_{N-M+1} \end{bmatrix}$$
(11)

As a result of the derivation of Equations (4) and (5), the equations can be written as below. These equations are known as the subspace matrix input-output equations used for identification [26,27].

$$Y_f = \Gamma_M X_f + H_M^d U_f + H_N^s E_f \tag{12}$$

$$Y_p = \Gamma_M X_p + H_M^d U_p + H_N^s E_p \tag{13}$$

$$X_f = A^M X_p + \Delta^d_M U_p + \Delta^s_M E_p \tag{14}$$

 $\Gamma_M \in \mathbb{R}^{Ml imes n}$ can be described as the extended observability matrix, $\Delta_M^d \in \mathbb{R}^{n imes Mm}$ as reversed extended controllability matrix (deterministic), and $\Delta_M^s \in \mathbb{R}^{n \times Ml}$ as the reversed extended controllability matrix (stochastic) [20,28]. Y_f can be written with Equations (12)–(14) as below:

$$Y_f = \left[\Gamma_m A^m \Gamma_m^{\dagger} - A^m (H_m + \Delta_m)\right] \begin{bmatrix} Y_p \\ U_p \end{bmatrix} + H_m U_f + (\Delta_M^s - A^m {\Gamma_m}^{\dagger} H_M^s) E_p + H_M^s E_f \quad (15)$$

Since the effect of E_f is constant white noise, and cause of the stability of a Kalman filter, Equation (15) can be written to give an optimal prediction expression of the system output Y_f as follows:

$$\hat{Y}_f = L_w W_p + L_u U_f \tag{16}$$

where $W_p = \begin{bmatrix} Y_p & U_p \end{bmatrix}^T$, U_f consists of past inputs and outputs and future inputs, respectively. $L_w \in \mathbb{R}^{Ml \times M(l+m)}$ is the subspace matrix corresponding to the past input and output states, and $L_u \in \mathbb{R}^{Ml \times Mm}$ is the subspace matrix corresponding to the future inputs. Future output values in Equation (16), as well as the system's future input, can be formulated as a linear combination of the system's past input and output states. The system's behavior will then be described using Equation (16), rather than going back to the identification techniques that yield the traditional transfer function or state-space description of the system.

The following least squares problem is solved to calculate L_w and L_u from the Hankel matrices.

$$min \left\| Y_f - \begin{bmatrix} L_w & L_u \end{bmatrix} \begin{bmatrix} W_p \\ U_f \end{bmatrix} \right\|_F^2$$
(17)

This problem can be solved from the orthogonal projection of the row space of Y_f into the row space of the matrices $W_p = \begin{bmatrix} Y_p & U_p \end{bmatrix}^T$. This can be defined by Equation (18) as follows:

$$\hat{Y}_f = Y_f / \begin{bmatrix} W_p \\ U_f \end{bmatrix}$$
(18)

2.4. Data-Driven Predictive Controller

The data-driven predictive control algorithm uses the linear subspace predictor and the cost function of MPC algorithm.

M and *N* are lengths of data. Furthermore, $y_d(k+1)$, y(k+1), u(k-M) are the desired output *r*, output, and the input, respectively. All I/O data are stored in a database and then can be used again in control.

The MPC algorithm cost function form [29,30] can be written with the prediction and control horizon N_p and N_c equal to f as follow:

$$J = \left(\sum_{k_p=1}^{N_p} \left(\hat{Y}_{t+k_p} - r_{t+k_p} \right)^T W_Q \left(\hat{Y}_{t+k_p} - r_{t+k_p} \right) + \sum_{k_c=1}^{N_c} \Delta U_{t+k_c}^T W_R \Delta U_{t+k} \right)$$
(19)

where W_Q and W_R are the weight matrices, r_t is the reference signal at the current time t, N_p and N_c are the prediction and control horizon, respectively. N_c maybe less than or equal to the prediction horizon N_p ($N_c \leq N_p$ or $N_c \leq f$).

We maintain to improve the basics of DDPC via rewriting the cost function of MPC Equation (19) in quadratic form. Using Equation (16) and the reference signal of r_t , we can update the cost function as follows:

$$J = \left(L_w \Delta W_p + L_u^{N_c} \Delta U_{N_c} + Y_t - r_{t+1}\right)^T W_Q \times \left(L_w \Delta W_p + L_u^{N_c} \Delta U_{N_c} + Y_t - r_{t+1}\right) + \Delta U_{N_c}^T W_R \Delta U_{N_c}$$
(20)

If we solve the cost function, the control rule can be written as follows:

$$\Delta U_{N_c} = -\left(\left(L_u^{N_c}\right)^T W_Q\left(L_u^{N_c}\right) + W_R\right)^{-1} \times \left(L_u^{N_c}\right)^T W_Q\left(L_w \Delta W_p + (Y_t - r_{t+1})\right) \\ = -K_{\Delta W_p, N_c} \Delta W_p - K_{e, N_c}(Y_t - r_{t+1}).$$
(21)

where $-K_{\Delta W_p,N_c}$ and K_{e,N_c} are the weight of the past data vector and the weight of the tracking error, respectively.

At each time condition, only the first element of ΔU_{N_c} is used to calculate the control input U_{t+1} , which complies with ΔU_{t+1} . Hence, abbreviating the first *m* rows in Equation (21) gives as below:

$$\Delta U_{t+1} = -K_{\Delta W_p} \Delta W_p - K_e (Y_t - r_{t+1})$$
(22)

With,

$$K_{\Delta W_p} = \begin{bmatrix} I_m & 0_{m \times (M-1)m} \end{bmatrix} K_{\Delta W_p, N_c}$$
(23)

$$K_e = \begin{bmatrix} I_m & 0_{m \times (M-1)m} \end{bmatrix} K_{e,N_c}$$
(24)

where I_m is an identity matrix of size m while $0_{i \times j}$ is a zero matrix with i rows and j columns. Consequently, the control input U_t can be written as follows:

$$U_t = U_{t-1} + \Delta U_t \tag{25}$$

2.5. DDPC with Considering Constraints via Quadratic Programming

The ability of MPC and other predictive control algorithms to include constraints in the final control solution is one of their advantages. In this section, a constrained DDPC algorithm is provided, taking the constraints in Equation (26) into account.

$$F_{m}\Delta u_{min} \leq \Delta u_{Nc} \leq F_{m}\Delta u_{max}$$

$$F_{m}u_{min} \leq u_{Nc} \leq F_{m}u_{max}$$

$$F_{l}y_{min} \leq \Delta y_{f} \leq F_{l}\Delta y_{max}$$

$$F_{l}y_{min} \leq y_{f} \leq F_{l}y_{max}$$
(26)

Here F_m and F_l are defined as $F_m = \begin{bmatrix} I_m & I_m & \dots & I_m \end{bmatrix}^T \in \mathbb{R}^{N_c m \times m}$, $F_l = \begin{bmatrix} I_l & I_l & \dots & I_l \end{bmatrix}^T$ with identity matrices I_m and I_l . Further, $u_{Nc} = \begin{bmatrix} u_{t+1}^T & u_{t+2}^T & \dots & u_{t+N_c}^T \end{bmatrix}^T$.

The control signal is optimized in the constrained DDPC algorithm while taking into account the constraints placed on the cost function specified in the earlier sections. The inequalities shown in Equation (27) are reached when the constraints defined in Equation (26) are rewritten as a function of [2,20].

$$\underbrace{\begin{bmatrix} I_{N_{c}m} \\ -I_{N_{c}m} \\ \Gamma_{m} \\ -\Gamma_{m} \\ L_{u}^{N_{c}} \\ -L_{u}^{N_{c}} \\ \Gamma_{l}L_{u}^{N_{c}} \\ -\Gamma_{l}L_{u}^{N_{c}} \end{bmatrix}}_{A_{QP}} \Delta u_{Nc} \leq \underbrace{\begin{bmatrix} F_{m}\Delta u_{max} \\ -F_{m}\Delta u_{min} \\ F_{m}u_{max} - F_{m}u_{t} \\ -F_{m}u_{min} + F_{m}u_{t} \\ F_{l}\Delta y_{max} - L_{w}\Delta w_{p} \\ -F_{l}\Delta y_{min} + L_{w}\Delta w_{p} \\ F_{l}y_{max} - F_{l}y_{t} - \Gamma_{l}L_{w}\Delta w_{p} \\ -F_{l}y_{min} + F_{l}y_{t} + \Gamma_{l}L_{w}\Delta w_{p} \end{bmatrix}}$$
(27)

It can be discovered by optimizing the cost function for the DDPC algorithm given in Equation (28) with constraints.

$$\min_{\Delta u_{Nc}} \frac{1}{2} \Delta u_{Nc}^T H \Delta u_f + \Delta u_{Nc}^T f \ s.t. \ A_{QP} \Delta u_{Nc} \le b_{QP}$$
(28)

The quadratic programming (QP) algorithm can be used to solve this optimization process. The QP algorithm determines the ideal control signal Δu_{Nc} while accounting for the constraints.

2.6. Experimental Setup

For the execution of the parameter estimation algorithms, the Matlab/Simulink environment is used. STM32F4107 CPU was used to run control algorithms. The dual-channel H bridge L298 driver IC was used for DC motors and a 9V Li-Po battery was used as the voltage supply for the motor. The Matlab/Simulink environment was also utilized during the studies to gather and store data. Communication is established between the microprocessor and the Matlab/Simulink environment with the UART protocol. In each experiment, control signals or reference position values were sent from the Matlab/Simulink environment to the CPU using the serial communication protocol. The employed experimental setup is shown in Figure 3.



Figure 3. Experimental setup.

The data-driven predictive control method was validated on a real-time system by different experiments. The system position was directly measured using a linear potentiometer.

2.7. Passive and Active Rehabilitation

The design of both active and passive rehabilitation tasks requires estimation or measurement of the external force exerting on the exoskeleton's endpoint. In this study, a micro load cell is placed between end point of the linear actuator shaft and fork joint to measure external force (F_{ex}). External force can be used to stimulate a virtual mass-spring-damper system, as shown in Figure 4 and its mechanical parameters can be adjusted depending on the type and degree of rehabilitation required.



Figure 4. Virtual mass-spring-damper system.

When the measured external force is applied to the virtual system using Equation (29), the virtual system's response x_d , can be found.

$$m_d \ddot{x}_d + b_d \dot{x}_d + k_d x_d = F_{ex} \tag{29}$$

For active rehabilitation task, the virtual system's position response, x_d , can be used to deviate to the desired reference from the actuator's actual position, x_1 (Equation (30)). In this instance, the behavior of the controller is a regulator and keeps the actuator in its actual stroke position. If the external force is greater than zero, the desired reference is different from the actual position of the actuator.

$$x_r' = x_1 - x_d \tag{30}$$

For passive rehabilitation, x_d can be used to create a deviation from the predefined trajectory x_r , as shown in Equation (31), and the controller can make use of this desired reference.

$$x'_r = x_r - x_d \tag{31}$$

It is possible to decide how the virtual mass-spring-damper system responds to external force and carries out passive, active, or assistive rehabilitation tasks by setting the parameters within the acceptable range of values.

3. Results and Discussion

3.1. Experimental Results of Subspace Prediction Algorithm

During the tests, the robot was not subjected to any external force and the exoskeleton is tested on a healthy human hand. The predicted model was compared with the state-space

model obtained by the output error method; those applications and results are explained in past studies [23]. By using the model horizon parameters p = 30, f = 10, the subspace estimation model parameters (Lu_r and Lw_r) are calculated. The Lu and Lw obtained with the same values as the previous p and f from the system are then calculated using the u_t input signal to be tested, the Lu and Lw are then calculated using the same values as the previous p and f from the system (Figure 5. In the graphics, the results calculated using the reference (Lu_r and Lw_r) models and the actual Lu and Lw models are compared.



Figure 5. Subspace prediction validation procedure.

The output error-based model (OEbM) output is compared with the sub-space model (SPbM) estimation result obtained with the *u* input signal ($u = A_0 sin\omega_0 t$, $\omega_0 = 0.5$ rd/sn ve $A_0 = 5$) shown in the Figure 6. The linear actuator's stroke length, which is considered the system output, is represented by the *y*-axis on the graph as position.



Figure 6. Selection of the sinus input signal $\omega_0 = 0.5$ rd/s and $A_0 = 5$, and comparison of results. (a) Single sinus signal, (b) Comparison of results.

The percentage error function in Equation (32) was obtained between the OEbM output, which was used as the reference model, and the SPbM output to evaluate the outcomes of the model estimate test, and this number was chosen as the performance criterion.

$$e = \frac{\sum_{i=1}^{n} (y_{OEbM} - y_{SPbM}/y_{OEbM})}{n}$$
(32)

This experiment's error value (e) was calculated as 1.1871%. Table 1 provides the percentage error values of the test results with additional specified input signals.

Experiments	$\boldsymbol{\omega}_{0}\left(\mathbf{rd/s} ight)$	A_0	Error (e)
1	0.3	5	0.0187
2	0.4	5	0.0266
3	0.5	5	0.0337
4	0.6	5	0.0395
5	0.7	5	0.0727
6	0.8	5	0.0825
7	1	5	0.1431
8	2	5	0.9273

Table 1. Experiments with a single frequency input signal ($u = A_0 sin \omega_0 t$).

The range of $\omega_0 = 0.3$ rd/s, where the input signal runs throughout the full stroke in a single alternate, and $\omega_0 = 2$ rad/s, which gives 5% displacement on the motor stroke, is tested using a single frequency component. The first eight experiments revealed a linear correlation between the modeling success and the parameters of the frequency component of the input signal. As a result, success increased and error decreased at higher frequency values, while success increased at lower frequency values. Since the full stroke length could not be studied in the experiments with high frequency *u* input signals, it can be argued that all the system's characteristics could not be apprehended in them. This has an impact on modeling success.

The following experiments are performed with a sinusoidal input signal with two separate frequency components ($u = A_0 \sin \omega_0 t + A_1 \sin \omega_1 t$). The experiment that was performed using input signals with $\omega_0 = 0.2$ rd/s and $A_0 = 2$, $\omega_1 = 0.8$ rd/s and $A_1 = 3$ (Experiment 10) results are shown in Figure 7.



Figure 7. Results of experiment 10: (a) input and output signal, (b) results.

It was noted that in experiments using the u input signal with two frequency components, the average success rate increased. When experiments 9 through 12 in Table 2 are examined, it becomes clear that the error function produces results that are similar across the ranges of experimental parameters. It was noted that the error function is negatively impacted by the separation between the two component frequencies.

Table 2. Experiments with a two-frequency *u* input signal ($u = A_0 sin\omega_0 t + A_1 sin\omega_1 t$).

Experiment	$\pmb{\omega}_0\left(\mathbf{rd/s} ight)$	$\boldsymbol{\omega}_1 ~(\mathbf{rd/s})$	A_0	A_1	Error (e)
9	0.3	0.4	2	3	0.0127
10	0.2	0.8	2	3	0.0267
11	0.2	0.8	1	4	0.0360
12	0.5	0.1	2	2	0.0235

The following experiments (13–16) use *u* input signals that have three distinct frequency components ($u = A_0 sin\omega_0 t + A_1 sin\omega_1 t + A_2 sin\omega_2 t$), as shown in Figure 8. The



experiments conducted are more useful than the earlier experiments, as shown in Table 3. Different frequency components are found to increase the success of modeling.

Figure 8. Results of experiment 10: (a) input and output signals, (b) results.

Table 3. Experiments with a three-frequency u input signal ($u = A_0 sin\omega_0 t + A_1 sin\omega_1 t + A_2 sin\omega_2 t$).

Experiment	$\boldsymbol{\omega}_0 \; (\mathbf{rd/s})$	$\pmb{\omega}_1 \; (\mathbf{rd/s})$	$\omega_2 \ (rd/s)$	A_0	A_1	A_2	Error
13	0.3	0.4	0.7	2	2	2	0.0088
14	0.3	0.4	0.1	2	2	2	0.0239
15	0.3	0.4	0.1	2	2	5	0.0251
16	0.5	0.4	0.1	2	2	5	0.0211

The following tests were conducted using input signals that contained a scanning frequency as shown in Figure 9. These input signals are configurations for input signals that begin with low-frequency components and increase throughout the designated range. The average success is higher in these experiments, as shown in Table 4. This input signal, which has components in several different frequency ranges, helps to clarify the system's characteristics.





Figure 9. Results of experiment 20: (a) input and output signal, (b) results.

Table 4.	Experiments	s with inpu	t signals †	that configure	ed by	scanning	frequency

Experiment	k_0	A_0	e (Error)
17	0.003	4	0.0149
18	0.005	5	0.0200
19	0.01	5	0.0083
20	0.01	3	0.0048

3.2. Experimental Results of Data-Driven Predictive Controller

The predicted model parameters and control parameters *Ke* and $K\Delta w p^{1x2p}$ are calculated. The step function and a sinusoidal trajectory response (2 π radians from position 0

to 50%) are examined to analyze the parameters affecting the control algorithm, and the results are discussed.

3.2.1. Effect of Data Length on Control Success

Figure 10 shows the relationship between the calculated control coefficients and the control success as a function of the data size of the input signal (u), which is used as the estimation input signal. In experiments, only the input signal u's data length is changed, while the other parameters, p = 30, f = 10, Q = 1, R = 2, and Nc = 5, remain constant. As a result, it was found that as the number of data increases, overshoot decreases, increasing the success of the control. The control coefficients from experiments with the same frequency components but fewer data are seen to negatively affect the success as the number of data decreases. The overshoot increases as N decreases. The quantity of data has no meaningful effect on the rise time.



Figure 10. Step response according to u signals of different N data lengths.

3.2.2. Analysis of Model Horizon Parameters (p, f)

The system state equations along the chosen horizon are the basis of the subspacebased estimation model. Using the input signal from the most successful experiment (Experiment 20) from the previous test results, the test results referring to the horizon parameter success characteristic are presented in this section.

As a result of parameter estimation using various historical model horizon values, the control parameters are displayed in Table 5. Experiments are carried out with constant parameters f = 10, Q = 1, R = 2, and Nc = 5 by varying the model horizon p between 30 and 60. The effect of the previous model horizon on the system response is investigated in these experiments. Rise time and percent overshoot are used as performance factors in step function experiments. The success factor in trajectory tracking is the mean of the squares of the errors. Table 5 details all these values.

Model Horizon (v)	Ke	Mean of KAzum ^{1x2p}	Standard Deviation	Response of Step Function		Tracking Performance
$\operatorname{Hom}(\varphi)$		$OI K \Delta w p$	$OI K\Delta w p$	Rising Time (s)	Overshoot %	mse
30	0.1908	0.0730	0.2499	3.88	0.17	4.3138
40	0.2853	0.0852	0.2493	5.77	0.20	18.4100
50	0.3746	0.0904	0.2399	3.84	0.18	1.4763
60	0.3780	0.1045	0.2202	* None	* None	16.9600

Table 5. Control parameters calculated using *p*.

* In this experiment, no rise or overshoot was observed during the experiment.

Without considering the experiment taken as p = 50 in the table, it can be said that the other values show that the step response and trajectory tracking success are negatively

impacted by an increase in the *p* value as shown in Figure 11. However, in the experiment where *p* is set at 50, success increases once more. The control is seen to be totally broken in the following experiment. This assumes that there might be a linear relationship between control success and the past modeling horizon, *p*. When the response of the system to the sine waveform is analyzed, it is seen that even the worst result (*p* = 60) follows the trajectory with a certain error (mse = 16.96 from Table 5).



Figure 11. Success of different past model horizon (*p*) values in tracking trajectories and success of response of step function.

In the following experiments, the past model horizon p, and other parameters (p = 30, Q = 1, R = 2, and Nc = 5) are held constant to examine the control success of the future model horizon f. Rise time and percent overshoot are considered performance factors in experiments using the step function. The average of the squares of the errors is used as a performance metric for trajectory tracking success. In Table 6, each of these values is described in detail.

Table 6. Control parameters defined using *f*.

Future	Ke	Mean	Standard Deviation of $KA_{ava}^{1x^{2}p}$	Response of S	Step Function	Tracking Performance
Horizon ()		$OI K \Delta w p$	$OI K\Delta w p$	Rising Time (s)	mse	
5	0.0476	0,0066	0.0222	3.38	0.0962	9.0828
10	0.1908	0.0733	0.2400	3.85	0.0322	4.3138
15	0.2191	0.0996	0.2946	3.94	0.0220	1.2027
20	0.2385	0.1580	0.3942	4.36	0.0018	3.9737
25	0.1688	0.1188	0.2980	4.48	0.0254	10.5345
50	0.1951	0.1897	0.4622	6.91	0.0524	60.1988

When the tables and graphics are examined, it is observed that there is a nonlinear relationship between the model future horizon f and control success. The overshoot and rise times are close to each other in experiments where the f value of the future horizon is between 20 and 25. The increase in the mean value and standard deviation of Ke and $K\Delta wp$ has a negative effect on the rise time. The rise time increases as the f parameter increases, but the average overshoot decreases. The response of trajectory also shows that the overshoot is high in the experiment where f is set to 5 (Figure 12). This overshoot is a result of the *Ke* value obtained in this experiment being much lower than in the other experiments. Based on gathered data, it is seen that, even if the number of future horizons is chosen as only 5, the steady-state stability is considered as critically stable.



Figure 12. Effect of different future model horizon (*f*) values on the success of tracking the step response and trajectory.

The tables display control coefficients based on the parameters used in the experiments. The graph in Figure 13 shows the control parameters (*Ke*) for all p and f values in the determined range (5–100). When the *Ke* value is between 0.2 and 0.3, it is apparent from the experiments, the results of which are evaluated, that the success rate is high. Additionally, the tables display the mean and standard deviation values of the $K\Delta wp$ parameter calculated in successful results. Figures 13 and 14 can be used to analyze the suitable numeric selections of p and f parameters for these values. The color scale of the p and f pairs providing successful *Ke* values is shown in Figure 13. It is clear from all the graphs that the values of p and f for the successful control task can be selected from a range of 5 to 80 and 20 to 60, respectively.



Figure 13. *Ke* and $K\Delta wp$ calculated for the 5–100 range of *p* and *f* values.



Figure 14. The standard deviation value of $K \Delta w p$ calculated for the 5–100 range of p and f values.

3.3. The Effect of Q and R Parameters on the Control System Succession

It is shown in equation 16 that the parameters Q and R in the general MPC cost function are the weights of the reference error and the differential control signal, respectively. These weights have a direct impact on the success of the control because the data-driven predictive control rule is based on the MPC cost function. The success of trajectory tracking and step response are examined using the experiments in this section with the Q and R parameters. With p = 30, f = 10, and Nc = 5, various Q and R parameters are tested in experiments.

The *Q* parameter and the rise time have a linear relationship, as can be seen when Table 7 is examined. In addition, the ratio of the *Q* parameter to *R* also affects the rise time. It is clearly seen that the *R* parameter affects the overshoot as seen in Figure 15. The overshoot increased up to a maximum value of 8% in the experiment where *R* was set to be 50. These tests led to the conclusion that selecting a small *Q* value would have a positive effect on the rise time as shown in Figure 16. The experiments also confirmed that selecting a low value (*Q* = 1) and choosing the parameter *R* within the optimal ranges (10,200) have a positive impact on the response of step and trajectory tracking.

Table 7. Effect of *Q* and *R* parameters on control response.

Q R	R Ke	Mean $f K A \pi u m^{1} x^{2} p$	Standard Deviation	Response of St	Tracking Performance		
			or Kaup	of Kaup	Rising Time (s)	Overshoot %	mse
5	2	0.7694	0.1969	0.8251	* None	* None	288.1818
4	2	0.5693	0.1513	0.6184	5.654	0.0385	43.5554
3	2	0.4975	0.1354	0.5512	6.631	0.0084	65.2003
2	2	0.4329	0.1202	0.4776	3.093	0.0688	33.0765
1	2	0.2589	0.0806	0.2981	3.031	0.0050	11.8783
1	3	0.1897	0.0648	0.2269	3.528	0.0118	6.9044
1	5	0.1293	0.0510	0.1649	3.189	0.0286	3.7065
1	10	0.0807	0.0398	0.1152	3.493	0.0490	2.7420
1	20	0.0549	0.0337	0.0889	3.902	0.0626	2.4434

* In this experiment, no rise or overshoot was observed during the experiment.



Figure 15. Effect of the *R* parameter on step and trajectory response.

The control horizon *Nc* in the data-driven predictive control algorithm is described in Equation (16). Unlike *f*, which is the model horizon, *Nc* specifies the size of the space from which the system control signal u_c should be calculated. It is used by the control algorithm to compute the system control signal u_{Nc} , which is the first component of Δu_{Nc} along *Nc*. These tests were conducted with the following parameters: p = 30, f = 10, Q = 1, and R = 2. The impact of *Nc* on step function response and trajectory tracking response is shown in Table 8 and Figure 17. It is clear from the tables and graphs that the rise time has a linear relationship with *Nc*'s trajectory tracking response and step function response.



Figure 16. Effect of the *Q* parameter on step and trajectory response.

Table 8. Effect of *Nc* parameter on control response.

Nc	Ke	Mean of KAzum ^{1x2p}	Standard Deviation of $K \Lambda z u n^{1 \times 2 p}$	Response of S	Step Function	Tracking Performance
		of KBup	$OI K \Delta w p$	Rising Time (s)	Overshoot %	mse
5	0.1688	0.1188	0.2980	3.232	0.0254	10.5345
10	0.1799	0.0798	0.2429	3.224	0.0152	8.5312
15	0.2220	0.0768	0.2669	3.210	0.0084	3.9609
20	0.2589	0.0806	0.2981	3.180	0.0084	11.8783
25	0.3423	0.0870	0.3562	3.192	0.0684	17.3509



Figure 17. According to Nc, the response of step function and trajectory.

3.4. Passive and Active Rehabilitation

The patient is completely passive during passive rehabilitation tasks. The patient's hand is completely guided by the exoskeleton. The patient did not exert any force on the exoskeleton for the first 10 s, as shown in Figure 18. The controller in this case uses $x'_r = x_d$. When a maximum counterforce of 7–8 N was applied in the final 10th second, the virtual system response x_d deviates from the reference x'_r . This makes it possible for the system to react to the patient in accordance with any unexpected issues that might arise on the patient's finger. The response stiffness of the system to the patient's hand can be adjusted by appropriately adjusting the parameters of the virtual mechanical system as shown in Equation (26).



Figure 18. Passive rehabilitation task.

The controller functions as a regulator during the active rehabilitation process and tries to keep the current position of the actuator stroke. The reference shifted, as shown in Figure 19, because of the patient applying an external force of about -6N in the 12th second. The patient made the flexion movement with a stable force applied to the system. The extension movement was carried out by exerting force in the opposite direction after the 14th second. The system can be operated at greater forces by modifying the virtual mechanical system's parameters. The system can now perform in resistive mode as a result. The system operates in assistive mode by taking x_d in negative.



Figure 19. Active rehabilitation task.

The study of the robustness of MPC can be approached in a variety of ways. The first focuses on the closed-loop systems' robustness when created utilizing the nominal system (i.e., neglecting uncertainty). The second makes an effort to achieve robustness within the context of conventional model predictive control by taking into account all feasible realizations of the uncertainty. The third approach addresses this by introducing feedback in the min–max optimal control problem solved online [31]. In this study, according to the results obtained in some experiments, it was evaluated that the system is robust against uncertainties. For example, in the analysis of model horizon parameter experiments, it was observed that the system followed the trajectory even at the worst coefficients (p = 60, f = 5). The model control values, *Ke* and *K*\Delta*wp*, are determined using this model and calculated as optimum values, showing that the controller is robust enough to handle uncertainties.

The subspace prediction estimation results to be made with the ongoing past data online will guarantee that the controller operates feasibly and continuously during the rehabilitation tasks. Along with model parameters calculated throughout a specific horizon with sub-space prediction, the model includes all model uncertainties associated with the patient's exoskeleton and measurement noises related to force and position. The use of data-driven estimation methods and model-free control algorithms will improve the effectiveness of studies with patients who can be evaluated in a wide range of spectrums as opposed to computing the model of a biomechanical structure, such as the human hand, with conventional approximate methods.

4. Conclusions

In this work, a data-driven predictive control is developed for a hand exoskeleton robot used for rehabilitation. The designed control rule was used in a set of experiments, and the results were presented. The experiments are intended to examine how the parameters affecting the suggested control algorithm influence the success of the control. A data-driven predictive control algorithm is optimization-based and certain constraints are added to the problem during the optimization process; it then suggests the best solution within the boundaries set by those constraints. The rehabilitation process aims to regain the patient's lost mobility by having them perform repeated exercises that are suited to their situation.

In our experiments to evaluate the performance of the proposed control algorithm, data length is investigated for subspace prediction, and it is expressed that, within a certain range, data length was linearly related to modeling success. On the success of controlling for the DDPC rule, the effects of the p, f, Nc, Q, and R parameters are examined and discussed separately. When all the test results are evaluated, we can conclude that the suggested solution is suitable for rehabilitation processes because it offers the best solutions, while still considering some limitations.

It was shown that the exoskeleton controller operates in passive, active, and assistive modes with the benefit of an auxiliary reference created using the measured external force.

Author Contributions: Conceptualization, E.K. and G.A.; methodology, G.A.; software, G.A.; validation, E.K. and G.A.; formal analysis, E.K.; investigation, G.A.; resources, E.K.; data curation, E.K.; writing—original draft preparation, G.A.; writing—review and editing, E.K.; visualization, G.A.; supervision, E.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Ethical review and approval were waived for this study due to no applicable.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

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

References

- 1. Yue, Z.; Zhang, X.; Wang, J. Hand rehabilitation robotics on poststroke motor recovery. *Behav. Neurol.* **2017**, 2017, 3908135. [CrossRef] [PubMed]
- Akgun, G.; Cetin, A.E.; Kaplanoglu, E. Exoskeleton design and adaptive compliance control for hand rehabilitation. *Trans. Inst. Meas. Control* 2020, 42, 493–502. [CrossRef]
- 3. Du Plessis, T.; Djouani, K.; Oosthuizen, C. A review of active hand exoskeletons for rehabilitation and assistance. *Robotics* **2021**, 10, 40. [CrossRef]
- 4. Reitan, I.; Dahlin, L.B.; Rosberg, H.E. Patient-reported quality of life and hand disability in elderly patients after a traumatic hand injury—A retrospective study. *Health Qual. Life Outcomes* **2019**, *17*, 148. [CrossRef]
- 5. Dovat, L.; Lambercy, O.; Gassert, R.; Maeder, T.; Milner, T.; Leong, T.C.; Burdet, E. HandCARE: A cable-actuated rehabilitation system to train hand function after stroke. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2008**, *16*, 582–591. [CrossRef]
- Kabir, R.; Sunny, M.S.H.; Ahmed, H.U.; Rahman, M.H. Hand Rehabilitation Devices: A Comprehensive Systematic Review. *Micromachines* 2022, 13, 1033. [CrossRef]
- Perry, J.C.; Trimble, S.; Machado, L.G.C.; Schroeder, J.S.; Belloso, A.; Rodriguez-de-Pablo, C.; Keller, T. Design of a spring-assisted exoskeleton module for wrist and hand rehabilitation. In Proceedings of the 2016 38th Annual International Conference, Orlando, FL, USA, 16–20 August 2016.
- Tsoi, Y.H.; Xie, S.Q. Impedance control of ankle rehabilitation robot. In Proceedings of the 2008 IEEE International Conference on Robotics and Biomimetics, Bangkok, Thailand, 22–25 February 2009; pp. 840–845.
- 9. Li, X.; Liu, Y.H.; Yu, H. Iterative learning impedance control for rehabilitation robots driven by series elastic actuators. *Automatica* **2018**, *90*, 1–7. [CrossRef]
- Culmer, P.R.; Jackson, A.E.; Makower, S.; Richardson, R.; Cozens, J.A.; Levesley, M.C.; Bhakta, B.B. A control strategy for upper limb robotic rehabilitation with a dual robot system. *IEEE/ASME Trans. Mechatron.* 2009, 15, 575–585. [CrossRef]

- 11. Jafarov, E.M.; Parlakçi, M.N.A.; Istefanopulos, Y. A new variable structure PID-controller design for robot manipulators. *IEEE Trans. Control Syst. Technol.* 2004, 13, 122–130. [CrossRef]
- 12. Mo, L.; Feng, P.; Shao, Y.; Shi, D.; Ju, L.; Zhang, W.; Ding, X. Anti-Disturbance Sliding Mode Control of a Novel Variable Stiffness Actuator for the Rehabilitation of Neurologically Disabled Patients. *Front. Robot. AI* **2022**, *9*, 9. [CrossRef]
- Riani, A.; Madani, T.; Benallegue, A.; Djouani, K. Adaptive integral terminal sliding mode control for upper-limb rehabilitation exoskeleton. *Control Eng. Pract.* 2018, 75, 108–117. [CrossRef]
- 14. Chen, Z.; Li, Z.; Chen, C.P. Disturbance observer-based fuzzy control of uncertain MIMO mechanical systems with input nonlinearities and its application to robotic exoskeleton. *IEEE Trans. Cybern.* **2016**, *47*, 984–994. [CrossRef]
- 15. Rahmani, M.; Rahman, M.H. An upper-limb exoskeleton robot control using a novel fast fuzzy sliding mode control. *J. Intell. Fuzzy Syst.* **2019**, *36*, 2581–2592. [CrossRef]
- 16. Yu, H.; Huang, S.; Chen, G.; Thakor, N. Control design of a novel compliant actuator for rehabilitation robots. *Mechatronics* **2013**, 23, 1072–1083. [CrossRef]
- 17. Chen, J.; Huang, Y.; Guo, X.; Zhou, S.; Jia, L. Parameter identification and adaptive compliant control of rehabilitation exoskeleton based on multiple sensors. *Measurement* **2020**, *159*, 107765. [CrossRef]
- 18. Ulkir, O.; Akgun, G.; Nasab, A.; Kaplanoglu, E. Data-Driven Predictive Control of a Pneumatic Ankle Foot Orthosis. *Adv. Electr. Comput. Eng.* **2021**, *21*, 65–74. [CrossRef]
- 19. Maciejowski, J.M. Predictive Control: With Constraints; Pearson Education: London, UK, 2002.
- 20. Mardi, N.A. Data-Driven Subspace-Based Model Predictive Control. Doctoral Dissertation, RMIT University, Melbourne, Australia, 2010.
- Ulkir, O.; Akgun, G.A.Z.İ.; Kaplanoglu, E. Real-time implementation of data-driven predictive controller for an artificial muscle. Stud. Inform. Control 2019, 28, 189–200. [CrossRef]
- 22. Rossiter, J.A. Model-Based Predictive Control: A Practical Approach; CRC Press: Boca Raton, FL, USA, 2017.
- 23. Akgün, G. Data Driven Predictive Control of Exoskeleton for Rehabilitation. Doctoral Dissertation, Marmara Universitesi, Istanbul, Turkey, 2019. (In Turkish).
- 24. Akgun, G.; Kaplanoglu, E.; Cetin, A.E.; Ulkir, O. Mechanical design of exoskeleton for hand therapeutic rehabilitation. *Quest J. J. Res. Mech. Eng.* **2018**, *4*, 9–17.
- 25. Van Overschee, P.; De Moor, B. Subspace Identification for Linear Systems: Theory—Implementation—Applications; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2012.
- Kadali, R.; Huang, B.; Rossiter, A. A data driven subspace approach to predictive controller design. *Control Eng. Pract.* 2003, 11, 261–278. [CrossRef]
- 27. González, A.H.; Ferramosca, A.; Bustos, G.A.; Marchetti, J.L.; Fiacchini, M.; Odloak, D. Model predictive control suitable for closed-loop re-identification. *Syst. Control Lett.* **2014**, *69*, 23–33. [CrossRef]
- Meidanshahi, V.; Corbett, B.; Adams, T.A., II; Mhaskar, P. Subspace model identification and model predictive control based cost analysis of a semicontinuous distillation process. *Comput. Chem. Eng.* 2017, 103, 39–57. [CrossRef]
- Venkat, A.N.; Hiskens, I.A.; Rawlings, J.B.; Wright, S.J. Distributed MPC strategies with application to power system automatic generation control. *IEEE Trans. Control Syst. Technol.* 2008, 16, 1192–1206. [CrossRef]
- 30. Kaplanoğlu, E.; Arsan, T.; Varol, H.S. Predictive control of a constrained pressure and level system. *Turk. J. Electr. Eng. Comput. Sci.* **2015**, *23*, 641–653. [CrossRef]
- Mayne, D.Q.; Rawlings, J.B.; Rao, C.V.; Scokaert, P.O. Constrained model predictive control: Stability and optimality. *Automatica* 2000, 36, 789–814. [CrossRef]