energies

Article

Performance Analysis of Mars-Powered Descent-Based Landing in a Constrained Optimization Control Framework

Adnan Khalid [1,†], Mujtaba Hussain Jaffery [2], Muhammad Yaqoob Javed [2], Adnan Yousaf [3], Jehangir Arshad [2,*,†], Ateeq Ur Rehman [4,*], Aun Haider [1], Maha M. Althobaiti [5], Muhammad Shafiq [6,*] and Habib Hamam [7,8,9]

[1] Department of Electrical Engineering, Sialkot Campus, University of Management and Technology Lahore, Sialkot 51310, Pakistan; adnan.khalid@skt.umt.edu.pk (A.K.); aun.haider@skt.umt.edu.pk (A.H.)
[2] Electrical and Computer Engineering Department, COMSATS University Islamabad, Lahore 54000, Pakistan; m.jaffery@cuilahore.edu.pk (M.H.J.); yaqoob.javed@cuilahore.edu.pk (M.Y.J.)
[3] Department of Electrical Engineering, Superior University, Lahore 54000, Pakistan; adnan.yousaf@superior.edu.pk
[4] Department of Electrical Engineering, Government College University, Lahore 54000, Pakistan
[5] Department of Computer Science, College of Computing and Information Technology, Taif University, Taif 21944, Saudi Arabia; Maha_m@tu.edu.sa
[6] Department of Information and Communication Engineering, Yeungnam University, Gyeongsan 38541, Korea
[7] Faculty of Engineering, Université de Moncton, Moncton, NB E1A3E9, Canada; habib.hamam@umoncton.ca or spectrumcenter21@gmail.com
[8] Spectrum of Knowledge Production & Skills Development, Sfax 3027, Tunisia
[9] School of Electrical Engineering, Department of Electrical and Electronic Engineering Science, University of Johannesburg, Johannesburg 2006, South Africa
* Correspondence: jehangirarshad@cuilahore.edu.pk (J.A.); ateqrehman@gmail.com (A.U.R.); shafiq@ynu.ac.kr (M.S.); Tel.: +92-321-4141912 (J.A.)
† These authors contributed equally to this work.

Abstract: It is imperative to find new places other than Earth for the survival of human beings. Mars could be the alternative to Earth in the future for us to live. In this context, many missions have been performed to examine the planet Mars. For such missions, planetary precision landing is a major challenge for the precise landing on Mars. Mars landing consists of different phases (hypersonic entry, parachute descent, terminal descent comprising gravity turn, and powered descent). However, the focus of this work is the powered descent phase of landing. Firstly, the main objective of this study is to minimize the landing error during the powered descend landing phase. The second objective involves constrained optimization in a predictive control framework for landing at non-cooperative sites. Different control algorithms like PID and LQR have been developed for the stated problem; however, the predictive control algorithm with constraint handling's ability has not been explored much. This research discusses the Model Predictive Control algorithm for the powered descent phase of landing. Model Predictive Control (MPC) considers input/output constraints in the calculation of the control law and thus it is very useful for the stated problem as shown in the results. The main novelty of this work is the implementation of Explicit MPC, which gives comparatively less computational time than MPC. A comparison is done among MPC variants in terms of feasibility, constraints handling, and computational time. Moreover, other conventional control algorithms like PID and LQR are compared with the proposed predictive algorithm. These control algorithms are implemented on quadrotor UAV (which emulates the dynamics of a planetary lander) to verify the feasibility through simulations in MATLAB.

Keywords: Mars landing; explicit model predictive control; unmanned aerial vehicle (UAV); powered descent

check for updates

Citation: Khalid, A.; Jaffery, M.H.; Javed, M.Y.; Yousaf, A.; Arshad, J.; Ur Rehman, A.; Haider, A.; Althobaiti, M.M.; Shafiq, M.; Hamam, H. Performance Analysis of Mars-Powered Descent-Based Landing in a Constrained Optimization Control Framework. *Energies* **2021**, *14*, 8493. https://doi.org/10.3390/en14248493

Academic Editor: Muhammad Naveed Iqbal

Received: 11 November 2021
Accepted: 9 December 2021
Published: 16 December 2021

1. Introduction

Space exploration has been one of the main research areas of science for the last few decades. With the increase in the space exploration race, the observation of different planets in space has gained popularity. The nearest neighbor to Earth is Mars, which has attracted human beings more than any other planet. In this context, Mars landing, and exploration missions started in the 1970s. Most of them failed whereas few ended successfully. Mars landing sites for initial missions had high altitudes whereas Mars missions in the future will land at low altitudes. Thus, more precise landing technology is required for future missions. Therefore, this research would be very helpful to present predictive control algorithms for future Mars landings. Every single mission to Mars has these four stages: launch from Earth; the cruise stage, which takes about 6 to 9 months; entry, descent, and landing (EDL); and then finally, the surface missions start. EDL is a crucial stage as it is responsible for the success of the complete mission. It has three phases: the entry phase, parachute phase, and terminal descent phase. The descent phase is further divided into two parts: gravity turn and the powered descent. The focus of this work is the powered descent-based landing of Mars.

- Firstly, the major objective of this study is to minimize landing error during the powered descent phase.
- The second objective involves constrained optimization in a predictive control framework for landing at non-cooperative sites.
- In this work, the landing dynamics of the planetary lander are emulated by a vertical takeoff and landing (VTOL) unmanned aerial vehicle (UAV).

This work is focused on developing a constraint optimization predictive control algorithms for powered descent-based Mars landing. The proposed control laws are validated on an Earth-based testbed in a simulation environment. The testbed used for this purpose is the quadrotor UAV. Due to its dynamic nature, the quadrotor can emulate a planetary lander. It is used to perform landings similar to the Mars-powered descent. During the landing phase, the lander comes across obstacles and hazards. So, there is a need to implement such a control algorithm that can predict such hazardous constraints and generate optimal solutions such that the constraints are not violated. Therefore, MPC is good for use under such conditions as it can predict future output constraints in advance and can also respect input constraints. However, there are a few issues while implementing MPC, including its computation time and feasibility. MPC can take a significant amount of time to generate an optimal solution at every sampling time. As an alternative to this problem, another predictive algorithm is proposed in this work. This alternative predictive control algorithm is known as Explicit MPC (EMPC).

- The MPC and it is variant, as stated above, are implemented on quadrotor UAVs for powered descent.
- The proposed explicit predictive control algorithm is explored and compared with a simple predictive algorithm for Mars landing, which is the main novelty of this work.
- PID and LQR control algorithms are also implemented for the powered descent landing phase to obtain a comparative analysis with predictive control algorithms.

Like guidance and navigation, control is also important for the safe landing of planetary landers. Different control algorithms have been explored for Mars EDL missions. They have been used for trajectory generation and trajectory tracking as well. Similarly, in [1], a tracker-based reference control was designed. At first, a reference model was planned offline. The algorithm mentioned above with a compensator was used to track the desired profiles. The feasibility and robust behavior of the algorithm were verified using simulations. In [2], an active disturbance compensation algorithm is given. ADRL is an emerging control algorithm that is independent of the accurate model of the system. Simulations were performed based on the MSL case study. The results were compared with PID, which showed that ADRL is better for tracking problems as compared to PID. Whereas in [3], a terminal Sliding Mode Control (SMC) algorithm is proposed to track the desired trajectories. The stability of the stated control algorithm was verified through the Lyapunov-based approach. A state observer was used in this work for estimating the uncertainty. These two approaches combined converge the system states to the reference trajectory.

Similarly, in [4], the powered descent phase of Mars is focused on. An adaptive neural network approach is used to precisely track the desired reference. Using Lyapunov stability, it is proved that the errors for the presented algorithm are uniformly bound. The approach was good at following the desired trajectories and was robust against wind disturbances and uncertainties. In [5], a detailed overview of Mars guidance and control technologies is given. All the past technologies used for such missions are analyzed and future opportunities are discussed briefly. The primary goal of this work was the atmospheric entry of Mars. An active fault-tolerant control method for Mars entry based on a neural network is proposed. At first, an online neural network is used to for fault detection and isolation. Secondly, a neural network-based PID controller is implemented for fault-tolerant Mars entry. The conventional PID control algorithm is switched to the neural network-based PID controller whenever a fault is detected. The effectiveness of the proposed algorithm was verified by computer simulation.

Recently, in [6], a barrier Lyapunov function (BLF)-based SMC is presented to overcome the Mars landing problem. With the combination of the novel logarithm-type BLF, the SMC proved stable in the presence of external disturbance and input saturation. The MSL Mission-based Mars entry scenario was used for the simulation to verify the verification. The results proved the effectiveness of the BLF-based terminal SMC design. Whereas in [7], a tracking control scheme is presented for Mars landing under uncertainty. At first, a novel fixed-time nonsingular terminal sliding mode (FTNTSM) surface is developed, which gives fast convergence. Secondly, based on the FTNTSM surface, a continuous adaptive fixed-time nonsingular terminal sliding mode control (AFTNTSMC) method is designed. Under this control scheme, fixed-time convergence of the tracking error is achieved. The effectiveness of the proposed control algorithm was verified through numerical simulations. Furthermore, in [8], a deep reinforcement learning-based guidance and control method are proposed for Six Degree-of-Freedom (DOF) planetary landing. Simulation was performed, which showed the effectiveness of the proposed guidance and control methods for the stated problem. It showed a robust performance in the presence of noise and parametric uncertainty. Similarly, in [9], an integrated estimation and control algorithm for Mars landing is presented for accurate navigation and control in the presence of uncertainties. This method is based on a variable structure control framework. The performance of this method was analyzed via Monte Carlo simulations in the presence of different uncertainties. The results were compared with the results of an extended Kalman filter and PID controller algorithm. The proposed algorithm showed a robust and precise performance and proved its superiority against conventional algorithms.

Like other control algorithms, optimal and predictive control algorithms have also been used for Mars landing problems. Like in [10], a comparative analysis was performed based on simulations between one and two parametric shooting predictive methods. Additionally, an identification approach that adapts to the varied environment was used. The computer simulations showed the second technique is more accurate. In [11], a trajectory optimization algorithm for the power descent landing phase is presented considering the disturbances. A composite solution based on a non-linear controller and a disturbance compensator is presented. The robustness and feasibility were verified through simulations for Mars landing missions.

In [12], it is mentioned that real-time embedded MPC would be an important guidance and control strategy for the next generation of space missions. An embedded MPC for thrust vectored control is proposed for the ascent and pinpoint descent phase. The following simplifications are involved for system modeling: (a) planet rotation is ignored; (b) the flat surface of the planet is considered; (c) uniform gravity; and (d) aerodynamic effects are negligible. Furthermore, for optimization, an accelerated dual gradient projection (GPAD) algorithm is used for quadratic programming. Similarly, in [13], a composite controller is implemented for following the Mars trajectory. It involves MPC as an optimal trajectory tracking control algorithm and an observer-based feedforward compensator. The simulation results showed an improvement in the performance compared to previous results.

At last, in [14], a constrained dynamic algorithm is presented that reduces the expected cost whereas the chances of violating the constraints are bounded. Applications of this algorithm have been shown for tracking Mars and lunar landings through simulations. All these predictive algorithms used for the Mars landing gave satisfactory results. However, they lack a few aspects that are very necessary for a safe landing. For example, a few of them lack a constraint handling ability while in others, the computational time was not considered at all. Few of these predictive algorithms were only used for trajectory generation, whereas most were not focused on the powered descent phase. In powered descent, hazard avoidance features of safe landing must also be considered. Therefore, a predictive control algorithm with a constraint handling ability, an effective computational time, and focused on powered descent needs to be further explored.

➢ In most of the research mentioned above, only the entry phase of Mars, the landing, has been the focused area.
➢ In past research, the powered descent phase has not been focused on specifically.
➢ All the above-mentioned research lacks the ability of the constraint handling feature, which is very important for the Mars landing problem.

Like the above-mentioned planetary UAVs, similar control algorithms have been used for terrestrial UAVs as well. In this context, recently, an adaptive SMC was presented for attitude and altitude stabilization of a quadrotor in [15]. It was concluded that in comparison with conventional control strategies, better tracking performance and disturbance rejection can be achieved with the proposed control algorithm. Whereas, in [16], another SMC is proposed for the quadrotor tracking control problem to achieve robustness. The main advantage of this proposed method is that the chattering issue is resolved along with a fast and precise dynamic response. Meanwhile, all the errors present in the closed-sloop were proved to be bounded through Lyapunov analysis. Finally, the effectiveness of this control method was verified through simulations and experimentation.

The section division of this paper is as follows: Section 2 outlines the UAV test platforms whereas Section 3 discusses the control algorithms for Mars landing. Section 4 describes Mars landing control. Finally, Section 5 contains the conclusion of this research work.

2. UAV Test Platform

In the literature, there are examples of emulating the dynamics of the lander with other dynamical systems. A helicopter UAV was used for this in [17], where a six d.o.f platform was introduced to emulate a planetary lander. Similarly, the Autonomous Helicopter Testbed (AHT) is an experimental platform used to evaluate proposed technologies for future space missions. The Gantry Testbed (GT) was developed to test algorithms before implementing them on the AHT testbed. This can be controlled to follow desired trajectories. Furthermore, the Planetary Landing GNC Test Facility (PLGTF) was designed by European Space Agency (ESA) for the verification of precision landing GNC systems. It consists of a flying testbed that is outdoors along with a station on the ground, which is designed according to the powered descent of Mars and the Moon [18].

A testbed for a lunar mission specially designed for long-duration analysis and testing was presented in [19]. It is comprised of hardware and software, which exhibits the dynamics that closely approximate the original lunar landers. Similarly, the Landing Dynamic Test Facility (LDTF) was designed in collaboration with the Canadian Space Agency (CSA) and NGC's Laboratory, Sherbrooke, Canada in 2007. The main purpose of LDTF was to fill the gap between computer simulations and real and expensive rocket-based flight tests [20].

A quadrotor is a multirotor type of UAV. It has many advantages over traditional helicopters and fixed-wing aircraft due to its dynamic nature. It has the ability of VTOL, hovering, and flying in six degrees of freedom. Such flying abilities make it possible for quadrotors to take off and land in any environment, making them an ideal and relevant platform for emulating a planetary lander. It has four rotors and their arrangement relative to coordinate system has two configurations: the “x” and “+” [21]. The quadrotor has four inputs controlling six degrees of freedom output states as given by Equation (1). Three of them are translational while the other three are rotational outputs. It can follow the trajectories like a spacecraft. To implement optimal and predictive control, a state-space model of the system is required. The state-space model can be derived from linear or nonlinear equations of the system. In this part, a state-space model of the system is used to implement MPC. The state-space form can handle linear, nonlinear, and unstable systems as well. The non-linear equations of the quadrotor are given below by Equation (2):

$$6\,\mathrm{DOF} = [x,\ y,\ z,\ \phi,\ \theta,\ \psi] \tag{1}$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \\ \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} \cos(\theta)\cos(\psi)u - \cos(\varphi)\sin(\psi)v + \sin(\varphi)\sin(\theta)\cos(\psi)v + \sin(\varphi)\sin(\psi)w + \cos(\varphi)\sin(\theta)\cos(\psi)w \\ \cos(\theta)\sin(\psi)u + \cos(\varphi)\cos(\psi)v + \sin(\varphi)\sin(\theta)\sin(\psi)v - \sin(\varphi)\cos(\psi)w + \cos(\varphi)\sin(\theta)\sin(\psi)w \\ -\sin(\theta)u + \sin(\varphi)\cos(\theta)v + \cos(\varphi)\cos(\theta)w \\ K_\varphi S_\varphi + K_\theta \tan(\theta)\sin(\varphi)S_\theta + K_\psi \tan(\theta)\cos(\varphi)S_\psi \\ K_\theta \cos(\varphi)S_\theta - K_\psi \sin(\varphi)S_\psi \\ K_\theta \cos(\varphi)S_\theta - K_\psi \sin(\varphi)S_\psi \\ K_\theta \sin(\varphi)S_\theta / \cos(\theta) + K_\psi \cos(\varphi)S_\psi / \cos(\theta) \\ K_\theta \sin(\varphi)S_\theta / \cos(\theta) + K_\psi \cos(\varphi)S_\psi / \cos(\theta) \\ K_\varphi S_\varphi w - K_\psi S_\psi u + g\sin(\varphi)\cos(\theta) \\ K_\theta S_\theta u - K_\varphi S_\varphi v + g\cos(\varphi)\cos(\theta) + I_f w/m + F_{rotor}/m \end{bmatrix} \tag{2}$$

This non-linear model of a quadrotor is used from our past work [22], where x, y, z is the position; u, v, w represents the velocity; and φ, θ, ψ are the Euler angles. The linear state-space model of a quadrotor is derived from Equation (2) by applying small-angle approximation and is given by Equations (3)–(7):

$$\dot{x}_s = A{\cdot}x_s + B{\cdot}u_s\ y = C{\cdot}x_s + D{\cdot}u_s \tag{3}$$

$$x_s = \begin{bmatrix} x & y & z & \varphi & \theta & \psi & u & v & w \end{bmatrix} \tag{4}$$

$$u_s = \begin{bmatrix} S_\varphi & S_\theta & F_z & S_\psi \end{bmatrix} \tag{5}$$

In Equation (4), x, y, z are the position coordinates of the quadrotor; φ, θ, ψ are Euler angles; and u, v, w are the velocities of the quadrotor given in Equations (6) and (7):

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -g & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & g & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{I_f}{m} \end{bmatrix} \tag{6}$$

$$B = \begin{bmatrix} 0 & 0 & 0 & K_\varphi & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & K_\theta & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & K_\psi & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{K_t}{m} \end{bmatrix}^T \tag{7}$$

3. Control Algorithms for Mars Landing

3.1. Model Predictive Control

MPC is an optimal control algorithm and is suitable for multivariable and constrained systems [23]. It was introduced for use in process control, but recently, its applications has been expanded to other fields as well because increasing computational power of modern microprocessors has been presented as an alternative for conventional controllers. It is advantageous to use MPC instead of other controllers. For instance, it is feasible for different types of processes, can be implemented for multivariable problems, and has a fast dynamic response.

Furthermore, it can handle nonlinear systems and constraints offered by the system. With the dramatic increase in the computational power of the latest computing technology, MPC has become feasible, more than ever before. To implement the MPC, a mathematical model describing the dynamics of the system is compulsory. The model given in Section 2 is used for MPC. The MPC finds the manipulated variable by minimizing the cost function. This function compares the predicted output of the system with the reference. These output predictions are computed from the given model. A control action is computed at every sampling interval so that the cost function is minimized. MPC solves an open-loop control problem optimally and the algorithm is repeated in a receding horizon fashion at each sampling time, thus it provides a feedback loop and robustness against system uncertainties [24]. The main edge of MPC is its power to handle the constraints and incorporate them during optimization. The main objective is to change the optimal manipulated control input to be suitable for the condition with some active constraints. Based on the predictions, the quadratic cost function is minimized while considering the constraints at each step. This minimization of the cost function subject to constraint needs quadratic programming (QP). The solution of optimization would be infeasible if constraints are overly restrictive such that no solution exists. However, if the solution exists, it is referred to as a feasible solution [25]. The case of infeasibility for practical implementation of MPC to some applications could be dangerous. It may take a long time for QP to compute the feasible solution. There may be a case when no feasible solution exists. As a result, no control move is found, and the system may become unstable. The prediction horizon and control horizon are very important for the tuning of MPC. The prediction horizon is the number of samples for which MPC predicts the output of the plant whereas the control horizon is the number within the prediction horizon for which the control action can be affected as shown in Equation (8):

$$x_{k+1} = Ax_k + Bu_k \quad y_k = Cx_k + Du_k \tag{8}$$

Consider the control law for each sample is given by Equation (9), where K is the gain of feedback and c_k is the degrees of freedom (DOF) as shown in Equation (9):

$$u_{k+i} = \begin{cases} -K(x_{k+i} - x_{ss}) + c_{k+i};\ i \in \{0, \ldots, N_c - 1\} \\ -K(x_{k+i} - x_{ss});\ i \in \{N_c, N_c + 1, \ldots\} \end{cases} \tag{9}$$

Steady states u_{ss}, x_{ss} are the inputs and states required for tracking in the steady state. The main focus of MPC is generating the control signal using predictive output. This control signal is computed using a cost function that is similar to the linear-quadratic function given in Equation (10):

$$J = \sum_{k=0}^{N_p} x^T Q x + \sum_{k=0}^{N_p} \Delta\, u^T R\, \Delta u \tag{10}$$

$$\Delta u = [\Delta u1\ \Delta u2\ \Delta u3\ \Delta u3]^T \tag{11}$$

To obtain the optimal control signal, MPC minimizes the cost function given by the above equation, where Q shows the convergence to states. The value of Q and R is selected very carefully. Every physical system has some limitations known as constraints of the system. If the limit of the system is exceeded, this can result in permanent failure of the system. Thus, MPC restricts the input to the system so that it never goes beyond this limit, which is the most unique characteristic of this controller, as given in Equation (12):

$$y_{min} \le y \le y_{max}\ and\ u_{min} \le u \le u_{max} \tag{12}$$

3.2. Explicit Model Predictive Control

As MPC computes the optimal solution at each sampling time, this runtime computation is a time-consuming process. Thus, MPC implementation is sometimes difficult in some cases. To overcome this issue, the EMPC technique has been explored, where offline computation is used to minimize the time consumption during optimization [26]. EMPC uses lookup tables instead of online optimization and it minimizes the computation time. By employing multi-parametric quadratic programming, computations are performed offline in a repetitive manner, which significantly minimizes the computation time. In EMPC, it is impossible to derive the true optimal control law; hence, approximate approaches have been used. Thus, several different techniques are used like neural networks [27], set memberships identification [28], and piecewise affine (PWA) or linear approximations. In this work, EMPC with the piecewise affine function is of interest. EMPC saves the coefficients for PWA for each control region of the system states, where the PWA is constant [29]. It creates multiple regions against these coefficients. Thus, to obtain optimal control, the region with current states is determined and secondly, a mere evaluation of PWA from the PWA coefficients stored for the region is found. For a small number of regions, EMPC does not require much computational time compared to implicit MPC and is suitable for systems with fast dynamics [30]. However, when a larger number of regions is created than what is required for the controller, the memory increases and the computational time to search for the current control region also increases, but this is still better compared to online MPC computation. In MPC, quadratic programming (QP) is used to determine the optimal control variable at each control interval, which is the result of the nonlinear implicit function given in Equation (13):

$$u = f(x) \tag{13}$$

It is a time-consuming process, and the computational time may vary from one control interval to the next. In applications where solutions are required within a certain range of time, in microseconds, the implicit MPC approach is not suitable. In the explicit MPC control variable, the solution becomes a linear function of x as given by Equation (14):

$$u = Fx(k) + G \tag{14}$$

In Equation (14), F and G are the constant values, where, if x remains in a region that is already defined, then the QP solution is a linear function of x, but the F and G constants have different values. EMPC uses offline computations to find the polyhedral regions and control-law constants [31]. When the controller runs, EMPC follows these steps. At first, it estimates the states using measurements as in MPC. Then, it uses the estimated states and values of the other variables' region in which x(k) resides is identified. After that, it looks into the predefined F and G constants for that particular region. At last, it evaluates the desired linear function using the relation u = Fx(k) + G. A PWA state-space model is defined as given by Equation (15):

$$x_{k+1} = A_i x_k + B_i u_k \; y_k = C_i x_k + D_i u_k \tag{15}$$

where $\mathbf{x_k}$ and $\mathbf{u_k}$ belongs to $\mathbf{P_i}$ and $\mathbf{P_i}$ is the polyhedral region defined by states and inputs and it is given by Equation (16) below [31]. The values of the matrices $\mathbf{A_i}$, $\mathbf{B_i}$, and $\mathbf{C_i}$ are defined by the sates $\mathbf{x_k}$ of the system:

$$P_i = \{(x_k, u_k) | F_i x_k + G_i u_k \leq b_i, \; i = 1, \ldots, p\} \tag{16}$$

The results of these predictive control algorithms are also compared with PID and LQR control algorithms designed in our previous research [22].

4. Mars Landing Control

The powered descent phase is the main focus area in this study. First, the lander obtains the desired orientation and then descends vertically. Ideally, the lander only has a vertical velocity at this point. In powered descent, once the thrusters start, and they cannot be turned off. Therefore, the controller must look for both the minimum and maximum limits for thrust levels during computation. These thrust constraints should be incorporated during optimization. Another constraint for such propulsive landers is to spend a minimum amount of time near the surface during touchdown. This prevents the lander from facing hazards.

Instead of considering the Mars lander as a point mass, a quadrotor UAV can also be used instead to perform this terminal descent [32]. It is a very unique idea to implement, test, and evaluate different landing control algorithms on such a simple and reliable test platform before they are implemented on a real Mars lander. We are interested in evaluating the tracking performance of the proposed control algorithms with input constraints on thrust inputs and some output constraints on the trajectory of the Mars lander for the powered descent landing phase. The following assumptions need to be considered to test the proposed control algorithm: at first, a non-zero final vertical position for landing is considered [32]. Secondly, the curvature of Mars is negligible. Thirdly, centripetal and Coriolis forces are not considered. Furthermore, an optimal trajectory is considered for the powered descent phase of the Mars landing. Moreover, during the descent phase, the distance from the surface is very short, so a uniform gravitational model is used in the equations of motion. Finally, as the lander has very little velocity in the descent phase compared to the entry phase, the aerodynamic forces become insignificant and the thrust force from the descent engine is dominant at this point. In [33], details of the Mars landing for the Phoenix mission are given. The powered descent starting point has a mean value of 51.9 m in the case of phoenix. This height varies with different missions. It was 142 m in the case of MSL [34]. Thus, in the simulations, we considered this height to be 150 m. Keeping in view the ending point constraint of the landing, a height of 2 m above the surface level was selected. Two scenarios were selected to exhibit the powered descent on the quadrotor in the simulation environment as stated below. It is assumed in both scenarios that the sensors are given accurate information about the states of the system.

4.1. Scenario 1

In the first case, the quadrotor descends from a height of 150 m to a height of 2 m, with zero x, y starting positions. A reference trajectory is given to the quadrotor comprising x, y, and z values as required by the condition to evaluate the control algorithm. At first, the tracking performance of the PID control algorithm is checked and the results are plotted in Figure 1. The PID controller shows a good tracking performance. Although the PID algorithm shows some overshoot, overall, it shows good tracking ability. After evaluating the PID algorithm for powered descent, LQR is implemented next. The same trajectory is applied to the quadrotor as a reference with the LQR control algorithm and verified tracking performance. The results of the tracking performance with LQR are given in Figure 2, showing the desired and actual values along the z-axis. The results show a good tracking performance with the LQR controller. Though it lags in performance for tracking the z-axis by some value compared with PID, it does not overshoot like PID.

After the PID and LQR control algorithms, MPC is explored for the stated case of the Mars-powered descent. Figure 3 shows the performance results for the MPC algorithm. The MPC algorithm shows a good tracking ability with no overshoot as compared to the PID control algorithm. Another predictive control method is proposed in this study to have less computational time: the Explicit MPC. Thus, this predictive control algorithm is implemented on the quadrotor for powered descent in the same way as the previous control algorithms. Figure 4 shows the tracking ability and effectiveness of Explicit MPC for the stated landing scenario.

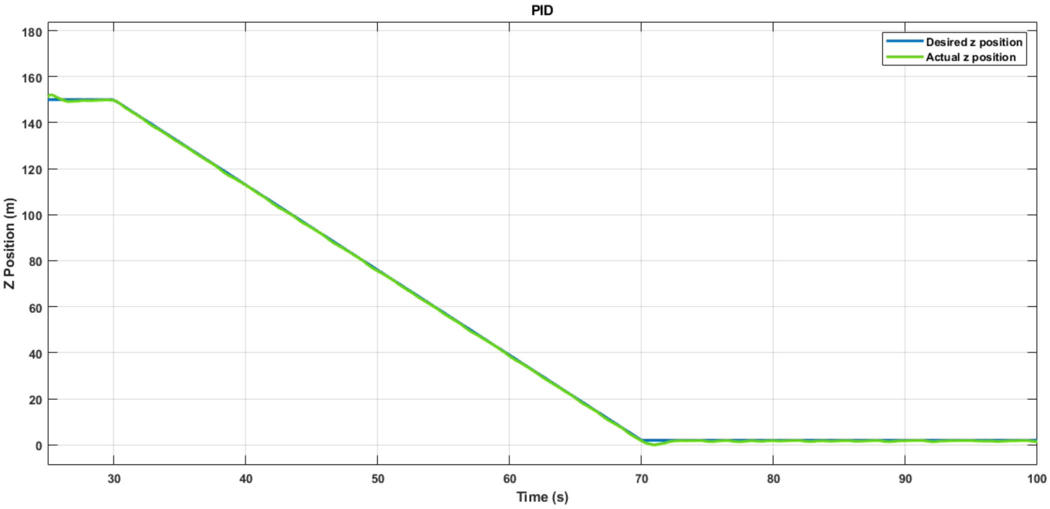

Figure 1. Z position with PID.

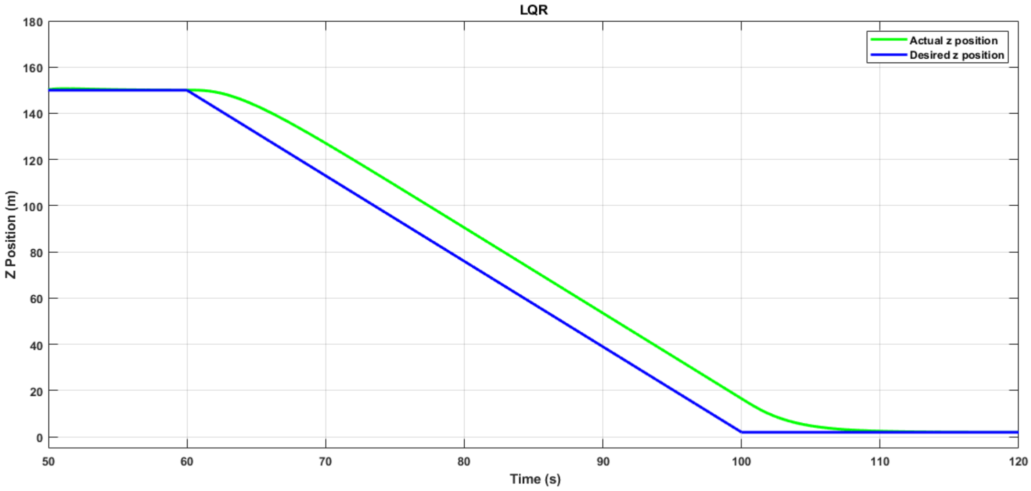

Figure 2. Z position with LQR.

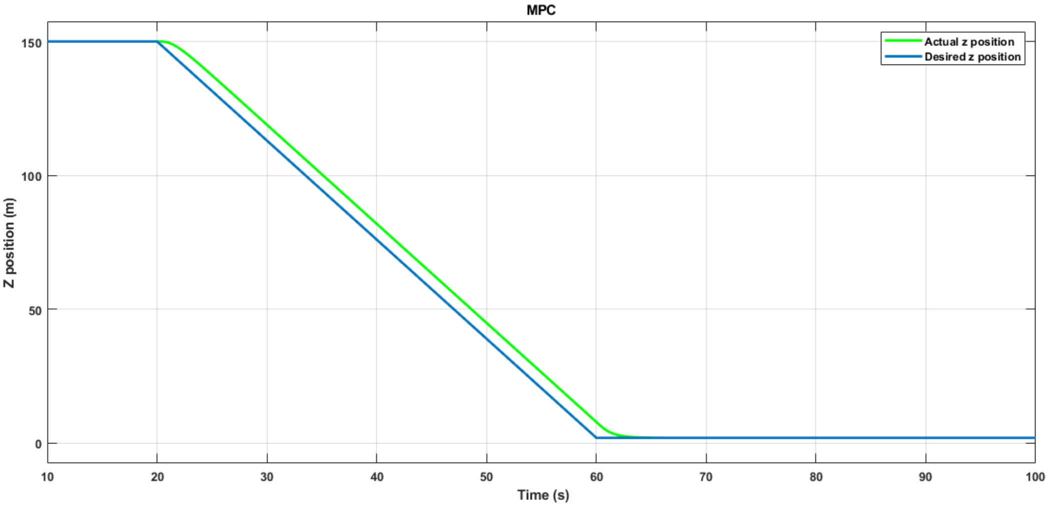

Figure 3. Z position with MPC.

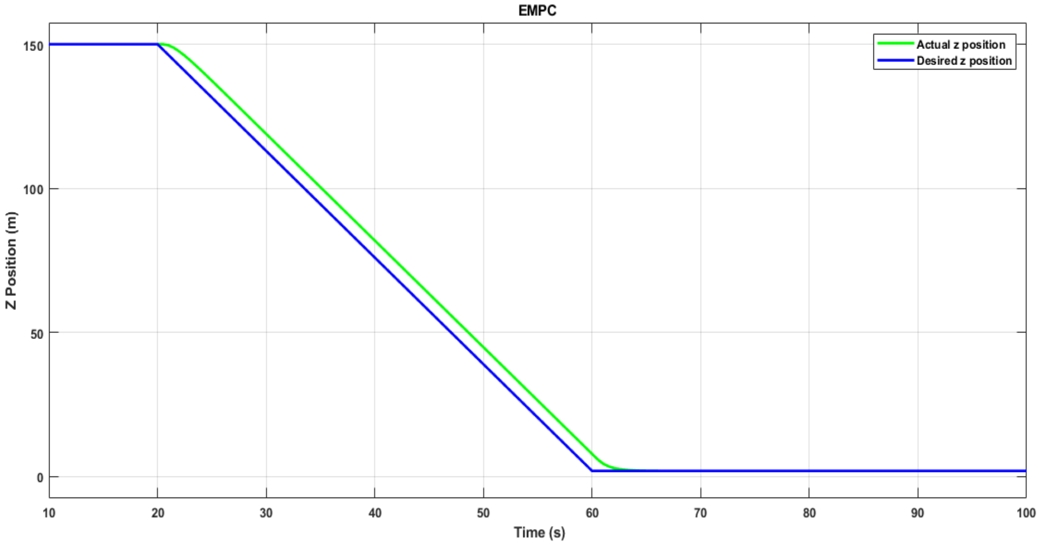

Figure 4. Z position with EMPC.

4.2. Scenario 2

In this section, another scenario is designed for Mars-powered where non-zero x and y positions are defined in the reference trajectory. The starting value for x and y is taken as 2 m whereas the starting value for the z-axis is the same as in the previous case. The lander faces disturbances and atmospheric uncertainty, which makes the lander divert from the planned trajectory. Thus, a disturbance signal is also applied along the x-axis, which disturbs the quadrotor to detract it from the desired path. Restriction of the horizontal position of the lander is required so that it cannot go beyond certain limits. Such a safe region is defined in a form of a square. Thus, the horizontal limits of 2 m along the x-axis and the y-axis are considered as output constraints. It is assumed that if the quadrotor goes beyond this boundary, it will face obstacles similar to what a planetary lander faces while landing on non-cooperative sites. Moreover, the thrust input is also limited to a level to validate MPC's input constraint handling ability. All the control algorithms used in the previous section are implemented in this section to check the performance for the stated scenario in this section. This scenario illustrates the importance of constraints handling.

At first, the PID control algorithm is implemented to track the stated trajectory with a disturbance at 30 s. It can be seen from Figure 5 that the PID control algorithm is unable to restrict the system under the boundary limits, which may cause an unsuccessful landing. Furthermore, PID is unable to restrict the input limitations of the system. The same is the case with LQR, which is also unable to show the constraint behavior. Although PID and LQR have a very good tracking ability, they do not show a good performance in a constraint environment. MPC fulfills the constraint drawbacks of PID and LQR as it can give a better performance in a constraint environment by considering the input and output constraints during optimization. Thus, the MPC algorithm is explored for this case to handle the constraint environment. During the design of the algorithm, an output constraint of 2 m is set for the x position of the quadrotor. Similarly, an input constraint for the thrust input of the quadrotor is implemented during the design of the predictive control algorithm. The constraint performance of the quadrotor is given in Figure 6. The output variable x of a quadrotor does not go beyond the boundary of 2 m. Similarly, the unconstraint and constraint thrust input is also shown in the fourth section of this figure. Moreover, the tracking performance along the y and z-axis is also shown in the same figure. After implementing MPC for the stated scenario in this section, EMPC is implemented as well. While implementing constraints, EMPC for a MIMO system is difficult compared to implicit MPC. Proper ranges for states, inputs, outputs, and references must be defined properly while designing an EMPC. An EMPC is implemented on the quadrotor for this scenario and the constraint performance is given in Figure 7. A constraint of 2 m is set for the x output of the system. Like MPC, it shows good constraint behavior and never goes beyond the boundary limits. When generating a control signal by EMPC, 992 solution regions are explored for the constraint system used in the simulation. These solutions are saved to the controller and it gives the optimal and predicted solution without calculation during the run time, thus it reduces the execution time.

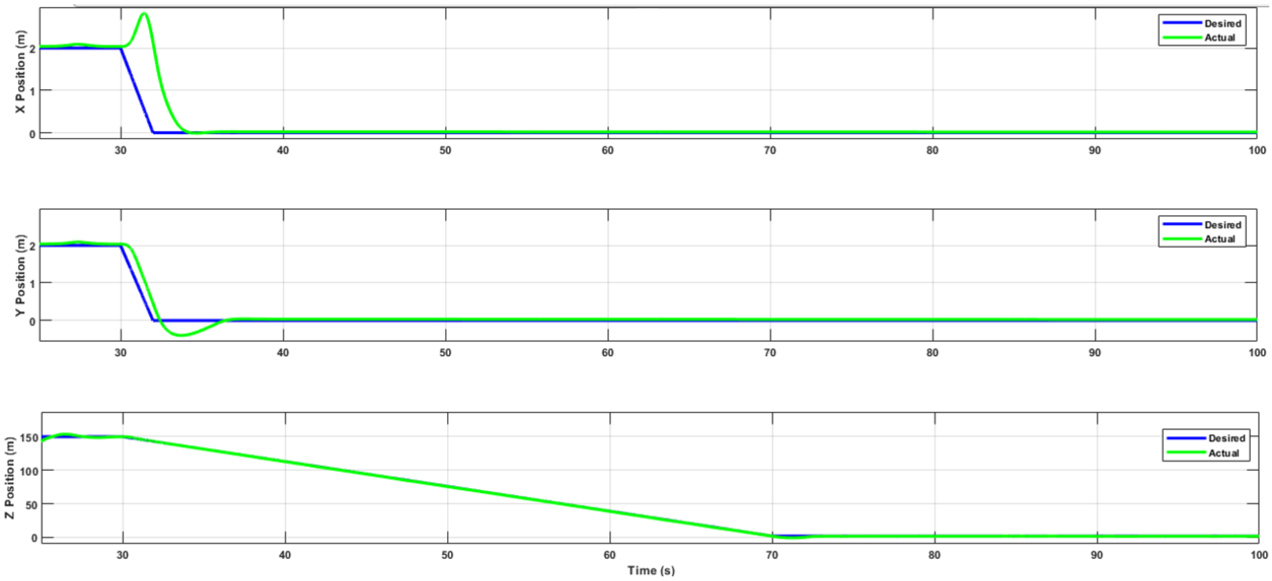

Figure 5. Z powered descent with PID.

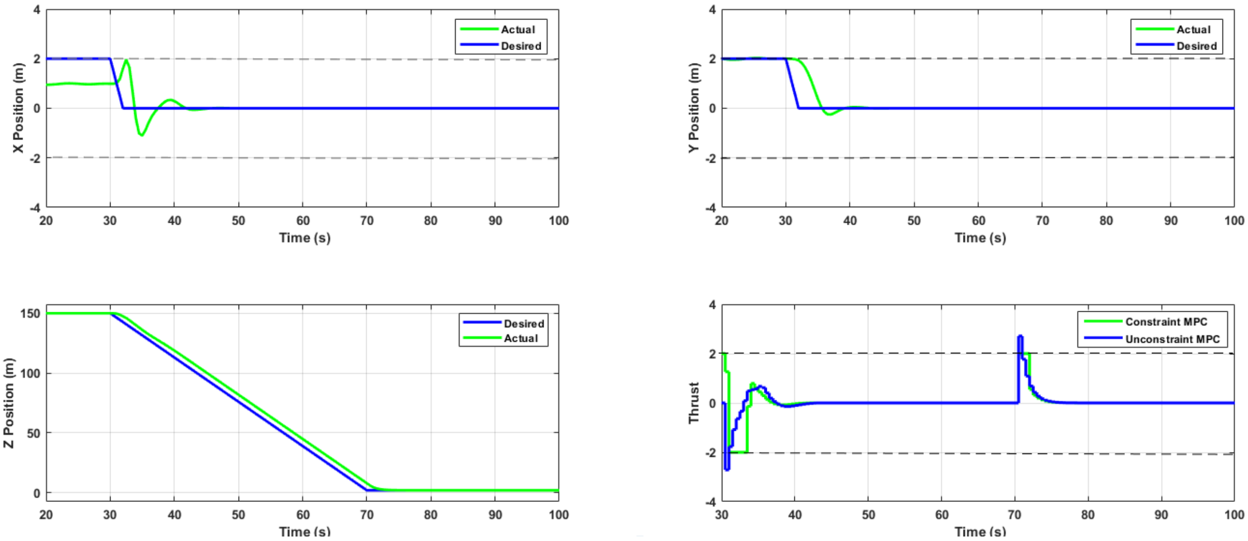

Figure 6. Z powered descent with MPC.

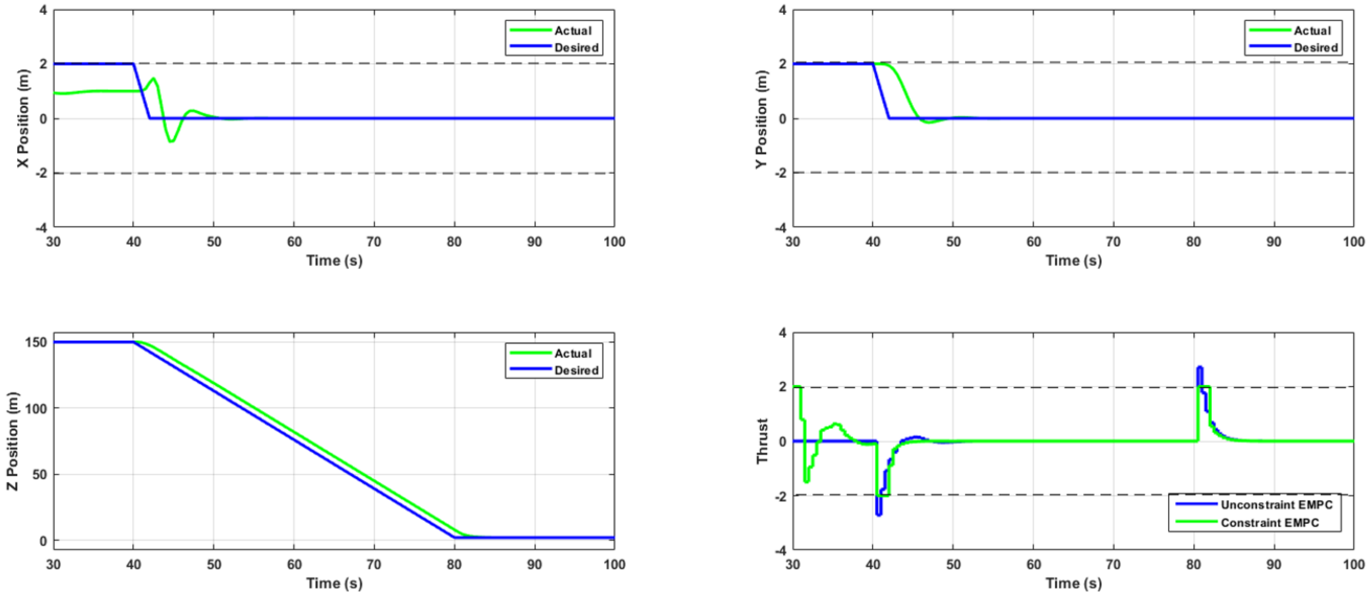

Figure 7. Z powered descent with EMPC.

- ➢ The successful constraint behavior of the proposed control algorithm on the stated scenarios shows the feasibility of these predictive algorithms for Mars-powered descent.
- ➢ The performance of the proposed predictive control algorithms for Mars-powered descend was evaluated based on the constraint handling, feasibility, and computational time.
- ➢ It is summarized in Table 1, which gives the computational time, constraints violations, and feasibility, which were the main objectives of this research.

Table 1. Performance evaluation.

Control Algorithm	Computational Time (ms)	Constraint Violations	Feasibility	Online Cost
Different [4]	N/A	Yes	Feasible	N/A
MPC [12]	0.51–8.31	No	Feasible	N/A
MPC	0.038	No	Feasible	1,200,000
EMPC	0.026	No	Feasible	N/A

5. Conclusions

The number of missions on other planets like Mars is increasing exponentially with time in which the landing is the most crucial part of the whole mission, which often fails. So, in this study, the Mars-powered descent phase of landing was addressed in detail and the proposed control algorithms were implemented for the stated problem. Simulations were performed with unconstraint and constraint input and outputs. It was shown that the proposed Model Predictive Control algorithms showed effective performance for constraint environments whereas conventional control algorithms failed to show such constraint behavior. Thus, the control algorithms explored in this study are most suitable for the Mars-powered descent phase. Finally, the performance of these algorithms was evaluated based on the constraint handling, computational time, and feasibility. It was verified that they are very good for constraint environments, have reasonably good computational time, and are more feasible for the Mars-powered descent landing phase. In the future, these control algorithms can also be applied for missions to the Moon or any other planetary body.

Author Contributions: Conceptualization, A.K., M.H.J. and M.Y.J.; methodology, A.Y., J.A. and A.U.R.; software, A.K., M.H.J. and M.Y.J.; validation, A.H., M.M.A., H.H. and M.S.; formal analysis, A.Y., J.A. and A.U.R.; investigation, A.H., M.M.A., H.H. and M.S.; resources, A.Y., J.A. and A.U.R.; data curation, A.H., M.M.A., H.H. and M.S.; writing—original draft preparation, A.K., M.H.J. and

M.Y.J.; writing—review and editing, A.H., M.M.A., H.H. and M.S.; visualization, A.Y., J.A. and A.U.R.; supervision, J.A. and A.U.R. and M.S.; project administration, A.Y., J.A.; funding acquisition, M.M.A., H.H. and M.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Taif University Researchers Supporting Project Number (TURSP-2020/328), Taif University, Taif, Saudi Arabia.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: We deeply acknowledge Taif University for supporting this research through Taif University Researchers Supporting Project Number (TURSP-2020/328), Taif University, Taif, Saudi Arabia.

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

References

1. Li, S.; Peng, Y. Command generator tracker based direct model reference adaptive tracking guidance for Mars atmospheric entry. *Adv. Space Res.* **2012**, *49*, 49–63. [CrossRef]
2. Xia, Y.; Chen, R.; Pu, F.; Dai, L. Active disturbance rejection control for drag tracking in mars entry guidance. *Adv. Space Res.* **2014**, *53*, 853–861. [CrossRef]
3. Dai, J.; Xia, Y. Mars atmospheric entry guidance for reference trajectory tracking. *Aerosp. Sci. Technol.* **2015**, *45*, 335–345. [CrossRef]
4. Kang, S.-W.; Bang, H.; Lee, S.-R. Adaptive backstepping radial basis function neural network controller design for a Mars lander during the powered descent phase. *Proc. Inst. Mech. Eng. Part G J. Aerosp. Eng.* **2018**, *232*, 2091–2107. [CrossRef]
5. Huang, Y.; Li, S.; Sun, J. Mars entry fault-tolerant control via neural network and structure adaptive model inversion. *Adv. Space Res.* **2019**, *63*, 557–571. [CrossRef]
6. Long, J.; Zhu, S.; Cui, P.; Liang, Z. Barrier Lyapunov function based sliding mode control for Mars atmospheric entry trajectory tracking with input saturation constraint. *Aerosp. Sci. Technol.* **2020**, *106*, 106213. [CrossRef]
7. Shen, G.; Xia, Y.; Zhang, J.; Cui, B. Adaptive fixed-time trajectory tracking control for Mars entry vehicle. *Nonlinear Dyn.* **2020**, *102*, 2687–2698. [CrossRef]
8. Gaudet, B.; Linares, R.; Furfaro, R. Deep reinforcement learning for six degree-of-freedom planetary landing. *Adv. Space Res.* **2020**, *65*, 1723–1741. [CrossRef]
9. Maryam, K.; Ahmadvand, R. Modified Variable Structure Estimation and Control for Constrained Landing on Mars. *Amirkabir J. Mech. Eng.* **2021**, *53*, 2.
10. Liang, Z.; Mease, K.D. Precision Guidance for Mars Entry with a Supersonic Inflatable Aerodynamic Decelerator. *J. Guid. Control Dyn.* **2019**, *42*, 1571–1578. [CrossRef]
11. Liu, R.; Li, S.; Chen, X.; Guo, L. Powered-descent trajectory optimization scheme for Mars landing. *Adv. Space Res.* **2013**, *52*, 1888–1901. [CrossRef]
12. Pascucci, C.A.; Bennani, S.; Bemporad, A. Model predictive control for powered descent guidance and control. In Proceedings of the European Control Conference, Linz, Austria, 15–17 July 2015; IEEE: New York, NY, USA, 2015; pp. 1388–1393.
13. Wu, C.; Yang, J.; Li, S.; Li, Q.; Guo, L. Disturbance observer based model predictive control for accurate atmospheric entry of spacecraft. *Adv. Space Res.* **2018**, *61*, 2457–2471. [CrossRef]
14. Ono, M.; Pavone, M.; Kuwata, Y.; Balaram, J. Chance-constrained dynamic programming with application to risk-aware robotic space exploration. *Auton. Robot.* **2015**, *39*, 555–571. [CrossRef]
15. Shao, X.; Sun, G.; Yao, W.; Liu, J.; Wu, L. Adaptive Sliding Mode Control for Quadrotor UAVs with Input Saturation. *IEEE/ASME Trans. Mechatron.* **2021**. [CrossRef]
16. Xu, L.; Shao, X.; Zhang, W. USDE-Based Continuous Sliding Mode Control for Quadrotor Attitude Regulation: Method and Application. *IEEE Access* **2021**, *9*, 64153–64164. [CrossRef]
17. Saripalli, S.; Sukhatme, G.S. A testbed for Mars precision landing experiments by emulating spacecraft dynamics on a model helicopter. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems Lausanne, Switzerland, 30 September–4 October 2002; IEEE: New York, NY, USA, 2002.
18. Guizzo, G.P.; Bertoli, A.; Della Torre, A.; Magistrati, G.; Mailland, F.; Vukman, I.; Philippe, C.; Jurado, M.M.; Ori, G.G.; Macdonald, M.; et al. Mars and Moon exploration passing through the European Precision Landing GNC Test Facility. *Acta Astronaut.* **2008**, *63*, 74–90. [CrossRef]
19. de Lafontaine, J.; Neveu, D.; Hamel, J.-F. Planetary Landing Dynamic Test Facility: Design and Applications. In Proceedings of the AIAA Guidance, Navigation and Control Conference, Minneapolis, MN, USA, 13–16 August 2012; p. 6341.
20. Zhang, X.; Li, X.; Wang, K.; Lu, Y. A survey of modelling and identification of quadrotor robot. In *Abstract and Applied Analysis*; Hindawi Publishing Corporation: London, UK, 2014; Volume 2014.

21. Maurya, H.L.; Behera, L.; Verma, N.K. Trajectory Tracking of Quad-Rotor UAV Using Fractional Order PI$^{\mu}$D$^{\lambda}$ Controller. In *Computational Intelligence: Theories, Applications and Future Directions*; Springer: Singapore, 2019; Volume 1, pp. 171–186.
22. Jaffery, M.H.; Shead, L.; Forshaw, J.L.; Lappas, V.J. Experimental quadrotor flight performance using computationally efficient and recursively feasible linear model predictive control. *Int. J. Control* **2013**, *86*, 2189–2202. [CrossRef]
23. Bahakim, S.S.; Ricardez-Sandoval, L.A. Simultaneous design and MPC-based control for dynamic systems under uncertainty: A stochastic approach. *Comput. Chem. Eng.* **2014**, *63*, 66–81. [CrossRef]
24. Vazquez, S.; Rodriguez, J.; Rivera, M.; Franquelo, L.G.; Norambuena, M. Model Predictive Control for Power Converters and Drives: Advances and Trends. *IEEE Trans. Ind. Electron.* **2017**, *64*, 935–947. [CrossRef]
25. Rossiter, J.A. *Model-Based Predictive Control: A Practical Approach*; CRC Press: Boca Raton, FL, USA, 2017.
26. Trinh, V.V.; Alamir, M.; Bonnay, P.; Bonne, F. Explicit Model Predictive Control via Nonlinear Piecewise Approximations. *IFAC-PapersOnLine* **2016**, *49*, 259–264. [CrossRef]
27. Pin, G.; Filippo, M.; Pellegrino, F.A.; Fenu, G.; Parisini, T. Approximate model predictive control laws for constrained nonlinear discrete-time systems: Analysis and offline design. *Int. J. Control* **2013**, *86*, 804–820. [CrossRef]
28. Fagiano, L.; Canale, M.; Milanese, M. Set membership approximation of discontinuous nonlinear model predictive control laws. *Automatica* **2012**, *48*, 191–197. [CrossRef]
29. Oberdieck, R.; Diangelakis, N.A.; Pistikopoulos, E.N. Explicit model predictive control: A connected-graph approach. *Automatica* **2017**, *76*, 103–112. [CrossRef]
30. Hredzak, B.; Agelidis, V.G.; Demetriades, G. Application of explicit model predictive control to a hybrid battery-ultracapacitor power source. *J. Power Sources* **2015**, *277*, 84–94. [CrossRef]
31. Rivotti, P. and Pistikopoulos, E.N. A dynamic programming-based approach for explicit model predictive control of hybrid systems. *Comput. Chem. Eng.* **2015**, *72*, 126–144. [CrossRef]
32. Rizwan, R.; Arshad, J.; Almogren, A.; Jaffery, M.H.; Yousaf, A.; Khan, A.; Rehman, A.U.; Shafiq, M. Implementation of ANN-Based Embedded Hybrid Power Filter Using HIL-Topology with Real-Time Data Visualization through Node-RED. *Energies* **2021**, *14*, 7127. [CrossRef]
33. Desai, P.N.; Prince, J.L.; Queen, E.M.; Schoenenberger, M.M.; Cruz, J.R.; Grover, M.R. Entry, Descent, and Landing Performance of the Mars Phoenix Lander. *J. Spacecr. Rocket.* **2012**, *48*, 798–808. [CrossRef]
34. San Martin, A.M.; Lee, S.W.; Wong, E.C. The Development of the MSL Guidance, Navigation, and Control System for Entry, Descent, And Landing. In Proceedings of the 23rd Space Flight Mechanics Meeting, Kauai, HI, USA, 10–14 February 2013.