



Article Preliminary Design of a Receding Horizon Controller Supported by Adaptive Feedback

Hazem Issa ^{1,*,†} and József K. Tar ^{1,2,3,*,†}

- ¹ Doctoral School of Applied Informatics and Applied Mathematics, Óbuda University, Bécsi út 96/B, H-1034 Budapest, Hungary
- ² Antal Bejczy Center for Intelligent Robotics, Óbuda University, Bécsi út 96/B, H-1034 Budapest, Hungary
- ³ John von Neumann Faculty of Informatics, Óbuda University, Bécsi út 96/B, H-1034 Budapest, Hungary
 - * Correspondence: hazem.issa@uni-obuda.hu (H.I.); tar.jozsef@nik.uni-obuda.hu (J.K.T.)
 - + These authors contributed equally to this work.

Abstract: Receding horizon controllers are special approximations of optimal controllers in which the continuous time variable is discretized over a horizon of optimization. The cost function is defined as the sum of contributions calculated in the grid points and it is minimized under the constraint that expresses the dynamic model of the controlled system. The control force calculated only for one step of the horizon is exerted, and the next horizon is redesigned from the measured initial state to avoid the accumulation of the effects of modeling errors. In the suggested solution, the dynamic model is directly used without any gradient reduction by using a transition between the gradient descent and the Newton–Raphson methods to achieve possibly fast operation. The optimization is carried out for an "overestimated" dynamic model, and instead of using the optimized force components the optimized trajectory is adaptively tracked by an available approximate dynamic model of the controlled system. For speeding up the operation of the system, various cost functions have been considered in the past. The operation of a 4-degrees-of-freedom SCARA robot using the simple sequential Julia language code realizing Euler integration.

check for updates

Citation: Issa, H.; Tar, J.K. Preliminary Design of a Receding Horizon Controller Supported by Adaptive Feedback. *Electronics* **2022**, *11*, 1243. https://doi.org/10.3390/ electronics11081243

Academic Editor: Jorge Pomares

Received: 22 March 2022 Accepted: 12 April 2022 Published: 14 April 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/). **Keywords:** motion control; nonlinear robot control; optimal control; receding horizon control; adaptive control; fixed point iteration; structured cost functions; reduced gradient; SCARA robot

1. Introduction

In control technology, various control methods are present in the inventory of possible solutions. The appropriate choice can be selected according to various particular and practical aspects related to a given task, and there is no way to generally state that a given method would be superior in comparison with others. Such properties as mathematical complexity, the need for computational power, the need for a more or less precise dynamic model of the controlled system, robustness against modeling errors and external disturbances, adaptivity, and possibilities for implementation can be considered when a control approach is chosen to tackle a given problem. In many cases, a simple PID-type controller invented in the 1940s [1] can do well. In robotics, the direct use of the dynamic model without inserting it into the mathematical framework of optimal controllers was initiated in the 1980s [2] in the concept of computed torque control (CTC). This approach had to cope with the problem of the existence of only imprecise dynamic models in the next ten years [3]. The robust variable structure/sliding mode controllers that became popular in the 1990s (e.g., [4]) are simple solutions that can solve the problem of modeling errors and unknown external disturbance. In a similar manner, resolved acceleration rate control (e.g., [5]) and acceleration feedback controllers (e.g., [6]) can be considered as improvements of the CTC controllers. The model-based back-stepping control (e.g., [7]) is sensitive to modeling imprecisions. The wide class of adaptive controllers tackle the imprecisions of the models in different manners. The widest subset uses Lyapunov's stability theorems

and keeps prevailing from the early 1990s to the present (e.g., [8]) as well as the model reference adaptive control (e.g., [9]). Normally, stability or asymptotic stability of these solutions are guaranteed for a huge set of possible parameters of which the appropriate ones can be selected on the basis of practical aspects often applying various versions of evolutionary computation as genetic algorithms, particle swarm optimization, simulated annealing, and so on. The comparison of the operation of these methods cannot be the subject of the present paper. In the sequel, we concentrate on the subset of the wide set of optimal controllers, the heuristic receding horizon control that allows the limitation of the control forces in its cost function. Furthermore, we consider its adaptive extension based on an alternative of the Lyapunov function-based approach.

Strong non-linearity is a natural feature of most physical, biological, economic, and engineering systems and most traditional software packages solving optimization problems can normally handle only linear time-invariant system models with typically quadratic cost function structures because this restricted subject area can be tackled by well-elaborated and efficient mathematical tools as the application of Riccati equations [10] (it provides the solution of special first-order quadratic differential equations by solving second-order linear ones), Schur's decomposition method that obtains the solution of quadratic matrix equations by solving linear ones [11,12]. For solving linear matrix inequalities in system and control theory a complete program was announced by Boyd et al. in 1994 [13] for which efficient MATLAB program packages have been developed [14]. The mainstream of the engineering research efforts aimed at the elaboration of approximate linear system models and quadratic cost functions for tackling optimization problems by the use of this efficient mathematical apparatus.

In [15], the linear matrix inequality (LMI) condition based on slack variables was used to reduce the high gains of controlling, resulting in using the robust \mathcal{H}_{∞} state feedback controllers. Although the study [16] proposed a new condition that is presented in the form of linear matrix inequality for designing the output feedback \mathcal{H}_{∞} controller. The published paper [17] used the formulation of LMI to design the quantized event-triggered tracking controller that guarantees the \mathcal{H}_{∞} tracking performance and maintain the system asymptotically stable.

However, for more complex dynamical models and specially structured cost functions the more general mathematical context does not allow such relatively "simple solutions". Instead of using ready-made program packages researchers have to develop their own program codes that are not supported by the rigorous and reliable quality guaranties of the MATLAB packages.

From a mathematical point of view, optimization can be formulated by the use of variation calculus. In the 1950s, i.e., in the advent of the appearance of powerful computers, Bellman introduced dynamic programming [18] that computationally is too greedy. The problem was later simplified by the introduction of a discrete, evenly scaled time-grid of resolution δt that is dense enough to allow numerical differentiation and Euler integration over it. The sum of the cost function contributions in the grid points of a horizon of discrete length H was minimized for a first-order dynamical system under the constraint $\frac{q(t_{i+1})-q(t_i)}{\delta t} \approx \dot{q}(t_i)$ in which the function $\dot{q}(t_i) = \mathcal{F}(q(t_i), u(t_i))$ describes the dynamic model of the controlled system, and $u(t_i)$ denotes the control force. By the use of the usual constraint function $g_i(q(t_i), q(t_{i+1}), u(t_i)) := \frac{q(t_{i+1}) - q(t_i)}{\delta t} - \mathcal{F}(q(t_i), u(t_i))$, a general cost function (with a simpler notation) $\sum_{\ell=1}^{H} \Phi(q_{\ell}, u_{\ell})$ has to be minimized over the horizon by varying the coordinates $\{q_2, \ldots, q_H\}$ (q_1 is given as the initial condition of the motion), and force terms $\{u_1, \ldots, u_{H-1}\}$ (because u_H has influence only on the next grid point at time t_{H+1}). The optimization must have done under the constraints $g_i(q_i, q_{i+1}, u_i) = 0$. It traditionally can be solved by the use of Lagrange's reduced gradient method by using Lagrange multipliers for gradient reduction that was introduced in the late 18th century for solving constrained problems in classical mechanics [19]. Later, it obtained ample applications from the 1960s with the development of computer technology that provided easy implementation possibilities (e.g., [20,21]). The scheme description is known as the

receding horizon control (RHC) which is a reliable, heuristic practical tool that has many applications (e.g., [22–27]). The adaptive version of RHC were investigated in many cases, such as in [28] wherein the used ARHC is based on Lyapunov's adaptation law, whereas in [29], the adaptive controller is based on the set membership identification algorithm, which iteratively calculates at each cycle a set of candidate plant models. The general ARHC is used in [30] along with particle swarm optimization (PSO). Implementing the sliding mode (SM) as an adaptive technique for ARHC is addressed in [31].

Because of the fact that the Lagrange multipliers normally have clear physical interpretation (e.g., [32]), and the strong analogy with the canonical equations of classical mechanics that provides solutions similar to the the flow of incompressible fluids, together with the plausible mathematical consequences of this approach, the constraint-based formulation of the problem generally prevailed, though it is not the computationally simplest and cheapest approach. These analogies are derived from considering the auxiliary function of the problem in (1)

$$\Psi(\{q\},\{\lambda\},\{u\}) := \sum_{\ell=1}^{H} \Phi(q_{\ell},u_{\ell}) - \sum_{\ell=1}^{H-1} \lambda_{\ell} g_{\ell}(q_{\ell},q_{\ell+1},u_{\ell}) \quad .$$
(1)

Evidently, $\Psi(\{q\}, \{\lambda\}, \{u\})$ is not bounded, and at the point where the gradient reduction algorithm stops, it satisfies the equations as $\frac{\partial \Psi}{\partial \lambda_j} = 0$, meaning that the solution satisfies the constraint conditions, $\frac{\partial \Psi}{\partial q_k} = 0$ that can be so interpreted that the reduced gradient is 0, and an additional condition $\frac{\partial \Psi}{\partial u_i} = 0$. These partial derivatives allow the interpretation of the appearance of the numerical approximation of a differential equation for λ , considering the q_i and λ_i pairs as canonical coordinate pairs, and interpreting Ψ as a Hamiltonian with the conservation property' $\dot{\Psi} \equiv 0$. The analogy with the flow of incompressible fluids is related to the fact that the canonical state propagation equations are related to symplectic transformations that conserve the volume of the phase space (Liouville's theorem, e.g., [33]).

The numerical algorithm that solves the above problem is commenced by finding a point on the constraint surface by using the Newton–Raphson algorithm [34], then making consecutive "small steps" along the reduced gradient $\nabla \Phi - \sum_{\ell} \lambda_{\ell} \nabla g_{\ell}$ in which the Lagrange multipliers are so chosen that for the constraint gradients it must be valid that $\forall j (\nabla g_j)^T (\nabla \Phi - \sum_{\ell} \lambda_{\ell} \nabla g_{\ell}) = 0$ (in this formulation the symbol ∇ contains $\frac{\partial}{\partial q}$ and $\frac{\partial}{\partial u}$ components). Gradient reduction needs the solution of this linear set of equations. The algorithm stops when the reduced gradient becomes zero.

It was realized that placing the dynamic model into the constraint term of the optimization task is rather a tradition than a necessity. If we do not insist on the above mentioned elegant formal analogies with classical mechanics, the complexity of the calculations can be considerably reduced. In the original approach the free variables of the optimization are the coordinate values $\{q\}$, and the force terms $\{F\}$ over the horizon, and the quantities that additionally have to be calculated are the $\{\lambda\}$ Lagrange multipliers for reduction of the gradient containing the partial derivatives according to the components $\{q\}$ and $\{F\}$. In [35], the structure of the auxiliary function was investigated in the case of a simple paradigm, and it was found that the appropriate solution is at its saddle point. Furthermore, instead of using a set of individual constraint functions for optimization as $\{g_{\ell} = 0\}$, the use of a single constraint term defined as $G := \sum_{\ell} g_{\ell}^2 = 0$ can be successfully applied with only one associated Lagrange multiplier that can very easily be computed. In [36], the use of the Lagrange multipliers was completely evaded, and the method's operation was illustrated by controlling the dynamic model of two connected mass points that were able to move in a given linear direction. In this approach, the free variables of the optimization are only the force terms $\{F\}$ over the horizon, the gradient in the optimization consists only of the $\frac{\partial}{\partial F}$ components, and the simple gradient descent method can be applied without any gradient reduction. Following this simple illustration, the method was used for simulating the treatment of illness type 1 diabetes mellitus in determining the necessary insulin ingress

rate, and the estimation of the evolution of the not observable internal model variables. In [37], this approach was considered for the RHC control of the Furuta pendulum [38], and in [39,40] application possibilities were considered in solving the inverse kinematic task of redundant robots. In this paper the application of our method is considered for a special robot model, the SCARA robot with specially designed cost function structure. The paper is structured as follows. In Section 2 the description of the optimization task without using constraint terms is expounded. In Section 3 the cost function's structure is detailed. The adaptive controller part described in Section 4. The simulation results are discussed in Section 5 in two major parts: without (Section 5.1) and with (Section 5.2) control force limitation. Finally, the purpose of the designed cost function with the used optimal adaptive control conclusions are addressed in the discussion placed into Section 6.

2. Formulation of the Optimization Problem without Using Constraint Terms

The direct formulation of the problem without the use of constraint terms for a second order system can be built up by the use of the principle of causality, the available approximate dynamic model, and the forward differences as follows.

- It is given the horizon for design in the discrete time steps $\{t_1, t_2, t_3, ..., t_H\}$ containing the state variables q, \dot{q} as $\{q_1, q_2, q_3, ..., q_H\}$, and $\{\dot{q}_1, \dot{q}_2, \dot{q}_3, ..., \dot{q}_H\}$, the exerted control forces $\{F_1, F_2, F_3, ..., F_H\}$, and the second coordinate time derivatives as $\{\ddot{q}_1, \ddot{q}_2, \ddot{q}_3, ..., \ddot{q}_H\}$;
- in the discrete time representation the above data contain certain redundancy because the relationship $\dot{q}_i \approx \frac{q_{i+1}-q_i}{\delta t}$ must be valid for $i \in \{1, 2, ..., H-1\}$, and $\ddot{q}_i \approx \frac{q_{i+2}-2q_{i+1}+q_i}{\delta t^2}$ for $i \in \{1, 2, ..., H-2\}$;
- the dynamic model of the system determines $\ddot{q}_i = \mathcal{F}(q_i, \dot{q}_i, F_i)$; and
- the initial conditions are determined by $\{q_1, q_2\}$ redundantly with $\{\dot{q}_1\}$.

According to the above considerations, by the use of the dynamic model, the independent variable of optimization F_1 with the initial conditions $\{q_1, q_2\}$ and the redundant $\{\dot{q}_1\}$ determine \ddot{q}_1 that determines q_3 and redundantly \dot{q}_2 . The next independent variable of the problem, i.e., F_2 with $\{q_2, q_3\}$ and the redundant $\{\dot{q}_2\}$ together determine \ddot{q}_2 that determines q_4 and redundantly \dot{q}_3 , etc. The last independent variable is F_{H-2} with $\{q_{H-2}, q_{H-1}\}$ and the redundant $\{\dot{q}_{H-2}\}$ that together determine \ddot{q}_{H-2} that determines q_H and redundantly \dot{q}_{H-1} .

In the mentioned approach, the independent variables are only $\{F_1, \ldots, F_{H-2}\}$, i.e., the number of the independent variables is much smaller than that in the problem formulated by the constraint-based approach. Furthermore, in the simulation program simple functions can be used that according to the initial conditions with the independent variables build up all the important values within the horizon. The cost function afterward can be computed from these values, and in principle the simple gradient descent method can be applied for the independent variables for optimization.

3. Solution for the Minimization

Both the simple gradient descent and the reduced gradient methods generally suffer from the lack of reliable information with regard to the question of what is the appropriately small step that simultaneously provides fast convergence, and good precision. It is well known that the Newton–Raphson algorithm, that can be applied if it is known that the minimum is zero, produces fast convergence. However, when the minimum is not zero it becomes divergent and is apt to make finite jumps around the argument of the minimum. In this paper, the following simple "tricks" were performed:

1. The minimization was commenced from zero force components, and started with a big step using the Newton–Raphson algorithm. A parameter $\alpha = 1$ was introduced in the first step.

- 2. Until the Newton–Raphson algorithm with the given step-length yielded a smaller cost value than the previously visited point, it proceeded forward with this big step length;
- 3. When in the suggested next point the function value found was bigger than that in the starting point, α has been halved, and α times the step-length of the Newton–Raphson algorithm was applied. If it yielded a better point, the calculated step-length was used; otherwise α was kept halved until it either provided a better next point, or it achieved a preset minimum value α_{min} at which point the algorithm stopped.

The speed of optimization certainly strongly depends on the properties of the cost functions in use. In [41,42] very complex cost functions were introduced that were based on polynomial behavior of the functions $\left(\frac{|x|}{\Delta}\right)^p$. If p > 1 such functions have little contribution if $|x| \ll \Delta > 0$ (this is an "error tolerant region"), and drastically increase as $|x| \gg \Delta$ (it corresponds to a "strongly penalized region"). However, numerical problems often arose. In this paper, for the generalized coordinate q_j the contribution in the function was set

as $C_{q_j} \frac{\left|(q_j^N(t)-q_j^O(t))/\delta\right|^{p+1}}{1+\left|(q_j^N(t)-q_j^O(t))/\Delta\right|^p}$. For the big tracking error, its graph looks like that of a linear function to evade numerical problems, whereas for small errors it is tolerantly flat and small (typical shapes are given in Figure 1). For the penalization of the driving force term

the appropriate contributions were given in the form $C_{u_j} \begin{cases} |F_j - \Delta_u|^{p_u} \text{ if } F_j > \Delta_u \\ 0 \text{ if } -\Delta_u \le F_j \le \Delta_u \\ |F_j + \Delta_u|^{p_u} \text{ if } F_j < -\Delta_u \end{cases}$. This

means that a range of forces were exerted without any penalization.



Figure 1. The schematic structure of the cost function shape for penalization (for arbitrary physical dimensions) $\delta = 10^{-5}$, $\Delta = 2000\delta$, and various *p* values: (**a**) in full scale; (**b**) zoomed in excerpt.

In this manner, a relatively fast algorithm was created for seeking the possible minimum of the cost function, because the linear part gives very fast convergence in the Newton–Raphson algorithm.

4. Utilization of the Result of Cost Minimization

Taking into account that the above minimization technique can yield strongly scattering force values, the optimization was realized for a very "heavy" model. In the dynamic models used in robotics the kinematic model parameters normally are precisely known but only imprecise knowledge is available on the distribution of the masses in the space. This fact was an issue in the use of the available dynamic model of the robot in the computed torque control of a PUMA robot in 1986 [2]. Various investigations in the 1990s made it clear that it is not possible to obtain very precise models and some consensus was achieved in [3]. However, it can be assumed that for achieving a given acceleration of the robot's components, for higher masses higher force or torque components are needed. Furthermore, for working against a higher gravitational acceleration higher force or torque components are also needed. For this reason, in Table 1, the length data are identical for the "heavy" and the "exact" models, but the masses, the inertia moments, and the gravitational acceleration are overestimated for the heavy model. Intuitively it can be expected that if a motion can be realized for the heavy components, it can be realized, too, if less weighty components have to be moved along the same trajectory.

Therefore, the force values that were calculated in the optimization process were dropped and the smoothed version $q^{Os}(t)$ of the optimized trajectory $q^{O}(t)$ was adaptively tracked by the use of an available approximate dynamic model of the actual system under control. For smoothing, a third-order solution inspired by [43] was applied as

$$\left(\Lambda_{filt} + \frac{\mathrm{d}}{\mathrm{d}t}\right)^3 q^{Os}(t) = \Lambda_{filt}^3 q^O(t), \ q^{Os}(t_0) = 0, \ \dot{q}^{Os}(t_0) = 0, \ \ddot{q}^{Os}(t_0) = 0, \ (2)$$

with $0 < \Lambda_{filt} = \text{constant}$ For very high frequencies it has the transfer function characteristic in the Laplace transform as $\propto s^{-3}$ (very drastic rejection), and at zero frequency it has the value 1.

This smoothed function was adaptively tracked by a fixed point iteration-based adaptive controller. This design is a further development of the computed torque control method that directly uses the dynamic model of the robot [2] without including it in the mathematical framework of the optimal controllers. It contains a kinematic block in which a special PID-type tracking strategy is defined in (3) as

$$\left(\Lambda + \frac{\mathrm{d}}{\mathrm{d}t}\right)^{3} e_{Int}(t) \equiv 0 \Rightarrow$$

$$\ddot{q}^{Des}(t) = \Lambda^{3} e_{Int}(t) + 3\Lambda^{2} e(t) + 3\Lambda \dot{e}(t) + \ddot{q}^{o}(t) ,$$
(3)

in which each feedback gain is determined by the use of a single positive constant parameter Λ , and the integrated error $e_{Int}(t) := \int_{t_0}^t (q^N(\xi) - q(\xi)) d\xi$. In simulations, the output of the kinematic block was moderated by the function $\ddot{q}^{Des} = \ddot{q}_{maxk} \tanh(\ddot{q}^{Des}_{Unmod}/\ddot{q}_{maxk})$ that limited its possible output approximately at \ddot{q}_{maxk} to evade numerical difficulties. This strategy has the property that following a relatively large initial overshoot and undershoot it asymptotically converges to zero. In the possession of the exact dynamic model this desired value directly can be introduced into this model to compute the necessary control forces. In the possession of only an approximate model, as the first element of a sequence, in the beginning of the control action, it is introduced into the approximate model. In this manner, an adaptive controller can be created that has a simpler mathematical background than the Lyapunov function-based design, and directly tackles the convergence of the individual error components to zero. (In most cases the Lyapunov function-based approach guarantees only the convergence of some quadratic norm made of these components. Non-quadratic Lyapunov functions can be constructed, too).

The adaptive deformation depends on the deformed signal applied in the previous step and on the observed response obtained by it (details are given in [44]). Under certain conditions, the sequence of the deformed signals converges to a fixed point that provides $\ddot{q}^{Des} \approx \ddot{q}$, i.e., in spite of the modeling errors, the kinematic tracking strategy is well approximated. The mathematical background of the convergence is based on Banach's Fixed Point Theorem' of contractive maps that map a Banach Space (i.e., a linear, normed, complete metric space) into itself [45]. For the realization of the adaptive deformation various "deformation functions" can be invented (e.g., given in [44,46–48]). In the present paper the version given in [48] is applied that so augments the array $\ddot{q}^{Des}(t) \in \mathbb{R}^4$ into $A \in \mathbb{R}^5$, the array $\ddot{q}(t - \delta t) \in \mathbb{R}^4$ array into $B \in \mathbb{R}^5$ that in the Frobenius sense $||A|| = ||B|| = R_a$, then constructs a rotation that rotates *B* into *A* with the angle φ , then augments the array $\ddot{q}^{Def}(t - \delta t) \in \mathbb{R}^4$ into $C \in \mathbb{R}^5$ so that $||C|| = R_a$, then rotates *C* with an interpolated rotational angle $\lambda_a \varphi$ ($\lambda_a \in (0, 1]$), and finally returns the first four components of this rotated array. Before returing, these components were moderated for very big (i.e., of the order of magnitude of $|\ddot{q}| \geq \ddot{q}_{maxa}$ values) by the function $\ddot{q}^{Def} = \ddot{q}_{maxa} \tanh(\ddot{q}^{Def}_{Proj}/\ddot{q}_{maxa})$.

As a result, in the physically interpreted projection space, i.e., in \mathbb{R}^4 , simultaneous rotation and shrinking/dilatation happens with the "deformed value" to be used in the time instant t. The approximate model and the actually controlled system approximately defines a response function' $\ddot{q} = f(\ddot{q}^{Def})$ that only slightly drifts with the state variables (q, \dot{q}) whereas the controller quickly and abruptly can modify its argument $\ddot{q}^{Def}(t)$. As a generalization of the idea of "monotonic increasing function" that realizes a mapping $\mathbb{R} \mapsto \mathbb{R}$, function $f(x) : \mathbb{R}^n \mapsto \mathbb{R}^n$ is called approximately direction-keeping if for an infinitesimally small $\Delta x \in \mathbb{R}^n$ the angle between the vectors $\Delta f := f(x + \Delta x) - f(x)$ and Δx is acute. Practically it means that a small modification of the input causes similar modification of the output. For instance, in car driving, this behavior is valid for the operation of the steering wheel, the brake, and the accelerator pedals, and for this reason an adaptive system such as a human driver can learn how to drive a car. It also gives satisfactory basis for the convergence of the deformed signals in the fixed point iteration process. The whole control process for having each optimal coordinate q_i^O , filtered optimal coordinate q_i^{Os} and the realized one q_i can be recognized through the flow chart in Figure 2.



Figure 2. The flow chart of the controlling mechanism.

In this paper, the above ideas were applied for a 4 DoF SCARA robot arm the dynamic model of which was taken from [49]. The generalized coordinates of the robots are q_1 [m] is the only prismatic joint, and q₂, q₃, q₄ are rotary ones measured in [rad] units. Accordingly, the generalized forces are F_1 [N] for the first joint, and F_2 , F_3 , F_4 have the dimension [N · m]. The equations of motion are given in (4)–(7),

$$F_1 = (m_1 + m_2 + M + M_{load})\ddot{q}_1 - g(M + m_1 + m_2 + M_{load}) , \qquad (4)$$

$$F_{2} = (m_{1}L_{1}^{2}/4 + m_{2}L_{1}^{2} + m_{2}L_{2}^{2}/2 + m_{2}L_{1}L_{2}\cos(q_{3}) + M_{load}(L_{1}^{2} + L_{2}^{2} + 2L_{1}L_{2}) + \Theta_{load})\ddot{q}_{2} + (m_{2}L_{2}^{2}/2 + m_{2}L_{1}L_{2}\cos(q_{3})/2 + M_{load}(L_{2}^{2} + L_{1}L_{2}\cos(q_{3})) + \Theta_{load})\ddot{q}_{3} + \Theta_{load}\ddot{q}_{4} - \dot{q}_{2}(m_{2}L_{1}L_{2}\sin(q_{3}) + M_{load}2L_{1}L_{2}\sin(q_{3}))\dot{q}_{3} - \dot{q}_{3}^{2}(m_{2}L_{1}L_{2}\sin(q_{3}) + M_{load}L_{1}L_{2}\sin(q_{3})) ,$$
(5)

$$F_{3} = (m_{2}L_{2}^{2}/2 + m_{2}L_{1}L_{2}\cos(q_{3})/2 + M_{load}(L_{2}^{2} + L_{1}L_{2}\cos(q_{3})) + \Theta_{load})\ddot{q}_{2} + (m_{2}L_{2}^{2}/4 + M_{load}L_{2}^{2} + \Theta_{load})\ddot{q}_{3} + \Theta_{load}\ddot{q}_{4} + \dot{q}_{2}^{2}L_{1}L_{2}sin(q_{3})(m_{2}/2 + M_{load}) ,$$

$$F_{4} = \Theta_{load}\ddot{q}_{2} + \Theta_{load}\ddot{q}_{3} + \Theta_{load}\ddot{q}_{4} + \Theta_{load}\ddot{q}_{4} .$$
(6)

$$F_4 = \Theta_{load}\ddot{q}_2 + \Theta_{load}\ddot{q}_3 + \Theta_{load}\ddot{q}_4 \quad . \tag{7}$$

The dynamic parameters of the heavy model used for optimization, the exact system model (not known by the controller), and the available approximate system model used for trajectory tracking are given in Table 1.

Parameter	Exact Model for Simulation	Heavy Model for Optimization	Approximate Model for Adaptive Control
M[kg] component's mass	10.0	15.0	12.0
$m_1[kg]$ component's mass	20.0	25.0	21.0
$m_2[kg]$ component's mass	10.0	13.0	12.0
M _{load} [kg] load's mass	50.0	55.0	52.0
$g[\mathbf{m} \cdot \mathbf{s}^{-2}]$ grav. accel.	9.81	10.0	9.0
$\theta_{load} [kg \cdot m^2]$ load's inertia moment	45.0	50.0	42.0
$L_1[m]$ arm length	2.0	2.0	2.0
$L_2[m]$ arm length	1.0	1.0	1.0

Table 1. The Dynamic Model Parameters in Equations (4)–(7).

5. Simulation Results

The necessary time-grid resolution depends on various factors such as the dynamics of the nominal trajectory to be tracked, the structure of the cost function applied, the parameters used in smoothing the optimized trajectory, and that of the adaptive controller that tracks the smoothed trajectory. In general, the precise mathematical analysis of these factors is quite time-consuming. From a practical point of view, making numerical simulations for a given problem seems to be an easier way. A simple check of reliability is the comparison of the results obtained for the sets { $\delta t = 10^{-4} s$, STEPS = 4000, HL = 12} (referred to as set 1) and $\{0.5\delta t, 2 \cdot STEPS, 2 \cdot HL\}$ (referred to as set 2) that physically corresponds to computing the same task with a finer time resolution. If the results obtained for the tracking precision and the control force needs can be well compared to each other, the original time resolution δt can be considered as acceptable. All simulations were made with a Dell vostro 1540 laptop operated by a CORETMi3 Intel processor under the Windows 10 Home 64-bit operating system. The simulation consists of two parts. The first part (in Section 5.1) was tested without force limitation and has two sub-simulations as seen in Section 5.1.1 where set 1 and set 2 are compared. In Section 5.1.2, we see the effect of increasing the stopping limit α . The second simulation part was tested with the limited forces by choosing the proper of the two examined sets in addition to the better value of α . Table 2 shows the controller parameters values that used during the all simulation parts.

		, arec
δt	Discrete time resolution	$10^{-4}[s]$
Λ	Trajectory tracking exponential coeff.	$36.0 [s^{-1}]$
Λ_{filt}	Trajectory smoothing exponential coeff.	800.0 $[s^{-1}]$
$C_{q_1} = C_{q_3} = C_{q_4}$	Cost contribution coeffs.	10 ⁶
C_{a_2}	Cost contribution coeff.	$2 imes 10^6$
δ^{12}	Cost parameter 1	10^{-5} [rad] or [m]
Δ	Cost parameter 2	$2000 \cdot \delta$
$C_{u_1} = C_{u_2} = C_{u_3} = C_{u_4}$	Force cost parameter 1	10^{4}
р	Cost parameter 3	1.1
Δ_u	Force cost parameter 2	varying $[N \cdot m]$ or $[N]$
p_u	Force cost parameter 3	1.5
R_a	Augmented arrays' Frobenius norm	$10^6 \big[m\cdot s^{-2}\big]$ or $\big[rad\cdot s^{-2}\big]$
H	Discrete horizon length	12
<i>q̃maxa</i>	Moderating factor in adaptive control	$10^4 \big[m\cdot s^{-2}\big]$ or $\big[rad\cdot s^{-2}\big]$
<i>q̃_{maxk}</i>	Moderating factor in kinematic block	$10^7 \bigl[m\cdot s^{-2}\bigr]$ or $\bigl[rad\cdot s^{-2}\bigr]$
α_{min}	Stopping limit in minimum seeking	10^{-2}
λ_a	Adaptive interpolation factor	1.0

Meaning

Value

Table 2. The Controller's Parameters.

Parameter

5.1. Simulations without Force Limitation

5.1.1. Investigations for Different Time Resolutions

In the simulation investigations, the operation of the method at first was considered for a cost function without control force penalizing terms (in this case $\forall j C_{u_j} = 0$). The common control parameters of these simulations are given in Table 2. Some sinusoidal motion was chosen for the nominal trajectory to be tracked.

It can be seen in Figures 3–6 that halving the time-resolution $\delta t = 10^{-4}$ did not produce significantly different results. The figures of the second time derivatives (Figures 7–10) and that of the control forces (Figures 11 and 12) show more significant differences. However, they definitely are in the same order of magnitude and reveal qualitative similarities. Figure 13 shows the perceptible difference of the required optimal forces values for the two sets. The adaptive abstract rotations has no noticeable differences (Figure 14). Naturally, the computational burden of the method strongly depends on the time resolution which can be noticed in the big differences of the computational time need in Figure 15. On this basis, it was determined that for a preliminary design the computationally less greedy set 1 parameters' setting will be used in the case of force limitations. It worth mentioning that the big initial transient is a typical consequence of a PID-type tracking policy formulated in (3).



Figure 3. Trajectory tracking without force limitation for q_1 and q_2 : (a) For the set 1. (b) For the set 2.



Figure 4. Trajectory tracking without force limitation for q_3 and q_4 . (a) For the set 1. (b) For the set 2.



Figure 5. Nominal optimized trajectory tracking errors $q^N - q^O$ without force limitation. (a) For set 1. (b) For set 2.



Figure 6. Optimized realized trajectory tracking errors $q^O - q$ without force limitation. (a) For set 1. (b) For set 2.



Figure 7. Second time derivatives without force limitation for the joint coordinates $q_1(t)$ and $q_2(t)$ in full scale. (a) For set 1. (b) For set 2.



Figure 8. Second time derivatives without force limitation for the joint coordinates $q_1(t)$ and $q_2(t)$ without the initial transient. (a) For set 1. (b) For set 2.



Figure 9. Second time derivatives without force limitation for the joint coordinates $q_3(t)$ and $q_4(t)$ in full scale. (a) For set 1. (b) For set 2.



Figure 10. Second time derivatives without force limitation for the joint coordinates $q_3(t)$ and $q_4(t)$ without the initial transient. (a) For set 1. (b) For set 2.



Figure 11. The adaptive control forces without force limitation in full scale. (a) For set 1. (b) For set 2.



Figure 12. The adaptive control forces without force limitation and without the initial transient. (a) For set 1. (b) For set 2.



Figure 13. The optimal forces without force limitation and without the initial transient. (**a**) For set 1. (**b**) For set 2.



Figure 14. The adaptive abstract rotations. (a) For set 1. (b) For set 2.



Figure 15. The computational time need of the main cycle without force limitation. (**a**) For set 1. (**b**) For set 2.

5.1.2. Computations for Less Precise Minimum Seeking

Based on previous comparison, set 1 will be chosen for the upcoming investigations. In the sequel, to decrease computational burden, the setting $\alpha_{min} = 10^{-1}$ value was used. This corresponds to a less precise approximation of the local minimum, but the noise filtering applied for adaptive tracking can tackle the effects of this increased imprecision. According to Figures 16 and 17, it can be stated that the RHC controller was able to generate a good optimized trajectory that was successfully tracked by the adaptive controller. Figures 18 and 19 reveal what happens in the background: due to the necessary adaptive deformation it was possible to track the smoothed optimal trajectory well. The strong smoothing depicted in (2) was absolutely necessary and resulted in the adaptive control forces given in Figure 20. The comparison of Figures 15a and 21b reveals quite considerable reduction in the necessary computational time. The optimal forces (Figure 22) are decreased noticeably, whereas the adaptive abstract rotation is within accepted range as it can be seen in Figure 21a.



Figure 16. Trajectory tracking without force limitation for the increased $\alpha_{min} = 0.1$. (a) For q_1 and q_2 . (b) For q_3 and q_4 .



Figure 17. Trajectory tracking errors without force limitation for the increased $\alpha_{min} = 0.1$. (a) For nominal-optimized: $q^N - q^O$. (b) For optimized-realized: $q^O - q$.



Figure 18. Second time derivatives without force limitation for the increased $\alpha_{min} = 0.1$ for the joint coordinates $q_1(t)$ and $q_2(t)$. (a) In full scale. (b) Without the initial transient.



Figure 19. Second time-derivatives without force limitation for the increased $\alpha_{min} = 0.1$ for the joint coordinates $q_3(t)$ and $q_4(t)$. (a) In full scale. (b) Without the initial transient.



Figure 20. The adaptive control forces without force limitation for the increased $\alpha_{min} = 0.1$. (a) In full scale. (b) Without the initial "transient".



Figure 21. The adaptive abstract rotations (**a**) and the computational time-need of the main cycle without force limitation (**b**) for the increased $\alpha_{min} = 0.1$.



Figure 22. The optimal forces without force limitation for the increased $\alpha_{min} = 0.1$.

5.2. With Force Limitation

These simulations were made by keeping the increased $\alpha_{min} = 0.1$ and by choosing the set 1; otherwise the parameters given in Table 2 were in use. The value of the parameter $\Delta_u = 495.0$ N or N · m considerably corrupted the optimized trajectory by not allowing it to exert the necessary control forces. Figures 23 and 24 reveal that in the optimization phase the force limitation seriously concerned the optimized trajectories $q_1^O(t)$, $q_2^O(t)$, $q_3^O(t)$, and $q_4^O(t)$, the smoothed versions of which were adaptively well tracked. Figure 25 shows considerable fluctuations in the optimized forces that, due to the smoothing applied, have limited effect on the adaptive tracking forces in Figure 26. According to Figure 27b the time need of the optimum seeking was kept at a relatively low level.



Figure 23. Trajectory tracking with force limitation $\Delta_u = 495.0$ N or N · m for the increased $\alpha_{min} = 0.1$. (a) For q_1 and q_2 . (b) For q_3 and q_4 .



Figure 24. Trajectory tracking errors with force limitation $\Delta_u = 495.0$ N or N \cdot m for the increased $\alpha_{min} = 0.1$: (a) For nominal-optimized: $q^N - q^O$. (b) For optimized-realized: $q^O - q$.



Figure 25. Second time derivatives with force limitation $\Delta_u = 495.0$ N or N · m for the increased $\alpha_{min} = 0.1$ (**a**) for the joint coordinates $q_1(t)$ and $q_2(t)$, (**b**) for the joint coordinates $q_3(t)$ and $q_4(t)$.



Figure 26. The control forces with force limitation $\Delta_u = 495.0$ N or N \cdot m for the increased $\alpha_{min} = 0.1$. (a) The adaptive control forces. (b) The optimized forces.



Figure 27. The adaptive abstract rotations (**a**) and the computational time need (**b**) of the main cycle with force limitation $\Delta_u = 495.0$ N or N · m for the increased $\alpha_{min} = 0.1$.

It is interesting to observe how sharply the parameter value Δ_u concerns the results for $p_u = 1.5$. If the limits of the penalty-free force region are increased from $\Delta_u = 495.0$ to $\Delta_u = 500.0$ N or N · m the optimized trajectory suffered far fewer distortions (Figure 28).



Figure 28. Trajectory tracking with force limitation $\Delta_u = 500.0$ N or N \cdot m for the increased $\alpha_{min} = 0.1$: (a) For q_1 and q_2 . (b) For q_3 and q_4 .

6. Discussion

The detailed steps in Sections 2–4 and simulation results can be summarized as follows:

- 1. In general it seems to be an interesting research area to consider the problem of adaptive optimal control design for a wider set that is exempt of the limitations of linear time-invariant dynamic system models and quadratic cost function contributions.
- 2. The prevailing general approach in this field is linear programming that tackles the problem in discrete time grid approach and the use of Lagrange's reduced gradient algorithm that is professionally implemented e.g., in the EXCEL's Solver package.

- 3. It is evident that the direct problem formulation applied for evading the use of the constraint-based formalism leads to considerable reduction in complexity and computational needs. The number of the independent variables of the original approach is considerably decreased. Instead of the variables $\{q\}, \{u\}$, only the variables $\{u\}$ remain the independent ones, and there is no need to calculate the Lagrange multipliers $\{\lambda\}$.
- 4. The problem of small steps in the case of the suggested solution can be tackled by a modification of the Newton–Raphson algorithm (generally it is an issue in using the reduced gradient method, too).
- 5. It was found that for keeping the computational time low, it is expedient to use not very precise minimum seeking.
- 6. The scattering of the force values and the related effects rather can be tackled by a simple noise-filtering approach applied for the optimized trajectory to be adaptively tracked.
- 7. The suggested adaptive controller can well track the smoothed signal.
- 8. The mathematical frameworks of optimization and adaptive tracking can be separated from each other in a simple manner.
- 9. The basic concept was that the force-limited optimization can be executed by the use of a heavy dynamic model. Therefore, the force limitation issues can be tackled in the optimization phase. It can be expected that the trajectory that was optimized for the heavy model can be tracked by an easier mechanical construction for the acceleration of the components of which smaller force or torque components can be expected. For tracking the optimized trajectory, a simple CTC type or an improved adaptive CTC type control strategy can be used that is free of the burden of the force-limitation issues.
- 10. The necessary time grid resolution depends on various factors as the dynamics of the nominal trajectory to be tracked, the structure of the cost function applied, the parameters used in smoothing the optimized trajectory, and that of the adaptive controller that tracks the smoothed trajectory.
- 11. In general all the above factors can be clarified via making numerical simulations for a given problem or problem class.
- 12. In the given investigations, the execution time was measured by the use of the given hardware that was a laptop with a single CPU and a multitasking operating system. Consequently, the measured data also contain the time sections during which the actual task was interrupted and the processor was working on other tasks. However, because during the calculations no heavy software application was running, these data provide approximate and reliable information. (Parallel or simultaneous'use for instance of a video player drastically modifies these observed data.)
- 13. The PID-based tracking policy still has huge initial signal swinging that should be reduced. In the literature, various fractional order controllers can be found that tackle this and similar problems (e.g., [50–56]). More general information can be obtained from resources [57,58].

On the basis of these considerations in the future works, it seems to be expedient to make investigations for various cost function types, dynamical systems, and nominal trajectory types, and hardware possibilities for realization.

Author Contributions: Conceptualization, H.I.; software, H.I.; validation, J.K.T., formal analysis, J.K.T.; writing—original draft preparation, J.K.T.; writing—review and editing, H.I.; supervision, J.K.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: Authors are highly acknowledge the support of the Doctoral School of Applied Informatics and Applied Mathematics, Óbuda University Budapest, Hungary.

Conflicts of Interest: The authors declare that they have no conflict of interest.

References

- 1. Bennett, S. Nicholas Minorsky and the automatic steering of ships. IEEE Control Syst. Mag. 1984, 4, 10–15. [CrossRef]
- Armstrong, B.; Khatib, O.; Burdick, J. The Explicit Dynamic Model and Internal Parameters of the PUMA 560 Arm. In Proceedings of the IEEE International Conference on Robotics and Automation, San Francisco, CA, USA, 7–10 April 1986; pp. 510–518.
- Corke, P.; Armstrong-Helouvry, B. A Search for Consensus Among Model Parameters Reported for the PUMA 560 Robot. In Proceedings of the IEEE International Conference on Robotics and Automation, San Diego, CA, USA, 8–13 May 1994; pp. 1608–1613.
- 4. Mohd Zaihidee, F.; Mekhilef, S.; Mubin, M. Robust speed control of PMSM using sliding mode control (SMC)—A review. *Energies* 2019, 12, 1669. [CrossRef]
- Sagara, S.; Ambar, R. Performance comparison of control methods using a dual-arm underwater robot-Computed torque based control and resolved acceleration control for UVMS. In Proceedings of the 2020 IEEE/SICE International Symposium on System Integration (SII), Honolulu, HI, USA, 12–15 January 2020, pp. 1094–1099.
- Hamandi, M.; Tognon, M.; Franchi, A. Direct acceleration feedback control of quadrotor aerial vehicles. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020, pp. 5335–5341.
- Bodó, Z.; Lantos, B. Integrating Backstepping Control of Outdoor Quadrotor UAVs. *Period. Polytech. Electr. Eng. Comput. Sci.* 2019, 63, 122–132. [CrossRef]
- Yuan, S.; Lv, M.; Baldi, S.; Zhang, L. Lyapunov-equation-based stability analysis for switched linear systems and its application to switched adaptive control. *IEEE Trans. Autom. Control* 2020, *66*, 2250–2256. [CrossRef]
- 9. Dogan, K.M.; Yucelen, T.; Haddad, W.M.; Muse, J.A. Improving transient performance of discrete-time model reference adaptive control architectures. *Int. J. Adapt. Control Signal Process.* **2020**, *34*, 901–918. [CrossRef]
- 10. Riccati, J. Animadversiones in aequationes differentiales secundi gradus (Observations regarding differential equations of the second order). *Actorum Erud. Quae Lipsiae Publicantur Suppl.* **1724**, *8*, 66–73.
- 11. Haynsworth, E. On the Schur Complement. Basel Math. Notes 1968, BMN 20, 17.
- 12. Laub, Alan. A Schur method for solving algebraic Riccati equations. IEEE Trans. Autom. Control 1979, 24, 913–921. [CrossRef]
- 13. Boyd, S.; Ghaoui, L.; Feron, E.; Balakrishnan, V. *Linear Matrix Inequalities in Systems and Control Theory*; SIAM Books: Philadelphia, PA, USA, 1994.
- 14. Gahinet, P.; Nemirovskii, A.; Laub, A.; Chilali, M. The LMI control toolbox. In Proceedings of 1994 33rd IEEE Conference on Decision and Control, Lake Buena Vista, FL, USA, 14–16 December 1994, Volume 3, pp. 2038–2041. [CrossRef]
- Koch, G.G.; Maccari, L.A.; Oliveira, R.C.; Montagner, V.F. Robust H_∞ State Feedback Controllers Based on Linear Matrix Inequalities Applied to Grid-Connected Converters. *IEEE Trans. Ind. Electron.* 2018, 66, 6021–6031. [CrossRef]
- 16. Chang, X.H.; Liu, R.R.; Park, J.H. A Further Study on Output Feedback *H*_∞ Control for Discrete-Time Systems. *IEEE Trans. Circ. Syst.* **2019**, *67*, 305–309.
- 17. Li, Z.M.; Chang, X.H.; Park, J.H. Quantized static output feedback fuzzy tracking control for discrete-time nonlinear networked systems with asynchronous event-triggered constraints. *IEEE Trans. Syst. Man Cyber. Syst.* **2019**, *51*, 3820–3831. [CrossRef]
- 18. Bellman, R. Dynamic Programming and a new formalism in the calculus of variations. *Proc. Natl. Acad. Sci. USA* **1954**, 40, 231–235. [CrossRef] [PubMed]
- 19. Lagrange, J.; Binet, J.; Garnier, J. *Mécanique Analytique (Analytical Mechanics)*; Binet, J.P.M., Garnier, J.G., Eds.; Ve Courcier: Paris, France, 1811.
- 20. Hestenes, M.R. Multiplier and gradient methods. J. Optim. Theory Appl. 1969, 4, 303–320. [CrossRef]
- 21. Powell, M.J. A method for nonlinear constraints in minimization problems. Optimization 1969, 283–298.
- 22. Mayne, D.Q.; Michalska, H. Receding horizon control of nonlinear systems. In Proceedings of the 27th IEEE Conference on Decision and Control, Austin, TX, USA, 7–9 December 1988; pp. 464–465.
- 23. Michalska, H.; Mayne, D.Q. Robust receding horizon control of constrained nonlinear systems. *IEEE Trans. Autom. Control* **1993**, 38, 1623–1633. [CrossRef]
- Bellingham, J.; Richards, A.; How, J.P. Receding horizon control of autonomous aerial vehicles. In Proceedings of the 2002 American control conference (IEEE Cat. No. CH37301), Anchorage, AK, USA, 8–10 May 2002; Volume 5, pp. 3741–3746.
- 25. Cagienard, R.; Grieder, P.; Kerrigan, E.C.; Morari, M. Move blocking strategies in receding horizon control. *J. Process Control* 2007, 17, 563–570. [CrossRef]
- 26. Kuwata, Y.; Richards, A.; Schouwenaars, T.; How, J.P. Distributed robust receding horizon control for multivehicle guidance. *IEEE Trans. Control Syst. Technol.* 2007, 15, 627–641. [CrossRef]
- 27. Mattingley, J.; Wang, Y.; Boyd, S. Receding horizon control. IEEE Control Syst. Mag. 2011, 31, 52–65.
- Igreja, J.; Lemos, J.; Silva, R. Adaptive receding horizon control of a distributed collector solar field. In Proceedings of the 44th IEEE Conference on Decision and Control, Seville, Spain, 15 December 2005; pp. 1282–1287.
- Tanaskovic, M.; Fagiano, L.; Smith, R.; Morari, M. Adaptive receding horizon control for constrained MIMO systems. *Automatica* 2014, 50, 3019–3029. [CrossRef]
- Lukina, A.; Esterle, L.; Hirsch, C.; Bartocci, E.; Yang, J.; Tiwari, A.; Smolka, S.A.; Grosu, R. ARES: adaptive receding-horizon synthesis of optimal plans. In *Proceedings of the International Conference on Tools and Algorithms for the Construction and Analysis of* Systems; Springer: Berlin/Heidelberg, Germany, 2017; pp. 286–302.

- 31. Evangelista, C.A.; Pisano, A.; Puleston, P.; Usai, E. Receding horizon adaptive second-order sliding mode control for doubly-fed induction generator based wind turbine. *IEEE Trans. Control Syst. Technol.* **2016**, *25*, 73–84. [CrossRef]
- 32. Karabulut, H. Physical meaning of Lagrange multipliers. Eur. J. Phys. Gen. Phys. 2007, 27, 709–718. [CrossRef]
- 33. Arnold, V. Mathematical Methods of Classical Mechanics; Springer: Berlin/Heidelberg, Germany, 1989.
- 34. Raphson, J. Analysis Aequationum Universalis (Analysis of Universal Equations); Nabu Press: Paris, France, 1702.
- Issa, H.; Tar, J. Speeding up the Reduced Gradient Method for Constrained Optimization. In Proceedings of the IEEE 19th World Symposium on Applied Machine Intelligence and Informatics (SAMI 2021), Herl'any, Slovakia, 21–23 January 2021; pp. 000485–000490. [CrossRef]
- Redjimi, H.; Tar, J.K. Approximate model-based state estimation in simplified Receding Horizon Control. Int. J. Circ. Syst. Signal Process. 2021, 15, 114–124. [CrossRef]
- Issa, H.; Khan, H.; Tar, J.K. Suboptimal Adaptive Receding Horizon Control Using Simplified Nonlinear Programming. In Proceedings of the 2021 IEEE 25th International Conference on Intelligent Engineering Systems (INES), Budapest, Hungary, 7–9 July 2021; pp. 000221–000228.
- Acosta, J. Furuta's Pendulum: A Conservative Nonlinear Model for Theory and Practise. *Math. Probl. Eng.* 2010, 2010, 1–29. [CrossRef]
- Issa, H.; Varga, B.; Tar, J.K. A receding horizon-type solution of the inverse kinematic task of redundant robots. In Proceedings of the 2021 IEEE 15th International Symposium on Applied Computational Intelligence and Informatics (SACI), Timisoara, Romania, 19–21 May 2021; pp. 000231–000236.
- Issa, H.; Varga, B.; Tar, J.K. Accelerated reduced gradient algorithm for solving the inverse kinematic task of redundant open kinematic chains. In Proceedings of the 2021 IEEE 15th International Symposium on Applied Computational Intelligence and Informatics (SACI), Timisoara, Romania, 19–21 May 2021; pp. 000387–000392.
- 41. Khan, H.; Tar, J.; Rudas, I.; Eigner, G. Adaptive Model Predictive Control Based on Fixed Point Iteration. WSEAS Trans. Syst. Control 2017, 12, 347–354.
- 42. Khan, H.; Tar, J.; Rudas, I.; Eigner, G. Iterative Solution in Adaptive Model Predictive Control by Using Fixed-Point Transformation Method. *Int. J. Math. Models Methods Appl. Sci.* 2018, 12, 7–15.
- 43. Lantos, B.; Bodó, Z. High Level Kinematic and Low Level Nonlinear Dynamic Control of Unmanned Ground Vehicles. *Acta Polytech. Hung.* **2019**, *16*, 97–117.
- Tar, J.; Bitó, J.; Nádai, L.; Tenreiro Machado, J. Robust Fixed Point Transformations in Adaptive Control Using Local Basin of Attraction. Acta Polytech. Hung. 2009, 6, 21–37.
- 45. Banach, S. Sur les opérations dans les ensembles abstraits et leur application aux équations intégrales (About the Operations in the Abstract Sets and Their Application to Integral Equations). *Fund. Math.* **1922**, *3*, 133–181. [CrossRef]
- Dineva, A.; Tar, J.; Várkonyi-Kóczy, A. Novel Generation of Fixed Point Transformation for the Adaptive Control of a Nonlinear Neuron Model. In Proceedigns of the IEEE International Conference on Systems, Man, and Cybernetics, (SMC 2015), Hong Kong, China, 10–13 October 2015; pp. 987–992.
- Dineva, A.; Tar, J.; Várkonyi-Kóczy, A.; Piuri, V. Generalization of a Sigmoid Generated Fixed Point Transformation from SISO to MIMO Systems. In Proceedigns of the IEEE 19th International Conference on Intelligent Engineering Systems, (INES 2015), Bratislava, Slovakia, 3–5 September 2015; pp. 135–140.
- Csanádi, B.; Galambos, P.; Tar, J.; Györök, G.; Serester, A. A Novel, Abstract Rotation-based Fixed Point Transformation in Adaptive Control. In Proceedigns of the 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC2018), Miyazaki, Japan, 7–10 October 2018; pp. 2577–2582.
- 49. Somló, J.; Lantos, B.; Cát, P. Advanced Robot Control; Akadémiai Kiadó: Budapest, Hugary, 2002.
- Monje, C.; Ramos, F.; Feliu, V.; Vinagre, B. Tip position control of a lightweight flexible manipulator using a fractional order controller. *Control Theory Appl. IET* 2007, 1, 1451–1460. [CrossRef]
- Ferreira, N.; Machado, J.T.; Tar, J. Fractional Control of Two Cooperating Manipulators. In Proceedings of the 6th IEEE International Conference on Computational Cybernetics, Stará Lesna, Slovakia, 27–29 November 2008; pp. 27–32.
- 52. Padula, F.; Visioli, A. Tuning rules for optimal PID and fractional order PID controllers. *J. Process Control* 2011, 21, 69–81. [CrossRef]
- Dumlu, A.; Erenturk, K. Trajectory tracking control for a 3-DOF parallel manipulator using fractional-order PI^λD^μ control. *IEEE Trans. Ind. Electron.* 2014, *61*, 3417–3426. [CrossRef]
- 54. Bruzzone, L.; Fanghella, P. Comparison of PDD^{1/2} and PD^μ Position Controls of a Second Order Linear System. In Proceedings of the 33rd IASTED International Conference on Modelling, Identification and Control (MIC 2014), Innsbruck, Austria, 17–19 February 2014; pp. 182–188.
- 55. Folea, S.; De Keyser, R.; Birs, I.R.; Muresan, C.I.; Ionescu, C. Discrete-Time Implementation and Experimental Validation of a Fractional Order PD Controller for Vibration Suppression in Airplane Wings. *Acta Polytech. Hung.* **2017**, *14*, 191–206.
- Tar, J.; Bitó, J.; Kovács, L.; Faitli, T. Fractional Order PID-Type Feedback in Fixed Point Transformation-Based Adaptive Control of the FitzHugh-Nagumo Neuron Model with Time-Delay. In Proceedings of the 3rd IFAC Conference on Advances in Proportional-Integral-Derivative Control, Ghent, Belgium, 9–11 May 2018; pp. 906–911.
- De Oliveira, E.C.; Tenreiro Machado, J.A. A Review of Definitions for Fractional Derivatives and Integral. *Math. Problems Eng.* 2014, 2014, 6. [CrossRef]
- 58. Tenreiro Machado, J.; Kiryakova, V. The Chronicles of Fractional Calculus. Fract. Calc. Appl. Anal. 2017, 20, 307–336. [CrossRef]