Min–Max Optimal Control of Robot Manipulators Affected by Sensor Faults

Milić, Vladimir; Kasać, Josip; Lukas, Marin

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:235:932535

Download date / Datum preuzimanja: **2025-01-07**

Repository / Repozitorij:

Repository of Faculty of Mechanical Engineering and Naval Architecture University of Zagreb

Article

Min–Max Optimal Control of Robot Manipulators Affected by Sensor Faults

Vladimir Milić *[iD], Josip Kasać [iD] and Marin Lukas [iD]

Faculty of Mechanical Engineering and Naval Architecture, University of Zagreb, HR-10000 Zagreb, Croatia
* Correspondence: vladimir.milic@fsb.hr; Tel.: +385-(0)1-6168-505

Abstract: This paper is concerned with the control law synthesis for robot manipulators, which guarantees that the effect of the sensor faults is kept under a permissible level, and ensures the stability of the closed-loop system. Based on Lyapunov's stability analysis, the conditions that enable the application of the simple bisection method in the optimization procedure were derived. The control law, with certain properties that make the construction of the Lyapunov function much easier—and, thus, the determination of stability conditions—was considered. Furthermore, the optimization problem was formulated as a class of problem in which minimization and maximization of the same performance criterion were simultaneously carried out. The algorithm proposed to solve the related zero-sum differential game was based on Newton's method with recursive matrix relations, in which the first- and second-order derivatives of the objective function are calculated using hyper-dual numbers. The results of this paper were evaluated in simulation on a robot manipulator with three degrees of freedom.

Keywords: robot manipulators; sensor faults; $\mathcal{L}_2$-gain; min–max optimization; Newton-like algorithm

1. Introduction

In recent years, the control of robot manipulators in the presence of faults, and also, in general, the control of nonlinear dynamic systems in the presence of faults, has been a very active area of research. The approaches have been primarily focused on systems affected by sensor faults [1–3], actuator faults [4,5] and simultaneously both sensor and actuator faults [6–9]. Many well-known advanced control methods have been proposed as a way to cope with fault occurrences: these methods include—but are not limited to—sliding mode control [10–13], adaptive control [14], model predictive control [15,16], artificial neural network control [17–19], fuzzy control [20], and hybrid control [21–24].

Furthermore, an optimal and robust control in the presence of sensor and actuator faults has, for the last decade, been a topic of import for researchers on the control of dynamical systems. Many authors consider the $\mathcal{L}_2$-gain of a nonlinear system (often called nonlinear $\mathcal{H}_\infty$, because, according to [25,26], the $\mathcal{L}_2$-gain is equivalent to the $\mathcal{H}_\infty$ norm of a linear system) to be a measure of the influence of the faults. A fixed $\mathcal{L}_2$-gain fault-tolerant controller, for a class of Lipschitz nonlinear system with actuator saturation, was designed in [27]. In [28], a data-driven output-feedback approach to the fault-tolerant control (FTC) problem, considering $\mathcal{L}_2$-gain properties, was proposed. A robust $\mathcal{H}_\infty$ FTC scheme to regulate the quadrotor system was adopted in [29]: the approach was based on linear matrix inequalities (LMI), so the overall dynamics of the quadrotor were linearized. In [30], backstepping and adaptive control methods based on $\mathcal{L}_2$-gain were combined together, to provide a passive fault-tolerant attitude controller of a nonlinear dynamic model of morphing aircraft.

All the above-mentioned works were based on the formulation of the problem in the form of LMI, or on the determination of the approximate solution of the associated Hamilton–Jacobi–Isaacs (HJI) equation. The application of LMI requires the linearization of

system dynamics, and therefore robustness cannot be guaranteed in all operating points; on the other hand, solutions of the HJI equation can be complex, and therefore difficult to apply in real control tasks.

The difficulties described above motivated us to conduct the research presented here. We formulated the control problem of a robot manipulator affected by sensor faults, as a typical representative of nonlinear dynamic systems, in the form of optimal $\mathcal{L}_2$-gain control and the related min–max optimization, and we approached its solution without LMI formalism or the need to approximate the solution of the HJI equation.

The main idea of the approach presented in this paper was based on the algorithmic procedure described in the authors' previous works [31,32]. The idea of using a Newton-like algorithm with recursive matrix relations to calculate exact gradients and Hessians was applied to the synthesis of a control system which aimed to overcome the sensor malfunctions whilst maintaining desirable stability and optimal $\mathcal{L}_2$-gain performance properties.

This paper considered the synthesis of the PID control law of a robot manipulator, which kept the influence of sensor faults below the permissible limits, and ensured the robust stability of the overall closed-loop system. The problem was presented as a two-player zero-sum differential game, with the objective function including a parameter in such a way that the control vector represented the "player" that minimized the objective function, while the vector containing the sensor faults represented the "player" that maximized the same objective function.

The algorithm proposed in this paper, in one part, was based on Lyapunov's stability analysis. The derived stability conditions were used in the optimization process for the appropriate initial setting of the algorithm parameters: these conditions were crucial in the application of the bisection method by which we determined the minimum $\mathcal{L}_2$-gain. Also, by using Lyapunov method, the stopping criteria of the algorithm was improved, by deriving inequalities dependent on PID controller gains that bound the distance of the current point from the saddle point, and thus added bound on the number of iterations.

The main contributions of this work are summarized as follows:

- A suitable mathematical tool and systematic algorithmic procedure for the synthesis of an optimal robust PID controller for robot manipulators affected by sensor faults was developed. The presented approach admitted a min–max formulation, and hence provided a guarantee of robustness, which was achieved by optimizing the worst-case performance. To the best of the authors' knowledge, such an algorithmic PID controller synthesis procedure had not previously been investigated.

- In many other similar optimal control-based algorithms that have been proposed in the literature (see, for example, [33,34] and references therein), the nonlinear system dynamics were treated as equality constraints, and were included in the optimization process, using the method of Lagrange multipliers: the results were HJI equations that were very difficult or almost impossible to solve. For this reason, many approximation methods have been developed (see, for example, [35–38] and references therein) in which actual computational complexity increased with the number of system states which needed to be estimated. In contrast to such approaches, the algorithm proposed in this paper had no high-dimensional structure: this followed from the fact that, instead of incorporating the robot dynamics in the closed loop with the controller directly into an objective function, and solving the corresponding HJI equation, the state variables and PID controller gains were coupled by recursive matrix relations, used to calculate the first- and second-order derivatives that appear in the Newton-like method.

Based on the contributions noted above, the importance of the research presented in this paper lies in the fact that the proposed control architecture does not rely directly on sensor fault estimation, in order to improve the desired positioning of the robot manipulator, and is closely related to nonlinear $\mathcal{L}_2$-gain robust control, where a PID-type controller, with its well-known simplicity of structure, is designed to be robust against a sensor fault in the system. Such a control strategy has the potential for practical applications, due to its

simplicity of design, it having less lag between sensor fault appearance and accommodation, and it not requiring significant computing resources.

The rest of the paper is organized as follows. In Section 2, the robot manipulator dynamics and their main properties are presented. In Section 3, the min–max optimal control problem, of a robot manipulator affected by sensor faults and in a closed loop with a PID controller, is defined. The main results are presented in Sections 4 and 5. Based on the theory of $\mathcal{L}_2$ stability and Lyapunov's approach, the stability conditions of the entire control system are established in Section 4. In Section 5, the bisection method, the Newton-like algorithm, the Adams discretization method and the recursive matrix calculation of gradients and Hessians are integrated into an efficient algorithmic procedure. The closed-loop response of the proposed control strategy and a robot manipulator with three degrees of freedom (3DOF) are evaluated in computer simulations, and the results are given and discussed in Section 6. Section 7 concludes the paper. There are three appendices included. Appendix A describes the notation used throughout the paper. In Appendix B, a detailed derivation of expressions for the kinetic and potential energy of the cylindrical robot manipulator is given. Appendix C contains the expressions for the evaluation of coefficients resulting from the mechanical properties of a cylindrical robot manipulator with 3DOF.

2. Dynamic Model and Properties

Consider a dynamic model of an *N*-DOF robot manipulator system of the form

$$\mathbf{M}(\mathbf{q})\,\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\,\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \mathbf{u}, \tag{1}$$

where $\mathbf{q} \in \mathbb{R}^N$ is the vector of generalized coordinates, $\mathbf{u} \in \mathbb{R}^N$ is the vector of control forces/torques applied to the system, $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{N \times N}$ is the inertia matrix, $\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} \in \mathbb{R}^N$ is the vector of centrifugal and Coriolis forces/torques and $\mathbf{g}(\mathbf{q}) \in \mathbb{R}^N$ is the vector of gravitational forces/torques. It is well-known that Equation (1) can be obtained through modeling a robot manipulator system by Euler–Lagrange equations.

For this paper, we considered a robot manipulator that had both rotational and translational generalized coordinates, and hence had the following properties (see, for example, [39–43]):

Property 1. *The matrix*

$$\dot{\mathbf{M}}(\mathbf{q}) - 2\mathbf{C}(\mathbf{q},\dot{\mathbf{q}}), \tag{2}$$

is skew-symmetric for all $\mathbf{q},\ \dot{\mathbf{q}} \in \mathbb{R}^N$*: this implies that*

$$\dot{\mathbf{M}}(\mathbf{q}) = \mathbf{C}(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})^{\mathrm{T}}; \tag{3}$$

Property 2. *The inertia matrix* $\mathbf{M}(\mathbf{q})$ *is a positive-definite symmetric matrix which satisfies*

$$a_1\|\boldsymbol{\xi}\|^2 \leq \boldsymbol{\xi}^{\mathrm{T}}\mathbf{M}(\mathbf{q})\boldsymbol{\xi} \leq \left(a_2 + c_2\|\mathbf{q}\| + d_2\|\mathbf{q}\|^2\right)\|\boldsymbol{\xi}\|^2, \tag{4}$$

for all $\boldsymbol{\xi}$ *and* $\mathbf{q} \in \mathbb{R}^N$*, where* $a_1, a_2 > 0$*, and* $c_2, d_2 \geq 0$*.*

Property 3. *There exist constants,* c_1 *and* d_1*, such that the Coriolis and centrifugal term,* $\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\,\dot{\mathbf{q}}$*, satisfies*

$$\|\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\,\dot{\mathbf{q}}\| \leq (c_1 + d_1\|\mathbf{q}\|)\|\dot{\mathbf{q}}\|^2, \tag{5}$$

for all $\mathbf{q}$ *and* $\dot{\mathbf{q}} \in \mathbb{R}^N$*, where* $c_1 > 0, d_1 \geq 0$*.*

As is well-known, the vector of gravitational forces/torques is a gradient of the potential energy. The considered class of robot manipulator was such that its potential energy depended linearly on translational generalized coordinates, while rotational generalized coordinates appeared as a trigonometric function, with period 2π. Hence, the following property was imposed:

Property 4. *There exist positive constants, k_{g1} and k_{g2}, such that the Jacobian of the gravity vector satisfies*

$$\left\| \frac{\partial \mathbf{g}(\mathbf{q})}{\partial \mathbf{q}} \right\| \leq k_{g1} + k_{g2}\|\mathbf{q}\|, \tag{6}$$

for all $\mathbf{q} \in \mathbb{R}^N$.

Remark 1. *If the system* (1) *has no translational generalized coordinates, then c_2, d_2, d_1, $k_{g2} = 0$. See, for example, [44–47], and references therein.*

3. Optimal Control Problem

As robot manipulators often work in extreme environments, their sensors are prone to faults during operation: for example, due to environmental noise, like a magnetic field. Faults in robot manipulator sensors often occur due to vibrations that cause disruptions in communication between the control unit and the sensor, or even a short circuit. Furthermore, intermittent sensor connection, bias in sensor measurement and sensor gain drop can also be presented as faults.

For this paper, we were interested in designing a decentralized system controller (1) that guaranteed the stability around a desired constant of generalized coordinate positions $\mathbf{q}_d \in \mathbb{R}^N$, while the influence of sensor faults was kept under a permissible level. In this context, we considered the least upper bound (supremum) of the ratio of the $\mathcal{L}_2$-norm of the vector of the output signals to be controlled, and the $\mathcal{L}_2$-norm of the vector representing sensor faults. It should be noted that considering the control problem in this way was actually equivalent to the $\mathcal{L}_2$-gain (or nonlinear $\mathcal{H}_\infty$) optimal control defined in [25,26].

In a sensor fault situation, the internal robot dynamic properties are not affected, but the controller of the robot manipulator (1) receives generalized coordinates that are not exact, and are given as follows [2,8,48,49]:

$$\bar{\mathbf{q}} = \mathbf{q} + \mathbf{F}_p\,\boldsymbol{\varphi}_p, \tag{7}$$

$$\dot{\bar{\mathbf{q}}} = \dot{\mathbf{q}} + \mathbf{F}_v\,\boldsymbol{\varphi}_v, \tag{8}$$

where $\boldsymbol{\varphi}_p \in \mathcal{L}_2\left(\left[t_0,\ t_f\right],\ \mathbb{R}^S\right)$ and $\boldsymbol{\varphi}_v \in \mathcal{L}_2\left(\left[t_0,\ t_f\right],\ \mathbb{R}^S\right)$ are the vector functions representing sensor faults, and $\mathbf{F}_p$ and $\mathbf{F}_v$ are the matrices with appropriate dimensions. The assumptions about the sensor fault terms were employed as follows:

Assumption 1. *There exist positive-definite time-dependent functions f_p and f_v, such that the sensor faults are bounded by*

$$\|\boldsymbol{\varphi}_p\| \leq f_p, \tag{9}$$

$$\|\boldsymbol{\varphi}_v\| \leq f_v. \tag{10}$$

Remark 2. *In many other works—for example, in [2,7,8,50]—sensor fault bounds are considered as known positive real constants. In this work, we did not treat these bounds as known specific constants, but as unknown functions that needed to be determined in such a way that they maximized the objective function, so that the proposed control strategy might be available not only for robotic manipulators but also for a wide class of nonlinear dynamical systems.*

A PID-type control law was proposed, with gravity vector compensation in the following form:

$$\mathbf{u} = -\mathbf{K}_P\,\tilde{\mathbf{q}} - \mathbf{K}_D\,\dot{\tilde{\mathbf{q}}} - \mathbf{K}_I\,v + \mathbf{g}(\mathbf{q}), \tag{11}$$

$$\dot{v} = \tilde{\mathbf{q}}, \tag{12}$$

where $\tilde{\mathbf{q}} = \bar{\mathbf{q}} - \mathbf{q}_d$ was a generalized coordinates errors vector, and $\mathbf{K}_P$, $\mathbf{K}_D$ and $\mathbf{K}_I$ were $N \times N$ positive–definite diagonal matrices. Note that, as $\mathbf{K}_P$, $\mathbf{K}_D$ and $\mathbf{K}_I$ were diagonal matrices, the proposed control law was formed in a fully decentralized fashion.

The robot manipulator system (1) with mixed rotational and translational generalized coordinates, affected by sensor faults (7) and (8) in a closed loop with a PID-type controller (11) and (12), can be represented by a block diagram, as shown in Figure 1. Each DOF of the robot manipulator was driven by servo motors with gears or pneumatic and hydraulic cylinders. The decentralized PID control law was implemented in the microcontroller, and point-to-point communication was established by electrical interface and physical layers. The received encoder data were used to close the feedback loop in the microcontroller.

Regarding the block diagram shown in Figure 1, in our approach the emphasis was not on the structure of the controller itself, but on the method for adjusting the gains of the controller. The PID controller structure was considered because it is the most common structure used in industrial applications.
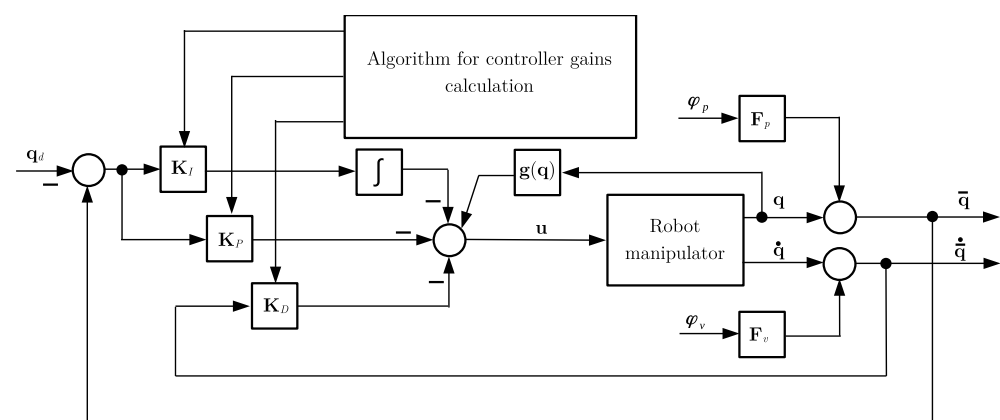

Figure 1. Block diagram of closed-loop system, with algorithm for controller gains calculation.

Remark 3. *It is important to emphasize the main difference between a conventional PID controller and our proposed one. In conventional applications of a PID controller, the gain values are independently and freely chosen by the designer, often using a trial-and-error method; however, in our approach, a proportional gain matrix $\mathbf{K}_P$, a derivative gain matrix $\mathbf{K}_D$ and an integral gain matrix $\mathbf{K}_I$ were tuned automatically and correlatively by a numerical algorithm that will be presented in detail in the following sections.*

Inserting (7) and (8) into (11) and (12) gave

$$\mathbf{u} = -\mathbf{K}_P\,\hat{\mathbf{q}} - \mathbf{K}_D\,\dot{\mathbf{q}} - \mathbf{K}_I\,\mathbf{v} - \mathbf{K}_P\,\mathbf{F}_p\,\boldsymbol{\varphi}_p - \mathbf{K}_D\,\mathbf{F}_v\,\boldsymbol{\varphi}_v + \mathbf{g}(\mathbf{q}), \tag{13}$$

$$\dot{\mathbf{v}} = \hat{\mathbf{q}} + \mathbf{F}_p\,\boldsymbol{\varphi}_p, \tag{14}$$

where $\hat{\mathbf{q}} = \mathbf{q} - \mathbf{q}_d$.

Finally, inserting (13) and (14) into (1) gave closed-loop equations in the following form:

$$\mathbf{M}(\mathbf{q})\,\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\,\dot{\mathbf{q}} = -\mathbf{K}_P\,\hat{\mathbf{q}} - \mathbf{K}_D\,\dot{\mathbf{q}} - \mathbf{K}_I\,\mathbf{v} - \mathbf{K}_P\,\mathbf{F}_p\,\boldsymbol{\varphi}_p - \mathbf{K}_D\,\mathbf{F}_v\,\boldsymbol{\varphi}_v, \tag{15}$$

$$\dot{\mathbf{v}} = \hat{\mathbf{q}} + \mathbf{F}_p\,\boldsymbol{\varphi}_p. \tag{16}$$

Based on the definition of finite $\mathcal{L}_2$-gain and the definition of $\mathcal{L}_2$ stability (see, for example, [25,51]), we defined the min–max optimal control problem of a robot manipulator affected by sensor faults, as follows:

Problem 1. *Determine the PID controller gains, which are the elements of the matrices $\mathbf{K}_P$, $\mathbf{K}_D$ and $\mathbf{K}_I$ in (11) and (12), and determine the "worst case" of sensor fault functions $\boldsymbol{\varphi}_p$ and $\boldsymbol{\varphi}_v$ in (7) and (8), such that the system (15) and (16) is stable around the desired constant generalized*

coordinates positions, the ratio of the $\mathcal{L}_2$-norm of the vector of the output signals is controlled and the $\mathcal{L}_2$-norm of the vector representing sensor faults is minimized. In other words, solve the following zero-sum differential game:

$$J_\mu^*(\mathbf{q}_0) = \min_{\mathbf{K}_P, \mathbf{K}_D, \mathbf{K}_I} \max_{\boldsymbol{\varphi}} \left\{ \|\hat{\mathbf{q}}\|_{\mathcal{L}_2}^2 + \|\mathbf{u}\|_{\mathcal{L}_2}^2 - \mu \|\boldsymbol{\varphi}\|_{\mathcal{L}_2}^2 \right\}, \tag{17}$$

where $\hat{\mathbf{q}}$, $\mathbf{K}_P$, $\mathbf{K}_D$, $\mathbf{K}_I$ and $\boldsymbol{\varphi} = \begin{bmatrix} \boldsymbol{\varphi}_p^{\mathrm{T}} & \boldsymbol{\varphi}_v^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$ are coupled via closed-loop system dynamics (15) and (16), and $\mu > 0$ is a finite $\mathcal{L}_2$-gain; furthermore, $\mathbf{q}_0$ is an a priori known vector of initial states of generalized coordinates.

4. $\mathcal{L}_2$ Stability Conditions

It is well-known from the dissipativity theory [52–54] that a nonlinear dynamic system is $\mathcal{L}_2$-stable if, for all initial conditions, all uncertainties $\mathbf{w}$ and all $t_f \geq t_0$ there exists a storage function V, which is also a Lyapunov function, such that the following inequality holds:

$$\dot{V} \leq \gamma^2 \mathbf{w}^{\mathrm{T}} \mathbf{w} - \mathbf{z}^{\mathrm{T}} \mathbf{z}, \tag{18}$$

where $\mathbf{z}$ is the performance variable, and $\gamma > 0$ is the finite $\mathcal{L}_2$-gain.

Note that the right-hand side of the inequality (18) in our case actually corresponded to the argument of the min–max operator in (17). Based on this observation, we give the following proposition, by which we set the stability conditions of the control system proposed in this paper:

Proposition 1. *If the following conditions are satisfied:*

(a)

$$\alpha \lambda_{\min}\{\mathbf{K}_D\} - \frac{1}{\alpha}\lambda_{\max}\{\mathbf{K}_I\} - \alpha^2 a_1 > 0, \tag{19}$$

(b)

$$\alpha a_1 + (c_1 + d_1 \|\mathbf{q}\|)\|\hat{\mathbf{q}}\| - \lambda_{\max}\{\mathbf{K}_D\} - \mu \leq 0, \tag{20}$$

$$\alpha \lambda_{\min}\{\mathbf{K}_I\} - \alpha\mu \leq 0, \tag{21}$$

$$1 - \lambda_{\max}\{\mathbf{K}_D\}\sigma_{\max}\{\mathbf{F}_v\} \leq 0, \tag{22}$$

$$\alpha - \alpha\lambda_{\max}\{\mathbf{K}_D\}\sigma_{\max}\{\mathbf{F}_v\} \leq 0, \tag{23}$$

for some $\alpha > 0$, where $\mu > 0$ is a finite $\mathcal{L}_2$-gain, and where a_1, c_1 and d_1 are defined in Properties 2 and 3, then the system (15) and (16) is locally $\mathcal{L}_2$-stable.

Proof. The first step was to transform the system (15) and (16) into a form with a zero steady state. The stationary state of the system (15) and (16) is $\dot{\mathbf{q}} = \mathbf{0}$, $\hat{\mathbf{q}} = \mathbf{0} \Rightarrow \mathbf{q} = \mathbf{q}_d$, so it was obtained thus:

$$\mathbf{K}_I \, v^* + \mathbf{K}_P \, \mathbf{F}_p \, \boldsymbol{\varphi}_p^* + \mathbf{K}_D \, \mathbf{F}_v \, \boldsymbol{\varphi}_v^* = \mathbf{0} \tag{24}$$

$$\dot{v} = \mathbf{F}_p \, \boldsymbol{\varphi}_p^*, \tag{25}$$

where v^*, $\boldsymbol{\varphi}_p^*$ and $\boldsymbol{\varphi}_v^*$ were steady state values.

Subtracting (16) and (25) gave $\hat{\mathbf{q}} = -\mathbf{F}_p\left(\boldsymbol{\varphi}_p - \boldsymbol{\varphi}_p^*\right)$; then, subtracting (15) and (24), we obtained the error equations in the following form:

$$\mathbf{M}(\mathbf{q})\,\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\,\dot{\mathbf{q}} + \mathbf{K}_D\,\dot{\mathbf{q}} + \mathbf{K}_I\,\hat{v} + \mathbf{K}_D\,\mathbf{F}_v\,\hat{\boldsymbol{\varphi}}_v = \mathbf{0}, \tag{26}$$

where $\hat{v} = v - v^*$ and $\hat{\boldsymbol{\varphi}}_v = \boldsymbol{\varphi}_v - \boldsymbol{\varphi}_v^*$.

Following the methodology from [39,46], as the first step in the construction of the Lyapunov function, the error Equations (26) were multiplied on the left side by $\alpha \hat{\mathbf{q}}^{\mathrm{T}} + \dot{\mathbf{q}}^{\mathrm{T}}$ with some $\alpha > 0$:

$$
\begin{aligned}
&\alpha \hat{\mathbf{q}}^{\mathrm{T}} \mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}} + \alpha \hat{\mathbf{q}}^{\mathrm{T}} \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + \alpha \hat{\mathbf{q}}^{\mathrm{T}} \mathbf{K}_D \dot{\mathbf{q}} + \alpha \hat{\mathbf{q}}^{\mathrm{T}} \mathbf{K}_I \hat{\boldsymbol{v}} + \alpha \hat{\mathbf{q}}^{\mathrm{T}} \mathbf{K}_D \mathbf{F}_v \hat{\boldsymbol{\varphi}}_v \\
&+ \dot{\mathbf{q}}^{\mathrm{T}} \mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}} + \dot{\mathbf{q}}^{\mathrm{T}} \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + \dot{\mathbf{q}}^{\mathrm{T}} \mathbf{K}_D \dot{\mathbf{q}} + \dot{\mathbf{q}}^{\mathrm{T}} \mathbf{K}_I \hat{\boldsymbol{v}} + \dot{\mathbf{q}}^{\mathrm{T}} \mathbf{K}_D \mathbf{F}_v \hat{\boldsymbol{\varphi}}_v = 0.
\end{aligned}
\tag{27}
$$

Some of the terms in the expression (27) can be written as follows:

$$
\dot{\mathbf{q}}^{\mathrm{T}} \mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}} = \frac{d}{dt} \left(\frac{1}{2} \dot{\mathbf{q}}^{\mathrm{T}} \mathbf{M}(\mathbf{q}) \dot{\mathbf{q}} \right) - \frac{1}{2} \dot{\mathbf{q}}^{\mathrm{T}} \dot{\mathbf{M}}(\mathbf{q}) \dot{\mathbf{q}},
\tag{28}
$$

$$
\alpha \hat{\mathbf{q}}^{\mathrm{T}} \mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}} = \frac{d}{dt} \left(\alpha \hat{\mathbf{q}}^{\mathrm{T}} \mathbf{M}(\mathbf{q}) \dot{\mathbf{q}} \right) - \alpha \dot{\mathbf{q}}^{\mathrm{T}} \mathbf{M}(\mathbf{q}) \dot{\mathbf{q}} - \alpha \hat{\mathbf{q}}^{\mathrm{T}} \dot{\mathbf{M}}(\mathbf{q}) \dot{\mathbf{q}},
\tag{29}
$$

$$
\alpha \hat{\mathbf{q}}^{\mathrm{T}} \mathbf{K}_D \dot{\mathbf{q}} = \frac{d}{dt} \left(\frac{1}{2} \alpha \hat{\mathbf{q}}^{\mathrm{T}} \mathbf{K}_D \hat{\mathbf{q}} \right),
\tag{30}
$$

$$
\alpha \hat{\mathbf{q}}^{\mathrm{T}} \mathbf{K}_I \hat{\boldsymbol{v}} = \frac{d}{dt} \left(\frac{1}{2} \alpha \hat{\boldsymbol{v}}^{\mathrm{T}} \mathbf{K}_I \hat{\boldsymbol{v}} \right),
\tag{31}
$$

$$
\dot{\mathbf{q}}^{\mathrm{T}} \mathbf{K}_I \hat{\boldsymbol{v}} = \frac{d}{dt} \left(\hat{\mathbf{q}}^{\mathrm{T}} \mathbf{K}_I \hat{\boldsymbol{v}} \right) - \hat{\mathbf{q}}^{\mathrm{T}} \mathbf{K}_I \hat{\mathbf{q}}.
\tag{32}
$$

Inserting (28)–(32) in (27) and, as $\dot{\mathbf{M}}(\mathbf{q}) - 2\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ was skew-symmetric (see Property 1), which implied $\dot{\mathbf{q}}^{\mathrm{T}} (\dot{\mathbf{M}}(\mathbf{q}) - 2\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})) \dot{\mathbf{q}} = 0$, we obtained:

$$
\begin{aligned}
&\frac{d}{dt} \left(\frac{1}{2} \dot{\mathbf{q}}^{\mathrm{T}} \mathbf{M}(\mathbf{q}) \dot{\mathbf{q}} + \alpha \hat{\mathbf{q}}^{\mathrm{T}} \mathbf{M}(\mathbf{q}) \dot{\mathbf{q}} + \frac{1}{2} \alpha \hat{\mathbf{q}}^{\mathrm{T}} \mathbf{K}_D \hat{\mathbf{q}} + \frac{1}{2} \alpha \hat{\boldsymbol{v}}^{\mathrm{T}} \mathbf{K}_I \hat{\boldsymbol{v}} + \hat{\mathbf{q}}^{\mathrm{T}} \mathbf{K}_I \hat{\boldsymbol{v}} \right) \\
&= \alpha \dot{\mathbf{q}}^{\mathrm{T}} \mathbf{M}(\mathbf{q}) \dot{\mathbf{q}} + \alpha \hat{\mathbf{q}}^{\mathrm{T}} \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})^{\mathrm{T}} \dot{\mathbf{q}} - \dot{\mathbf{q}}^{\mathrm{T}} \mathbf{K}_D \dot{\mathbf{q}} + \alpha \hat{\mathbf{q}}^{\mathrm{T}} \mathbf{K}_I \hat{\mathbf{q}} - \dot{\mathbf{q}}^{\mathrm{T}} \mathbf{K}_D \mathbf{F}_v \hat{\boldsymbol{\varphi}}_v - \alpha \hat{\mathbf{q}}^{\mathrm{T}} \mathbf{K}_D \mathbf{F}_v \hat{\boldsymbol{\varphi}}_v.
\end{aligned}
\tag{33}
$$

Based on the nonlinear differential form (33), it could be concluded that on the left-hand side we had the Lyapunov function candidate $V(\dot{\mathbf{q}}, \hat{\mathbf{q}}, \hat{\boldsymbol{v}})$, while on the right-hand side we had a candidate for the time derivative of the Lyapunov function $\dot{V}(\dot{\mathbf{q}}, \hat{\boldsymbol{v}}, \hat{\boldsymbol{\varphi}}_v)$.

The Lyapunov function candidate could be rewritten as follows:

$$
\begin{aligned}
V(\dot{\mathbf{q}}, \hat{\mathbf{q}}, \hat{\boldsymbol{v}}) &= \frac{1}{2} (\alpha \hat{\mathbf{q}} + \hat{\boldsymbol{v}})^{\mathrm{T}} \mathbf{M}(\mathbf{q}) (\alpha \hat{\mathbf{q}} + \hat{\boldsymbol{v}}) - \frac{1}{2} \alpha^2 \hat{\mathbf{q}}^{\mathrm{T}} \mathbf{M}(\mathbf{q}) \hat{\mathbf{q}} + \frac{1}{2} \alpha \hat{\mathbf{q}}^{\mathrm{T}} \mathbf{K}_D \hat{\mathbf{q}} \\
&+ \frac{1}{2} \left(\frac{1}{\sqrt{\alpha}} \hat{\mathbf{q}} + \sqrt{\alpha} \hat{\boldsymbol{v}} \right)^{\mathrm{T}} \mathbf{K}_I \left(\frac{1}{\sqrt{\alpha}} \hat{\mathbf{q}} + \sqrt{\alpha} \hat{\boldsymbol{v}} \right) - \frac{1}{2\alpha} \hat{\mathbf{q}}^{\mathrm{T}} \mathbf{K}_I \hat{\mathbf{q}}.
\end{aligned}
\tag{34}
$$

In order for the above expression to be positive-definite, it was necessary to determine the conditions under which

$$
-\frac{1}{2} \alpha^2 \hat{\mathbf{q}}^{\mathrm{T}} \mathbf{M}(\mathbf{q}) \hat{\mathbf{q}} + \frac{1}{2} \alpha \hat{\mathbf{q}}^{\mathrm{T}} \mathbf{K}_D \hat{\mathbf{q}} - \frac{1}{2\alpha} \hat{\mathbf{q}}^{\mathrm{T}} \mathbf{K}_I \hat{\mathbf{q}} > 0.
\tag{35}
$$

Using definitions of vector and matrix norms, bound of quadratic forms (see Notation, Appendix A) and, finally, applying Property 2 we obtained the following:

$$
\left(\frac{1}{2} \alpha \lambda_{\min}\{\mathbf{K}_D\} - \frac{1}{2\alpha} \lambda_{\max}\{\mathbf{K}_I\} - \frac{1}{2} \alpha^2 a_1 \right) \|\hat{\mathbf{q}}\|^2 > 0,
\tag{36}
$$

which was positive-definite if

$$
\alpha \lambda_{\min}\{\mathbf{K}_D\} - \frac{1}{\alpha} \lambda_{\max}\{\mathbf{K}_I\} - \alpha^2 a_1 > 0.
\tag{37}
$$

Next, according to the theory of $\mathcal{L}_2$ stability and the passivity properties of Euler–Lagrange systems, the following inequality needed to be satisfied:

$$\dot{V}(\dot{\mathbf{q}}, \hat{\mathbf{v}}, \hat{\boldsymbol{\varphi}}_v) \leq \mu \|\alpha\hat{\mathbf{q}} + \dot{\mathbf{q}}\|^2 - (\alpha\hat{\mathbf{q}} + \dot{\mathbf{q}})^{\mathrm{T}} \hat{\boldsymbol{\varphi}}_v, \tag{38}$$

where $\mu > 0$ was a finite $\mathcal{L}_2$-gain.

Comparing (38) to the right-hand side of (33), using Cauchy—Schwarz inequality, triangle inequality, definitions of vector and matrix norms, bound of quadratic forms (see Notation, Appendix A) and, finally, applying Properties 2 and 3, we got the following inequality:

$$(\alpha a_1 + (c_1 + d_1\|\mathbf{q}\|)\|\hat{\mathbf{q}}\| - \lambda_{\max}\{\mathbf{K}_D\} - \mu)\|\dot{\mathbf{q}}\|^2 + (\alpha\lambda_{\min}\{\mathbf{K}_I\} - \alpha\mu)\|\hat{\mathbf{q}}\|^2 + \\ (1 - \lambda_{\max}\{\mathbf{K}_D\}\sigma_{\max}\{\mathbf{F}_v\})\|\dot{\mathbf{q}}\|\,\|\hat{\boldsymbol{\varphi}}_v\| + (\alpha - \alpha\lambda_{\max}\{\mathbf{K}_D\}\sigma_{\max}\{\mathbf{F}_v\})\|\hat{\mathbf{q}}\|\,\|\hat{\boldsymbol{\varphi}}_v\| \leq 0. \tag{39}$$

The conditions that satisfied the above inequality were as follows:

$$\alpha a_1 + (c_1 + d_1\|\mathbf{q}\|)\|\hat{\mathbf{q}}\| - \lambda_{\max}\{\mathbf{K}_D\} - \mu \leq 0, \tag{40}$$

$$\alpha\lambda_{\min}\{\mathbf{K}_I\} - \alpha\mu \leq 0, \tag{41}$$

$$1 - \lambda_{\max}\{\mathbf{K}_D\}\sigma_{\max}\{\mathbf{F}_v\} \leq 0, \tag{42}$$

$$\alpha - \alpha\lambda_{\max}\{\mathbf{K}_D\}\sigma_{\max}\{\mathbf{F}_v\} \leq 0. \tag{43}$$

□

Remark 4. *As the condition (40) depended on the generalized coordinates of the system, it was concluded that the proposed controller guaranteed only local stability. For a more detailed analysis of stability, it would be necessary to determine the domain of attraction that guarantees asymptotic stability; furthermore, in order to ensure global stability, it would be necessary to introduce a nonlinear term in the control law (see, for example, [39]). Determining the domain of attraction and deriving the global stability conditions were beyond the scope of the research presented in this paper. The stability conditions derived in this work were calculated numerically, and served only for the appropriate initialization of the algorithm, thereby ensuring satisfactory convergence properties.*

5. Algorithm for the Controller Synthesis

In order to derive the algorithm for the controller synthesis, i.e., the algorithm to solve Problem 1, the PID control law (13) had to be written in parametrized form first. To do this, a vectorization operation was performed:

$$\begin{aligned}
\text{vec}(\mathbf{u}) &= -\text{vec}(\mathbf{K}_P\hat{\mathbf{q}}) - \text{vec}(\mathbf{K}_D\dot{\mathbf{q}}) - \text{vec}(\mathbf{K}_I\mathbf{v}) - \text{vec}(\mathbf{K}_P\mathbf{F}_p\boldsymbol{\varphi}_p) - \text{vec}(\mathbf{K}_D\mathbf{F}_v\,\boldsymbol{\varphi}_v) \\
&= -\left(\hat{\mathbf{q}}^{\mathrm{T}} \otimes \mathbf{I}\right)\text{vec}(\mathbf{K}_P) - \left(\dot{\mathbf{q}}^{\mathrm{T}} \otimes \mathbf{I}\right)\text{vec}(\mathbf{K}_D) - \left(\mathbf{v}^{\mathrm{T}} \otimes \mathbf{I}\right)\text{vec}(\mathbf{K}_I) \\
&\quad - \left(\boldsymbol{\varphi}_p^{\mathrm{T}}\mathbf{F}_p^{\mathrm{T}} \otimes \mathbf{I}\right)\text{vec}(\mathbf{K}_P) - \left(\boldsymbol{\varphi}_v^{\mathrm{T}}\mathbf{F}_v^{\mathrm{T}} \otimes \mathbf{I}\right)\text{vec}(\mathbf{K}_D) \\
&= -\left[\left(\hat{\mathbf{q}}^{\mathrm{T}} + \boldsymbol{\varphi}_p^{\mathrm{T}}\mathbf{F}_p^{\mathrm{T}}\right) \otimes \mathbf{I} \quad \left(\dot{\mathbf{q}}^{\mathrm{T}} + \boldsymbol{\varphi}_v^{\mathrm{T}}\mathbf{F}_v^{\mathrm{T}}\right) \otimes \mathbf{I} \quad \mathbf{v}^{\mathrm{T}} \otimes \mathbf{I}\right]\begin{bmatrix}\text{vec}(\mathbf{K}_P) \\ \text{vec}(\mathbf{K}_D) \\ \text{vec}(\mathbf{K}_I)\end{bmatrix}.
\end{aligned} \tag{44}$$

Then, by substituting $\mathbf{x} = \begin{bmatrix}\mathbf{x}_1^{\mathrm{T}} & \mathbf{x}_2^{\mathrm{T}} & \mathbf{x}_3^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}} = \begin{bmatrix}\hat{\mathbf{q}}^{\mathrm{T}} & \dot{\mathbf{q}}^{\mathrm{T}} & \mathbf{v}^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}$, from (15) and (16), we got the following first-order nonlinear differential equations:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{B}(\mathbf{x}, \boldsymbol{\varphi})\mathbf{k} + \mathbf{F}\boldsymbol{\varphi}_p, \tag{45}$$

where

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} \mathbf{x}_2 \\ -\mathbf{M}^{-1}(\mathbf{x}_1)\mathbf{C}(\mathbf{x}_1,\mathbf{x}_2)\mathbf{x}_2 \\ \mathbf{x}_1 \end{bmatrix},$$

$$\mathbf{B}(\mathbf{x},\boldsymbol{\varphi}_p) = \begin{bmatrix} \mathbf{0} \\ -\mathbf{M}^{-1}(\mathbf{x}_1)\mathbf{R}(\mathbf{x},\boldsymbol{\varphi}) \\ \mathbf{0} \end{bmatrix},$$

$$\mathbf{R}(\mathbf{x},\boldsymbol{\varphi}_p) = \begin{bmatrix} \left(\mathbf{x}_1^{\mathsf{T}} + \boldsymbol{\varphi}_p^{\mathsf{T}}\mathbf{F}_p^{\mathsf{T}}\right) \otimes \mathbf{I} & \left(\mathbf{x}_2^{\mathsf{T}} + \boldsymbol{\varphi}_v^{\mathsf{T}}\mathbf{F}_v^{\mathsf{T}}\right) \otimes \mathbf{I} & \mathbf{x}_3^{\mathsf{T}} \otimes \mathbf{I} \end{bmatrix},$$

$$\mathbf{k} = \begin{bmatrix} \text{vec}(\mathbf{K}_P) \\ \text{vec}(\mathbf{K}_D) \\ \text{vec}(\mathbf{K}_I) \end{bmatrix}, \tag{46}$$

$$\boldsymbol{\varphi} = \begin{bmatrix} \boldsymbol{\varphi}_p \\ \boldsymbol{\varphi}_v \end{bmatrix},$$

$$\mathbf{F} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{F}_p \end{bmatrix}.$$

Note that $\mathbf{u} = \mathbf{R}(\mathbf{x},\boldsymbol{\varphi})\mathbf{k}$.

In accordance with the previous substitution, expression (17) then became

$$J_\mu^*(\mathbf{x}_0) = \min_{\mathbf{k}} \max_{\boldsymbol{\varphi}} \left\{ \|\mathbf{x}_1\|_{\mathcal{L}_2}^2 + \|\mathbf{R}(\mathbf{x},\boldsymbol{\varphi})\mathbf{k}\|_{\mathcal{L}_2}^2 - \mu\|\boldsymbol{\varphi}\|_{\mathcal{L}_2}^2 \right\}, \tag{47}$$

where $\mathbf{x}$, $\mathbf{k}$ and $\boldsymbol{\varphi}$ were coupled via nonlinear differential Equations (45) and (46).

5.1. $\mathcal{L}_2$-Gain Minimization

The stability conditions from Proposition 1 suggested that a simple and well-known schematic bisection algorithm [55] could be applied, to minimize the $\mathcal{L}_2$-gain: the steps of this method were as follows.

Step initialization: Choose the initial elements of the vector $\mathbf{k}$, i.e., the initial PID controller gains, such that the lower μ_{l_0} and upper μ_{u_0} bound of the $\mathcal{L}_2$-gain satisfy the conditions in expression (19). Choose a small enough positive constant, ϵ, as the stopping criteria of the bisection method.

Step 1: Set $k := 1$.

Step 2: Set

$$\mu_k := \frac{1}{2}\left(\mu_{l_{k-1}} + \mu_{u_{k-1}}\right), \tag{48}$$

then calculate the value of function $J_{\mu_k}^*(\mathbf{x}_0)$ by solving the zero-sum differential game (47).

Step 3: If $|J_{\mu_k}^*(\mathbf{x}_0)| < \epsilon$, then stop; otherwise, if $J_{\mu_k}^*(\mathbf{x}_0) \le 0$ then $\mu_{l_k} := \mu_{l_{k-1}}$ and $\mu_{u_k} := \mu_k$; else, $\mu_{l_k} := \mu_k$ and $\mu_{u_k} := \mu_{u_{k-1}}$.

Step 4. Set $k := k+1$ and return to **Step 2**.

5.2. Zero-Sum Differential Game Solution

In the second step of the algorithmic procedure described in the previous subsection, it was necessary to solve the problem defined by expression (47). To solve this zero-sum differential game, we implemented the Newton method, which is described in the following steps.

Step initialization: Choose a small enough positive constant, ε, as the stopping criterion of the Newton algorithm.

Step 1: Set $j := 0$.

Step 2: Determine the search direction vector $\mathbf{s}_j$ by solving a system of linear equations using Cholesky factorization:

$$\begin{bmatrix} \nabla_{\mathbf{k}}^2 J & \nabla_{\mathbf{k},\boldsymbol{\varphi}}^2 J \\ -\nabla_{\boldsymbol{\varphi},\mathbf{k}}^2 J & -\nabla_{\boldsymbol{\varphi}}^2 J \end{bmatrix} \mathbf{s}_j = -\begin{bmatrix} \nabla_{\mathbf{k}}^{\mathrm{T}} J \\ -\nabla_{\boldsymbol{\varphi}}^{\mathrm{T}} J \end{bmatrix}, \tag{49}$$

where J is the argument of the min max operator in (47), $\nabla_{\mathbf{k}} J$, $\nabla_{\boldsymbol{\varphi}} J$, $\nabla_{\mathbf{k}}^2 J$, $\nabla_{\boldsymbol{\varphi}}^2 J$, and $\nabla_{\mathbf{k},\boldsymbol{\varphi}}^2 J$ are gradients and Hessians with respect to the vectors $\mathbf{k}$ and $\boldsymbol{\varphi}$, respectively. Note that the maximization with respect to $\boldsymbol{\varphi}$ is achieved simply by the minus sign in front of the gradient and Hessians.

Step 3: Use the line search strategy satisfying the Wolfe conditions [56] to compute the step-size $\eta_j > 0$.

Step 4: Calculate

$$\begin{bmatrix} \mathbf{k}_{j+1} \\ \boldsymbol{\varphi}_{j+1} \end{bmatrix} = \begin{bmatrix} \mathbf{k}_j \\ \boldsymbol{\varphi}_j \end{bmatrix} + \eta_j \mathbf{s}_j. \tag{50}$$

Step 5: If

$$\left\| \begin{bmatrix} \nabla_{\mathbf{k}} J & -\nabla_{\boldsymbol{\varphi}} J \end{bmatrix}^{\mathrm{T}} \right\|_{\infty} \le \varepsilon, \tag{51}$$

then stop; else, set $j := j + 1$ and go to **Step 2**.

As is well-known, Newton's method has a locally quadratic convergence to the saddle point, assuming that the initial point is close enough to the saddle point. In the approach we propose here, the proximity to the saddle point is ensured by the $\mathcal{L}_2$ stability conditions set in Proposition 1: this further implies that $\nabla_{\mathbf{k}}^2 J > \mathbf{0}$ and $-\nabla_{\boldsymbol{\varphi}}^2 J > \mathbf{0}$. As these Hessians are positive-definite submatrices on the main diagonal of the matrix on the left-hand side of (49), and submatrices $\nabla_{\mathbf{k},\boldsymbol{\varphi}}^2 J$ and $-\nabla_{\boldsymbol{\varphi},\mathbf{k}}^2 J$ are negative transpositions of each other, the matrix on the left-hand side of Equation (49) is asymmetric positive-definite and, therefore, the linear system (49) is well-defined; therefore, the previously described algorithmic procedure, which is based on the Newton method, produces:

$$\mathbf{k}_k \in \arg\min_{\mathbf{k}} \left\{ \|\mathbf{x}_1\|_{\mathcal{L}_2}^2 + \|\mathbf{R}(\mathbf{x},\boldsymbol{\varphi})\mathbf{k}\|_{\mathcal{L}_2}^2 - \mu_k \|\boldsymbol{\varphi}\|_{\mathcal{L}_2}^2 \right\}, \tag{52}$$

$$\boldsymbol{\varphi}_k \in \arg\max_{\boldsymbol{\varphi}} \left\{ \|\mathbf{x}_1\|_{\mathcal{L}_2}^2 + \|\mathbf{R}(\mathbf{x},\boldsymbol{\varphi})\mathbf{k}\|_{\mathcal{L}_2}^2 - \mu_k \|\boldsymbol{\varphi}\|_{\mathcal{L}_2}^2 \right\}, \tag{53}$$

in the k-th iteration of the bisection algorithm proposed in Section 5.1.

To perform the steps of the proposed Newton-like algorithm, we needed expressions for the gradients and Hessians that appear in (49). A detailed procedure for deriving recursive matrix relations for computing these gradients and Hessians is given in the next subsection.

5.3. Matrix Relations for Recursive Calculation of Gradients and Hessians

In order to derive recursive relations for calculating gradients and Hessians, we used the fact that derivatives are a measure of the sensitivity of functions to small changes in their variables. This suggested that, for their calculation, we could perform a time discretization of the system dynamics (45). In the approach presented in this paper, the discretization of system dynamics using the fourth-order Adams approximation with a small time step was applied. In [32] it is shown that an explicit Adams method can be conveniently transformed into discrete-time state space.

The explicit Adams approximation of system (45) can be simply written in the following discrete-time state-space form:

$$\hat{\mathbf{x}}(i+1) = \hat{\boldsymbol{\phi}}(\hat{\mathbf{x}}(i), \mathbf{k}(i), \boldsymbol{\varphi}(i)), \quad \hat{\mathbf{x}}(0) = \hat{\mathbf{x}}_0, \tag{54}$$

such that the time grid consists of points $t_i = t_0 + i\tau$ for $i = 0, 1, 2, \ldots, N-1$, where $\tau = (t_f - t_0)/N$ is the time step length, and $\hat{\mathbf{x}}$ is the extended $4n$-dimensional state vector (n is a dimension of state vector $\mathbf{x}$ in (45), and 4 is the order of the Adams method). More details on the Adams method can be found in [57].

The discrete-time form of the objective function resulted in

$$J(\mathbf{x}_0) = \tau \sum_{i=0}^{N-1} F(\hat{\mathbf{x}}(i), \mathbf{k}(i), \boldsymbol{\varphi}(i)), \tag{55}$$

where F was the sub-integral function of the argument of the min max operator in (47)

$$F(i) = \|\mathbf{x}_1(i)\|^2 + \|\mathbf{R}(\mathbf{x}(i), \boldsymbol{\varphi}(i))\mathbf{k}(i)\|^2 - \mu \|\boldsymbol{\varphi}(i)\|^2. \tag{56}$$

The gradient of the objective function (55), with respect to the $\mathbf{k}$ in the j-th iteration of the Newton algorithm (Section 5.2), was given by

$$\nabla_{\mathbf{k}(l)} J = \tau \sum_{i=0}^{N-1} \nabla_{\mathbf{k}(l)} F(i) = \tau \left(\nabla_{\mathbf{k}(l)} F(l) + \sum_{i=l+1}^{N-1} \nabla_{\mathbf{k}(l)} F(i) \right), \tag{57}$$

for $l = 0, 1, 2, \ldots, N-1$. Note that, because of the causality principle, the terms for $i < l$ were equal to zero.

As the terms under the sum on the right-hand side of (57) depended on $\mathbf{k}(l)$ implicitly through $\hat{\mathbf{x}}(i)$ for $i > l$, this gave

$$\nabla_{\mathbf{k}(l)} F(i) = \nabla_{\mathbf{k}(l)} \hat{\mathbf{x}}(i) \, \nabla_{\hat{\mathbf{x}}(i)}^{\mathrm{T}} F(i). \tag{58}$$

Using the chain rule for ordered derivatives, from (54) it followed that

$$\nabla_{\mathbf{k}(l)} \hat{\mathbf{x}}(i) = \nabla_{\mathbf{k}(l)} \hat{\mathbf{x}}(i-1) \, \nabla_{\hat{\mathbf{x}}(i-1)} \hat{\boldsymbol{\phi}}(i-1), \tag{59}$$

for $i = l+2, \ldots, N-1$.

To simplify the following expressions, we introduced

$$\sigma(l) = \sum_{i=l+1}^{N-1} \nabla_{\mathbf{k}(l)}^{\mathrm{T}} F(i). \tag{60}$$

Starting from $l = N-2; i = N-1$, it followed that

$$\sigma(N-2) = \nabla_{\mathbf{k}(N-2)}^{\mathrm{T}} F(N-1), \tag{61}$$

then, substituting (58) in (61)

$$\sigma(N-2) = \nabla_{\mathbf{k}(N-2)} \hat{\mathbf{x}}(N-1) \, \nabla_{\hat{\mathbf{x}}(N-1)}^{\mathrm{T}} F(N-1), \tag{62}$$

and substituting (59) in (62), and taking into account (54), we obtained

$$\sigma(N-2) = \nabla_{\mathbf{k}(N-2)} \hat{\boldsymbol{\phi}}(N-2) \, \nabla_{\hat{\mathbf{x}}(N-1)}^{\mathrm{T}} F(N-1). \tag{63}$$

The above procedure could be further continued for $l = N-3; i = N-1, N-2;$ $l = N-4; i = N-3, N-2, N-1$, etc., and the final recursive expressions for calculation of gradient $\nabla_{\mathbf{k}} J$ had the following form:

$$\boldsymbol{\omega}(N-1) = \mathbf{0}, \tag{64}$$

$$\boldsymbol{\omega}(l) = \nabla_{\hat{\mathbf{x}}(l+1)}^{\mathrm{T}} F(l+1) + \nabla_{\hat{\mathbf{x}}(l+1)} \hat{\boldsymbol{\phi}}(l+1) \cdot \boldsymbol{\omega}(l+1), \tag{65}$$

$$\boldsymbol{\sigma}(l) = \nabla_{\mathbf{k}(l)} \hat{\boldsymbol{\phi}}(l) \cdot \boldsymbol{\omega}(l), \tag{66}$$

$$\nabla_{\mathbf{k(l)}} J = \tau \left(\nabla_{\mathbf{k}(l)} F(l) + \boldsymbol{\sigma}^{\mathrm{T}}(l) \right), \tag{67}$$

for $l = N-2, N-3, \dots, 0$.

Before giving recursive expressions for computing the Hessians, we adopted the usual convention such that, for some function $\mathbf{z} = f(\mathbf{v})$, the Hessian is defined by $\nabla_{\mathbf{v}}^2 \mathbf{z} = \nabla_{\mathbf{v}}[\mathrm{vec}(\nabla_{\mathbf{v}} \mathbf{z})]$. Then, in order to calculate $\nabla_{\mathbf{k}}^2 J$ in the j-th iteration of the Newton algorithm (Section 5.2), the derivatives of Equations (64)–(67), with respect to vectors $\hat{\mathbf{x}}$ and $\mathbf{k}$, were taken. We obtained the following recursive matrix relation:

$$\mathbf{W}(N-1) = \mathbf{0}, \tag{68}$$

$$
\begin{aligned}
\mathbf{W}(l) = {} & \nabla_{\hat{\mathbf{x}}(l+1)} \left[\mathrm{vec} \left(\nabla_{\hat{\mathbf{x}}(l+1)}^{\mathrm{T}} F(l+1) \right) \right] \\
& + \nabla_{\hat{\mathbf{x}}(l+1)} \left[\mathrm{vec} \left(\nabla_{\hat{\mathbf{x}}(l+1)} \hat{\boldsymbol{\phi}}(l+1) \right) \right] \cdot \mathbf{W}(l+1),
\end{aligned}
\tag{69}
$$

$$\mathbf{S}(l) = \nabla_{\mathbf{k}(l)} \left[\mathrm{vec} \left(\nabla_{\mathbf{k}(l)} \hat{\boldsymbol{\phi}}(l) \right) \right] \cdot \mathbf{W}(l), \tag{70}$$

$$\nabla_{\mathbf{k}}^2(l) J = \tau \left(\nabla_{\mathbf{k}(l)} \left[\mathrm{vec} \left(\nabla_{\mathbf{k}(l)} F(l) \right) \right] + \mathbf{S}(l) \right), \tag{71}$$

for $l = N-2, N-3, \dots, 0$.

In the previous expressions, (64)–(71), the derivatives of the system dynamics (54) and sub-integral function (56), with respect to the vectors $\hat{\mathbf{x}}$ and $\mathbf{k}$,

$$
\begin{aligned}
& \nabla_{\hat{\mathbf{x}}(l+1)} \hat{\boldsymbol{\phi}}(l+1), \ \nabla_{\hat{\mathbf{x}}(l+1)} F(l+1), \ \nabla_{\mathbf{k}(l)} \hat{\boldsymbol{\phi}}(l), \ \nabla_{\mathbf{k}(l)} F(l), \\
& \nabla_{\hat{\mathbf{x}}(l+1)} \left[\mathrm{vec} \left(\nabla_{\hat{\mathbf{x}}(l+1)} \hat{\boldsymbol{\phi}}(l+1) \right) \right], \ \nabla_{\hat{\mathbf{x}}(l+1)} \left[\mathrm{vec} \left(\nabla_{\hat{\mathbf{x}}(l+1)}^{\mathrm{T}} F(l+1) \right) \right], \\
& \nabla_{\mathbf{k}(l)} \left[\mathrm{vec} \left(\nabla_{\mathbf{k}(l)} \hat{\boldsymbol{\phi}}(l) \right) \right], \ \nabla_{\mathbf{k}(l)} \left[\mathrm{vec} \left(\nabla_{\mathbf{k}(l)} F(l) \right) \right],
\end{aligned}
\tag{72}
$$

were calculated by the hyper-dual number method [58,59], which enabled us to get the first- and second-order derivatives in one step, accurately to machine epsilon, which cannot be achieved by the finite difference method. Compared to the widely used and popular method of adjoint automatic differentiation, the hyper-dual number method is more robust and easier to apply.

In the previous considerations from expression (54)–(71) the derivation of recursive matrix relations for the calculation of gradients and Hessians, with respect to the vector $\mathbf{k}$ containing the gains of the PID controller, are shown. Matrix relations for calculating gradients and Hessians, with respect to vector $\boldsymbol{\varphi}$ containing the sensor faults, can be obtained by the same procedure, with obvious changes in notation, and there is no need to give them here.

A flowchart of the entire algorithmic procedure described in Sections 5.1–5.3 is shown in Figure 2.
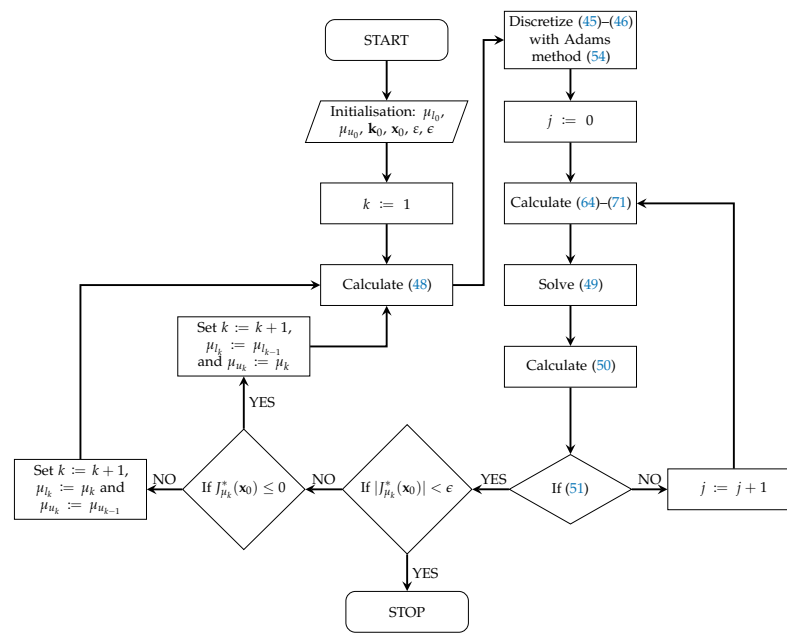
Figure 2. Flowchart of algorithmic procedure for calculation of PID controller gains and determination of the "worst case" of sensor fault functions.

6. Results and Discussions

6.1. Numerical Simulations

In this subsection, the simulation results for control of the robot manipulator, using the controller synthesis algorithm that is described in the previous sections, are presented.

Figure 3 shows the considered structure of a robot manipulator. The structure of a robot manipulator of this type was also the subject of consideration in [60,61]; however, for the sake of completeness, in this paper a more detailed derivation of the mathematical model is presented. Detailed expressions for kinetic and potential energy, and parameters that define the bounds of the robot's dynamic properties, are given (see Appendices B and C).

The first joint is rotary, which enables the rotation of the robot around the vertical axis, while the second and third are prismatic joints, which enable movement in the vertical and horizontal directions, respectively. Such robot manipulators have a workspace in the shape of a cylinder. Robot manipulators with a cylindrical structure, due to their compact design, are most often used for simple assembly tasks, handling machine tools, applying coatings, etc. In addition, such robots can be fast, so a compromise should be sought, because speed implies problems with rotational inertia, which can affect repeatability if the entire system is not configured in accordance with its capabilities. It is usual for cylindrical robots that the up-and-down motion is achieved by pneumatic or hydraulic cylinders, while the rotation is usually achieved by electric motors and gears.

To model the kinematics and dynamics we denoted by q_1 (rad) the rotational coordinate of the first joint, and by q_2 (m) and q_3 (m) the translational coordinates of the second and third joints, respectively.
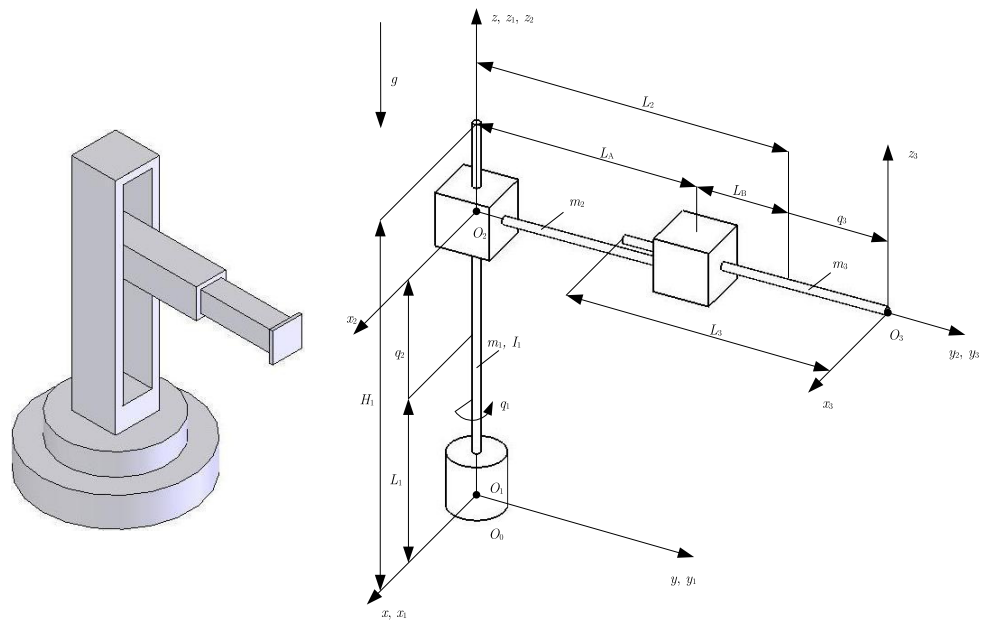
Figure 3. Computer model and schematic representation of a robot manipulator with three degrees of freedom.

First, the forward kinematic equations for the cylindrical structure of a robot manipulator with 3DOF, using the Denavit–Hartenberg convention were developed. The Denavit–Hartenberg parameters are given in Table 1, where $\theta = q_1$, $d_1 = 0$, $d_2 = L_1 + q_2$ and $d_3 = L_2 + q_3$.

Table 1. Denavit–Hartenberg parameters of cylindrical manipulator.

Parameter	θ_i	d_i	a_i	α_i
Joint 1	θ_1	d_1	0	0
Joint 2	0	d_2	0	$\pi/2$
Joint 3	0	d_3	0	0
Unit	rad	m	m	rad

The homogeneous transformation matrices were as follows:

$$\mathbf{A}_1 = Rot(z, q_1) = \begin{bmatrix} \cos(q_1) & -\sin(q_1) & 0 & 0 \\ \sin(q_1) & \cos(q_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{73}$$

$$\mathbf{A}_2 = Tran(0, 0, L_1 + q_2) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & L_1 + q_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{74}$$

$$\mathbf{A}_3 = Tran(0, L_2 + q_3, 0) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & L_2 + q_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{75}$$

The transformation matrices were as follows:

$$
{}^0\mathbf{T}_1 = \mathbf{A}_1 = \begin{bmatrix} \cos(q_1) & -\sin(q_1) & 0 & 0 \\ \sin(q_1) & \cos(q_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},
\tag{76}
$$

$$
{}^0\mathbf{T}_2 = \mathbf{A}_1\mathbf{A}_2 = \begin{bmatrix} \cos(q_1) & -\sin(q_1) & 0 & 0 \\ \sin(q_1) & \cos(q_1) & 0 & 0 \\ 0 & 0 & 1 & L_1 + q_2 \\ 0 & 0 & 0 & 1 \end{bmatrix},
\tag{77}
$$

$$
{}^0\mathbf{T}_3 = \mathbf{A}_1\mathbf{A}_2\mathbf{A}_3 = \begin{bmatrix} \cos(q_1) & -\sin(q_1) & 0 & -(L_2 + q_3)\sin(q_1) \\ \sin(q_1) & \cos(q_1) & 0 & (L_2 + q_3)\cos(q_1) \\ 0 & 0 & 1 & L_1 + q_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}.
\tag{78}
$$

where ${}^0\mathbf{T}_3$ was the corresponding Denavit–Hartenberg matrix and, based on the last column, we had the position vector of the robot end-effector in the Cartesian space, as follows:

$$
\begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = \begin{bmatrix} -(L_2 + q_3)\sin(q_1) \\ (L_2 + q_3)\cos(q_1) \\ L_1 + q_2 \end{bmatrix}.
\tag{79}
$$

The solution of the inverse kinematic problem could be obtained, based on the following equation:

$$
\mathbf{A}_1^{-1}\,{}^0\mathbf{T}_3 = \mathbf{A}_2\mathbf{A}_3,
\tag{80}
$$

$$
\begin{bmatrix} \star & \star & \star & p_x\cos(q_1) + p_y\sin(q_1) \\ \star & \star & \star & -p_x\sin(q_1) + p_y\cos(q_1) \\ \star & \star & \star & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & L_2 + q_3 \\ 0 & 0 & 1 & L_1 + q_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}.
\tag{81}
$$

Based on the last columns of the previous matrices on the left and right sides, we obtained:

$$
p_x\cos q_1 + p_y\sin(q_1) = 0 \implies q_1 = \arctan\left(-\frac{p_x}{p_y}\right),
\tag{82}
$$

$$
p_z = L_1 + q_2 \implies q_2 = -L_1 + p_z,
\tag{83}
$$

$$
-p_x\sin(q_1) + p_y\cos(q_1) = L_2 + q_3 \implies q_3 = -L_2 - p_x\sin(q_1) + p_y\cos(q_1).
\tag{84}
$$

In order to derive the corresponding dynamic equations of motion, the kinetic energy of the robot was first determined. The overall kinetic energy was calculated as the sum of kinetic energies from corresponding links, as follows:

$$
T = T_1 + T_2 + T_3,
\tag{85}
$$

where

$$
T_1 = \frac{1}{2}I_1\dot{q}_1^2,
\tag{86}
$$

$$
T_2 = \frac{1}{6}m_2\left(L_A^2\dot{q}_1^2 + 3\dot{q}_2^2\right),
\tag{87}
$$

$$
T_3 = \frac{m_3}{2L_3}\left[\frac{1}{3}L_3^3\dot{q}_1^2 + L_3^2(L_2 + q_3)\dot{q}_1^2 + L_3\left((L_2 + q_3)^2\dot{q}_1^2 + \dot{q}_2^2 + \dot{q}_3^2\right)\right].
\tag{88}
$$

Next, the overall potential energy was calculated, as the sum of potential energies from corresponding links, as follows:

$$U = U_1 + U_2 + U_3, \tag{89}$$

where

$$U_1 = \frac{1}{2}m_1 g H_1, \tag{90}$$

$$U_2 = m_2 g (L_1 + q_2), \tag{91}$$

$$U_3 = m_3 g (L_1 + q_2). \tag{92}$$

The Euler–Lagrange equations of motion for a considered robot manipulator were given by

$$\tau_i = \frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}_i}\right) - \frac{\partial T}{\partial q_i} + \frac{\partial U}{\partial q_i}, \quad i = 1, 2, 3, \tag{93}$$

where T was the total kinetic energy defined by (85)–(88), U was the total potential energy defined by (89)–(92), and τ_i was the torque/force applied to the i-th robot link. Energy dissipation in rotational joints, and viscous friction in translational joints were neglected.

The Equation (93) can be written in matrix form (1), such that $\mathbf{q} = [q_1 \ q_2 \ q_3]^{\mathrm{T}}$, $\mathbf{u} = [\tau_1 \ \tau_2 \ \tau_3]^{\mathrm{T}}$ and

$$\mathbf{M}(\mathbf{q})\,\ddot{\mathbf{q}} = \begin{bmatrix} M_{11} & 0 & 0 \\ 0 & m_2 + m_3 & 0 \\ 0 & 0 & m_3 \end{bmatrix}, \tag{94}$$

$$M_{11} = I_1 + L_2 L_3 m_3 + L_2^2 m_3 + \frac{1}{3}\left(L_A^2 m_2 + L_3^2 m_3\right) + (L_3 m_3 + 2L_2 m_3)q_3 + m_3 q_3^2, \tag{95}$$

$$\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\,\dot{\mathbf{q}} = \begin{bmatrix} (L_3 m_3 + 2L_2 m_3 + 2m_3 q_3)\dot{q}_1\,\dot{q}_3 \\ 0 \\ \frac{1}{2}(-L_3 m_3 - 2L_2 m_3 - 2m_3 q_3)\dot{q}_1^2 \end{bmatrix}, \tag{96}$$

$$\mathbf{g}(\mathbf{q}) = \begin{bmatrix} 0 \\ g(m_2 + m_3) \\ 0 \end{bmatrix}. \tag{97}$$

A detailed derivation of expressions for the kinetic and potential energy of the considered robot manipulator is given in Appendix B.

The numerical values of the parameters relevant to the derivation of the dynamic model of the considered robot, using the Euler–Lagrange formalism, are given in Table 2.

Table 2. Numerical values of the robot's constant parameters.

Parameter	L_2	L_3	L_A	m_2	m_3	I_1	g
Value	0.5	0.3	0.4	2.5	1	0.1	9.81
Unit	m	m	m	kg	kg	kgm^2	m/s^2

Furthermore, for the appropriate initialization and numerical efficiency of the proposed algorithmic procedure for PID controller synthesis, we needed the constant values of the coefficients defined in properties (4)–(6). The numerical values of these parameters are given in Table 3, and the expressions for determining them for the considered cylindrical robot are given in Appendix C.

Table 3. Numerical values of the coefficients defined in properties (4)–(6).

Parameter	a_1	a_2	c_2	d_2	c_1	d_1
Value	0.2408	0.6633	1.3	1.0	0.9192	1.4142
Unit	kgm^2	kgm^2	kgm	kg	kgm	kg

The parameters of the algorithm proposed in this paper, for this particular robot manipulator, were set as follows. The vector of initial conditions of the generalized coordinates was $\mathbf{q}_0 = \mathbf{0}$. The initial values of the PID controller gaining satisfying stability conditions were chosen as

$$\mathbf{K}_{P0} = \begin{bmatrix} 120 & 0 & 0 \\ 0 & 100 & 0 \\ 0 & 0 & 50 \end{bmatrix}, \quad \mathbf{K}_{D0} = \begin{bmatrix} 110 & 0 & 0 \\ 0 & 50 & 0 \\ 0 & 0 & 30 \end{bmatrix}, \quad \mathbf{K}_{I0} = \begin{bmatrix} 50 & 0 & 0 \\ 0 & 50 & 0 \\ 0 & 0 & 20 \end{bmatrix}. \tag{98}$$

The stop criterion of the bisection method was chosen as $\epsilon = 10^{-3}$. The criterion for stopping Newton's method was set as $\varepsilon = 10^{-4}$. The final time was $t_f = 2$ sec., and the number of optimization time intervals was $N = 2000$, so that the sampling interval was $\tau = 0.002$ sec.

The desired positions of the generalized coordinates were selected as follows: $q_{d1} = \pi/2$; $q_{d2} = 0.2$ m; $q_{d3} = 0.1$ m. By running the algorithm, we obtained the following PID controller gains:

$$\mathbf{K}_P = \begin{bmatrix} 126.6367 & 0 & 0 \\ 0 & 103.1782 & 0 \\ 0 & 0 & 10.9068 \end{bmatrix}, \quad \mathbf{K}_D = \begin{bmatrix} 50.3292 & 0 & 0 \\ 0 & 4.0766 & 0 \\ 0 & 0 & 42.3659 \end{bmatrix},$$
$$\mathbf{K}_I = \begin{bmatrix} 45.9922 & 0 & 0 \\ 0 & 79.7230 & 0 \\ 0 & 0 & 33.0817 \end{bmatrix}, \tag{99}$$

and the minimum $\mathcal{L}_2$-gain, $\mu = 532.3096$.

The time responses of the rotational coordinates of the first joint q_1 and the translational coordinates of the second and third joints, q_2 and q_3, respectively, are presented in Figure 4. It can be seen from the figures that the transient response lasted approximately 0.8 seconds. The duration of the transient response could be further improved by adding weighting matrices into the objective function (47). When selecting these matrices, one should strive to reach a compromise between the time required to achieve the desired position, and the maximum value of the control torque/force, such that the effect of the sensor faults is kept under a permissible level, and ensures the stability of the closed-loop system.

Figure 5 illustrates the time dependence of the positioning errors. It can be seen that the errors from the desired positions were approximately 10^{-3}. These errors could be further reduced by reducing the parameters ϵ and ε of the bisection and Newton methods. Reducing these parameters would mean an increase in the number of iterations and, thus, the execution time of the algorithm. Furthermore, the positioning errors could also be reduced by choosing a smaller sampling time τ of the Adams method; however, as is well-known, there is a minimal value of τ that guarantees numerical stability.
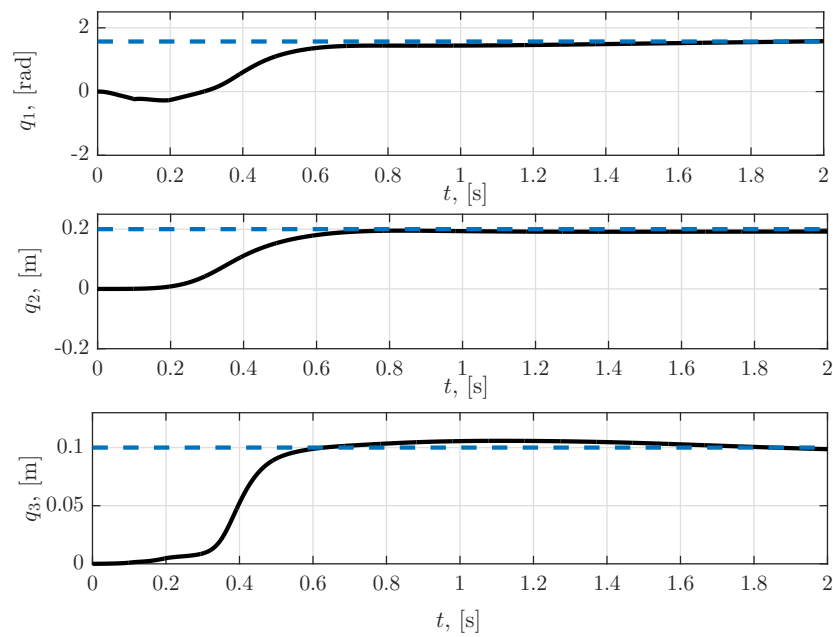
Figure 4. Time responses of the rotational coordinates of the first joint, and of the translational coordinates of the second and third joints.
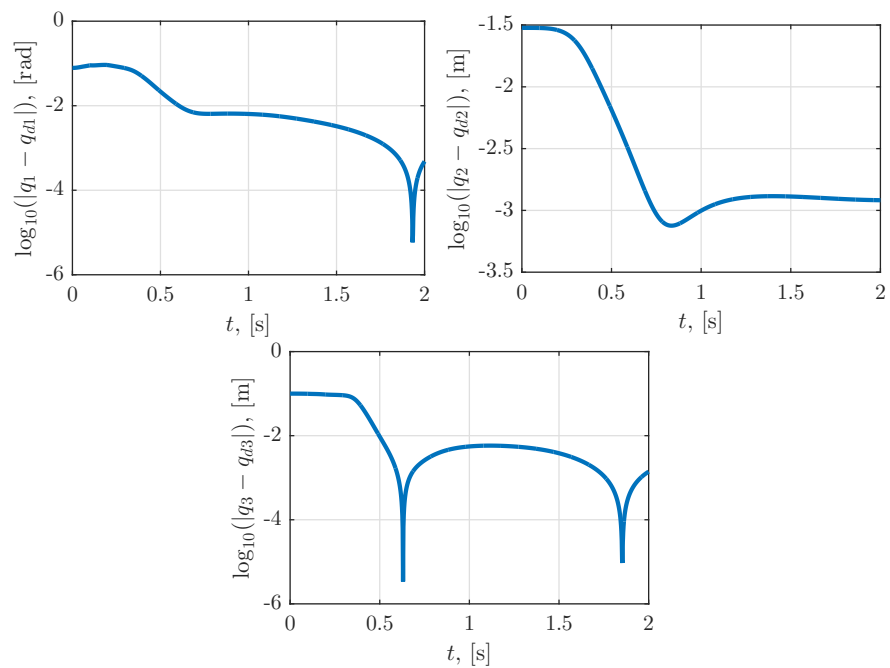

Figure 5. Time dependence of the positioning errors.

Figure 6 shows the time dependence of the force and torques applied to the robot links. Furthermore, Figure 7 illustrates the simulation results for the time dependence of the sensor faults calculated by the algorithm proposed in Sections 5.1 and 5.2. Note that f_{p1}, f_{p2}, f_{p3}, f_{v1}, f_{v2} and f_{v3} are elements of vector $\boldsymbol{\varphi}$, which is incorporated in the objective function of the zero-sum differential game (17), i.e., (47) and, since $\boldsymbol{\varphi}$ is the maximizing player, the sensor fault functions shown in Figure 7 affected the robot manipulator system in the worst possible manner. The obtained results can be interpreted as a sensor failure due to a deviation belonging to the class of bounded $\mathcal{L}_2$ functions.

It is obvious from the figures that the robot manipulator reached the desired positions at an acceptable settling time, with a negligible steady state error and with acceptable

amounts of applied forces and torques to achieve the desired motion: therefore, it can be concluded that the proposed control strategy is efficient in the presence of sensor faults.
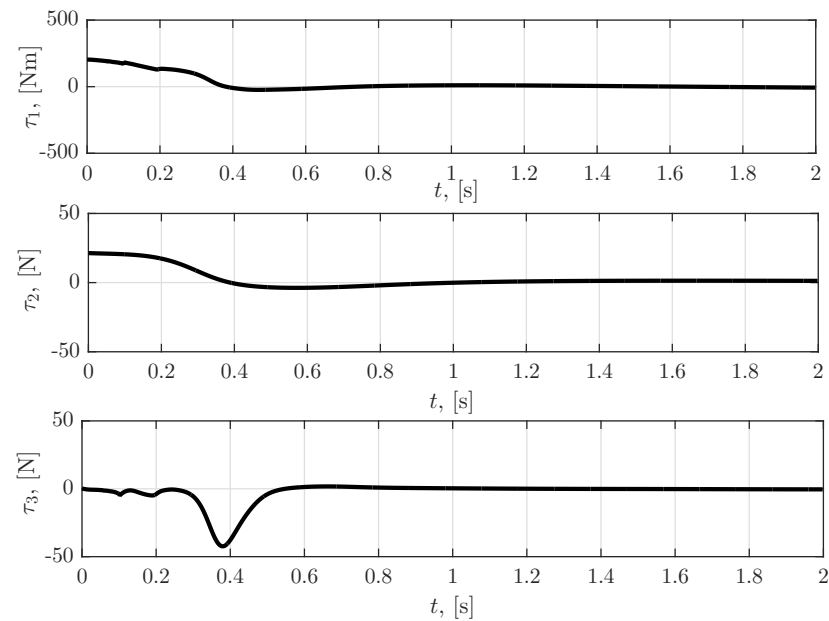

Figure 6. Time dependence of the force and torque applied to the robot links.
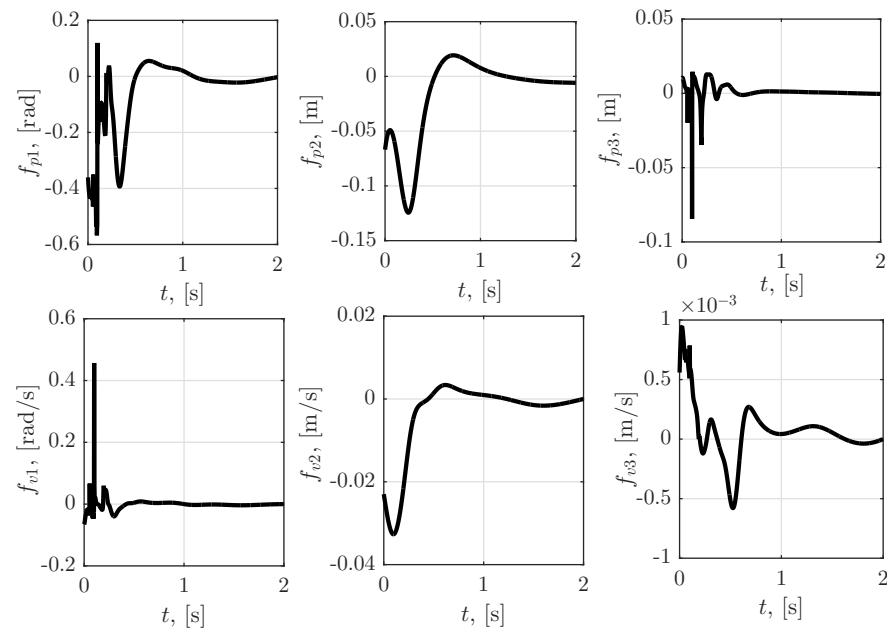

Figure 7. Time dependence of the "worst case" sensor faults obtained by proposed algorithms.

6.2. Discussion of Comparison with Other Methods

Here, we discuss the features of our approach, in comparison to other similar approaches.

In fault-tolerant control approaches based on LMI formalism (see, for example, [6,7,23,29] and references therein), the nonlinear dynamics of the robot manipulator must be linearized, which means that robustness cannot be guaranteed in all operating points. Then, the decentralized PID controller must be transformed into a state feedback controller or an output feedback controller, and the optimization problem is presented in the form of an LMI, in which it is necessary to determine the elements of the positive-definite Lyapunov matrix and the transformation matrix. In the case of a robot with 3DOF, the Lyapunov matrix is

6×6 and the transformation matrix is 3×6: this means that the problem has at least $21 + 9$ optimization parameters. In our approach, the PID controller gains were directly optimized, which meant that we only had 9 optimization parameters. Furthermore, the Mehrotra-type predictor–corrector variant of interior-point method algorithms is commonly used to solve the LMI problem: in our approach, we use Newton's method, which—near the saddle point of the associated differential game—produces a sequence that quadratically converges to a desired solution, while the Mehrotra-type predictor–corrector has linear convergence. Thus, compared to methods based on solving LMI, the approach proposed in this paper has less optimization variables, and requires less iterations.

Many of the approaches to fault-tolerant control (see, for example, [2,6,12] and references therein), in addition to the synthesis of the controller, also imply a certain detection or estimation of the faults that appear in the nonlinear dynamic system: this actually means additional computational requirements that increase with the number of state variables that need to be estimated due to the synthesis of an additional subsystem. Compared to these approaches, our approach does not require an additional subsystem to determine how to modify the controller structure and gains, but focuses on the $\mathcal{L}_2$-gain robust stability of the robot manipulator, considering the worst-case scenario of sensor faults rather than the desired performance for each fault occurrence scenario.

The algorithmic procedure presented in this paper is intended for the offline solution of the zero-sum differential game related to the $\mathcal{L}_2$-gain optimal control problem, with the explicit calculation of the PID controller gains: therefore, the computational complexity of our approach is not as much of a limitation as for approaches intended for online execution, such as model predictive control (see, for example, [15,16] and references therein).

7. Conclusions

In this paper, the tuning of PID-type controller gains for robot manipulators affected by sensor faults, using an algorithmic procedure that gives an explicit solution to the $\mathcal{L}_2$-gain optimality criterion and related zero-sum differential game, is presented. The main contribution of the paper includes the integration of the simple bisection method, the Newton method for solution of related zero-sum differential games without solving the HJI equation, the Adams method for time discretization, and hyper-dual numbers, to provide an effective method for control law synthesis. By applying Lyapunov stability analysis, and the dissipativity theory and passivity properties of systems described by Euler–Lagrange equations, we derived local $\mathcal{L}_2$ stability conditions that we used for appropriate initialization of the bisection method, and for ensuring and controlling the convergence of Newton's method. The simulation results for control of the 3DOF cylindrical robot manipulator showed that the proposed algorithmic procedure could efficiently calculate the controller gains, in order to position the system affected by sensor faults, as desired.

The extension of the proposed approach could be continued in the following directions:

- Although the case of sensor faults is considered in this paper, the proposed algorithm for control law synthesis could easily be extended to the case of dynamic systems affected by actuator faults, without significant increase in its complexity.
- Improvements in the control strategy proposed in this paper could also go towards the synthesis of a complete model-free control law, i.e., control law without gravity vector compensation: this would complicate the derivation of the stability conditions, and the controller gains should be presented as nonlinear functions of generalized coordinates.
- Instead of the initial conditions being known in advance, one could consider a case where the initial conditions were treated as an unknown uncertainty, i.e., the variables of the min–max optimization problem that maximizes the objective function.
- From the numerical optimization algorithms point of view, to perform a detailed analysis and comparison between a proposed algorithm and other existing methods, such as genetic algorithm or particle swarm optimization, in terms of convergence, accuracy and computational efficiency.

- From the point of view of its application to a robot manipulator affected by sensor faults, to carry out a detailed experimental analysis and comparison between a proposed control strategy and other existing related fault-tolerant control approaches.

Abbreviations

The following abbreviations are used in this manuscript:

DOF	degrees of freedom
FTC	fault-tolerant control
HJI	Hamilton–Jacobi–Isaacs
LMI	linear matrix inequalities
PID	proportional–integral–derivative

Appendix A. Notation

Matrices and vectors are represented in bold upper and bold lower case, respectively. Scalars are represented in italic lower case. $\mathbf{I}$ is an identity matrix, and $\mathbf{0}$ is a null matrix. The dimensions of the matrices and vectors can generally be determined trivially by the context. The symbols ∇ and ∇^2 stand for the gradient and Hessian, respectively. The symbol $^\mathrm{T}$ denotes transposition.

The $\mathrm{vec}(\cdot)$ is an operator that stacks the columns of a matrix one underneath the other. The Kronecker product of the two matrices $\mathbf{A}$ $(m \times n)$ and $\mathbf{B}$ $(p \times q)$, denoted by $\mathbf{A} \otimes \mathbf{B}$, is an $mp \times nq$ matrix defined by $\mathbf{A} \otimes \mathbf{B} = (a_{ij}\mathbf{B})_{ij}$. The definitions of the matrix differentials calculus, and the algebras related to Kronecker products, can be found in [62,63].

The Euclidean norm of vector is defined as $\|\mathbf{x}\| = \sqrt{\mathbf{x}^\mathrm{T}\mathbf{x}}$. $\lambda_{\min}\{\mathbf{A}\}$ and $\lambda_{\max}\{\mathbf{A}\}$ represent the smallest and the largest eigenvalues, respectively, of the symmetric positive-definite matrix $\mathbf{A}$. The induced norm of matrix $\mathbf{A}$ is defined as $\|\mathbf{A}\| = \sqrt{\lambda_{\max}\{\mathbf{A}^\mathrm{T}\mathbf{A}\}} = \sigma_{\max}\{\mathbf{A}\}$, where $\sigma_{\max}\{\mathbf{A}\}$ represents the largest singular value of matrix $\mathbf{A}$. If $\mathbf{A}$ is a symmetric positive-definite matrix, it implies $\|\mathbf{A}\| = \lambda_{\max}\{\mathbf{A}\}$; furthermore, for a symmetric positive-definite matrix, it holds that $\lambda_{\min}\{\mathbf{A}\}\|\mathbf{x}\|^2 \le \mathbf{x}^\mathrm{T}\mathbf{A}\mathbf{x} \le \lambda_{\max}\{\mathbf{A}\}\|\mathbf{x}\|^2$.

$\mathcal{L}_2(I, \mathbb{R}^n)$ stands for the standard Lebesgue spaces of vector-valued square-integrable and essentially bounded functions mapping an interval $I \subset \mathbb{R}$ to $\mathbb{R}^n$: this space is equipped with an $\mathcal{L}_2$ norm defined by $\| \cdot \|_{\mathcal{L}_2} = \sqrt{\int_{t_0}^{t_f} \| \cdot \|^2 dt}$. We avoided explicitly showing the dependence of the variables from the time when not needed.

Appendix B. Detailed Derivation of Expressions for the Kinetic and Potential Energy of the Cylindrical Robot

The movement of the first link of the robot can be viewed as a rotation around an axis, so the kinetic energy can be determined as follows:

$$T_1 = \frac{1}{2} I_1 \dot{q}_1^2, \tag{A1}$$

where I_1 is dynamic moment of inertia $I_1 = \frac{1}{2} m_1 R^2$. The potential energy of the first link is defined by

$$U_1 = -m_1 \mathbf{g}^T \mathbf{p}_1, \tag{A2}$$

where $\mathbf{p}_1 = \begin{bmatrix} 0 & 0 & \frac{H_1}{2} & 1 \end{bmatrix}^T$ is the vector of the position of the center of gravity of the infinitesimal mass of the first link in relation to the fixed coordinate system, and $\mathbf{g} = \begin{bmatrix} 0 & 0 & -g & 0 \end{bmatrix}^T$ is the vector of gravitational acceleration. By inserting them into the Equation (A2), we get

$$U_1 = \frac{1}{2} g m_1 H_1. \tag{A3}$$

The kinetic energy of the second link is

$$T_2 = \frac{1}{2} \int_{(m)} v_2^2 dm_2. \tag{A4}$$

If we assume that the second link is homogeneous, then it follows:

$$\frac{dm_2}{m_2} = \frac{du_2}{L_A} \Rightarrow dm_2 = \frac{m_2}{L_A} du_2, \tag{A5}$$

where dm_2 is infinitesimal mass and du_2 is infinitesimal displacement. Substituting expression (A5) into (A4) and changing the bounds of integration, it follows:

$$T_2 = \frac{m_2}{2L_A} \int\limits_0^{L_A} v_2^2 du_2. \tag{A6}$$

The position of the dm_2 in the robot coordinate system is determined by the position vector

$$\mathbf{R}_2 = \begin{bmatrix} 0 & u_2 & 0 & 1 \end{bmatrix}^T. \tag{A7}$$

The position vector of the mass dm_2, with respect to the fixed coordinate system, is

$$\mathbf{p}_2 = {}^0\mathbf{T}_2 \mathbf{R}_2. \tag{A8}$$

By inserting expressions (77) and (A7) into expression (A8), we get

$$\mathbf{p}_2 = \begin{bmatrix} -u_2 \sin(q_1) & u_2 \cos(q_1) & L_1 + q_2 & 1 \end{bmatrix}^T. \tag{A9}$$

The velocity of the mass dm_2 is

$$\mathbf{v}_2 = \frac{d\mathbf{p}_2}{dt} = \begin{bmatrix} -u_2 \dot{q}_1 \cos(q_1) & -u_2 \dot{q}_1 \sin(q_1) & \dot{q}_2 & 0 \end{bmatrix}^T, \tag{A10}$$

and the square of the velocity is

$$v_2^2 = \mathbf{v}_2 \cdot \mathbf{v}_2 = u_2^2 \dot{q}_1^2 \cos^2(q_1) + u_2^2 \dot{q}_1^2 \sin^2(q_1) + \dot{q}_2^2. \tag{A11}$$

Inserting Equation (A11) into (A6), and integrating from 0 to L_A, it follows:

$$
\begin{aligned}
T_2 &= \frac{m_2}{2L_A} \left[\frac{L_A^3}{3} \dot{q}_1^2 \cos^2(q_1) + \frac{L_A^3}{3} \dot{q}_1^2 \sin^2(q_1) + L_A \dot{q}_2^2 \right] \\
&= \frac{m_2}{2L_A} \left[\frac{L_A^3}{3} \dot{q}_1^2 \left(\cos^2(q_1) + \sin^2(q_1) \right) + L_A \dot{q}_2^2 \right] \\
&= \frac{1}{6} m_2 \left(L_A^2 \dot{q}_1^2 + 3 \dot{q}_2^2 \right).
\end{aligned}
\tag{A12}
$$

The potential energy of the second link is calculated as follows:

$$U_2 = -m_2 \mathbf{g}^T \mathbf{p}_2, \tag{A13}$$

where the vector $\mathbf{p}_2$ is determined by (A9) for $u_2 = L_A/2$, so the potential energy is equal to

$$U_2 = -m_2 \begin{bmatrix} 0 & 0 & -g & 0 \end{bmatrix} \begin{bmatrix} -\frac{1}{2}L_A \sin(q_1) \\ \frac{1}{2}L_A \cos(q_1) \\ L_1 + q_2 \\ 1 \end{bmatrix} = gm_2(L_1 + q_2). \tag{A14}$$

The same assumptions about the homogeneity of the second link and the linearity of the mass distribution are applied to the calculation of the kinetic and potential energy of the third link.

The kinetic energy of the third link is

$$T_3 = \frac{1}{2} \int_{(m)} v_3^2 dm_3, \tag{A15}$$

and, with the $dm_3 = \frac{m_3}{L_3} du_3$, it follows:

$$T_3 = \frac{m_3}{2L_3} \int_{-L_3}^{0} v_3^2 du_3. \tag{A16}$$

The position vector of the infinitesimal mass dm_3 in the robot coordinate system is

$$\mathbf{R}_3 = \begin{bmatrix} 0 & -u_3 & 0 & 1 \end{bmatrix}^T. \tag{A17}$$

The position vector of the mass dm_3, with respect to the fixed coordinate system, is equal to

$$\mathbf{p}_3 = {}^0\mathbf{T}_3\mathbf{R}_3 = \begin{bmatrix} u_3 \sin(q_1) - (L_2 + q_3) \sin(q_1) \\ (L_2 + q_3) \cos(q_1) - u_3 \cos(q_1) \\ L_1 + q_2 \\ 1 \end{bmatrix}, \tag{A18}$$

where the transfer matrix from the third coordinate system to the fixed coordinate system is defined by (78). The velocity of the mass dm_3 is

$$\mathbf{v}_3 = \frac{d\mathbf{p}_3}{dt} = \begin{bmatrix} u_3\dot{q}_1 \cos(q_1) - (L_2 + q_3)\dot{q}_1 \cos(q_1) - \dot{q}_3 \sin(q_1) \\ u_3\dot{q}_1 \sin(q_1) - (L_2 + q_3)\dot{q}_1 \sin(q_1) + \dot{q}_3 \cos(q_1) \\ \dot{q}_2 \\ 0 \end{bmatrix}, \tag{A19}$$

while the square of the velocity is

$$v_3^2 = \mathbf{v}_3 \cdot \mathbf{v}_3 = \dot{q}_1^2(L_2 - u_3 + q_3)^2 + \dot{q}_2^2 + \dot{q}_3^2. \tag{A20}$$

By inserting Equation (A20) into (A16), and after integrating within the limits from 0 to $-L_3$, we get kinetic energy of the third link as follows:

$$T_3 = \frac{m_3}{2L_3} \left[\frac{1}{3}L_3^3\dot{q}_1^2 + L_3^2(L_2 + q_3)\dot{q}_1^2 + L_3\left((L_2 + q_3)^2\dot{q}_1^2 + \dot{q}_2^2 + \dot{q}_3^2\right) \right]. \tag{A21}$$

The vector of the position of the center of gravity of the third link with respect to the fixed coordinate system is obtained by inserting $u_3 = L_3/2$ into (A18):

$$\mathbf{p}_3 = \begin{bmatrix} \frac{1}{2}L_3\sin(q_1) - (L_2 + q_3)\sin(q_1) \\ (L_2 + q_3)\cos(q_1) - \frac{1}{2}L_3\cos(q_1) \\ L_1 + q_2 \\ 1 \end{bmatrix} \tag{A22}$$

The potential energy of the third link is calculated as follows:

$$U_3 = -m_3\mathbf{g}^T\mathbf{p}_3, \tag{A23}$$

according to which, we get

$$U_3 = -m_3\begin{bmatrix} 0 & 0 & -g & 0 \end{bmatrix}\begin{bmatrix} \frac{1}{2}L_3\sin(q_1) - (L_2 + q_3)\sin(q_1) \\ (L_2 + q_3)\cos(q_1) - \frac{1}{2}L_3\cos(q_1) \\ L_1 + q_2 \\ 1 \end{bmatrix} \tag{A24}$$

$$= gm_3(L_1 + q_2).$$

Appendix C. Properties of the Dynamic Model of the Cylindrical Robot

Based on the inertia matrix defined by (94) and (95), using expression (4) with $\mathbf{q} = [q_1\ q_2\ q_3]^T$ and $\boldsymbol{\xi} = [\xi_1\ \xi_2\ \xi_3]^T$, the parameters a_1, a_2, c_2 and d_2 are determined as follows.

On the left-hand side of expression (4), we have

$$\left[I_1 + L_2L_3m_3 + \frac{1}{3}\left(L_A^2m_2 + L_3^2m_3\right) + L_3m_3q_3 + m_3(L_2 + q_3)^2 \right]\xi_1^2$$

$$+ (m_2 + m_3)\xi_2^2 + m_3\xi_3^2$$

$$\geq \left(-\frac{(L_3m_3 + 2L_2m_3)^2 - 4m_3p}{4m_3} \right)\xi_1^2 + (m_2 + m_3)\xi_2^2 + m_3\xi_3^2 \tag{A25}$$

$$\geq \min\left\{ \left(-\frac{(L_3m_3 + 2L_2m_3)^2 - 4m_3p}{4m_3} \right), m_2 + m_3, m_3 \right\}\left(\xi_1^2 + \xi_2^2 + \xi_3^2\right),$$

where

$$p = I_1 + L_2L_3m_3 + L_2^2m_3 + \frac{1}{3}(L_A^2m_2 + L_3^2m_3). \tag{A26}$$

from which it follows:

$$a_1 = \min\left\{ \left(-\frac{(L_3m_3 + 2L_2m_3)^2 - 4m_3p}{4m_3} \right), m_2 + m_3, m_3 \right\} \Rightarrow$$

$$a_1 = \min\left\{ I_1 + \frac{1}{12}L_3^2m_3 + \frac{1}{3}L_A^2m_2, m_3 \right\} \tag{A27}$$

Furthermore, on the right-hand side of expression (4), we have

$$\underbrace{\left[I_1 + L_2L_3m_3 + L_2^2m_3 + \frac{1}{3}\left(L_A^2m_2 + L_3^2m_3\right) \right]}_{p}\xi_1^2$$

$$+ \left[(L_3m_3 + 2L_2m_3)q_3 + m_3q_3^2 \right]\xi_1^2 + (m_2 + m_3)\xi_2^2 + m_3\xi_3^2 \tag{A28}$$

$$\leq \left[a_2 + c_2\sqrt{q_1^2 + q_2^2 + q_3^2} + d_2(q_1^2 + q_2^2 + q_3^2) \right]\left(\xi_1^2 + \xi_2^2 + \xi_3^2\right).$$

We can write the above expression as follows:

$$\left[a_2 - p + c_2 \sqrt{q_1^2 + q_2^2 + q_3^2} - (L_3 m_3 + 2L_2 m_3)q_3 + d_2(q_1^2 + q_2^2 + q_3^2) - m_3 q_3^2 \right] \xi_1^2$$
$$+ \left[a_2 - (m_2 + m_3) \right] \xi_2^2 + \left[a_2 - m_3 \right] \xi_3^2 \geq 0, \tag{A29}$$

from which it follows:

$$a_2 \geq p, \quad a_2 \geq m_2 + m_3, \quad a_2 \geq m_3, \quad c_2 \geq L_3 m_3 + 2L_2 m_3, \quad d_2 \geq m_3, \tag{A30}$$

that is, we have

$$a_2 = \min\{p, m_3\}, \quad c_2 = L_3 m_3 + 2L_2 m_3, \quad d_2 = m_3. \tag{A31}$$

Based on the Coriolis vector (96), using expression (5), the parameters c_1 and d_1 are determined as follows.

First, we will square the expression (5):

$$\|\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}\|^2 \leq (c_1 + d_1\|\mathbf{q}\|)^2 \|\dot{\mathbf{q}}\|^4. \tag{A32}$$

On the left side of (A32) we have

$$\|\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}\|^2 = [\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}]^{\mathsf{T}} \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} = h^2 \left(\dot{q}_3^2 + \frac{1}{4}\dot{q}_1^2 \right) \dot{q}_1^2, \tag{A33}$$

where

$$h = L_3 m_3 + 2L_2 m_3 + 2m_3 q_3. \tag{A34}$$

Inserting (A33) into (A32) gives the following inequality:

$$h^2 \left(\dot{q}_3^2 + \frac{1}{4}\dot{q}_1^2 \right) \dot{q}_1^2 \leq b \left(\dot{q}_1^2 + \dot{q}_2^2 + \dot{q}_3^2 \right)^2, \tag{A35}$$

where

$$b = \left(c_1 + d_1 \sqrt{q_1^2 + q_2^2 + q_3^2} \right)^2. \tag{A36}$$

Furthermore, from (A35), it follows

$$\left(b - \frac{1}{4}h^2 \right) \dot{q}_1^4 + 2 \left(b - \frac{1}{2}h^2 \right) \dot{q}_1^2 \dot{q}_3^2 + b \left(\dot{q}_2^4 + \dot{q}_3^4 \right)^2 + 2b\dot{q}_1^2\dot{q}_2^2 \geq 0, \tag{A37}$$

which will be satisfied if

$$b \geq \frac{1}{4}h^2,$$
$$c_1 + d_1 \sqrt{q_1^2 + q_2^2 + q_3^2} \geq \frac{\sqrt{2}}{2} [L_3 m_3 + 2L_2 m_3 + 2m_3 q_3], \tag{A38}$$

so that, in the end, we get

$$c_1 = \frac{\sqrt{2}}{2} L_3 m_3 + \sqrt{2} L_2 m_3, \quad d_1 = \sqrt{2} m_3. \tag{A39}$$

The parameter k_{g2} from (6) is equal to zero, because the considered cylindrical robot structure does not have translational degrees of freedom of motion at an angle that changes in relation to the gravitational field. As the gravitational vector defined by (97) is not a function of the controlled coordinates, it also follows from (6) that the parameter k_{g1} is equal to zero.

References

1. Baioumy, M.; Pezzato, C.; Ferrari, R.; Corbato, C.H.; Hawes, N. Fault-tolerant control of robot manipulators with sensory faults using unbiased active inference. In Proceedings of the 2021 European Control Conference (ECC), Delft, The Netherlands, 29 June–2 July 2021 ; pp. 1119–1125.
2. Kang, Y.; Yao, L.; Wu, W. Sensor fault diagnosis and fault tolerant control for the multiple manipulator synchronized control system. *ISA Trans.* **2020**, *106*, 243–252. [CrossRef] [PubMed]
3. Boukhari, M.R.; Chaibet, A.; Boukhnifer, M.; Glaser, S. Proprioceptive sensors' fault tolerant control strategy for an autonomous vehicle. *Sensors* **2018**, *18*, 1893. [CrossRef] [PubMed]
4. Xiao, B.; Cao, L.; Xu, S.; Liu, L. Robust tracking control of robot manipulators with actuator faults and joint velocity measurement uncertainty. *IEEE/ASME Trans. Mechatron.* **2020**, *25*, 1354–1365. [CrossRef]
5. Nguyen, N.P.; Xuan, M.N.; Hong, S.K. Actuator fault detection and fault-tolerant control for hexacopter. *Sensors* **2019**, *19*, 4721. [CrossRef] [PubMed]
6. Ashraf, M.A.; Ijaz, S.; Javaid, U.; Hussain, S.; Anwaar, H.; Marey, M. A robust sensor and actuator fault tolerant control scheme for nonlinear system. *IEEE Access* **2022**, *10*, 626–637. [CrossRef]
7. Jani, F.; Hashemzadeh, F.; Baradarannia, M.; Kharrati, H. Robust event-triggered finite-time control of faulty networked flexible manipulator under external disturbance. *J. Vib. Control* **2022**, *29*, 317–333. [CrossRef]
8. Ma, H.; Yang, G. Simultaneous fault diagnosis for robot manipulators with actuator and sensor faults. *Inf. Sci.* **2016**, *366*, 12–30. [CrossRef]
9. Tan, C.P.; Edwards, C. Sliding mode observers for robust detection and reconstruction of actuator and sensor faults. *Int. J. Robust Nonlinear Control* **2003**, *13*, 443–463. [CrossRef]
10. Zuev, A.; Zhirabok, A.N.; Filaretov, V.; Protsenko, A. Fault identification in electric servo actuators of robot manipulators described by nonstationary nonlinear dynamic models using sliding mode observers. *Sensors* **2022**, *22*, 317. [CrossRef]
11. Ghaf-Ghanbari, P.; Mazare, M.; Taghizadeh, M. Active fault-tolerant control of a Schönflies parallel manipulator based on time delay estimation. *Robotica* **2021**, *39*, 1518–1535. [CrossRef]
12. Truong, T.N.; Vo, A.T.; Kang, H.J.; Van, M. A novel active fault-tolerant tracking control for robot manipulators with finite-time stability. *Sensors* **2021**, *21*, 8101. [CrossRef]
13. Van, M.; Ceglarek, D. Robust fault tolerant control of robot manipulators with global fixed-time convergence. *J. Frankl. Inst.* **2021**, *358*, 699–722. [CrossRef]
14. Cao, Y.; Song, Y.D. Adaptive PID-like fault-tolerant control for robot manipulators with given performance specifications. *Int. J. Control* **2020**, *93*, 377–386. [CrossRef]
15. Jung, W.; Bang, H. Fault and failure tolerant model predictive control of quadrotor UAV. *Int. J. Aeronaut. Space Sci.* **2021**, *22*, 663–675. [CrossRef]
16. Karras, G.C.; Fourlas, G.K. Model predictive fault tolerant control for omni-directional mobile robots. *J. Intell. Robot. Syst.* **2020**, *97*, 635–655. [CrossRef]
17. Zhang, F.; Wu, W.; Song, R.; Wang, C. Dynamic learning-based fault tolerant control for robotic manipulators with actuator faults. *J. Frankl. Inst.* **2023**, *360*, 862–886. [CrossRef]
18. Zhang, S.; Yang, P.; Kong, L.; Chen, W.; Fu, Q.; Peng, K. Neural networks-based fault tolerant control of a robot via fast terminal sliding mode. *IEEE Trans. Syst. Man Cybern. Syst.* **2021**, *51*, 4091–4101. [CrossRef]
19. Dybkowski, M.; Klimkowski, K. Artificial neural network application for current sensors fault detection in the vector controlled induction motor drive. *Sensors* **2019**, *19*, 571. [CrossRef]
20. You, G.; Xu, B.; Cao, Y.; Hou, X.; Zhao, S.; Liao, W. Adaptive fuzzy finite-time fault-tolerant control design for non-linear systems under sensor faults. *IET Control Theory Appl.* **2022**, *16*, 1560–1572. [CrossRef]
21. Zhang, L.; Liu, H.; Tang, D.; Hou, Y.; Wang, Y. Adaptive fixed-time fault-tolerant tracking control and its application for robot manipulators. *IEEE Trans. Ind. Electron.* **2022**, *69*, 2956–2966. [CrossRef]
22. Van, M.; Ge, S.S. Adaptive fuzzy integral sliding-mode control for robust fault-tolerant control of robot manipulators with disturbance observer. *IEEE Trans. Fuzzy Syst.* **2021**, *29*, 1284–1296. [CrossRef]
23. Makni, S.; Bouattour, M.; El Hajjaji, A.; Chaabane, M. Robust fault tolerant control based on adaptive observer for Takagi-Sugeno fuzzy systems with sensor and actuator faults: Application to single-link manipulator. *Trans. Inst. Meas. Control* **2020**, *42*, 2308–2323. [CrossRef]
24. Wang, H.; Xie, S.; Zhou, B.; Wang, W. Non-fragile robust $\mathcal{H}_\infty$ filtering of Takagi-Sugeno fuzzy networked control systems with sensor failures. *Sensors* **2020**, *20*, 27. [CrossRef] [PubMed]
25. Van Der Schaft, A. $\mathcal{L}_2$-*Gain and Passivity Techniques in Nonlinear Control*; Springer: London, UK, 1996.
26. Van Der Schaft, A. $\mathcal{L}_2$-gain analysis of nonlinear systems and nonlinear state feedback $\mathcal{H}_\infty$ control. *IEEE Trans. Autom. Control* **1992**, *37*, 770–784. [CrossRef]
27. Zuo, Z.; Wang, Y.; Yang, W. $\mathcal{L}_2$-gain fault tolerant control of singular Lipschitz systems in the presence of actuator saturation. *Int. J. Robust Nonlinear Control* **2015**, *25*, 1751–1766. [CrossRef]
28. Wang, J.S.; Yang, G.H. Data-driven output-feedback fault-tolerant $\mathcal{L}_2$ control of unknown dynamic systems. *ISA Trans.* **2016**, *63*, 182–195. [CrossRef]

29. Li, C.; Jing, H.; Bao, J.; Sun, S.; Wang, R. Robust $\mathcal{H}_\infty$ fault tolerant control for quadrotor attitude regulation. *Proc. Inst. Mech. Eng. Part I J. Syst. Control Eng.* **2018**, *232*, 1302–1313. [CrossRef]

30. Yuan, L.H.; Wang, L.D.; Xu, J.T. Adaptive fault-tolerant controller for morphing aircraft based on the $\mathcal{L}_2$ gain and a neural network. *Aerosp. Sci. Technol.* **2023**, *132*, 107985. [CrossRef]

31. Milic, V.; Arandia-Kresic, S.; Lobrovic, M. An application of Newton-like algorithm for $\mathcal{H}_\infty$ proportional–integral–derivative controller synthesis of seesaw-cart system. *Trans. Inst. Meas. Control* **2022**, *44*, 1777–1793. [CrossRef]

32. Milic, V.; Kasac, J.; Novakovic, B. An analytical fuzzy-based approach to $\mathcal{L}_2$-gain optimal control of input-affine nonlinear systems using Newton-type algorithm. *Int. J. Syst. Sci.* **2015**, *46*, 2448–2460. [CrossRef]

33. Aliyu, M.D.S. An improved iterative computational approach to the solution of the Hamilton–Jacobi equation in optimal control problems of affine nonlinear systems with application. *Int. J. Syst. Sci.* **2020**, *51*, 2625–2634. [CrossRef]

34. Sun, W.; Pan, Y.; Lim, J.; Theodorou, E.A.; Tsiotras, P. Min-max differential dynamic programming: Continuous and discrete time formulations. *J. Guid. Control. Dyn.* **2018**, *41*, 2568–2580. [CrossRef]

35. Xi, A.; Cai, Y. A nonlinear finite-time robust differential game guidance law. *Sensors* **2022**, *22*, 6650. [CrossRef]

36. Liu, D.; Xue, S.; Zhao, B.; Luo, B.; Wei, Q. Adaptive dynamic programming for control: A survey and recent advances. *IEEE Trans. Syst. Man Cybern. Syst.* **2021**, *51*, 142–160. [CrossRef]

37. Mu, C.; Wang, K. Approximate-optimal control algorithm for constrained zero-sum differential games through event-triggering mechanism. *Nonlinear Dyn.* **2019**, *95*, 2639–2657. [CrossRef]

38. Vamvoudakis, K.G.; Modares, H.; Kiumarsi, B.; Lewis, F.L. Game theory-based control system algorithms with real-time reinforcement learning: How to solve multiplayer games online. *IEEE Control Syst. Mag.* **2017**, *37*, 33–52.

39. Kasac, J.; Novakovic, B.; Majetic, D.; Brezak, D. Global positioning of robot manipulators with mixed revolute and prismatic joints. *IEEE Trans. Autom. Control* **2006**, *51*, 1035–1040. [CrossRef]

40. Pervozvanski, A.; Freidovich, L. Robust stabilization of robotic manipulators by PID controllers. *Dyn. Control* **1999**, *9*, 203–222. [CrossRef]

41. Hodgson, S.P.; Stoten, D.P. Robustness of the minimal control synthesis algorithm to non-linear plant with regard to the position control of manipulators. *Int. J. Control* **1999**, *72*, 1288–1298. [CrossRef]

42. Gunawardana, R.; Ghorbel, F. On the boundedness of the Hessian of the potential energy of robot manipulators. *J. Robot. Syst.* **1999**, *16*, 613–625. [CrossRef]

43. Ghorbel, F.; Gunawardana, R. A uniform bound for the Jacobian of the gravitational force vector for a class of robot manipulators. *J. Dyn. Syst. Meas. Control* **1997**, *119*, 110–114. [CrossRef]

44. Zhou, N.; Xia, Y. Coordination control of multiple Euler-Lagrange systems for escorting mission. *Int. J. Robust Nonlinear Control* **2015**, *25*, 3515–3830. [CrossRef]

45. Su, Y. Global continuous finite-time tracking of robot manipulators. *Int. J. Robust Nonlinear Control* **2009**, *19*, 1871–1885. [CrossRef]

46. Kasać, J.; Novaković, B.; Majetić, D.; Brezak, D. Passive finite-dimensional repetitive control of robot manipulators. *IEEE Trans. Control Syst. Technol.* **2008**, *16*, 570–576. [CrossRef]

47. Kelly, R.; Santibanez, V.; Loria, A. *Control of Robot Manipulators in Joint Space*; Springer: London, UK, 2005.

48. Caccavale, F.; Marino, A.; Muscio, G.; Pierri, F. Discrete-time framework for fault diagnosis in robotic manipulators. *IEEE Trans. Control Syst. Technol.* **2013**, *21*, 1858–1873. [CrossRef]

49. Namvar, M.; Aghili, F. Failure detection and isolation in robotic manipulators using joint torque sensors. *Robotica* **2010**, *28*, 549–561. [CrossRef]

50. Chen, Y.; Guo, B. Sliding mode fault tolerant tracking control for a single-link flexible joint manipulator system. *IEEE Access* **2019**, *7*, 83046–83057. [CrossRef]

51. Khalil, H.K. *Nonlinear Systems*; Prentice-Hall Inc.: Upper Saddle River, NJ, USA , 2002.

52. Hill, D.; Moylan, P. The stability of nonlinear dissipative systems. *IEEE Trans. Autom. Control* **1976**, *21*, 708–711. [CrossRef]

53. Yuliar, S.; James, M.R.; Helton, J.W. Dissipative control systems synthesis with full state feedback. *Math. Control Signals Syst.* **1998**, *11*, 335–356. [CrossRef]

54. Willems, J.C. Dissipative dynamical systems part I: General theory. *Arch. Ration. Mech. Anal.* **1972**, *45*, 321–351. [CrossRef]

55. Beck, A.; Ben-Tal, A.; Teboulle, M. Finding a global optimal solution for a quadratically constrained fractional quadratic problem with applications to the regularized total least squares. *SIAM J. Matrix Anal. Appl.* **2006**, *28*, 425–445. [CrossRef]

56. Nocedal, J.; Wright, S.J. *Numerical Optimization*; Springer Science + Business Media, LLC: New York, NY, USA, 2006.

57. Hairer, E.; Nørsett, S.P.; Wanner, G. *Solving Ordinary Differential Equations I—Nonstiff Problems*, 2nd ed.; Springer: Berlin, Germany, 2008.

58. Fike, J.A. Multi-Objective Optimization Using Hyper-Dual Numbers. Ph.D. Thesis, Stanford University, Stanford, CA, USA, 2013.

59. Fike, J.; Alonso, J. The development of hyper-dual numbers for exact second-derivative calculations. In Proceedings of the 49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition, Orlando, FL, USA, 4–7 January 2011 ; pp. 1–17.

60. De Souza, D.A.; Batista, J.G.; Vasconcelos, F.J.S.; Dos Reis, L.L.N.; Machado, G.F.; Costa, J.R.; Junior, J.N.N.; Silva, J.L.N.; Rios, C.S.N.; Júnior, A.B.S. Identification by recursive least squares with Kalman filter (RLS-KF) applied to a robotic manipulator. *IEEE Access* **2021**, *9*, 63779–63789. [CrossRef]

61. Batista, J.; Souza, D.; dos Reis, L.; Barbosa, A.; Araújo, R. Dynamic model and inverse kinematic identification of a 3-DOF manipulator using RLSPSO. *Sensors* **2020**, *20*, 416. [CrossRef]

62. Graham, A. *Kronecker Products and Matrix Calculus: With Applications*; Ellis Horwood Limited: West Sussex, UK, 1981.

63. Brewer, J.W. Kronecker products and matrix calculus in system theory. *IEEE Trans. Circuits Syst.* **1978**, *25*, 772–781. [CrossRef]