Adaptive Robust Controller Design-Based RBF Neural Network for Aerial Robot Arm Model

Izzat Al-Darraji 1,2, Dimitrios Piromalis 3, Ayad A. Kakei 2, Fazal Qudus Khan 4, Milos Stojmenovic 5, Georgios Tsaramirsis 6 and Panagiotis G. Papageorgas 7,*

1 Automated Manufacturing Department, University of Baghdad, Baghdad 10081, Iraq; dr.izzat79@gmail.com
2 Mechanical Engineering Department, University of Kirkuk, Kirkuk 36001, Iraq; dr.izzat79@uokirkuk.edu.iq
3 Department of Industrial Design and Production Engineering, University of West Attica, 12244 Athens, Greece; piromali@uniwa.gr
4 Department of Information Technology, Faculty of Computing and IT, King Abdulaziz University, Jeddah 21589, Saudi Arabia; fqkhan@kau.edu.sa
5 Department of Computer Science and Electrical Engineering, Singidunum University, 11000 Belgrade, Serbia; mstojmenovic@singidunum.ac.rs
6 Abu Dhabi Women’s Campus, Higher Colleges of Technology, Abu Dhabi 25026, United Arab Emirates; gtsaramirsis@hct.ac.ae
7 Department of Electrical and Electronics Engineering, University of West Attica, 12244 Athens, Greece
* Correspondence: ppapag@uniwa.gr

Abstract: Aerial Robot Arms (ARAs) enable aerial drones to interact and influence objects in various environments. Traditional ARA controllers need the availability of a high-precision model to avoid high control chattering. Furthermore, in practical applications of aerial object manipulation, the payloads that ARAs can handle vary, depending on the nature of the task. The high uncertainties due to modeling errors and an unknown payload are inversely proportional to the stability of ARAs. To address the issue of stability, a new adaptive robust controller, based on the Radial Basis Function (RBF) neural network, is proposed. A three-tier approach is also followed. Firstly, a detailed new model for the ARA is derived using the Lagrange–d’Alembert principle. Secondly, an adaptive robust controller, based on a sliding mode, is designed to manipulate the problem of uncertainties, including modeling errors. Last, a higher stability controller, based on the RBF neural network, is implemented with the adaptive robust controller to stabilize the ARAs, avoiding modeling errors and unknown payload issues. The novelty of the proposed design is that it takes into account high nonlinearities, coupling control loops, high modeling errors, and disturbances due to payloads and environmental conditions. The model was evaluated by the simulation of a case study that includes the two proposed controllers and ARA trajectory tracking. The simulation results show the validation and notability of the presented control algorithm.

Keywords: aerial robot arms; modeling of aerial robot arms; adaptive controller; robust controller; sliding mode controller; RBF neural network; stability of aerial robot arms; trajectory tracking; high modeling errors

1. Introduction

In the last ten years, a significant amount of research has focused on aerial robots. An aerial robot has the advantage of flight within a high level of maneuverability and without including humans in its platform for implementing a required task. Different types of aerial robots exist for a wide range of applications, such as urban search [1], disasters [2], agriculture [3], forests [4], libraries [5], etc. Nowadays, due to the preferable level of mobility offered by aerial robots, robot arms are equipped with aerial robots to form aerial manipulators, which are used for different applications, such as inspection and transportation [6–9]. Aerial manipulators are drones equipped with Aerial Robot Arms (ARAs). The aerial robot arms are usually located on the lower part of the aerial robot [10].
An ARA with two degrees of freedom system can be designed with a quadcopter [11]. This designed quadcopter aerial manipulator is developed for remote inspecting, manipulating, and transporting. Another example of a two degree of freedom ARA can be found in [12], where the designed aerial manipulator is developed for aerial manipulations using a hexarotor. In [13], a winged aerial manipulator of dual two degrees of freedom ARAs is designed for assisting flying and manipulating objects. The main novelty was a new design that supported holding and manipulating objects while in horizontal flight. Their approach was inspired by nature and, moreover, by birds. In [14], a seven degree of freedom ARA equipped with a helicopter platform for manipulating, using a fully actuated redundant robot arm, is presented. They developed a practical aerial manipulating system that generates de-oscillations in terms of low frequency (Phase Circles). In turn, the phased circle is removed by a special coupling that is developed between the seven degrees of freedom redundant ARA and the helicopter. In [15], a new aerial manipulator with a lightweight arm is designed, which can be applied in repairing high-altitude positions. In some situations, such as in [16], the ARA is designed to be above the quadcopter for inspecting bridges. ARAs have many other applications and can be extended to include space robots, such as satellites, for the refilling of tanks during flight or de-orbiting [17].

Due to the high level of nonlinearities in the equation of motion of an ARA, the control design of an ARA is a challenging issue. Hence, various kinematic formulas, dynamic models, and control techniques have been introduced in this field. These are discussed in the related work section of this paper. None of the previous controller models took into account high nonlinearities, coupling control loops, high modeling errors, or disturbances due to payloads and environmental conditions.

In this paper, a two degree of freedom serial ARA of rotating joints that is mounted on the body of an aerial robot is studied. Unlike the traditional robot manipulators found on fixed or mobile ground robots, the kinematics of the ARA is obtained based on Euler angles and the Denavit–Hartenburg method while considering that the base of the ARA is a float. The dynamics of the ARA are derived in detail using the Lagrange–d’Alembert method. An adaptive robust controller-based Radial Basis Function (RBF) neural network is designed to treat all the mentioned issues in ARA manipulators. The methodology used in this work is summarized in Figure 1.

In this paper, a two degree of freedom serial ARA of rotating joints that is mounted on the body of an aerial robot is studied. Unlike the traditional robot manipulators found on fixed or mobile ground robots, the kinematics of the ARA is obtained based on Euler angles and the Denavit–Hartenburg method while considering that the base of the ARA is a float. The dynamics of the ARA are derived in detail using the Lagrange–d’Alembert method. An adaptive robust controller-based Radial Basis Function (RBF) neural network is designed to treat all the mentioned issues in ARA manipulators. The methodology used in this work is summarized in Figure 1.

First, an adaptive robust controller is developed, based on the ARA nominal model and sliding mode control, to overcome the problem of low modeling errors. Second, the RBF neural network of a Gaussian function type is used in the first controller to manage the problem of high modeling errors and disturbances. An additional study is implemented with the second controller that uses RBF neural networks to show the advantage of including RBF neural networks. The validity of the designed controllers is proved using numerical simulations.

The rest of this paper is organized as follows. The related work is discussed in Section 2. Kinematics and dynamics analyses of the ARA are derived in Section 3. Then, in Section 4, a robust controller is developed to manipulate the problem of uncertainties using the nominal mode. Next, in Section 5, the RBF neural network is implemented into adaptive robust controllers to avoid control chattering in situations of high uncertainty (due to modeling error) and disturbances. In Section 6, stability analysis based on a Lyapunov function is discussed. In Section 7, the simulation results are presented. Finally, Section 8 concludes this study by discussing the results and recommendations for future works.
First, an adaptive robust controller is developed, based on the ARA nominal model and sliding mode control, to overcome the problem of low modeling errors. Second, the RBF neural network of a Gaussian function type is used in the first controller to manage the problem of high modeling errors and disturbances. An additional study is implemented with the second controller that uses RBF neural networks to show the advantage of including RBF neural networks. The validity of the designed controllers is proved using numerical simulations.

The rest of this paper is organized as follows. The related work is discussed in Section 2. Kinematics and dynamics analyses of the ARA are derived in Section 3. Then, in Section 4, a robust controller is developed to manipulate the problem of uncertainties. The correctness of the controller is proved by the simulation.

2. Related Work

This section discusses a state of the art, existing work on ARM control. In [18], the momentum of the dynamic model of ARA is considered for an aerial robot by proposing feedback of a torque control type. They developed controllers to show its stability under the assumption of the non-singularity of the aerial robot. A method called “product of exponentials” that considers the ARA position level is implemented in [19] to obtain the kinematics of the ARA and to calculate joint angles of ARA rapidly. In [20], an adaptive controller is developed for the ARA considering the existence of uncertainty in both kinematics and dynamics. They introduce a controller that regulates the attitude and tracks the path, of the end effector continuously. In [21], an adaptive controller with a
novel formula is presented with existing uncertainty in the ARA model. Furthermore, the combination of the error of trajectory and interaction force is discussed. In [22], a robust controller using a Lyapunov technique is presented for the ARA with a model of underactuated type. This control algorithm manages the complexity included in the dynamics of the ARA due to uncertainty in the parameters. In addition, the introduced robust control algorithm eliminates the need for measuring the acceleration of the base of ARAs. In [23], a workstation including a robotic arm named “BLOCKS’ WORLD” is developed. The concept of artificial intelligence is implemented to control the robotic arm using voice. In [24], an aerial robot is developed that can undertake unsupervised missions for indoor search and rescue. The neural network is implemented in image processing to provide mission capability of a dynamic type. In [25], a new control algorithm for an aerial robot based on the human brain principle is introduced. The control algorithm applies a neural network, swarm, and hybrid swarm intelligence. The neural network is provided the ability to control the aerial robot autonomously. A summary of the related work is presented in Table 1.

<table>
<thead>
<tr>
<th>Paper Reference Number</th>
<th>Kinematic Calculation</th>
<th>Dynamics Model</th>
<th>Disturbances due to Environmental Conditions</th>
<th>Disturbances due to Payload</th>
<th>Coupling Control Loops</th>
<th>High Modeling Errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>[19]</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>[20]</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>[21]</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>[22]</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>This paper</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Equipping aerial robots with ARAs generates several new challenges related to the ability to control ARAs. This is because the dynamics of an ARA are nonlinear, with multiple inputs/outputs and coupled control loops with the aerial robot itself. Moreover, modeling an ARA in control engineering depends on the principles of physics that are derived from the ARA model simplifications. Therefore, it can be assumed that the developed models of ARA do not sufficiently represent the physical system. Another issue to be considered in the control of ARAs is the disturbances due to payloads and environment. For instance, consider an ARA task of moving an object in the air. The ARA is flying without any payload in the beginning. When it carries an object of unknown physical properties, it leads to the necessity of designing a controller that can address such upload conditions. Furthermore, the changes in environmental conditions—e.g., winds—will act as another kind of disturbance that leads to changes in the ARA dynamics. Developing a controller to ensure high performance while taking into account nonlinearity, coupling, modeling errors, and disturbances is a challenge in the ARA control design process. Consequently, advanced controllers, such as adaptive robust controllers, of non-fixed gains are advantageous since they provide firm performance without depending on ARA models, payload, and environmental changes.

3. Modeling

3.1. Kinematics Analysis

Position and velocity analysis are the two main parts of kinematic modeling. The position of the aerial robot arm links is obtained based on the Denavit–Hartenburg method [26]. Figure 2 presents the kinematics parameters of the system. As can be seen from Figure 2, a two-link aerial robot arm is attached to the body of an aerial robot, assuming the reference frame $O_{ra0}$ is located on $O_{ra}$. 
such upload conditions. Furthermore, the changes in environmental conditions—e.g., winds—will act as another kind of disturbance that leads to changes in the ARA dynamics. Developing a controller to ensure high performance while taking into account nonlinearity, coupling, modeling errors, and disturbances is a challenge in the ARA control design process. Consequently, advanced controllers, such as adaptive robust controllers, of non-fixed gains are advantageous since they provide firm performance without depending on ARA models, payload, and environmental changes.

3. Modeling

3.1. Kinematics Analysis

Position and velocity analysis are the two main parts of kinematic modeling. The position of the aerial robot arm links is obtained based on the Denavit–Hartenburg method [26]. Figure 2 presents the kinematics parameters of the system. As can be seen from Figure 2, a two-link aerial robot arm is attached to the body of an aerial robot, assuming the reference frame \( O_{\text{Earth}} \) is located on \( O_{\text{ar}} \).

![Figure 2. The aerial robot system and the kinematics parameters.](image)

Let \( O_{\text{Earth}} \), \( O_{\text{ar}} \), \( O_{\text{ra}i} \) be the origin of the reference frames—Earth, aerial robot body, and robot arm links, respectively—where \( i = 0, \ldots, 2 \) represents the joint index of ARA. The joint angles of the aerial robot arm are \( \Gamma_{\text{ra}} = [\sigma_1 \sigma_2]^T \). The body of the aerial robot is orientated by the Euler angles: roll about x-axis \( \phi \), pitch about y-axis \( \theta \), yaw about z-axis \( \psi \); \( \Gamma_{\text{ar}} = [\phi \theta \psi] \). By applying the principle of Euler angles [27], the rotation of the aerial robot, with respect to the earth frame, is determined with the following rotation matrix:

\[
R_{\text{Earth}ar} = \begin{bmatrix}
c\psi & c\theta & c\psi s\phi - s\psi c\phi & c\psi s\theta c\phi + s\psi s\phi \\
s\psi c\theta & s\psi c\theta & s\psi c\theta c\phi + s\psi s\phi & s\psi s\theta c\phi - c\psi s\phi \\
-s\theta & 0 & 0 & c\theta \\
0 & 0 & 0 & 1
\end{bmatrix},
\]

where the c and s denote sine and cosine. The homogeneous transformation matrix that calculates the position and orientation of reference frames \( O_{\text{ra}1} \) with respect to \( O_{\text{ra}0} \) follows:

\[
A_1 = \begin{bmatrix}
c_1 & -s_1 & 0 & a_1c_1 \\
s_1 & c_1 & 0 & a_1s_1 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}.
\]

For \( O_{\text{ra}2} \) with respect to \( O_{\text{ra}1} \):

\[
A_2 = \begin{bmatrix}
c_2 & -s_2 & 0 & a_2c_2 \\
s_2 & c_2 & 0 & a_2s_2 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}.
\]
and $O_{ra2}$ with respect to $O_{ra0}$.

$$T_2^0 = A_1A_2 = \begin{bmatrix} c_{12} & -s_{12} & 0 & a_1c_1 + a_2c_{12} \\ s_{12} & c_{12} & 0 & a_1s_1 + a_2s_{12} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$ \(\text{(4)}\)

Based on the following assumptions:

- The position of the aerial robot body—i.e., reference frame $O_{ar}$ position—with respect to the earth frame is:
  $$p_{\text{Earth}} = [X \ Y \ Z]^T.$$ \(\text{(5)}\)

- The position of the first link of robot arm end—i.e., reference frame $O_{ra1}$—relative to the earth frame is:
  $$p_{\text{Earth}}^{ra(1)} = p_{\text{Earth}} + R_{\text{Earth}}p_{ra}^{ar}(1).$$ \(\text{(6)}\)

- The position of the end effector i.e., reference frame $O_{ra2}$ relative to the earth frame is:
  $$p_{\text{Earth}}^{ra(2)} = p_{\text{Earth}} + R_{\text{Earth}}p_{ra}^{ar}(2).$$ \(\text{(7)}\)

The values of position vectors $p_{ra}^{ar}(1)$ and $p_{ra}^{ar}(2)$ are obtained from the first three rows of the third column of the matrix in Equations (2) and (4), respectively, as below:

$$p_{ra}^{ar}(1) = [a_1c_1 \ a_1s_1 \ 0]^T,$$ \(\text{(8)}\)

$$p_{ra}^{ar}(2) = [a_1c_1 + a_2c_{12} \ a_1s_1 + a_2s_{12} \ 0]^T.$$ \(\text{(9)}\)

By taking the derivatives of Equations (5)–(7), the linear velocities of the aerial robot body, aerial robot arm (link 1, link 2) will be determined as:

$$\dot{p}_{\text{Earth}} = [\dot{X} \ \dot{Y} \ \dot{Z}]^T,$$ \(\text{(10)}\)

$$\dot{p}_{ra}^{ar}(1) = \dot{p}_{\text{Earth}} + R_{\text{Earth}}w_{ra}^{ar} \Gamma_{ra},$$ \(\text{(11)}\)

$$\dot{p}_{ra}^{ar}(2) = \dot{p}_{\text{Earth}} + R_{\text{Earth}}w_{ra}^{ar} \Gamma_{ra}.$$ \(\text{(12)}\)

After considering the linear velocities of the aerial robot arm, with respect to the aerial robot body, in terms of Jacobian matrices and the skew symmetric matrix for rotation matrix, the above equations will be:

$$\dot{p}_{ra}^{ar}(1) = \dot{p}_{ar} + R_{ar}^{Earth} \text{SSM} \left(w_{ar}^{Earth} \right) p_{ra}^{ar}(1) + R_{ar}^{Earth} J_{w1} \dot{\Gamma}_{ra},$$ \(\text{(13)}\)

$$\dot{p}_{ra}^{ar}(2) = \dot{p}_{ar} + R_{ar}^{Earth} \text{SSM} \left(w_{ar}^{Earth} \right) p_{ra}^{ar}(2) + R_{ar}^{Earth} J_{w2} \dot{\Gamma}_{ra}.$$ \(\text{(14)}\)

On the other hand, the angular velocity of the aerial robot body with respect to the earth frame is:

$$w_{ar}^{Earth} = R_{ar}^{Earth} \begin{bmatrix} 1 & 0 & -\phi \\ 0 & c\phi & c\phi s\theta \\ 0 & -s\phi & c\phi \end{bmatrix} \dot{\Gamma}_{ar},$$ \(\text{(15)}\)

Then, the angular velocities of links with respect to the earth frame are:

$$w_{ra}^{Earth}(1) = w_{ar}^{Earth} + R_{ar}^{Earth} J_{w1} \dot{\Gamma}_{ra},$$ \(\text{(16)}\)

$$w_{ra}^{Earth}(2) = w_{ar}^{Earth} + R_{ar}^{Earth} J_{w2} \dot{\Gamma}_{ra}.$$ \(\text{(17)}\)
where Jacobian matrices of the linear and angular velocities are calculated as:

\[
J_v1 = \begin{bmatrix}
-a_1s_1 & 0 \\
a_1c_1 & 0
\end{bmatrix},
\]

\[
J_v2 = \begin{bmatrix}
-a_1s_1 - a_2s_{12} & -a_2s_{12} \\
a_1c_1 + a_2c_{12} & a_2c_{12}
\end{bmatrix},
\]

\[
J_w1 = \begin{bmatrix}
1 & 0
\end{bmatrix},
\]

\[
J_w2 = \begin{bmatrix}
1 & 1
\end{bmatrix}.
\]

3.2. Dynamics Analysis

The dynamic model of the aerial robot arm is obtained by applying the formula of Lagrange–d’Alembert [28]. This formula is explained as:

\[
\frac{d}{dt} \left( \frac{\partial (K - U)}{\partial \Gamma_{ra}} \right) - \frac{\partial (K - U)}{\partial \Gamma_{ra}} = T_{in} - T_{ext},
\]

where \( K \) and \( U \) represent the kinetic and potential energies of the aerial robot arm, respectively. \( T_{in} \) and \( T_{ext} \) are the torques of the joints and disturbances, respectively. The vector of joint torques is:

\[
T_{in} = \begin{bmatrix}
\tau_1 \\
\tau_2
\end{bmatrix}.
\]

The vector of the disturbances is:

\[
T_{ext} = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & a_1s_1 + a_2s_{12} & 0 & 0 & 0 \\
a_1s_1 + a_2s_{12} & a_1c_1 + a_2c_{12} & 0 & 0 & 0 & 0
\end{bmatrix} V.
\]

\( V \) is the vector of the interaction force and moments shown in Figure 3 that resulted from holding objects by the end effector.

\[
V = [F_{int,x} F_{int,y} F_{int,z} M_1 M_2 M_3]^T,
\]

where \( F_{int,x} \), \( F_{int,y} \), and \( F_{int,z} \) are the interaction force in the \( x \), \( y \), and \( z \) directions, respectively. On the other hand, \( M_1 \), \( M_2 \), and \( M_3 \) denote the moment resulting from the held object on the \( x \), \( y \), and \( z \) axes, respectively.
The total kinetic energy is the summation of the kinetic energy of link 1 \( (m_{ra(1)}) \) mass and link 2 \( (m_{ra(2)}) \) mass:

\[
K = K_{ra(1)} + K_{ra(2)},
\]

\[
K_{ra(1)} = \frac{1}{2} m_{ra(1)} \dot{r}_{ra(1)}^2 + \frac{1}{2} M_{ra(1)} \dot{\varphi}_{ra(1)}^2,
\]

\[
K_{ra(2)} = \frac{1}{2} m_{ra(2)} \dot{r}_{ra(2)}^2 + \frac{1}{2} M_{ra(2)} \dot{\varphi}_{ra(2)}^2.
\]

\( \dot{\varphi} \) is the constant inertia matrix at the corresponding reference frames of the aerial robot body and the links of the ARA. The potential energy of the aerial robot arm is the summation of the potential energy of its links:

\[
U = U_{ra(1)} + U_{ra(2)},
\]

\[
U_{ra(1)} = m_{ra(1)} g h_{ra(1)},
\]

\[
U_{ra(2)} = m_{ra(2)} g h_{ra(2)},
\]

\[
h_{ra(1)} = -a_1 \cos(\sigma_1),
\]

\[
h_{ra(2)} = -a_1 \cos(\sigma_1) - a_2 \cos(\sigma_2).
\]

By inserting Equations (21) and (24) into Equation (18) yields the following dynamic equation of the ARA:

\[
M_{ra}(\varphi_{ra}) \ddot{\varphi}_{ra} + C_{ra}(\varphi_{ra}, \dot{\varphi}_{ra}) \dot{\varphi}_{ra} + G_{ra}(\varphi_{ra}) = T_{in} - T_{ext},
\]

where the terms \( M_{ra}(\varphi_{ra}), C_{ra}(\varphi_{ra}, \dot{\varphi}_{ra}), G_{ra}(\varphi_{ra}), T_{in}, \) and \( T_{ext} \) denote the centrifugal, Coriolis, gravitational forces, input torques, and disturbances of the ARA joints, respectively.

4. Robust Control Based on Nominal Mode

Considering the dynamic formula of the aerial robot in Equation (25), practically, there are usually modeling errors in obtaining the mathematical expression of \( M_{ra}(\varphi_{ra}), C_{ra}(\varphi_{ra}, \dot{\varphi}_{ra}), G_{ra}(\varphi_{ra}) \). Hence, there are usually unknowns that can be represented as:

\[
M_{ra}(\varphi_{ra}) = E_M_{ra} + M_C(\varphi_{ra}),
\]

\[
C_{ra}(\varphi_{ra}, \dot{\varphi}_{ra}) = E_C_{ra} + C_C(\varphi_{ra}, \dot{\varphi}_{ra}),
\]

\[
G_{ra}(\varphi_{ra}) = E_G_{ra} + G_C(\varphi_{ra}),
\]

where \( E_M_{ra}, E_C_{ra}, \) and \( E_G_{ra} \) are the constant mathematical representations of \( M_{ra}(\varphi_{ra}), C_{ra}(\varphi_{ra}, \dot{\varphi}_{ra}), \) and \( G_{ra}(\varphi_{ra}) \), respectively. The \( M_C(\varphi_{ra}), C_C(\varphi_{ra}, \dot{\varphi}_{ra}), \) and \( G_C(\varphi_{ra}) \) are the modeling errors of \( M_{ra}(\varphi_{ra}), C_{ra}(\varphi_{ra}, \dot{\varphi}_{ra}), \) and \( G_{ra}(\varphi_{ra}) \), respectively. Sliding mode control is a vital technique for controlling nonlinear systems that have moderate modeling errors robustly [29]. To design the robust controller, for a desired position \( \Gamma_{rad}(t) \) and actual position value \( \Gamma_{ra}(t) \), let us first assume the error in tracking as:

\[
e(t) = \Gamma_{rad}(t) - \Gamma_{ra}(t).
\]

Then, for \( \rho > 0 \), the sliding mode formula is designed as:

\[
\Xi = \dot{\Xi} + \mu e.
\]

In terms of robustness \( \Xi \), the velocity and acceleration of the ARA assumed as \( \Gamma_{ra}\Xi = \Xi(t) + \Gamma_{ra}(t), \Gamma_{ra}\Xi = \Xi(t) + \Gamma_{ra}(t) \), respectively. From these two formulas and
Equations (29) and (30), the values of velocity and acceleration of the ARA while considering the slide mode control are: \( \dot{\bar{\bar{\tau}}}_{ra} = \ddot{\bar{\bar{\tau}}}_{rad} + \mu e \) and \( \ddot{\bar{\bar{\tau}}}_{ra} = \dddot{\bar{\bar{\tau}}}_{rad} + \mu \dot{e} \).

Now, referring to Equation (24), the input torque to the ARA is calculated as:

\[
T_{in} = M_{ra}(\Gamma_{ra})\bar{\bar{\tau}}_{ra} + C_{ra}\left(\bar{\bar{\tau}}_{ra}, \bar{\bar{\dot{\tau}}}_{ra}\right)\bar{\bar{\dot{\tau}}}_{ra} + G_{ra}(\Gamma_{ra}) + T_{ext} = M_{ra}(\Gamma_{ra})\bar{\bar{\dot{\tau}}}_{ra} + \bar{\bar{\bar{\tau}}}_{ra} + C_{ra}\left(\bar{\bar{\tau}}_{ra}, \bar{\bar{\dot{\tau}}}_{ra}\right)\bar{\bar{\dot{\tau}}}_{ra} + G_{ra}(\Gamma_{ra}) - M_{ra}(\Gamma_{ra})\bar{\bar{\dot{\tau}}}_{ra} - C_{ra}\left(\bar{\bar{\tau}}_{ra}, \bar{\bar{\dot{\tau}}}_{ra}\right)\bar{\bar{\dot{\tau}}}_{ra} + G_{C}(\Gamma_{ra}) + E_{M_{ra}}\bar{\bar{\dot{\tau}}}_{ra} + E_{C_{ra}}\bar{\bar{\dot{\tau}}}_{ra} + E_{G_{ra}} - M_{ra}(\Gamma_{ra})\bar{\bar{\dot{\tau}}}_{ra} - C_{ra}\left(\bar{\bar{\tau}}_{ra}, \bar{\bar{\dot{\tau}}}_{ra}\right)\bar{\bar{\dot{\tau}}}_{ra} + T_{ext}
\]

In terms of the nominal torque part \( T_{N} \), robust torque part \( T_{\|} \), proportional gain \( K_p > 0 \), and integral gain \( K_i > 0 \), the input torque to the ARA is:

\[
T_{in} = T_{N} + T_{\|} + K_p \bar{\bar{\dot{\tau}}} + K_i \int \bar{\bar{\dot{\tau}}} dt,
\]

where:

\[
M_{C}(\Gamma_{ra})\bar{\bar{\dot{\tau}}}_{ra} + C_{C}\left(\bar{\bar{\tau}}_{ra}, \bar{\bar{\dot{\tau}}}_{ra}\right)\bar{\bar{\dot{\tau}}}_{ra} + G_{C}(\Gamma_{ra}).
\]

Then:

\[
T_{\|} = K_2 \text{sgn}(\bar{\bar{\dot{\tau}}}).
\]

Inserting Equations (33) and (34) into Equation (32) and substituting the latter into Equation (31) results in the following dynamics equation of the ARA:

\[
M_{ra}(\Gamma_{ra})\bar{\bar{\dot{\tau}}} + C_{ra}\left(\bar{\bar{\tau}}_{ra}, \bar{\bar{\dot{\tau}}}_{ra}\right)\bar{\bar{\dot{\tau}}} + K_i \int \bar{\bar{\dot{\tau}}} dt = -K_p \bar{\bar{\dot{\tau}}} - K_2 \text{sgn}(\bar{\bar{\dot{\tau}}}) + E_{M_{ra}}\bar{\bar{\dot{\tau}}} + E_{C_{ra}}\bar{\bar{\dot{\tau}}} + E_{G_{ra}} + K_2 \text{sgn}(\bar{\bar{\dot{\tau}}}).
\]

The drawback of the robust controller is that the nominal model of the ARA should be calculated. Besides this, the big errors in the model yield a large \( K_2 \). In turn, high control chattering is produced \([30,31]\). To avoid this issue, an RBF neural network will be applied.

5. Adaptive RBF Neural Network Control

An inexact ARA model causes tracking errors \([32]\). To enhance the tracking performance of our designed robust controller in the presence of high modeling errors and disturbances of the ARA, an RBF neural network, as an artificial intelligence approach, is implemented due to its learning and mapping abilities \([33,34]\). The RBF neural network, as shown in Figure 4, is applied to compensate for the errors in \( M_{NN}(\Gamma_{ra}), C_{NN}\left(\bar{\bar{\tau}}_{ra}, \bar{\bar{\dot{\tau}}}_{ra}\right) \), and \( G_{NN}(\Gamma_{ra}) \).

![Figure 4. Block diagram of the proposed technique.](image-url)
$G_{NN}(\Gamma_{ra})$ is applied to model $M_{ra}(\Gamma_{ra})$, $C_{ra}(\Gamma_{ra}, \dot{\Gamma}_{ra})$, and $G_{ra}(\Gamma_{ra})$. Regarding the NN weight and hidden layers, the ideal weight value of RBF is assumed $W_{Mr_a}$, $W_{Cr_a}$, and $W_{Gr_a}$, while the output of the hidden layers is $H_{Mr_a}$, $H_{Cr_a}$, and $H_{Gr_a}$.

Thus,

$$M_{ra}(\Gamma_{ra}) = E_{Mra} + M_{NN}(\Gamma_{ra}),$$

$$C_{ra}(\Gamma_{ra}) = E_{Cra} + C_{NN}(\Gamma_{ra}),$$

$$G_{ra}(\Gamma_{ra}) = E_{Gra} + G_{NN}(\Gamma_{ra}),$$

Inserting the above equations in the dynamics formula of the ARA—i.e., Equation (25)—results in:

$$M_{ra}(\Gamma_{ra})\ddot{\Gamma}_{ra} + C_{ra}(\Gamma_{ra}, \dot{\Gamma}_{ra})\dot{\Gamma}_{ra} + G_{ra}(\Gamma_{ra}) = E_{Mra}\dot{\Gamma}_{ra} + E_{Cra}\dot{\Gamma}_{ra} + E_{Gra} + T_{ext} +
$$

$$M_{NN}(\Gamma_{ra})\ddot{\Gamma}_{ra} + C_{NN}(\Gamma_{ra}, \dot{\Gamma}_{ra})\dot{\Gamma}_{ra} + G_{NN}(\Gamma_{ra}, \dot{\Gamma}_{ra}) = E_{Mra}\dot{\Gamma}_{ra} + E_{Cra}\dot{\Gamma}_{ra} + E_{Gra} +
$$

$$\left[\{W_{Mr_a}\}^T\cdot\{H_{Mr_a}(\Gamma_{ra})\}\right]\dot{\Gamma}_{ra} + \left[\{W_{Cr_a}\}^T\cdot\{H_{Cr_a}(\Gamma_{ra}, \dot{\Gamma}_{ra})\}\right]\dot{\Gamma}_{ra} + \left[\left[\left\{W_{Gr_a}\right\}^T\cdot\{H_{Gr_a}(\Gamma_{ra})\}\right]\right]$$

Then, the estimation of $M_{NN}(\Gamma_{ra})$, $C_{NN}$, and $G_{NN}$ will be included to confirm data and model forecasting for modeling errors. In terms of RBF, the estimation of mismatch between the calculated and nominal values is supposed as:

$$\dot{M}_{NN}(\Gamma_{ra}) = \left[\{\hat{W}_{Mr_a}\}^T \cdot \{E_{Mra}(\Gamma_{ra})\}\right],$$

$$\dot{C}_{NN}(\Gamma_{ra}) = \left[\{\hat{W}_{Cr_a}\}^T \cdot \{E_{Cra}(\Gamma_{ra}, \dot{\Gamma}_{ra})\}\right],$$

$$\dot{G}_{NN}(\Gamma_{ra}) = \left[\{\hat{W}_{Gr_a}\}^T \cdot \{E_{Gra}(\Gamma_{ra})\}\right],$$

where $\hat{W}_{Mr_a}, \hat{W}_{Cr_a},$ and $\hat{W}_{Gr_a}$ are estimates of $W_{Mr_a}, W_{Cr_a},$ and $W_{Gr_a},$ respectively. Consequently, the control algorithm of the estimated model is:

$$T_m = \dot{M}_{NN}(\Gamma_{ra})\ddot{\Gamma}_{ra} + \dot{C}_{NN}(\Gamma_{ra}, \dot{\Gamma}_{ra})\dot{\Gamma}_{ra} + G_{NN}(\Gamma_{ra}).$$

By following the same procedure for the design of a robust controller in Section 4, we obtain:

$$M_{ra}(\Gamma_{ra})\ddot{\Xi} + C_{ra}(\Gamma_{ra}, \dot{\Xi})\dot{\Xi} + K_i \int \Xi dt = -K_p\Xi - K_s\text{sgn}(\Xi) +
$$

$$\left[\{\hat{W}_{Mr_a}\}^T \cdot \{H_{Mr_a}(\Gamma_{ra})\}\right]\dot{\Xi} + \left[\{\hat{W}_{Cr_a}\}^T \cdot \{H_{Cr_a}(\Gamma_{ra}, \dot{\Xi})\}\right]\dot{\Xi} + \left[\left[\left\{\hat{W}_{Gr_a}\right\}^T \cdot \{H_{Gr_a}(\Gamma_{ra})\}\right]\right] + E_{Mra}\dot{\Xi} + E_{Cra}\dot{\Xi} + E_{Gra}$$

where:

$$\tilde{W}_{Mr_a} = W_{Mr_a} - \hat{W}_{Mr_a},$$

$$\tilde{W}_{Cr_a} = W_{Cr_a} - \hat{W}_{Cr_a},$$

$$\tilde{W}_{Gr_a} = W_{Gr_a} - \hat{W}_{Gr_a}.$$
where \( i = 1, 2 \); \( \ell_{M_{ra}}, \ell_{C_{ra}}, \ell_{C_{rai}} \) are symmetric matrices of positive and constant elements. To check the ability of our designed controller, explained in this section of operation for large modeling errors, a stability analysis is presented.

6. Stability Analysis

Checking the stability of the ARA manipulator is implemented in the following. Let us assume the integration-type Lyapunov formula [35], which is applied to establish the stability of the ARA:

\[
V = \frac{1}{2} \sum_{i=1}^{2} \left( \frac{1}{\alpha} \int_{0}^{1} \left( f_{t} \hat{x}_{t} \right)^{2} \right) K_{i} \left( \int_{0}^{1} \left( f_{t} \hat{x}_{t} \right)^{2} \right) + \frac{1}{2} \sum_{k=1}^{2} \dot{W}_{M_{rai}}^{T} f_{M_{rai}}^{-1} \dot{W}_{M_{rai}} + \frac{1}{2} \sum_{k=1}^{2} \dot{W}_{C_{rai}}^{T} f_{C_{rai}}^{-1} \dot{W}_{C_{rai}},
\]

(48)

and its derivative:

\[
\dot{V} = \sum_{i=1}^{2} \left[ \frac{1}{\alpha} \int_{0}^{1} \left( f_{t} \hat{x}_{t} \right)^{2} \right] \dot{K}_{i} \left( \int_{0}^{1} \left( f_{t} \hat{x}_{t} \right)^{2} \right) + \sum_{k=1}^{2} \dot{W}_{M_{rai}}^{T} f_{M_{rai}}^{-1} \dot{W}_{M_{rai}} + \sum_{k=1}^{2} \dot{W}_{C_{rai}}^{T} f_{C_{rai}}^{-1} \dot{W}_{C_{rai}},
\]

(49)

The dynamics of the two-rotating-joints ARA has the property of skew-symmetric—i.e., \( \sum (M_{ra} - 2C_{ra}) = 0 \).

Thus:

\[
\dot{V} = \sum_{i=1}^{2} \left[ \frac{1}{\alpha} \int_{0}^{1} \left( f_{t} \hat{x}_{t} \right)^{2} \right] \dot{K}_{i} \left( \int_{0}^{1} \left( f_{t} \hat{x}_{t} \right)^{2} \right) + \sum_{k=1}^{2} \dot{W}_{M_{rai}}^{T} f_{M_{rai}}^{-1} \dot{W}_{M_{rai}} + \sum_{k=1}^{2} \dot{W}_{C_{rai}}^{T} f_{C_{rai}}^{-1} \dot{W}_{C_{rai}}.
\]

(50)

We inserting the dynamics equation of the ARA into the above equation:

\[
\dot{V} = -\sum_{i=1}^{2} \dot{K}_{i} \left( \int_{0}^{1} \left( f_{t} \hat{x}_{t} \right)^{2} \right) \dot{W}_{M_{rai}} - \sum_{i=1}^{2} \left[ \frac{1}{\alpha} \int_{0}^{1} \left( f_{t} \hat{x}_{t} \right)^{2} \right] \dot{W}_{C_{rai}}. \frac{1}{\alpha} \int_{0}^{1} \left( f_{t} \hat{x}_{t} \right)^{2} \dot{K}_{i} \left( \int_{0}^{1} \left( f_{t} \hat{x}_{t} \right)^{2} \right) + \sum_{k=1}^{2} \dot{W}_{M_{rai}}^{T} f_{M_{rai}}^{-1} \dot{W}_{M_{rai}} + \sum_{k=1}^{2} \dot{W}_{C_{rai}}^{T} f_{C_{rai}}^{-1} \dot{W}_{C_{rai}}.
\]

(51)

The adaptive rule will be included in the derivative of Lyapunov stability by inserting adaptive law, mentioned in Equations (45)–(47) into the above Equation (51) to obtain the final representation of derivative of Lyapunov stability as:

\[
\dot{V} = -\sum_{i=1}^{2} \dot{K}_{i} \left( \int_{0}^{1} \left( f_{t} \hat{x}_{t} \right)^{2} \right) \dot{W}_{M_{rai}} - \sum_{i=1}^{2} \left[ \frac{1}{\alpha} \int_{0}^{1} \left( f_{t} \hat{x}_{t} \right)^{2} \right] \dot{W}_{C_{rai}}. \frac{1}{\alpha} \int_{0}^{1} \left( f_{t} \hat{x}_{t} \right)^{2} \dot{K}_{i} \left( \int_{0}^{1} \left( f_{t} \hat{x}_{t} \right)^{2} \right) + \sum_{k=1}^{2} \dot{W}_{M_{rai}}^{T} f_{M_{rai}}^{-1} \dot{W}_{M_{rai}} + \sum_{k=1}^{2} \dot{W}_{C_{rai}}^{T} f_{C_{rai}}^{-1} \dot{W}_{C_{rai}} - \sum_{k=1}^{2} \dot{E}_{M_{rai}} \hat{r}_{rai} \dot{r}_{rai} + E_{C_{rai}} \hat{r}_{rai} \dot{r}_{rai} + E_{C_{rai}} \dot{r}_{rai} \leq -\sum_{i=1}^{2} \dot{K}_{i} \left( \int_{0}^{1} \left( f_{t} \hat{x}_{t} \right)^{2} \right) \dot{W}_{M_{rai}} - \sum_{i=1}^{2} \left[ \frac{1}{\alpha} \int_{0}^{1} \left( f_{t} \hat{x}_{t} \right)^{2} \right] \dot{W}_{C_{rai}}. \frac{1}{\alpha} \int_{0}^{1} \left( f_{t} \hat{x}_{t} \right)^{2} \dot{K}_{i} \left( \int_{0}^{1} \left( f_{t} \hat{x}_{t} \right)^{2} \right) + \sum_{k=1}^{2} \dot{W}_{M_{rai}}^{T} f_{M_{rai}}^{-1} \dot{W}_{M_{rai}} + \sum_{k=1}^{2} \dot{W}_{C_{rai}}^{T} f_{C_{rai}}^{-1} \dot{W}_{C_{rai}} - \sum_{k=1}^{2} \dot{E}_{M_{rai}} \hat{r}_{rai} \dot{r}_{rai} + E_{C_{rai}} \hat{r}_{rai} \dot{r}_{rai} + E_{C_{rai}} \dot{r}_{rai} \leq 0.
\]

(52)
Now, let us assume the minimum eigenvalues of $K_p$ is $\lambda_{\min}$, then the proportional gain $K_p$ is restricted with the following range $\lambda_{\min}(K_p) \int_0^t \Psi^T \Psi \leq \int_0^t \Psi^T K_p \Psi \leq V(0)$ and the following properties:

- Property 1:
  \[ V(0) > 0, \lambda_{\min}(K_p) > 0, \text{ and } e \in L^2_{\infty}. \] Hence, $e \in L^2_{\frac{1}{2}} \cap L^2_{\alpha \nu} \to 0$ as $t \to \infty$, and $\dot{e} \in L^2_{\alpha \nu}$.

- Property 2:
  \[ \dot{V} \leq -\Psi^T K_p \Psi \leq 0 \text{ and } 0 \leq V \leq V(0), \forall t \geq 0. \] Hence, $V(t) \in L_{\infty}$ presents that $\int_0^t \Psi^T dT_{\text{in}}, \tilde{W}_{\text{M}}, \tilde{W}_{G}, \tilde{W}_{C}$ are limited. That is $\tilde{W}_{\text{M}}, \tilde{W}_{G}, \tilde{W}_{C}$ are limited.

- Property 3:
  \[ e \in L^2_{\frac{1}{2}} \cap L^2_{\alpha \nu}, \dot{e} \in L^2_{\alpha \nu}, \text{ and } \Gamma_{\text{rad}}, \Gamma_{\text{rad}}^T \in L^2_{\infty}. \] Hence, presents that $\Gamma_{\text{rad}}, \Gamma_{\text{rad}}^T \in L^2_{\infty}$ and $T_{\text{in}} \in L^2_{\infty}$.

- Property 4:
  Since $\Psi \in L^2_{\frac{1}{2}}$ and $\dot{\Psi} \in L^2_{\infty}$, then, $\Psi \to 0$ when $t \to \infty$. So, $\dot{e} \to 0$ as $t \to \infty$.

This satisfies the fact that the ARA system will be stable for high modeling errors.

7. Simulation Results

To prove the validity of the derived model and the suggested controller, a simulation process is implemented. The objective of the simulation is to emphasize the advantages of the developed controller via two stages. In the first stage, the nominal model is assumed. The second stage takes into consideration the high modeling errors. The physical values of the ARA, which are settled in the tests, are presented in Table 2.

Table 2. The suggested physical parameters of ARA.

<table>
<thead>
<tr>
<th>Name</th>
<th>Symbol</th>
<th>Value (Unit)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length of link 1</td>
<td>$L_1$</td>
<td>1 (m)</td>
</tr>
<tr>
<td>Length of link 2</td>
<td>$L_2$</td>
<td>1 (m)</td>
</tr>
<tr>
<td>Width of link 2</td>
<td>$w_1$</td>
<td>0.2 (m)</td>
</tr>
<tr>
<td>Width of link 2</td>
<td>$w_2$</td>
<td>0.2 (m)</td>
</tr>
<tr>
<td>Height of link 2</td>
<td>$h_1$</td>
<td>0.3 (m)</td>
</tr>
<tr>
<td>Height of link 2</td>
<td>$h_2$</td>
<td>0.3 (m)</td>
</tr>
<tr>
<td>Mass of link 1</td>
<td>$m_1$</td>
<td>1 (Kg)</td>
</tr>
<tr>
<td>Mass of link 2</td>
<td>$m_2$</td>
<td>1 (Kg)</td>
</tr>
<tr>
<td>Gravitational acceleration</td>
<td>$g$</td>
<td>9.81 (m/s$^2$)</td>
</tr>
</tbody>
</table>

Assuming the links of type rigid homogeneous rectangular bar of length $L$ along the $x$-axis, width $w$ along the $y$-axis, and height $h$ along the $z$-axis [36], the inertial matrix is obtained as:

\[
I_i = \begin{bmatrix}
\frac{m}{12} (w^2 + h^2) & 0 & 0 \\
0 & \frac{m}{12} (l^2 + h^2) & 0 \\
0 & 0 & \frac{m}{12} (l^2 + w^2)
\end{bmatrix}.
\] (53)

Hence, for each of link 1 and link 2, the inertia values are obtained as:

\[
I_{xx} = \frac{m}{12} (w^2 + h^2) = 0.01 \text{ Kg.m}^2, \\
I_{yy} = \frac{m}{12} (l^2 + h^2) = 0.09 \text{ Kg.m}^2, \\
I_{zz} = \frac{m}{12} (l^2 + w^2) = 0.08 \text{ Kg.m}^2.
\] (54)
7.1. Nominal Model

Consider the dynamic equation of the ARA explained in Equation (25) of disturbance torque equal to $T_{\text{ext}} = 15\text{sgn}(\hat{T}_{\text{ra}})$ and assume the initial angle and rate of angle states of the ARA are $\hat{\theta}_{\text{ra initial}} = [0.1, -0.11]^T$ and $\dot{\theta}_{\text{ra initial}} = [0, 0]^T$, respectively. On the other hand, it is assumed the uncertainty of the calculated dynamic model is low. Hence, the calculated mathematical dynamic model system is 0.9 of the nominal model. The required trajectory of the ARA joint 1 and joint 2 is set to $\alpha_{1,d} = 0.2\sin(\pi t)$ and $\alpha_{2,d} = 0.3\sin(\pi t)$, respectively. Applying the derived controller of Section 4 in equations (32, 33, 34). In this controller, the gains and parameters are set as: $K_p = \text{diag}[90, 90]$, $K_i = \text{diag}[90, 90]$, $K_\mu = \text{diag}[20, 20]$, $\mu = \text{diag}[3, 3]$. The tracking of the position and speed of ARA joint 1 and joint 2 are presented in Figures 5 and 6, respectively.

![Figure 5. States of the ARA joint 1.](image1)

![Figure 6. States of the ARA joint 2.](image2)

Consequently, it is shown that the controller is robust under low uncertainties. In which, ARA joint 1 and joint 2 have followed the required trajectories in definitive reali-
tion. Comparing the simulation results for the tracking trajectory in Figures 5 and 6, it is obvious that ARA joint 2 is more stable and robust than ARA joint 1.

7.2. Non-Nominal Model

Assume that the ARA initial angle and rate of angle states of the ARA are \( \Gamma_{\text{initial}} = \begin{bmatrix} 0.1 & -0.11 \end{bmatrix}^T \) and \( \dot{\Gamma}_{\text{initial}} = \begin{bmatrix} 0 & 0 \end{bmatrix}^T \), respectively. The payload of the ARA is assumed unknown and exerts a disturbance \( T_{\text{ext}} = 25 \text{sgn}(\dot{\Gamma}_r) \). The required trajectory of the ARA joint 1 and joint 2 is set to \( \alpha_{1,d} = 0.2 \sin(\pi t) \) \( \alpha_{2,d} = 0.3 \sin(\pi t) \), respectively. Applying the derived controller of Section 5. In this controller, the gains and parameters are set as: \( K_p = \text{diag}[90,90], K_i = \text{diag}[90,90], K_\Sigma = \text{diag}[20,20], \mu = \text{diag}[3,3] \). Regarding the adaptive law, the gains are assumed \( l_{M_{\text{ra}}} = 6, l_{C_{\text{ra}}} = 13, l_{G_{\text{ra}}} = 7 \). The structure of the RBF neural network is: the number of inputs equals 2, the number of hidden layers is 4, and the number of outputs is 1. The Gaussian function is set with the following parameters: \( c_1 = \begin{bmatrix} -1.3 & -1.2 & -0.6 & 0 & 0.6 & 1.1 & 1.6 \end{bmatrix}^T \), \( b_1 = [14] \). All the initial weights are set to zero. The uncertainty in the dynamic model of ARA is assumed to be high with the following values:

\[
M_{\text{ra}}(\Gamma_{ra}) = \begin{bmatrix} 4.8 \cos(\alpha_2) & 1.9 \cos(\alpha_2) \\ 1.8 \cos(\alpha_2) & 0.4 \end{bmatrix}, \\
C_{\text{ra}} = \begin{bmatrix} -1.3 \alpha_2 \sin(\alpha_2) & -1.3 \alpha_1 \sin(\alpha_2) \\ -1.3 \alpha_1 \sin(\alpha_3) & 0 \end{bmatrix}, \\
G_{\text{ra}} = \begin{bmatrix} 39 \cos(\alpha_1) + 42 \cos(\alpha_1 + \alpha_2) \\ 42 \cos(\alpha_1 + \alpha_2) \end{bmatrix}.
\]

Figures 7 and 8, explain the performance of the tracking trajectory of ARA joint 1 and joint 2, respectively. It is shown in these two figures, i.e., Figures 6 and 7, respectively, errors of tracking in ARA joint 1 and joint 2 appear at the starting of the path before the time 2 s. Then, after time 2 s, both the required and actual trajectories are roughly overlapped. Hence, the adaptive RBF neural network control technique provides perfect tracking performances in situations where the uncertainty of the ARA dynamic is high and the payload is unknown. The required supplied torques at each ARA joint by the controller is presented in Figure 9. The required trajectory, as it existed in practical ARA manipulators, is not insistently sensational, as shown in Figure 7 and 8, respectively. Consequently, The change in the estimated terms of the dynamic system \( \dot{M}_{\text{ra}}(\Gamma_{ra}), \dot{M}_{\text{ra}}(\Gamma_{ra}, \dot{\Gamma}_{ra}) \), and \( \dot{G}_{\text{NN}}(\Gamma_{ra}) \) presented in Figure 10 do not meet \( M_{\text{ra}}(\Gamma_{ra}), C_{\text{ra}}(\Gamma_{ra}, \dot{\Gamma}_{ra}) \), and \( G_{\text{ra}}(\Gamma_{ra}) \), respectively.

![Figure 7. Position of ARA joint 1 and joint 2.](image-url)
high and the payload is unknown. The required supplied torques at each ARA joint by the controller is presented in Figure 9. The required trajectory, as it existed in practical ARA manipulators, is not insistently sensational, as shown in Figure 7 and 8, respectively. Consequently, the change in the estimated terms of the dynamic system $M_{\theta}(\Gamma_{\theta})$, $M_{\theta}(\Gamma_{\theta},\Gamma_{\theta'})$, and $G_{\theta}(\Gamma_{\theta})$ presented in Figure 10 do not meet $M_{\theta}(\Gamma_{\theta})$, $C_{\theta}(\Gamma_{\theta},\Gamma_{\theta'})$, and $G_{\theta}(\Gamma_{\theta})$, respectively.

Figure 7. Position of ARA joint 1 and joint 2.

Figure 8. Speed of ARA joint 1 and joint 2.

Figure 9. The control inputs of ARA joints.

Figure 10. Estimation of dynamic system terms.

8. Conclusions

This work presented a new controller design and simulation of an Aerial Robot Arm (ARA) that is capable of high-performing trajectory tracking under variable payload and environmental conditions. The new control algorithm takes into account high nonlinearities, coupling control loops, high modeling errors, and disturbances due to payloads and environmental conditions. The work was presented in the following steps: Firstly, a step-by-step dynamics system modeling technique for the ARA manipulator, called the Lagrange–d'Alembert principle, was introduced in detail. Then, the sliding mode algorithm was applied as a robust nominal controller. An adaptive RBF neural network was developed and applied for the robust nominal controller to handle both the high modeling errors and disturbances. Additionally, the sliding mode technique was applied to the controller, obtaining robustness to realize the required tracking execution while the RBF neural network was implemented for border-raised uncertainty. The proposed model was evaluated by two simulations. The first simulation evaluated the robustness in terms of examining the tracking trajectory of the ARA. The second simulation demonstrated the ability of the final control algorithm to stabilize and track the execution of the ARA in the presence of a high number of modeling errors and disturbances.
8. Conclusions

This work presented a new controller design and simulation of an Aerial Robot Arm (ARA) that is capable of high-performing trajectory tracking under variable payload and environmental conditions. The new control algorithm takes into account high non-linearities, coupling control loops, high modeling errors, and disturbances due to payloads and environmental conditions. The work was presented in the following steps: Firstly, a step-by-step dynamics system modeling technique for the ARA manipulator, called the Lagrange–d’Alembert principle, was introduced in detail. Then, the sliding mode algorithm was applied as a robust nominal controller. An adaptive RBF neural network was developed and applied for the robust nominal controller to handle both the high modeling errors and disturbances. Additionally, the sliding mode technique was applied to the controller, obtaining robustness to realize the required tracking execution while the RBF neural network was implemented for border-raised uncertainty. The proposed model was evaluated by two simulations. The first simulation evaluated the robustness in terms of examining the tracking trajectory of the ARA. The second simulation demonstrated the ability of the final control algorithm to stabilize and track the execution of the ARA in the presence of a high number of modeling errors and disturbances adaptively and robustly. The simulation results showed the validation and notability of the presented control algorithm. The main limitation of the proposed approach is that it does not take into account physical parameters such as the shape and texture of the object that is manipulated or moved. This can affect the overall simulation results. This issue is planned to be addressed in the future by using image recognition and depth sensors that will generate a representation of the 3D structure of the objects to be manipulated or moved. Another limitation is that the number of hidden layers and parameters of the Gaussian function could potentially affect the tracking of errors. In the future, we plan to apply an optimization technique to find the optimum RBF neural structure that will minimize the tracking errors. Additionally, alternative approaches such as deep learning networks could be applied instead of a neural network.

Author Contributions: Conceptualization, A.A.K.; Data curation, D.P.; Methodology, I.A.-D., D.P., A.A.K., and M.S.; Software, I.A.-D., D.P., F.Q.K., M.S., and G.T.; Supervision, D.P., and G.T.; Writing—original draft, I.A.-D., A.A.K., F.Q.K., M.S., G.T., and P.G.P.; Writing—review & editing, I.A.-D., D.P., A.A.K., F.Q.K., G.T., and P.G.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

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

Nomenclature

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>O_ra</td>
<td>The reference frame of ARA</td>
</tr>
<tr>
<td>O_Earth</td>
<td>The reference frame of earth</td>
</tr>
<tr>
<td>T_ra</td>
<td>Vector of ARA joint angles</td>
</tr>
<tr>
<td>σ_i</td>
<td>The angle of ARA joints</td>
</tr>
<tr>
<td>ϕ</td>
<td>Roll Euler angle</td>
</tr>
<tr>
<td>θ</td>
<td>Pitch Euler angle</td>
</tr>
<tr>
<td>ψ</td>
<td>Yaw Euler angle</td>
</tr>
<tr>
<td>J_v,i</td>
<td>Jacobian matrices of the linear velocities</td>
</tr>
<tr>
<td>J_w,i</td>
<td>Jacobian matrices of the angular velocities</td>
</tr>
<tr>
<td>K</td>
<td>Kinetic energies of the ARA</td>
</tr>
<tr>
<td>U</td>
<td>Potential energies of the ARA</td>
</tr>
<tr>
<td>T_in</td>
<td>Input torques of the ARA joints</td>
</tr>
<tr>
<td>T_ext</td>
<td>torques of the disturbances</td>
</tr>
<tr>
<td>V</td>
<td>vector of interaction force and moments</td>
</tr>
</tbody>
</table>
F_{int, x}, F_{int, y}, and F_{int, z} \quad \text{the interaction force in the x, y, and z-direction, respectively}

M_{1}, M_{2}, and M_{3} \quad \text{the moment which is resulted by the held object on the x, y, and z-axis, respectively}

M_{ra} (\Gamma_{ra}) \quad \text{Centrifugal force}

C_{ra} (\Gamma_{ra}, \Gamma_{ra}) \quad \text{Centrifugal force}

G_{ra} (\Gamma_{ra}) \quad \text{Gravitational force}

E_{Mra} \quad \text{modeling errors of } M_{ra} (\Gamma_{ra})

E_{Cra} \quad \text{modeling errors of } C_{ra} (\Gamma_{ra}, \Gamma_{ra})

E_{Gra} \quad \text{modeling errors of } G_{ra} (\Gamma_{ra})

e(t) \quad \text{error in tracking}

T_{\Sigma} \quad \text{nominal torque part}

\Sigma \quad \text{robust torque part}

References


2. Liu, Y.; Gao, J.; Zhao, J.; Shi, X. A New Disaster Information Sensing Mode: Using Multi-Robot System with Air Dispersal Mode. Sensors 2018, 18, 3589. [CrossRef]


