

Initial analysis and development of an automated maintenance system for Agrivoltaics plants

Farima Hajiahmadi^{1,2}, Mohammad Dehghani^{3*}, Payam Zarafshan^{4,5}, S. Ali A. Moosavian⁶, S. Reza Hassan-Beygi¹

(1. Department of Agro-Technology, College of Aburaihan, University of Tehran, Tehran, 3391653755, Iran;

2. Department of Earth and Space Sciences, Columbus State University, Columbus, GA 31907, USA;

3. Department of Food Technology, College of Aburaihan, University of Tehran, Tehran, 3391653755, Iran;

4. Department of Mechatronics Engineering, Faculty of New Sciences and Technologies, Tehran, 1439956191, Iran;

5. School of Engineering, Design and Built Environment, Western Sydney University, NSW 2751 Australia;

6. Department of Mechanical Engineering, K. N. Toosi University of Technology, Tehran, 1969764499, Iran)

Abstract: Agrivoltaics power generation is the simultaneous use of agricultural land and photovoltaic panels. In such cases, the panels are placed more sparsely, compared to conventional photovoltaic plants. Furthermore, the panels might be mounted at higher heights, to provide the required space for the farm. Due to this arrangement of the panels in wide areas with crops on the farm, maintenance of the panels is a hard task. The major part of maintenance is to clean the panels, since the accumulation of farm dust and bird excrement can reduce the efficiency of the panels. In Agrivoltaics plants, the panels can be scattered, thus it is not economical for each panel to have a dedicated cleaning robot. This paper presents the initial design and analysis of an automated system for cleaning such power plants. This system consists of a robotic AGV, which transfers a panel cleaner between the photovoltaic panels mounted over the farms. The initial design is presented. Then, to anticipate the required power and control system, the carrier kinetics and kinematics are modeled and validated through simulations. Then, position and trajectory controllers are designed and simulations are presented. The simulations show desired performance for the introduced controlled system.

Keywords: Agrivoltaics power plants; automated cleaning; AGV robot; simulation.

Citation: Hajiahmadi F., M. Dehghani, P. Zarafshan, S. Ali A. Moosavian, S. R. Hassan-Beygi. Initial analysis and development of an automated maintenance system for Agrivoltaics plants. *Agricultural Engineering International: CIGR Journal*, 25(3): 130-145.

1 Introduction

Solar power is one of the most important energy sources in the world. One major problem that reduces

the energy efficiency of solar panels is the dust accumulation on the panels. This problem gets worse where the solar power plant is placed near agricultural terrains, mining sites, and other polluted areas. For instance, the agricultural dust can be blown to nearby areas and form a thick layer of dust which covers the surfaces of panels. Studies have shown that up to 15% of energy loss is due to the mentioned problems

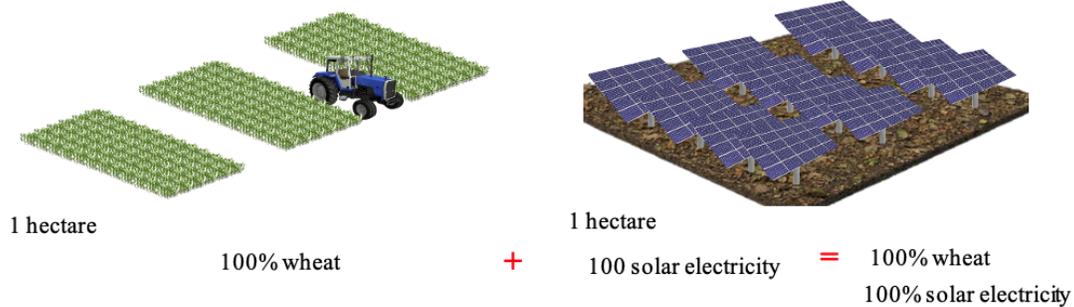
Received date: 2022-03-03 **Accepted date:** 2023-06-28

***Corresponding author: Mohammad Dehghani**, Department of Food Technology, College of Aburaihan, University of Tehran, Tehran, 3391653755, Iran. Email: udehghani@ut.ac.ir.

(Mondal and Bansal, 2015). This problem is specifically important for Agrivoltaics, or AgroPhotovoltaic (APV) systems. In APV's, the solar panels can be placed between the plants, over the plants in fields, or over greenhouses. These systems can be used where land is scarce and very valuable; or where the plants need to have a shelter or a partial shadow against the direct sunlight. Figure 1 depicts an APV with solar panels mounted over the crops. If one

hectare of land is only used for crop production, 100% of a typical land crop revenue is achieved. Likewise, if one hectare is only used for solar power production, 100% of a typical power plant revenue is achieved. Using an agrivoltaics system, the efficiency of crops and electricity production reduces to almost 80%. However, since both are produced on the same land, the total production revenue goes up to 160%.

Separate Land Use on 2 Hectare Cropland



Combined Land Use on 2 Hectare Cropland: Efficiency increases over 60%

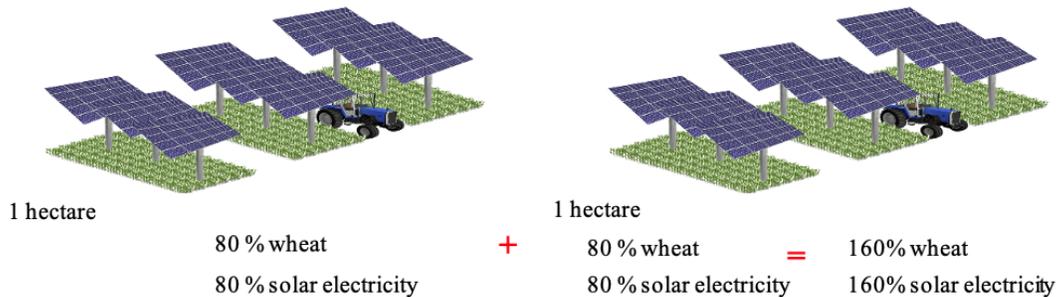


Figure 1 Simultaneous use of land for agriculture and for electricity production

Maintenance of solar panels can lead to electric shocks and severe injuries for workers. In manual cleaning, the quality of the cleaning is low, and the cost and risks are high. Heavy cleaning machineries can also be challenging, mostly because of their large size and inflexibility. Such machines are sometimes equipped with large, low-flexible arms to access high-altitude photovoltaic panels, which can be dangerous for the solar panels and for the laborers. Using robots for panel cleaning is a novel way for the maintenance of such power plants. Table. 1 introduces different ways to clean solar panels. Each method has its pros and cons. The use of automated systems for this task is rising, as automated systems are widely used for many

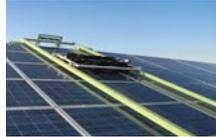
agricultural tasks, such as irrigation (Ebrahimi et al., 2018), pollination (Mazinani et al., 2021b, Mazinani et al., 2021a), surveillance (Haddadi et al., 2022), etc.

As shown in the table, the robots can be constrained in a frame. While the frame can have lateral motion, the robot can move up and down, inside the frame. Some robots can move freely on the panels in any required direction. Furthermore, manipulators can be used, which are mounted on vehicles. Such manipulators can be controlled automatically or manually. This method has the risk of applying too much force on the panels. Furthermore, it requires wide roads among the rows (Syafiq et al., 2018, Kumar et al., 2018). For agrivoltaic systems, where the land is used

for crops and the panels are elevated, this method is not useful. Using robots as automated systems for clearing panels in solar plants consisting of tens or hundreds of rows has been studied in various references. In these methods, each row of panels has its dedicated cleaning robot (Tadayon, 2015, Deb and Brahmabhatt, 2018,

Moshe Saraf, 2015). By designing and simulating several suitable controllers, the cleaning performance of several systems has been studied (Jaradat et al., 2015, Tranca et al., 2017, Cheah et al., 2003, Dubowsky and Papadopoulos, 1993).

Table 1 Different methods for cleaning solar solar panels (Khadka, N. et al., 23 July 2020)

Constrained Robot			
<p>Weaknesses</p> <ul style="list-style-type: none"> Install the frame on the solar panels Robot cannot move from one row to another row alone Requires an operator to move and install the structure High cost 	<p>Strengths</p> <ul style="list-style-type: none"> Cleaning without water Cleaning without damaging agricultural products. High accuracy 		<p>(Benjaminsen, 2021)</p>
			<p>(Ecoppia, 2018)</p>
Lateral Robot			
<p>Weaknesses</p> <ul style="list-style-type: none"> Robot cannot move from one row to another row alone Requires an operator to move and install the structure High cost 	<p>Strengths</p> <ul style="list-style-type: none"> Cleaning without water Cleaning without damaging agricultural products High accuracy 		<p>(Miraikikai, 2019)</p>
			<p>(Boson, 2019)</p>
Mobile Base Robot			
<p>Weaknesses</p> <ul style="list-style-type: none"> Robot cannot move from one row to another row alone Requires an operator to move and install the structure High cost 	<p>Strengths</p> <ul style="list-style-type: none"> Cleaning without water Cleaning without damaging agricultural products High accuracy 		<p>(HyCleaner, 2020)</p>
			<p>(Serbot, 2020)</p>
Manipulators			
<p>Weaknesses</p> <ul style="list-style-type: none"> Damaging to agricultural products Cleaning with water Requires an operator Low accuracy Using fossil fuel for cleaning solar panels It needs an operator to move and clean solar panels 	<p>Strengths</p>		
			<p>(Burgaleta et al., 2012)</p>

As mentioned, in order to overcome dust accumulation on the plates, it is necessary to clean them at regular times. The rate of dust accumulation depends on the weather, the type of dust and soil in the area, the presence of birds, and the type of agricultural activities and crops. Due to these factors, an agrivoltaics system requires to be cleaned more frequently and a typical PV system. In some photovoltaic systems, robots are used for cleaning the panels. In such cases, each row might have a dedicated

cleaning robot, mounted on it. However, in an agrivoltaics, the number of panel rows might be high, and the rows might be short. Hence, it is not economical to dedicate one cleaning robot for each row. As depicted in Figure 2, one economical solution is to transfer one cleaning robot between panel rows. To this end, a robotic carrier is required which can take the cleaning robot from one row, move along the field, transfer the cleaning robot and give it to the next row for cleaning.

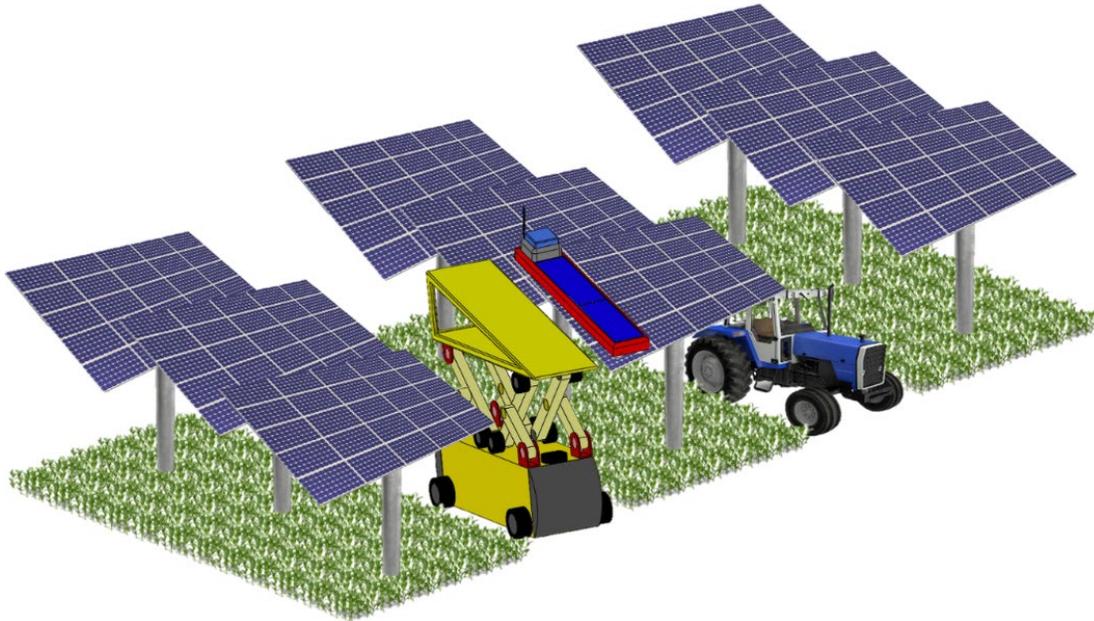


Figure 2 The introduced robotic solution for an agrivoltaic plant

Considering the stated advantages and disadvantages of the previous methods, in this article, a compound cleaning system is introduced which consists of two parts. One part is the cleaning robot, and the other is the carrier robot, as depicted in Figure 3. This robotic solution would automatically perform cleaning of the whole power plant. In addition, all panels can be cleaned by one cleaning robot, which reduces the total cost of the cleaning system. Therefore, the capabilities of these solutions can be stated as follows:

- 1- The carrier robot can adjust to panels at different heights.
- 2- The carrier robot can adjust to panels mounted at

different angles.

- 3- This robotic solution can work fully automated.
- 4- The carrier robot is small and moves in the designed passage of the farm.

In this study, a robotic carrier system is proposed as a part of an automated solution for agrivoltaics panel cleaning, as depicted in Figure 2 and Figure 3. As the initial study of this system, it is modeled, controlled, and simulated. First, the kinematics and dynamics of this system are modeled in section 2. Then, in section 3, the obtained mathematical model is verified using ADAMS software. In section 4, the robotic carrier is controlled, to achieve the required position, height, and angle of the panels. In order to achieve smooth and

accurate motion, without sudden accelerations and oscillations, different dynamic model-based and kinematics model-based controllers are designed and compared. According to the simulations, the robot can

move successfully from one row to another. Therefore, exploiting this solution, all panels can be cleaned by only one cleaning robot without damaging agricultural products.

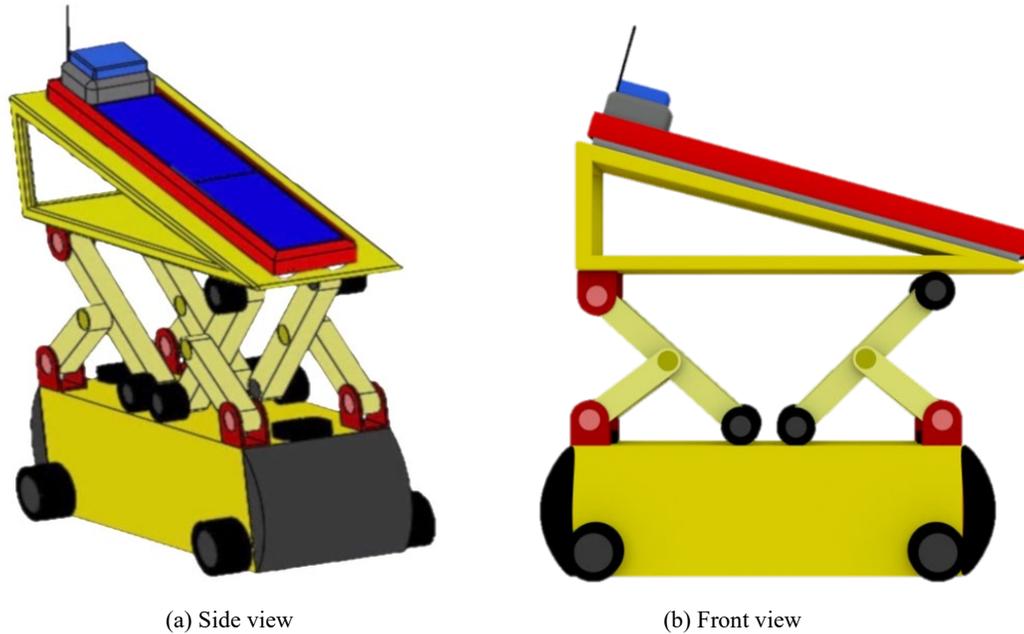


Figure 3 The proposed design of the robotic carrier system,;

2 Materials and method

2.1 Dynamic modelling

The kinematics characteristics of the carrier robot are introduced in Figure 4. The robot can move forward/backward and steer right or left; it can also adjust the upper table height and angle, using the two scissor mechanisms, as shown. For these movements, the general coordinates can be considered as Equation (1).

$$q = [x \ \theta_1 \ \theta_2 \ z \ \varphi]^T \tag{1}$$

where φ is the robot angle in the horizontal xz plane. Regarding the carrier goal, the workspace coordinates of the upper table (the end-effector) can be considered as $(x_{ee}, y_{ee}, z_{ee}, \theta_{ee}, \varphi)$. These coordinates must be controlled for the cleaning robot to be positioned beside the panel rows. Hence, the upper table is considered as the End Effector of the carrier robot. The parts and dimensions of the carrier robot are shown in Figure 4. As shown, a is the length between

the bases of the front and rear scissors.

2.2 System kinematics

Using the kinematics shown in Figure 4, the position of the end effector can be derived. For abbreviation, $\sin \theta_i$ and $\cos \theta_i$ are represented as S_i and C_i . so the end-effector position can yield as Equation (2).

$$\mathbf{x}_{ee} = \begin{bmatrix} x + (l_{ee} C_\varphi) \sqrt{2((2l_1 S_1 - 2l_2 S_2)^2 / a^2 + 1)} \\ 2l_2 S_2 + (l_{ee} (2l_1 S_1 - 2l_2 S_2)) / \sqrt{2a((2l_1 S_1 - 2l_2 S_2)^2 / a^2 + 1)} \\ z - (l_{ee} S_\varphi) / \sqrt{2((2l_1 S_1 - 2l_2 S_2)^2 / a^2 + 1)} \end{bmatrix} \tag{2}$$

where, l_{ee} is the table length; a is the length of the robot (distance between motor 2 and motor 3 axes); l_1 and l_2 are lengths of links 1 and 2 respectively; θ_1 and θ_2 are respectively the angle of the front and rear scissors. Using the derivatives of Equation 1 with respect to the general coordinates $(x, \theta_1, \theta_2, z, \varphi)$, the Jacobian matrices for x, y, z position of links are derived as follows. So, for the End Effector, the Jacobian matrix is:

$$\mathbf{J}_{cc} = \begin{bmatrix} 1 & -(l_{1cc} C_\phi C_1 (2l_1 S_1 - 2l_2 S_2)) / \sqrt{(a^2 ((2l_1 S_1 - 2l_2 S_2)^2 / (a^2 + 1))^2} \\ 0 & (l_{1cc} C_1) / \sqrt{(a^2 ((2l_1 S_1 - 2l_2 S_2)^2 / (a^2 + 1)) - (l_{1cc} C_1 (2l_1 S_1 - 2l_2 S_2)^2) / \sqrt{(a^2 ((2l_1 S_1 - 2l_2 S_2)^2 / (a^2 + 1))^2} \\ 0 & (l_{1cc} S_1 C_\phi (2l_1 S_1 - 2l_2 S_2)) / \sqrt{(a^2 ((2l_1 S_1 - 2l_2 S_2)^2 / (a^2 + 1))^2} \\ & (l_{1cc} S_1 C_\phi (2l_1 S_1 - 2l_2 S_2)) / (a^2 \sqrt{(2l_1 S_1 - 2l_2 S_2)^2 / a^2 + 1}) \\ 2l_2 C_2 - (l_2 l_{cc} C_2) / (a \sqrt{(2l_1 S_1 - 2l_2 S_2)^2 / (a^2 + 1)}) + (l_2 l_{cc} C_2 (2l_1 S_1 - 2l_2 S_2)^2) / (a^3 \sqrt{(2l_1 S_1 - 2l_2 S_2)^2 / a^2 + 1}) \\ & - (l_2 l_{cc} C_2 S_\phi (2l_1 S_1 - 2l_2 S_2)) / (a^2 \sqrt{(2l_1 S_1 - 2l_2 S_2)^2 / a^2 + 1}) \\ & \begin{bmatrix} 0 & -(l_{cc} S_1) / 2 \sqrt{(2l_1 S_1 - 2l_2 S_2)^2 / a^2 + 1} \\ 0 & 0 \\ 1 & -(l_{cc} C_1) / 2 \sqrt{(2l_1 S_1 - 2l_2 S_2)^2 / a^2 + 1} \end{bmatrix} \end{bmatrix} \quad (3)$$

Similarly, the Jacobian matrices for orientations of the links is obtained where ω refers to angular speeds as:

$$\mathbf{J}_{\omega cc} = \begin{bmatrix} 0 & (2l_1 c_1 s_\phi) / (a((2l_1 s_1 - 2l_2 s_2)^2 / a^2 + 1)) & -(2l_2 c_2 s_\phi) / (a((2l_1 s_1 - 2l_2 s_2)^2 / a^2 + 1)) & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & (2l_1 c_\phi c_1) / (a((2l_1 s_1 - 2l_2 s_2)^2 / a^2 + 1)) & -(2l_2 c_\phi c_2) / (a((2l_1 s_1 - 2l_2 s_2)^2 / a^2 + 1)) & 0 & 0 \end{bmatrix} \quad (4)$$

Using Equations 2 and 3, Jacobian matrices are related to link speeds, which are introduced in (Hajiahmadi et al., 2019b; Hajiahmadi et al., 2019a),

the mass center velocity V_i and angular velocity ω_i of each link can be obtained as:

$$v_i = J_{v_i} \dot{q} \quad \omega_i = J_{\omega_i} \dot{q} \quad (5)$$

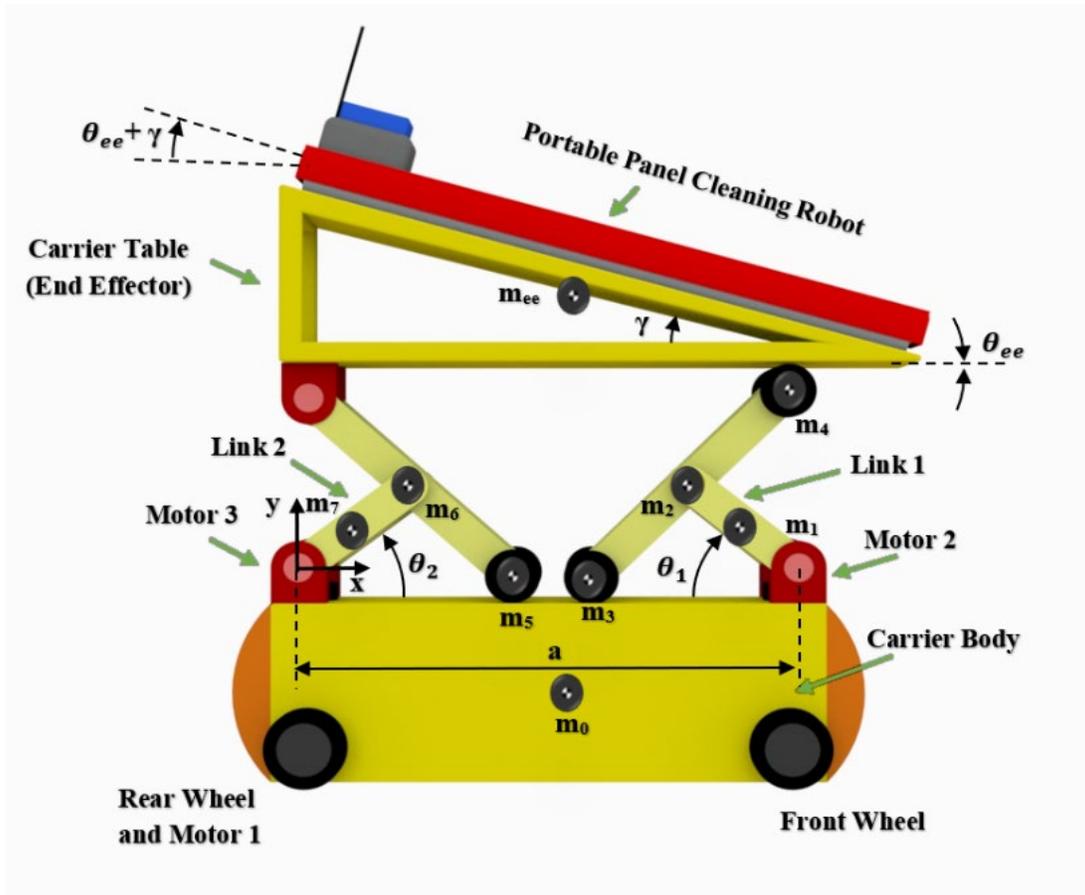


Figure 4 The designed carrier robot with its dynamic parameters

2.3 System dynamics

Based on Lagrange formulation, the dynamics model can be derived (John J. Craig, 2006; Mark W. Spong, 1st ed., 2019.). The kinetic energy can be represented in matrix form as:

$$K.E = \frac{1}{2} \dot{q}^T M \dot{q} = \frac{1}{2} \dot{q}^T \left[\sum_{i=1}^n (m_i J_{v_i}^T J_{v_i} + J_{\omega_i}^T I_{ci} J_{\omega_i}) \right] \dot{q} \quad (6)$$

Where, the general coordinates are chosen as $q = [x, \theta_1, \theta_2, z, \varphi]^T$. The positive definite inertia matrix \mathbf{M} is achieved in which, I_{ci} is the second moment of inertial and m_i the mass of the i^{th} link:

$$M = \sum_{i=1}^n (m_i J_{v_i}^T J_{v_i} + J_{\omega_i}^T I_{ci} J_{\omega_i}) \quad (7)$$

To consider the gravitational potential energies, the elevation of all mass centers must be considered. The generalized gravitational forces can be represented by the vector \mathbf{G} , as:

$$G = -(J_{v_1}^T (m_1 g) + J_{v_2}^T (m_2 g) + \dots + J_{v_n}^T (m_n g)) \quad (8)$$

Finally, the forces related to centrifugal and Coriolis effects are considered as \mathbf{V} vector as:

$$C_{kj} = \sum_{i=1}^n c_{ijk} (q) \dot{q}_i = \sum_{i=1}^n \frac{1}{2} \left\{ \frac{\partial M_{kj}}{\partial q_i} + \frac{\partial M_{ki}}{\partial q_j} + \frac{\partial M_{ij}}{\partial q_k} \right\} \quad (9)$$

$$V = c \dot{q}$$

Considering $\boldsymbol{\tau}$ as the vector of forces and torques related to the general coordinates, the closed form of the dynamics model of the robot is derived as:

$$M(q) \ddot{q} + V(q, \dot{q}) + G(q) = \boldsymbol{\tau} \quad (10)$$

2.4 Dynamic model verification

In this section, the derived dynamic model is verified using software simulation, using ADAMS (Azami et al., 2017, Hafezipour et al., 2013). First, the robot is modeled in SOLIDWORKS and then in ADAMS. For comparison, the ADAMS model is exported to MATLAB/Simulink. Then, for validation, robot motions are simulated. For a motion simulation, a predefined motion is given to the ADAMS model as the input. Then, the required motor forces/torques are

achieved as the simulation output. Finally, the results are compared with the model from Equation 9. In order to use the ADAMS simulator, first, the models of all robot links are built including mass, inertia, and the dimension of each link. Then, the constraints, forces, torques, and motions are defined and applied to the model. Finally, since our model calculations are performed in MATLAB, the ADAMS model is exported to MATLAB/Simulink.

A few simulations were run to make sure the mathematically derived model and the computer model give the same results. Here, the results of one simulation is depicted. For this simulation, the carrier is moved as considered in Equation 10.

$$\mathbf{q} = \left(3 \left(1 - \cos \frac{\pi t}{20} \right), \frac{\pi}{4} + 0.3 \sin \frac{4\pi t}{3}, \frac{\pi}{4} - 0.3 \sin \pi t, -3 \sin \frac{\pi t}{20}, \frac{\pi}{2} - \frac{\pi t}{20} \right) \quad (11)$$

The simulation time is chosen to be 20 seconds, which is enough to find out the possible simulation errors.

$$0 \leq t_{\text{simulation}} \leq 20 \quad (12)$$

The simulation also depends on initial velocities. Hence, the initial values of speeds are required for the ADAMS model. Here, Equation 10 is considered as the reference, and the initial rate \dot{q} is determined as

$$\dot{q}_0 = \left(0, 0.65\pi, -0.1\pi, -\frac{3\pi}{20}, -\frac{\pi}{20} \right) \quad (13)$$

The resulting forces, achieved from the ADAMS model and the mathematical model, are depicted and compared in Figure 5.

2.5 Robot control

As depicted in Figure 2, the carrier robot is designed to take the cleaning robot from one row, transfer it and give it to the next row. Along this path on the field, the road is not perfectly smooth. The cleaning robot is shipped over the carrier top table, along this path. Therefore, the carrier must move with a safe speed, without sudden accelerations. It is not safe

for this system to move while the table is erected; otherwise, it can lose its balance. Hence, the top table should retract down before the carrier moves. Then, the table needs to adjust its height and angle for the next row of panels. For adjustment of the carrier robot, actuators are required at its joints. Here, two actuators are required for the forward/backward and steering motion of the robot in the xz plane. Two actuators are required for the front and rear scissors, represented by θ_1 and θ_2 . These two actuators can adjust the table height and angle. As a part of the design process of the robotic system, it is necessary to study its control performance and the requirements for its actuation forces.

The carrier robot is not a linear system. For instance, the angles of scissors are not linearly related to the robot table height and angle. Actually, this kinematics is highly nonlinear. Different control methods can be considered for this nonlinear system. Here, model-based controllers are chosen for control analysis, using the model derived in the previous section. In general, a kinetics-model-based controller can provide the best response in dynamic systems; provided that the model is accurate and the computation costs are acceptable for the processor of the system. Regarding the computational costs, a simpler method can be a kinematics-model-controller. In this section, using the kinetics model, a computed torque method (CTM) controller will be developed. Likewise, using the kinematics model, transpose Jacobian (TJ) method will be used for controller design. In this section, to design and simulate the controllers, a designed path is considered for the carrier robot. Figure 6 depicts a schematic of a desired path on the field. The initial configuration is represented by $\mathbf{q}_0 = [x_0 \ \theta_{10} \ \theta_{20} \ z_0 \ \varphi_0]^T$. The desired trajectory is considered as a function of time. To move in circular path, x and z must follow harmonic functions; and, angle φ must grow linearly in time. For

a smooth motion of the table, scissor angles θ_1 and θ_2 are defined as quintic functions. Their coefficients are determined based on the initial and the final values. A desired trajectory for 20 seconds is represented in Equation 13, where t represents time in seconds.

$$\begin{aligned} q &= \left(3\left(1 - \cos\frac{\pi t}{20}\right), \theta_1, \theta_2, -3\sin\frac{\pi t}{20}, \frac{\pi}{2} - \frac{\pi t}{20} \right) \\ \theta_1 &= 0.349 + 0.0025t^3 - 0.0003t^4 \\ \theta_2 &= 0.349 + 0.0025t^3 - 0.0003t^4 \end{aligned} \quad (14)$$

The workspace coordinates are chosen as $X = [x_{ee}, y_{ee}, z_{ee}]^T$, where x_{ee} and y_{ee} and z_{ee} show the position of the table center respectively. The Jacobian matrices are defined in (2) and (3), and can be used as as Equation 14, to transfer velocities from the joint space q to the workspace X .

$$\dot{X} = J_{ee} \dot{q} \quad (15)$$

Here, the kinetics model is derived in the Lagrange form, as represented in Equation 9 in the space of generalized coordinates. In order to represent the dynamic model in the spatial space, X can be derived as a function of q . For instance, using Equation 14, $\dot{\mathbf{q}}$ can be replaced by the rate of X as:

$$\dot{q} = J_{ee}^{-1} \dot{X} \quad (16)$$

Therefore, using the Jacobian matrices, (9) can be represented in the workspace as

$$M_x \ddot{X} + V_x + G_x = F_x \quad (17)$$

Where F_x represents forces in the workspace; and

$$\begin{aligned} M_x(q) &= J_{ee}^{-T}(q)M(q)J_{ee}^{-T}(q) \\ V_x(q, \dot{q}) &= J_{ee}^T(q)(V(q, \dot{q}) - M(q)J_{ee}(q)\dot{q}) \\ G_x &= J_{ee}^{-T}(q)G(q) \end{aligned} \quad (18)$$

More details on modeling derivation of the matrices are presented in (Hajiahmadi et al., 2019b, Hajiahmadi et al., 2019a). As mentioned before, this model is validated using ADAMS and MATLAB co-simulation.

2.6 TJ and MTJ controllers

TJ control is a kinematic model-based control

approach (Moosavian and Papadopoulos, 1997, Moosavian and Papadopoulos, 2007). In this approach, the actuation forces and torques are determined using the Jacobian matrix, and the workspace errors. First, the workspace error, $e_x=X_{des}-X$, is used to determine the required workspace forces as:

$$F_x = K_v \dot{e}_x + k_p e_x \tag{19}$$

where, K_v is the derivative coefficient matrix; and K_p is the proportional coefficient matrix. Then, using the Jacobian matrix, the joint forces/torques are determined as:

$$\tau = J_{ee}^T F_x \tag{20}$$

According to the inertia of the robot and the external forces, the control coefficients of Equation 8 can be chosen, and adjusted by trial and error. Here, to compensate for the table weight, K_p the y-direction is chosen to be 30000 N m⁻¹. For the x-direction, the whole robot moves horizontally. Therefore, K_p in the x-direction is chosen to be 10000 N m⁻¹. Furthermore, for K_p respecting the table angle, a stiffness of 500 N m rad⁻¹ is chosen. The same amounts are considered for K_v . The coefficients are listed in Table 2, and the matrices are obtained as Equation 20. Finally, a saturation limit of 200 N is considered for the forces.

$$\begin{aligned} K_p &= \text{diag}(k_{px} + k_{py} + k_{p\theta}) \\ K_v &= \text{diag}(k_{vx} + k_{vy} + k_{v\theta}) \end{aligned} \tag{21}$$

Table 2 Coefficients of the TJ controller

$K_{px}=10000 \text{ N m}^{-1}$	$K_{py}=30000 \text{ N m}^{-1}$	$K_{pz}=500 \text{ N m}^{-1}$
$K_{vx}=10000 \text{ Ns m}^{-1}$	$K_{vy}=30000 \text{ Ns m}^{-1}$	$K_{vz}=500 \text{ Ns m}^{-1}$

As mentioned before, the controller is simulated using co-simulation of MATLAB/Simulink and ADAMS. The simulation results for trajectory tracking by the designed controller are depicted in Figure 7. In this simulation, an initial error is also introduced compared to the desired trajectory. For this initial condition, an error of -10 cm in x, 0.35 cm in y, and an

error of -0.2 cm in z are considered. The results show acceptable performance. The forces and errors are appropriate. However, the steady-state errors are not zero.

where, the amount of \dot{e}_{max} can be determined by try and error. The controller coefficients may need readjustment after h is introduced to the TJ equation. Here, the coefficients of x- and y- and z- directions are reduced, as in Table 3. Like the previous time, co-simulation of MATLAB/Simulink and ADAMS is used and an initial error is considered. Also, the same saturation limit is assumed for the forces. The simulation results are depicted in Figure 8. The diagrams show that the controller is successful in eliminating the steady-states errors.

Table 3 Coefficients of the MTJ controller

$k_{px}=1000 \text{ N m}^{-1}$	$k_{py}=20000 \text{ N m}^{-1}$	$k_{pz}=500 \text{ N m}^{-1}$
$k_{vx}=1000 \text{ Ns m}^{-1}$	$k_{vy}=20000 \text{ Ns m}^{-1}$	$k_{vz}=500 \text{ Ns m}^{-1}$

The steady-state errors achieved by the TJ controller cannot be eliminated as a fundamental characteristic of this control method. To eliminate these errors and achieve better responses, Modified TJ (MTJ) can be used. Using MTJ, the steady-state errors can be eliminated with minimum computational efforts (Karimi and Moosavian, 2010; Khalaji and Moosavian, 2015). For MTJ control, the control law (Equation 18) is modified by the introduction of a new term, $h(t)$, as:

$$F_x = k_v \dot{e}_x + k_p e_x + h(t) \tag{22}$$

where, h is a feedback linearization term. Calculations of such terms can require a thorough dynamics analysis. However, in the MTJ method, h is simply defined as:

$$h(t) = F_x \Big|_{t-\Delta t} \tag{23}$$

In Equation 22, the left term represents the value of force at a previous time interval, according to the chosen Δt . Furthermore, to avoid sudden disturbances, a correction factor k can be used, as:

$$\mathbf{h}(t) = k\mathbf{F}_x \Big|_{t-\Delta t} \tag{24}$$

When the errors are too large, it is better to use the TJ algorithm by considering $K=0$. Otherwise, when the errors are small enough, and the steady-state error is the main concern, the h factor can be used, having $k=1$. This algorithm can be presented as:

$$k = \begin{cases} 0, & e > e_{\max} \text{ or } \dot{e} > \dot{e}_{\max} \\ 1, & e < e_{\max} \text{ or } \dot{e} < \dot{e}_{\max} \end{cases} \tag{25}$$

Furthermore, to avoid sudden chattering, a smooth version of Equation 24 may be used, such as:

$$\mathbf{k} = \exp\left(-\left|\frac{\mathbf{e}}{\mathbf{e}_{\max}}\right| - \left|\frac{\dot{\mathbf{e}}}{\dot{\mathbf{e}}_{\max}}\right|\right) \tag{26}$$

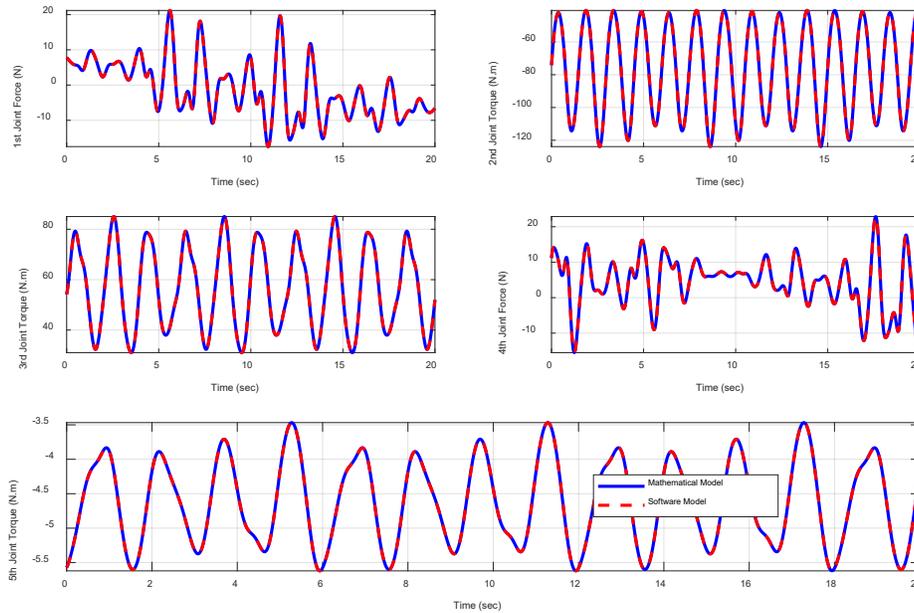


Figure 5 Simulation results for validation

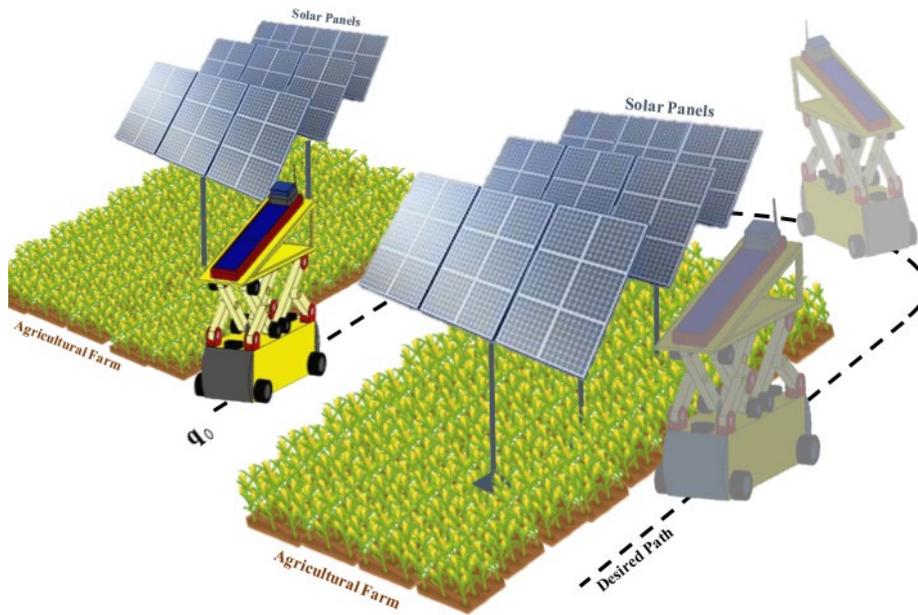


Figure 6 Desired path on the field

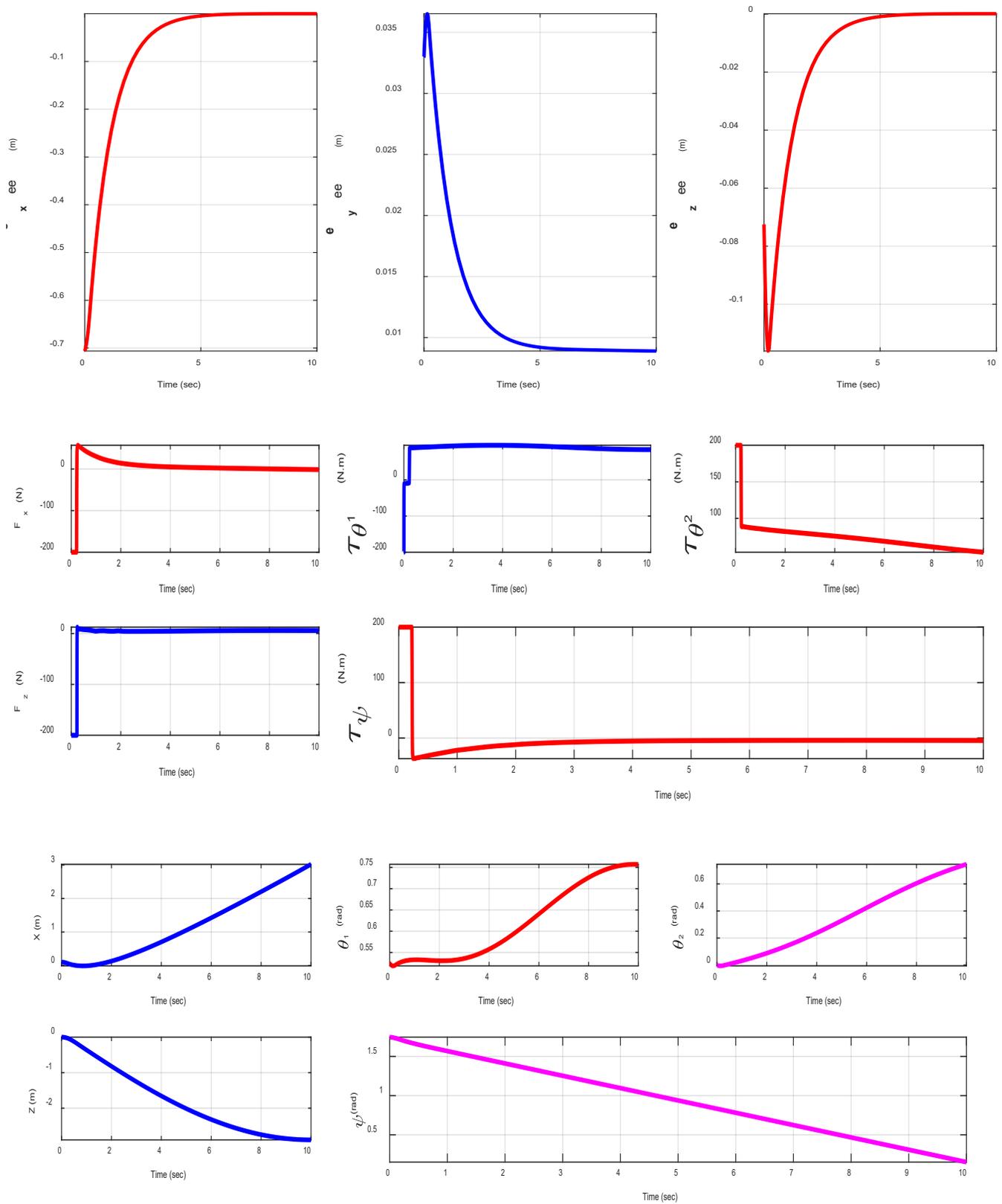


Figure 7 Results of TJ control simulation

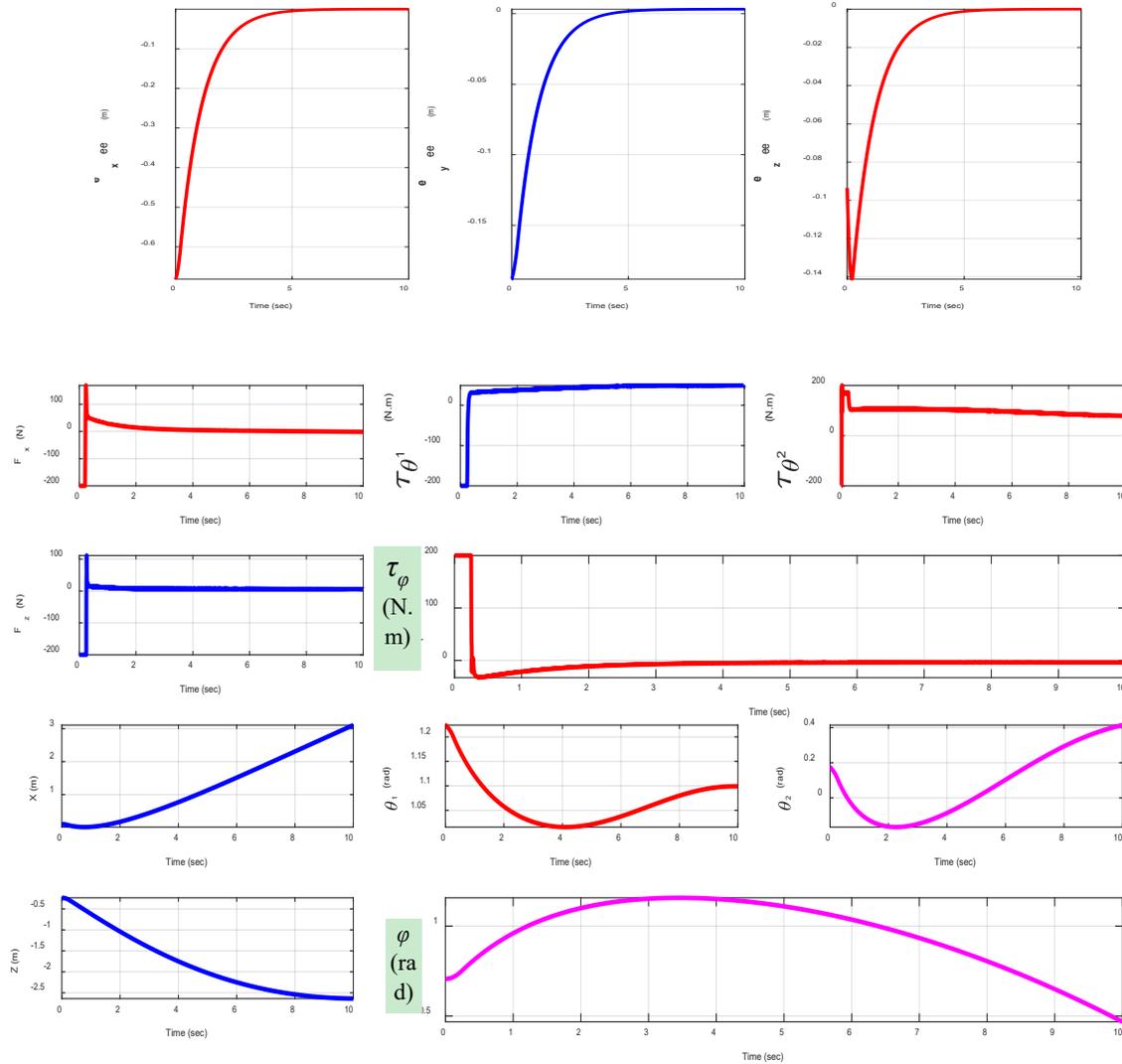


Figure 8 Results of MTJ control simulation

2.7 Computed torque method controller

In this section, as a kinetic model-based approach, a CTM controller is designed for the robot. This control method is a feedback linearization approach, using the dynamic model of the system. Hence, the controller is supposed to be accurate, but with the cost of heavy calculations of the dynamics model (Tso et al., 1991; Slotine and Weiping, 1988). Considering the joint space, the error is defined as:

$$e = q_d - q \tag{27}$$

The CTM control diagram is depicted in Figure 9. This diagram is built in MATLAB/Simulink. MATLAB performs the feedbacks and the kinetics model calculations. The robot block, depicted by a

picture of the robot, is made by ADAMS for simulation. As depicted, using the desired motion and the errors e , the control acceleration \ddot{q} is determined, as:

$$\ddot{\mathbf{q}}_c = \mathbf{q}_d + \mathbf{k}_v \dot{\mathbf{e}} + \mathbf{k}_p \mathbf{e} \tag{28}$$

Then, the required motor forces/torques and calculated, using the kinetics model, as:

$$\boldsymbol{\tau} = \mathbf{M}\ddot{\mathbf{q}}_c + \mathbf{V} + \mathbf{G} \tag{29}$$

Using this method, the nonlinearity of the control feedback and the computed torque is eliminated. Then, the response of the controlled system can be achieved as:

$$\ddot{\mathbf{e}} + \mathbf{k}_v \dot{\mathbf{e}} + \mathbf{k}_p \mathbf{e} = 0 \tag{30}$$

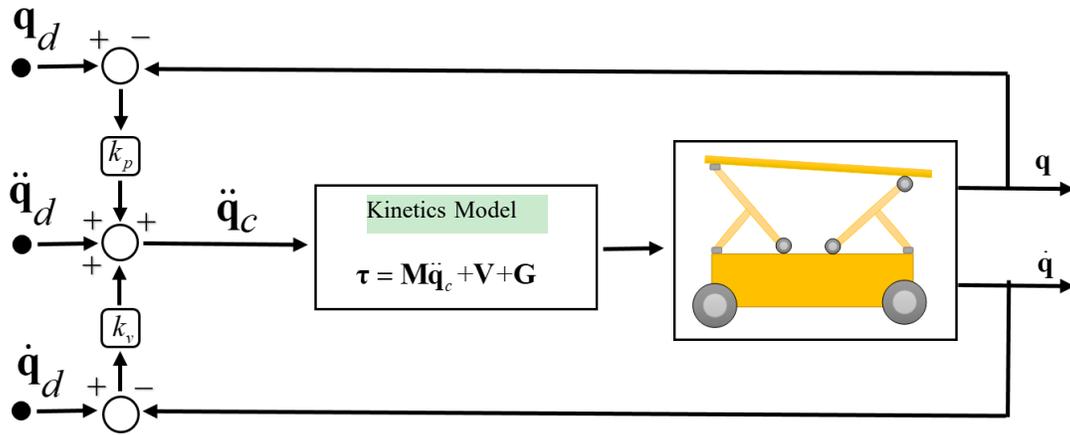
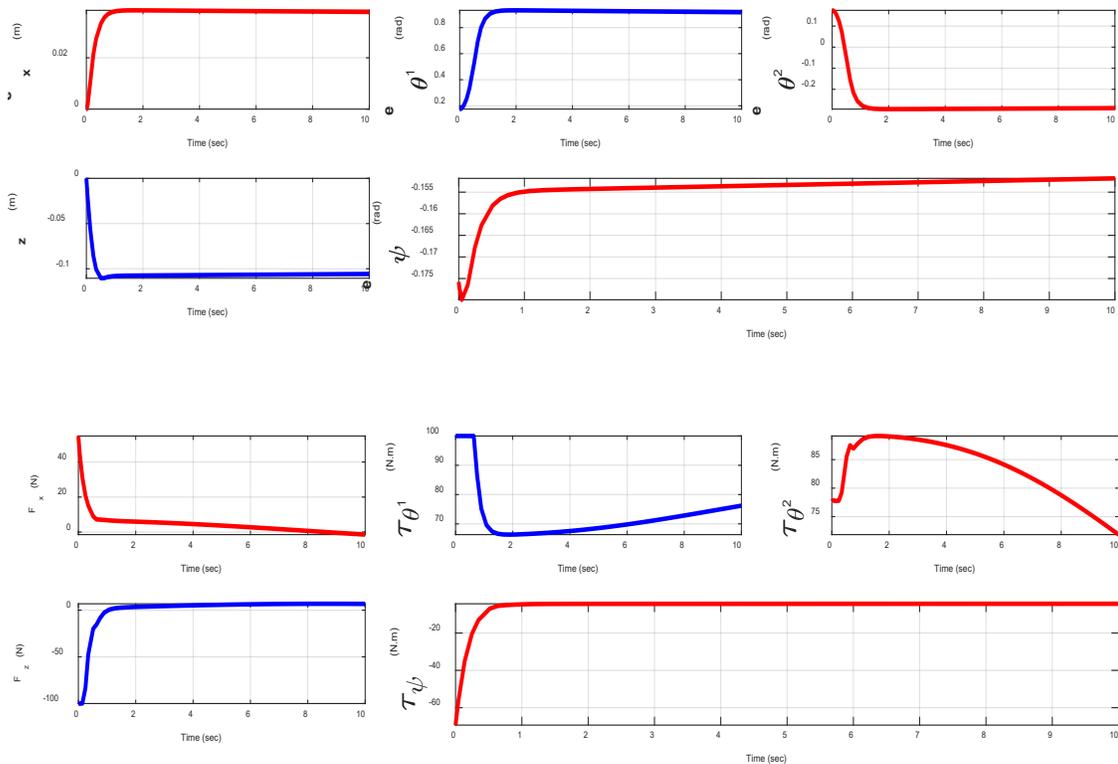


Figure 9 Simulink diagram of the CTM controller

Here, K_p and K_v are the corresponding controller coefficients. These coefficients determine the system response of Equation 28. Thus, they can be determined by pole placement, or according to the desired overshoot and settling time for the system. Here, a settling time of 4 sec is chosen, according to the

dimensions of the robot and its weight. In order to achieve a smooth response, the overshoot is chosen to be close to zero. Using this information, the values of K_p and K_v are chosen. After observing many simulations, the coefficients are chosen to be 5.



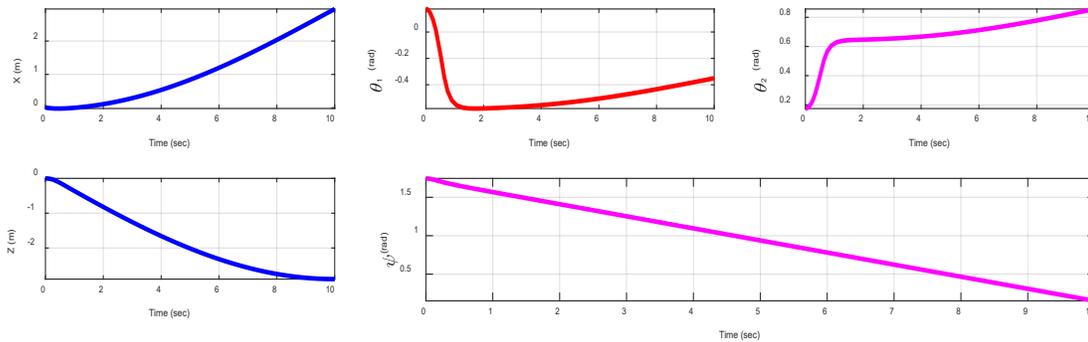


Figure 10 Results of CTM control simulation

For simulation, the desired trajectory was introduced in Equation 13. The co-simulation of MATLAB and ADAMS is used and the same initial errors are applied, as for the previous section. The obtained results for errors in tracking $x, \theta_1, \theta_2, z, \varphi$ and the required forces/torques are shown in Figure 10. As depicted, the errors are eliminated faster and more smoothly compared to the MTJ controller. Since the CTM controller uses the kinetics model, this better performance is expected. As the cost of this better performance, the control equations are more complicated, and more computational cost is required.

3 Conclusion

In this paper, a carrier robot was proposed, for automated cleaning of agrivoltaic plants. The conceptual design of the carrier robot was presented, along with its dynamic model, controller design, and simulations. First, the system geometry and its kinematic model were presented. Then, the Lagrange method was used to derive the dynamic model of the robot. For the model verification, the robot was also modeled in ADAMS. The results achieved from ADAMS were compared to the presented dynamics model. Three tests were simulated, and the comparison of results proved the validity of the proposed model. The simulations, control performance, and actuation forces should be studied, in order to study the concept and its proof. Therefore, two model-based controllers were designed and simulations for the desired task

were performed. One controller was based on the kinematics model and the other was based on the kinetics model, to determine the control actuation forces/torques. For simulation, a co-simulation of MATLAB/Simulink and ADAMS was designed and exploited. First, Transposed Jacobian controllers were introduced and designed, using the kinematics model. Using the MTJ method, the steady-state errors were eliminated with low computational costs. Then, based on the kinetics model, a CTM controller was designed. This method is a feedback linearization method, with reliable performance, but with the cost of high computational efforts of the kinetic model. The simulation results showed the satisfactory performance of the proposed automated robotic system. Furthermore, the required actuation forces in the task were determined, to be used for the detailed design of the robotic carrier.

References

- Azami, N., Zarafshan, P., Kermani, A. M., Khashehchi, M. and Kouravand, S. Design and analysis of an armed-otorotor to prune trees near the power lines. Proceedings of the 16th International Conference of Iranian Aerospace Society, Terhan, Iran, 2017: 1-6.
- Benjaminsen, Christina. 2017. An energy-efficient cleaning robot. Norwegian SciTech News. Available at: <https://norwegianscitechnews.com/2017/06/energy-efficient-cleaning-robot>. Accessed: July 2023.
- Boson. 2019. Robotic Solar Cleaning Systems. Boson Robotics Available at: <http://bosonrobotics.us>. Accessed November 2021.

- Burgaleta, Miguel et al. "Bilingualism at the core of the brain. Structural differences between bilinguals and monolinguals revealed by subcortical shape analysis." *NeuroImage* 125 (2016): 437-445.
- Cheah, C. C., M. Hirano, S. Kawamura, and S. Arimoto. 2003. Approximate Jacobian control for robots with uncertain kinematics and dynamics. *IEEE Transactions on Robotics and Automation*, 19(4): 692-702.
- Craig, J J. Ping Hsu, and S. Shankar Sastry. 1987. Adaptive control of mechanical manipulators. *The International Journal Of Robotics Research* 6 (2): 16–28.
- Craig, J J. 2006., *Introduction to Robotics Mechanics and Control*. Pearson Educacion.
- Deb, D., and N. L. Brahmhatt. 2018. Review of yield increase of solar panels through soiling prevention, and a proposed water-free automated cleaning solution. *Renewable and Sustainable Energy Reviews*, 82: 3306-3313.
- Dubowsky, S., and E. Papadopoulos. 1993. The kinematics, dynamics, and control of free-flying and free-floating space robotic systems. *IEEE Transactions on Robotics and Automation*, 9(5): 531-543.
- Ebrahimi, A., P. Zarafshan, S. R. Hassan-Beygi, M. Dehghani, and S. E. Hashemy. 2018. Design and analysis of a solar linear move irrigation system. In *6th RSI International Conference on Robotics and Mechatronics (ICRoM 2018)*, 382-387. Tehran, Iran, 23-25 October.
- Ecoppia. 2018. The world's leader in robotic solar panel cleaning. Ecoppia. Available at: <https://www.ecoppia.com/about>. Accessed: November 2021.
- Haddadi, S. J., Zarafshan, P. and Dehghani, M. 2022. A coaxial quadrotor flying robot: Design, analysis and control implementation. *Aerospace Science and Technology*, 120, 107260.
- Hafezipour, M., E. Saffari, P. Zarafshan, and S. A. A. Moosavian. 2013. Manipulation control of multi-body free-floating space robot based on software combination. *2013 First RSI/ISM International Conference on Robotics and Mechatronics (ICRoM)*, 2013. IEEE, 271-276.
- Hajiahmadi, F., M. Dehghani, P. Zarafshan, S. A. A. Moosavian, and S. R. Hassan-Beygi. 2019a. Trajectory control of a robotic carrier for solar power plant cleaning system. In *2019 7th International Conference on Robotics and Mechatronics (ICRoM)*, 463-468. Tehran, Iran, 20-21 November.
- Hajiahmadi, F., P. Zarafshan, M. Dehghani, S. A. A. Moosavian, and S. R. Hassan-Beygi. 2019b. Dynamics modeling and position control of a robotic carrier for solar panel cleaning system. In *2019 7th International Conference on Robotics and Mechatronics (ICRoM)*, 613-618. Tehran, Iran, 20-21 November.
- HyCleaner. 2020. hyCLEANER SOLAR facelift. Available at: <https://hycleaner.eu/en/produkte/hycleaner-black-solar>. Accessed November 2021.
- Jaradat, M. A., M. Tauseef, Y. Altaf, R. Saab, H. Adel, N. Yousuf, and Y. H. Zurigat. 2015. A fully portable robot system for cleaning solar panels. In *2015 10th International Symposium on Mechatronics and its Applications (ISMA)*, 1-6. Sharjah, United Arab Emirates, 8-10 December.
- Karimi, M., and S. A. A. Moosavian. 2010. Modified transpose effective jacobian law for control of underactuated manipulators. *Advanced Robotics*, 24(4): 605-626.
- Khadka, N., A. Bista, B. Adhikari, A. Shrestha, D. Bista, and B. Adhikary. 2020. Current practices of solar photovoltaic panel cleaning system and future prospects of machine learning implementation. *IEEE Access*, 8: 135948-135962.
- Khalaji, A. K., and S. A. A. Moosavian. 2015. Modified transpose Jacobian control of a tractor-trailer wheeled robot. *Journal of Mechanical Science and Technology*, 29: 3961-3969.
- Kumar, N. M., K. Sudhakar, M. Samykano, and S. Sukumaran. 2018. Dust cleaning robots (DCR) for BIPV and BAPV solar power plants-A conceptual framework and research challenges. *Procedia Computer Science*, 133: 746-754.
- Mark W., S. H. Spong and M. Vidyasagar 1st ed., 2019. *Robot Modeling and Control* .
- Mazinani, M., P. Zarafshan, M. Dehghani, H. Etezadi, K. Vahdati, and G. Chegini. 2021a. Modeling and control of a pollinator flying robot. In *2021 9th RSI International Conference on Robotics and Mechatronics (ICRoM 2021)*, 548-553. Tehran, Iran, 17-19 November.
- Mazinani, M., M. Dehghani, P. Zarafshan, H. Etezadi, K. Vahdati, and G. Chegini. 2021b. Design and manufacture of an aerial pollinator robot for walnut trees. In *2021 9th RSI International Conference on Robotics and Mechatronics (ICRoM 2021)*, 445-450. Tehran, Iran, 17-19 November.
- Miraikikai. 2019. MIRAI Products. Available at: <https://miraikikai.wixsite.com/miraikikai/products-e>. Accessed November 21, 2021.
- Mondal, A. K., and K. Bansal. 2015. A brief history and future aspects in automatic cleaning systems for solar

- photovoltaic panels. *Advanced Robotics*, 29(8): 515-524.
- Moosavian, S. A. A., and E. Papadopoulos. 1997. Control of space free-flyers using the modified transpose Jacobian algorithm. In *Proc. of the 1997 IEEE/RSJ International Conference on Intelligent Robot and Systems. Innovative Robotics for Real-World Applications. IROS'97*, 1500-1505. Grenoble, France, 11-11 September.
- Moosavian, S. A. A., and E. Papadopoulos. 2007. Modified transpose Jacobian control of robotic systems. *Automatica*, 43(2): 1226-1233.
- Moshe Saraf, Solar Panel Cleaning System. Mar. 24, 2015. US 8,984,704 B2.
- Saied Tadayon, Robot for solar farms. Aug. 6, 2015. US 2015/0217443 A1
- Serbot. 2020. Gekko Solar Hightec Robot. SERBOT SWISS INNOVATIONS. Available at: <https://www.serbot.ch/en/solar-panels-cleaning/gekko-solar-robot>. Accessed November 2021.
- Slotine, J. J. E., and W. Li. 1988. Adaptive manipulator control: A case study. *IEEE Transactions on Automatic Control*, 33(11): 995-1003.
- Syafiq, A., A. K. Pandey, N. N. Adzman, and N. A. Rahim. 2018. Advances in approaches and methods for self-cleaning of solar photovoltaic panels. *Solar Energy*, 162: 597-619.
- Tranca, D. C., D. Rosner, and A. V. Palacean. 2017. Autonomous flexible low power industrial IoT controller for solar panels cleaning systems. In *2017 21st International Conference on Control Systems and Computer Science (CSCS)*, 106-112. Bucharest, Romania, 29-31 May.
- Tso, S. K., Y. Xu, and H. Y. Shum. 1991. Variable structure model reference adaptive control of robot manipulators. In *Proc. of the 1991 IEEE International Conference on Robotics and Automation (ICRA)*, 2148-2153. Sacramento, CA, USA, 9-11 April