Robotica: (2020) volume 38, pp. 2039–2059. © Cambridge University Press 2020
doi:10.1017/S0263574719001814

Design, Modeling, and Control of a New Manipulating Climbing Robot Through Infrastructures Using Adaptive Force Control Method

V. Boomeri and H. Tourajizadeh*

Mechanical Engineering Department, Faculty of Engineering, Kharazmi University, Tehran, Iran
E-mail: vahidboomeri@gmail.com

(Accepted 10 December, 2019. First published online: January 14, 2020)

SUMMARY

In this paper, design, modeling, and control of a grip-based climbing robot are performed, which consists of a triangular chassis and three actuating legs. This robot can climb through any trusses, pipeline, and scaffolds structures and can perform any inspectional and operational tasks in the high height which decreases the falling danger of operation and increases the safety of the workers. The proposed robot can be substituted for the workers to decrease the risk of death danger and increase the safety of the operation. Since these kinds of infrastructures are truss shaped, the traditional wheel-based climbing robots are not able to travel through these structures. Therefore, in this paper, a grip-based climbing robot is designed to accomplish the climbing process through the trusses and infrastructures in order to perform inspecting and manipulating tasks. Hence, a proper mechanism for the mentioned robot is designed and its related kinematic and kinetic models are developed. Robot modeling is investigated for two different modes including climbing and manipulating phases. Considering the redundancy of the proposed robot and the parallel mechanism employed in it, the active joints are selected in a proper way and its path planning is performed to accomplish the required missions. Concerning the climbing mode, the required computed torque method (CTM) is calculated by the inverse dynamics of the robot. However, for the manipulation mode, after path planning, two controlling strategies are employed, including feedback linearization (FBL) and adaptive force control, and their results are compared as well. It is shown that the latter case is preferable since the external forces implemented on the end effector tool is not exactly predetermined and thus, the controller should adapt the robot with the exerted force pattern of the manipulator. The modeling correctness is investigated by performing some analytic and comparative simulation scenarios in the MATLAB and comparing the results with the MSC-ADAMS ones, for both climbing and manipulating phases. The efficiency of the designed controller is also proved by implementing an unknown force pattern on the manipulator to check its efficiency toward estimating the mentioned implemented forces and compensating the errors. It is shown that the designed robot can successfully climb through a truss and perform its operating task by the aid of the employed adaptive controller in an accurate way.

KEYWORDS: Climbing robot; Manipulation; Hybrid robot; Modeling; Adaptive force control.

1. Introduction

Nowadays, inspection and manipulation of industrial equipment in different environments have a significant role in the repair and maintenance industry. Enormous risks threaten the human operators

* Corresponding author. E-mail: Tourajizadeh@khu.ac.ir

who are responsible for performing the mentioned tasks depending on the environment in which the inspection or operation is supposed to be performed in high height. The danger of falling from a high height, the risk of chemical toxins, etc., force us to use mobile robots instead of human operators. Some of these hazards are compensated recently, thanks to the invention of climbing robots. Most of the climbing robots are designed to ascend over the walls and flat surfaces for which wheel-based and adhesive-based robots can cover the mentioned mission. However, there are some structures like trusses, pipeline structures, and scaffold in which other mechanisms are required to be employed in such these environments. Hence, by consideration of the fast development of technology and industry, employing mobile robots that can climb through the infrastructures is extremely necessary and inevitable. Large industrial environments such as metallic bridges, nuclear plants, telecom masts, and truss-shaped civil structures installed in gyms and amphitheater ceilings, astronomy space stations, pipelines in refineries, military facilities, and civil scaffolds are the main target of the application of climbing robots. Most of the existing and traditional climbing robots are designed to climb the walls in which a flat surface is necessary for climbing and this flat surface is not usually available in most of the mentioned structures such as trusses and scaffold-like infrastructures. Therefore, a grip-based mechanism is a proper design for a climbing robot which can satisfy the mentioned requirement for climbing through the structures. Thus, some climbing robots are designed and manufactured in order to climb the mentioned structures. These robots can perform operational tasks in high heights, including maintenance, constructing such as welding and riveting. Also, periodic inspectional tasks can be performed by climbing robots such as scavenging and inspecting the components in height, such as searching for destructions such as a crack, pipe leaks, and breakdowns in a pipeline structure. This is obvious that performing these operations by a human in such high heights is extremely dangerous even for skilled workers and can result in deadly catastrophes like falling, electric shock dangers, etc. The statistics of deaths in work associated with these kinds of dangers are booked about 25% according to world statistics. Moreover, there are a lot of tasks that should be done in the environments which are essentially dangerous for human life or human presence is forbidden in these environments like nuclear plants ducts, pipelines in petrochemical industries, etc. Hence, utilization of climbing robots which can ascend truss-shaped structures and scaffold-like structures is a logical alternative way.

Climbing robots can be categorized into four main subgroups: grip-based robots, wheel-based robots, insect robots, and adhesive robots that each of them is capable of covering special tasks according to their structures. For ascending walls or flat terrains, adhesive robots can be employed, known as wall-climbing robots. ROMA II[1] which is manufactured in RoboticsLa is an example of such a wall-climbing robot. Another multipurpose robot that is equipped with suction pumps is W-Climbot.[2] The grip-based robots are proposed for ascending trusses which can bypass the obstacles, with great maneuverability. ROMA I[3] is a grip-based robot which is designed for climbing infrastructures and metallic bridges for maintenance and is capable of switching its locomotion along with the horizontal and vertical directions. The 3D-CLIMBER[4] can climb through the pipes with branches and can also perform inspections, fastening, and welding tasks. Shady3D[5] with a passive bar with a six-degree of freedoms (DOF) manipulation control is a modular climbing robot and is able to ascend through the trusses. RiseV3[6] is a quadruped mechanism that provides the possibility of moving on poles and trees with its special design. Stewart mechanism is also proposed in pole climbing robot (PCR)[7] with the ability to climb through the curved ducts at 90°curvature angle with the capability of crossing branches in duct structures for both tubular profiles and rectangular cross-section profiles. Pobot V2 with natural self-locking capability[8] is manufactured to climb the poles. Climbot[9] is a grip-based climbing robot inspired by sloths and apes and is designed for electrical poles maintenance such as changing the lamps. A hybrid PCR is designed in ref. [10] for ascending through the poles in which the robot's mechanism is a combination of serial and parallel mechanisms. The UT-PCR[11] is a wheel-based climbing robot to perform maintenance operations on poles such as coloring the poles, washing the lamps, etc. The wheel-based robots can also be used for stair climbing purposes. A new kind of stair climbing robot is designed and manufactured in ref. [12] for crossing over the urban stairs with overhangs. In order to meet the mentioned target a novel design of wheel is proposed and investigated. A multilimbed climbing robot in ref. [13] is modeled and simulated in which Jacobian of the robot is used for position and force control of generalized coordinates. Libra[14] is designed at MIT university in order to climb through the infrastructures related to space crafts and space stations. Its mechanism is made up of one main body with three limbs connected by some joints.

https://doi.org/10.1017/S0263574719001814 Published online by Cambridge University Press

Modeling of a robot is the basic step of its design and manufacturing especially for complicated mobile robots with manipulators that have extreme nonlinear formulation. For example, dynamic modeling of a wheeled mobile robot is performed in ref. [15] using Gibbs–Appell (G-A) formulation to avoid Lagrange-multipliers' calculations caused by nonholonomic constraints. Another modeling is performed in ref. [16] for a space manipulator (SM) which is a parallel robot with three limbs. Another parallel kinematic machine (PKM) is considered in ref. [17] and its related kinematics is derived using transformation matrices while its kinetics is extracted employing virtual work principle and Lagrange equation. Abedinnasab et al. studied a parallel redundant actuated Stewart universal prismatic spherical (UPS) and developed its corresponding model in order to compare it with a nonredundant actuated robot with the same configuration.[18] Since its Jacobean matrix is nonsquare, the minimum norm method is utilized to obtain the unique solution for the inverse kinematics. As mentioned, covering operational tasks is one of the interesting features that can be added to the climbing robots. Here a manipulator needs to be designed to perform an operational task, which either is carrying a load or implementing a force on a target point, while the end effector acts on the environment in a path. For the redundant manipulators, numerous paths can be chosen for the same trajectory of end effector. The path planning can be generated in an optimum way to save energy and bear the force acting on end effector on a specific path in which the effects of these implementing forces on the main manipulator actuators are inevitable. The effects of these forces can be treated as payloads that are dependent on the end-effector trajectory in manipulators installed on mobile robots. For this case, the optimal trajectory control method is accomplished in ref. [19] in which the optimal path and its related maximum payload are derived by using two-point boundary value problem. For a redundant manipulator[20] according to the nonsquare Jacobean matrix, dynamic load-carrying capacity has different definitions rather than the regular ones. Here an optimal path can be derived for carrying the maximum payload since the inverse kinematics of the robot has not a unique solution. The same manipulator is optimized also using the closed-loop optimal controller in ref. [21] using state-dependent Riccati equation method. Also for flexible manipulators installed on the mobile carriers, optimal path planning is studied in ref. [22] for the first time in which Pontryagin's minimum principle method is employed to generate an optimal path for a mobile flexible manipulator. In some cases, the exact value of the load or force is not determined which significantly affects the designed nonlinear robot controller. Here the challenge of model uncertainty and proper estimation is concerned. Actually, in these cases either the exact amount of the load should be known or the external force estimation is essentially required in order to set and exert the proper amount of the motors torque by the controller to compensate the effects of mentioned exerting loads on the manipulator end effector. For Puma 600, the estimation of both inertia parameters related to the load and other physical items of the robot is performed by using the least square method.[23] One of the most suitable control methods for compensating the uncertainty problems is the adaptive control method. Most of the present adaptive control research for robots are focused on estimating the physical robot parameters. Adaptive control for a manipulating arm is illustrated by separating known and unknown parameters into two distinct matrices to estimate unknown parameters according to the Lyapunov stability algorithm by using sliding surface theorem.[24] The same adaptive controlling strategy is employed in ref. [25] for estimation of the parameters of a parallel robot with redundant joints. The estimation is performed for evaluating the real friction of the joints, and it is shown that this adaptive controller can significantly improve the calculated controlling input using the computed torque method (CTM). Adaptive controller for estimation of the load is designed for a flexible manipulator in ref. [26] in which two methods of variable controlling gains including least squares identification algorithm and self-tuning pole placement controller are compared. This study is performed for pick and placement processes in which sever load change is inevitable at their manipulators. In ref. [27], an adaptive controller is proposed and implemented for robust manipulating of the robots with unmolded dynamic to track the desired trajectory. Optimality and stability of the designed controller is proved by the aid of game-algebraic Riccati equation and the validation procedure is tested on a three-DOF robot manipulator. For some multilegged robots which can be usable for locomotion and manipulation, according to their task that should be done under the environment contact, Jacobian transpose control is applied, called coordinated Jacobian transpose control[28] in which this calculation is valid because in such robots the transmission gear ratio is high and the dynamic equations can be ignored similar to the work done on Libra climbing robot. The effect of external forces on the robot while the manipulation process is performing is inevitable that can impress the accuracy of the robot. Hence, the robustness

https://doi.org/10.1017/S0263574719001814 Published online by Cambridge University Press

of the robot should be conserved against these external effects in which their amounts are uncertain. Actually, for climbing robots which are supposed to perform manipulative tasks, the effect of external forces is usually ignored while the climbing robots are more vulnerable to instability in altitude. Most of the designed controllers for climbing robots are based on simple CTM.[29] In ref. [30] locomotion control of a biped robot is investigated by defining the trajectory of each joint using inverse kinematics. The joint trajectories change, just when the phase of motion switches. For a two-legged biped wall climbing robot of ref. [31] actuated by vacuum cups with five DOFs, a controller is employed to compensate the gravity force and inertia loads. Generalized mass adaptive function is used to control the mentioned robot. Also for a gait traveler robot which is one of the most usual application of walking underactuated biped robots, the optimal path planning is performed in ref. [32] in order to achieve the trajectory with less amount of energy consumption.

As seen most of the mentioned climbing robots which are designed and manufactured so far, need a flat or curvature surface through which a wheel, limb, or suction vacuum mechanism needs to be designed for their locomotion module. For the vast amount of structures which are made up of trusses, the lack of a proper mechanism which could be employed in such climbing robots can be observed clearly. This robot is essentially required for conducting the inspectional or manipulation operations instead of human workers in order to decrease the risk of falling danger. Considering the redundancy of the proposed robot, the robot can maneuver well through complex infrastructures by the capability of obstacle avoidance and also the related path planning should be done to save energy consumption. The other advantage of our proposed is its great stability which is provided by the aid of the designed parallel mechanism employed in the robot. To meet this goal two legs are supposed to grasp two support points in order to increase the climbing reliability. Also, the capability of performing operational tasks is neglected in most of the previous research. Moreover, the climbing robots with the ability of manipulation should be controlled in a robust way against the disturbances and exterior factors, while their stability should also be guaranteed during carrying high payloads. This ability is strongly desired since such robots need to provide a specific force for their operation at the same time with exact positioning for performing their grasping process. The adaptability of the manipulator with respect to the external forces and the carried payload is another important item that has been ignored in the literature. Therefore, in this paper, a proper climbing robot is designed and proposed to fulfill all of the mentioned expectances. The robot has a grip-based locomotive mechanism, thus, it can ascend through the trusses and infrastructures. Finally, an adaptive controller is designed for the robot while the manipulative operations are performing to support the robustness of the robot against the external disturbances related to the operational tasks and to guarantee its stability during its mission. To meet the mentioned goal, in the next section, the required mechanism of the robot is designed and its maneuverability and mobility are discussed. Afterward, in Section 3, the kinematic and kinetic model of the proposed robot is extracted by which the controlling procedure can be accomplished. In Section 3, the proper active joints are selected considering the redundancy of the robot and the proper path planning of the robot climbing is performed. Using the model of Section 3 together with the planned path of Section 4, two controlling strategies based on FBL and adaptive force controller are designed and implemented on the robot in Section 5 to compare their efficiency on the robot performance. Finally, the correctness of the proposed model is verified in Section 6 by MATLAB simulation and comparing their related results with MSC-ADAMS. Also, the efficiency of the designed adaptive force controller is proved in this section by conducting some comparative simulation scenarios in MATLAB-Simulink. It can be seen that the proposed climbing robot equipped by the designed controller can successfully accomplish an operational task in high infrastructure.

2. Robot Design

2.1. Mobility and maneuverability

According to the expected requirements mentioned in the literature, for our desired application, the base idea of the robot morphology is inspired by the monkey's movement on trees. They travel through the trees by grasping the branches and swinging through them. A monkey grasps two branches simultaneously by one foot and tail as two support points and uses its free hand for reaching to a new branch in order to grasp it as new support. It switches the supporting feet, hands, or tail roles

https://doi.org/10.1017/S0263574719001814 Published online by Cambridge University Press

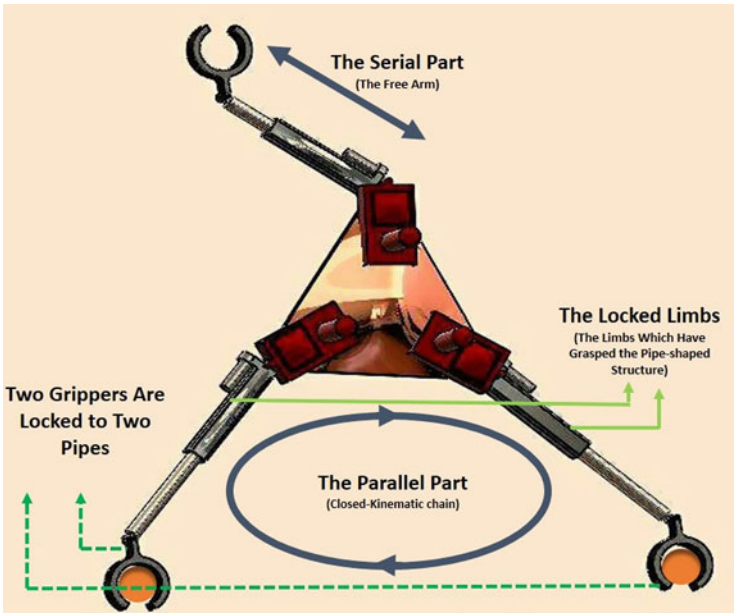

Fig. 1. The computer-aided design (CAD) view of the designed hybrid robot by introducing the subparts.

to freehand member role or *vice versa* alternately for continuing its movement on the trees. Hence, the proposed robot here has three limbs, each of which has three DOF (two rotational and one prismatic joint) and a triangular base through which the limbs are attached to. The schematic view of the designed robot which is similar to a monkey is shown in Fig. 1. Thus, the robot has a hybrid structure with five DOFs in a planar workspace, including three DOFs for the lower part and two DOFs for the upper free arm. The role of the upper arm switches alternatively as the lower arms of the robot and *vice versa*. Therefore, the robot needs to be actuated by five actuators including three prismatic jacks and two rotational motors in 2D planar workspace where these active joints are exchanged in roles in robot locomotion. In other words, during locomotion, active joints may become passive joints or *vice versa*. Redundant DOFs are installed in the serial module of the robot in order to increase the mobility and maneuverability of the robot toward bypassing the obstacles and increasing its workspace.

2.2. Manipulation and locomotion phases

The robot functionality is defined by two main phases according to its mission, locomotion phase and 2 manipulation phase. Locomotion phase means the robot travels from one point to another point like walking through the scaffold. Manipulation phase means the robot moves with its five DOFs while two grippers are locked to the frame and it acts as a fixed robot. These two phases are actually coupled since manipulation movement is a portion of locomotion. Indeed, a series of five DOFs manipulating movements chained together results in a locomotion phase. During this process, fixed limbs switch to free arm and *vice versa* alternatively to perform a special operation or to take another branch according to its momentary phase. For locomotion phase, the mentioned stages repeat until one cycle of the robot's locomotion would be accomplished. This process is illustrated in Fig. 2.

Both of locomotion and manipulation phases are considered in this paper. The locomotion process is performed using CTM, while the manipulation process is supported using a robust closed-loop system of the adaptive controller to neutralize the destructive effect of the external forces implemented during the operational task. The robot's modeling and controlling are performed using MATLAB-Simulink. In the end, the model is verified by analytic simulations and compared with MSC-ADAMS simulation results. It will be observed that with the assist of the proposed robot, climbing through the trusses and manipulation process are practically possible.

3. Modeling of Manipulation Phase

In this section, kinematic and kinetic model of the proposed robot is extracted for both manipulating and climbing phases. In kinematic modeling, the relationship between the workspace and joint space

https://doi.org/10.1017/S0263574719001814 Published online by Cambridge University Press

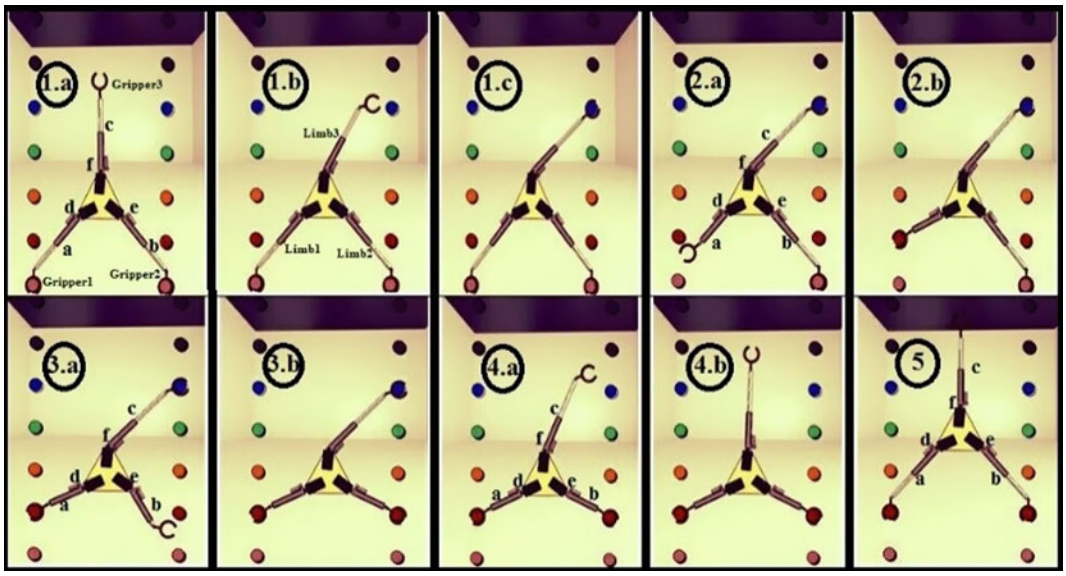

Fig. 2. The steps (gaits) one to five of robot's locomotion on infrastructure in its one locomotion cycle.

of the designed hybrid robot is extracted using the Jacobian matrix and its pseudo-inverse matrix for inverse kinematics calculation. This model is required to convert the desired movement of the robot in its workspace to the relevant required movement of its joint space. Also, the dynamic equation of the robot which is required for plant modeling of the system as the control method of system is a model-based controller. In order to control the robot in its two phases, CTM is employed for the locomotion phase and the model-based adaptive controller is employed for the manipulation phase. To meet this goal kinematic and kinetic model of the robot needs to be extracted.

3.1. Kinematics modeling

The planar robot is designed to move in a vertical plane and cover a high range of workspace. The shape of the proposed robot together with its associated parameters is shown in Fig. 3. The robot consists of three limbs and one triangular chassis as the main body. In the manipulation phase, two limbs are fixed to the branches by the aid of their grippers and another limb is free to perform an operational task. In manipulation, the closed chain of the robot is a parallel mechanism, while the free limb is a serial mechanism. These subparts are introduced in Fig. 1. In the climbing phase, the third free limb grasps another branch. Afterward, one of the two former fixed limbs is then released to grasp a new branch, and this cycle will be iterated until the robot reaches the desired height (Fig. 2). Ignoring the grippers of DOFs, five actuating joints are engaged for both manipulation and climbing phases. Three prismatic and two rotational joints are the main DOFs of the proposed robot. Thus, in order to provide the mobility of the robot in all of its climbing cycles, it is necessary to employ active actuators for all of the main joints; while in each stage, only five joints of the main DOFs are active and the other ones are passive. It should be noted that the proposed robot is a mobile one in which its limbs exchange their roles between free limb and locked limb during different steps of motion. Thus, no passive joint is permanently passive and no active joint is permanently active. In other words, the state of a joint to be a passive or active one depends on the instantaneous configuration of the robot which is determined by the designer. As mentioned, the robot is constrained since two limbs are fixed on the frame by the grasped grippers. As a result, the dependent generalized coordinates (DGCs) which are the passive joints should be defined as a function of independent generalized coordinates (IGCs). This procedure is prerequisite for extraction of the dynamic equations of the system when the robot is constrained. Also since the number of the closed kinematic chains of the parallel part, which is two according to the topologic vision of the front and back views of the closed kinematic chain, is not equal to the number of parallel system DOFs which is three DOFs, for the parallel part of the robot, the parallel mechanism of the robot is a nonfull parallel mechanism, which needs more challenging formulation to extract the related kinematic relations.[33] Considering the kinematics of the mentioned nonfull parallel mechanism in the manipulation phase, finding the relation among DGCs and IGCs and choosing the active joints is extremely significant. Here the parallel module of

https://doi.org/10.1017/S0263574719001814 Published online by Cambridge University Press

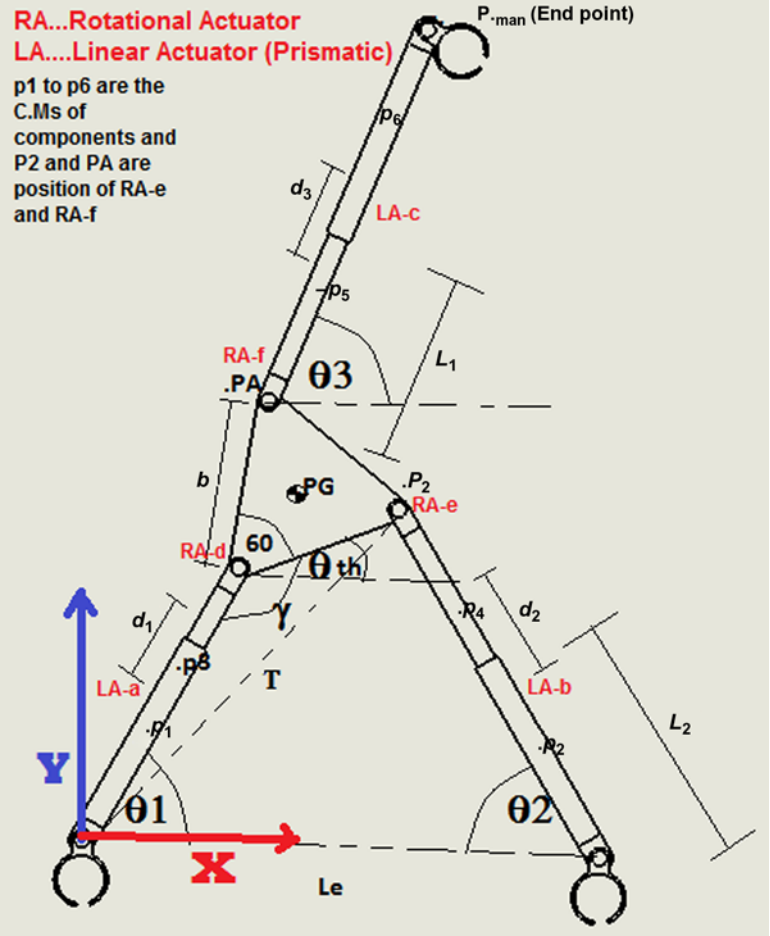

Fig. 3. Schematic view of the designed robot and its geometry parameters.

the robot is a nonfull parallel mechanism. Therefore, active joints should be chosen in an optimal way. Here according to Fig. 3, d_1, d_2, d_3, γ, and Θ_3 are chosen as the related IGCs. Note that the active joint layout is according to this choice that serial subpart's joints should be active (the free arm revolute and prismatic joints) and in the parallel subpart two prismatic joints and one of the revolute joints which are the connection between the chassis and limbs, should be considered as the active joint (γ here). Considering the geometrical configuration of the designed robot, DGCs can be defined as the functions of IGCs by geometrical calculation around the closed chain of the parallel mechanism, so the mentioned definition is fulfilled. We first consider this couple of equations along diagonal direction T shown in Fig. 3, where L_1 is the length of the jacks pistons, L_2 is the length of the cylinders, Le is the distance between the supporting grippers, and b is the length of the triangle sides. The chosen active joints as the IGCs are d_1, d_2, d_3, and γ, and θ_3 while θ_1, θ_2, and θ_{th} are DGCs or displacement of the passive joints. By solving a couple of equations along T, the function of one of the DGCs as the function of IGCs can be obtained as Eq. (1):

$$\theta_2 = a\cos\left\{\frac{(\mathrm{L}_1+\mathrm{L}_2+\mathrm{d}_2)^2-(\mathrm{L}_1+\mathrm{L}_2+\mathrm{d}_1)^2+Le^2-b^2+2(\mathrm{L}_1+\mathrm{L}_2+\mathrm{d}_1).b.\cos(\gamma)}{2(\mathrm{L}_1+\mathrm{L}_2+\mathrm{d}_2).Le}\right\}. \quad (1)$$

By considering the robot geometry along X-axis, it is possible easily to extract θ_1 and θ_{th} as the functions of IGCs which is shown in Eq. (2):

$$\theta_1 = a\cos\left(\frac{\pm\sqrt{(A.B)^2+B^4-(B.C)^2}+A.C}{A^2+B^2}\right) \Rightarrow \theta_{th}=\gamma-\pi$$
$$+\,a\cos\left(\frac{\pm\sqrt{(A.B)^2+B^4-(B.C)^2}+A.C}{A^2+B^2}\right), \quad (2)$$
$$A=(L_1+L_2+d_1)+b.\cos(\gamma-\pi);\ B=-b.\sin(\gamma-\pi);\ C=Le-(L_1+L_2+d_2)\cos\theta_2.$$

https://doi.org/10.1017/S0263574719001814 Published online by Cambridge University Press

Now the position of the center of masses (C.Ms) of each robot's component (p_i) and the end point of the manipulator coordinate ($P_{.man}$) shown in Fig. 3 can be obtained as the functions of IGCs. By replacing equations (1, 2) in θ_1, θ_2, and θ_{th} for the end point of manipulator, we have:

$$p_{.man} = \begin{bmatrix} -(L_1 + L_2 + d_2)\cos\theta_2 + Le \\ (L_1 + L_2 + d_2)\sin\theta_2 \end{bmatrix} - \begin{bmatrix} \frac{b}{2}.\cos\theta_{th} \\ \frac{b}{2}.\sin\theta_{th} \end{bmatrix} + \begin{bmatrix} -\frac{b}{2}\sqrt{3}\sin\theta_{th} \\ \frac{b}{2}\sqrt{3}\cos\theta_{th} \end{bmatrix} + \begin{bmatrix} (L_1 + L_2 + d_3)\cos\theta_3 \\ (L_1 + L_2 + d_3)\sin\theta_3 \end{bmatrix}. \tag{3}$$

In the above equation, d_3 and θ_3 are the other two IGCs related to the serial section of the robot. All points of P_2, P_G, P_A, and p_1 to p_6, and $P_{.man}$ are introduced in Fig. 3. Moreover, velocity kinematics can be extracted using the Jacobian matrix for each C.M shown in Fig. 3.

Positions of p_1 to p_6 are the C.M coordinates and φ_{iz} is the rotation of each component about the z-axis, perpendicular to the plane of robot motion. These coordinates can be defined as the function of main DOFs. Forward kinematics through which the velocity of the end point of the manipulator can be obtained is extracted as:

$$\begin{bmatrix} V_x \\ V_y \\ \omega_z \end{bmatrix}_{(3\times1)} = J_{(3\times5)} \begin{bmatrix} \dot{d}_1 & \dot{d}_2 & \dot{\gamma} & \dot{\theta}_3 & \dot{d}_3 \end{bmatrix}^T_{(5\times1)}. \tag{4}$$

where V and ω are the translational and rotational velocity of the gripper's end point in the free limb which indicates robot end point workspace, respectively. The left side of the equation is the workspace velocity of the manipulator end-effector while the right side is the vector of joint space velocity and J is the Jacobian matrix. By defining ϕ_Z, $p_{.man}$ as the orientation and position of the free arm's end point with respect to the main DOFs, the related Jacobian matrix can be derived in order to calculate the forward kinematics in Eq. (4).

3.2. Dynamics modeling

The dynamic equation of motion extracted from energy method is according to Eq. (5). In Eq. (5), q_n indicates IGCs of the system (d_1, d_2, γ, θ_3, and d_3):

$$\tau_n = D(q_n)\ddot{q}_n + C(q_n, \dot{q}_n)\dot{q}_n + g(q_n). \tag{5}$$

where D is inertia matrix, C is the Coriolis matrix, and the last term indicates the gravity vector. Using the above inverse dynamics, the required torques of the motor and also the required forces of the jacks can be easily calculated for the desired path using CTM. The above dynamic equation is extracted for the proposed robot of this paper using MATLAB and its terms and functions can be briefly introduced as below:

$$\begin{aligned} l\tau &= D(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q), \\ \begin{bmatrix} F_1 \\ F_2 \\ \tau_1 \\ \tau_2 \\ F_3 \end{bmatrix} &= D(d_1, d_2, \gamma, \theta_3, d_3)_{(5\times5)} \begin{bmatrix} \ddot{d}_1 \\ \ddot{d}_2 \\ \ddot{\gamma} \\ \ddot{\theta}_3 \\ \ddot{d}_3 \end{bmatrix} + C(d_1, d_2, \gamma, \theta_3, d_3, \dot{d}_1, \dot{d}_2, \dot{\gamma}, \dot{\theta}_3, \dot{d}_3)_{(5\times5)} \begin{bmatrix} \dot{d}_1 \\ \dot{d}_2 \\ \dot{\gamma} \\ \dot{\theta}_3 \\ \dot{d}_3 \end{bmatrix} \\ &\quad + g(d_1, d_2, \gamma, \theta_3, d_3)_{(5\times1)}. \end{aligned} \tag{6}$$

The above-modeled dynamics of the robot can also be used for simulating the forward dynamics as the plant of the system and extracting the actual movement of the robot using the implemented generalized forces. Also, this equation should be employed in a switching algorithm for covering the climbing phase. It should be noticed that the dynamics equations are extremely large and nonlinear which cannot be shown here.

https://doi.org/10.1017/S0263574719001814 Published online by Cambridge University Press

4. Modeling of Climbing Phase

For the climbing phase, the general kinematics and kinetics of the robot are the same as the manipulation phase except that an important consideration needs to be engaged in locomotion phase. In each step of motion, two legs are locked to the branches by grippers as the main supports of the robot and the other limb moves freely to reach the next branch; thus, during the climbing, the active and passive joints alter frequently depending on locomotion step which indicates a specific configuration of the robot in each step of motion. Thus, first of all, path planning is required to be performed to provide a suitable switching plan for the active and passive joints and realize the proper path. Then in each stage of the planned cycle as a locomotion step, the mentioned modeling is valid while the active and passive joints are exchanging their roles in each step of locomotion to follow a predefined path.

In the climbing phase, the robot is supposed to climb through a predefined pattern of the truss. This process can be divided into definite steps which will be iterated in each cycle of locomotion. Here five steps are considered for each cycle of climbing through which the altitude of the robot increases in each cycle according to Fig. 2. These cycles will be repeated until the robot reaches the height in which the operation is supposed to be conducted. In each cycle, the height of the robot increases by 0.1 m which is the vertical distance between the branches of truss and at the end of the cycle the robot reaches to its initial configuration. Considering the fact that the robot is hybrid, the active and passive joints of the robot needs to be switched alternatively in each step of climbing. Thus, the stated kinematics and kinetics formulation of the above section can be extended for all of the five steps of a cycle depending on the engaged active joints.

Generally, the complete algorithm of locomotion phase as seen in Fig. 2, can be divided into two main submotions including increasing the altitude (Fig. 2 pictures 1a to 4b) and guiding the robot to the initial configuration in the new height (Fig. 2 pictures 4b to 5) in order to repeat the cycle exactly. The movement of the free limb should be planned in a way that it starts from zero speed and zero acceleration, to move toward the target branch of the truss without passing through any singular point or obstacle and reaches to the target with the final speed and acceleration of zero. So there is no shock at the start and end points of motion. Then this free limb grasps the branch and the step will be iterated for another free limb of the robot with a new path and new active joints until a cycle is completed. Thus, the paths of $x,y,$ and ϕ_z for the end point of the free arm at gripper position ($P_{\cdot man}$) are defined using a polynomial of order five, so that six boundary conditions of the position, velocity, and acceleration can be satisfied simultaneously. Also using this order of polynomial, let us set zero velocity and acceleration for the initial and final points of the robot movement which results in a smooth motion. The path of motion in the Cartesian workspace is as Eq. (7) which is a polynomial function in terms of time for the motions along with x and y-axis and rotation ϕ_Z around z-axis:

$$\mathrm{P}_{ij}(\mathrm{t}) = \mathrm{a}_{1ij}t^5 + \mathrm{a}_{2ij}t^4 + \mathrm{a}_{3ij}t^3 + \mathrm{a}_{4ij}t^2 + \mathrm{a}_{5ij}t + \mathrm{a}_{6ij};\ for\ldots\ i = 1:4, j = x, y, \phi_Z. \tag{7}$$

where index i indicates the step (gait) of motion and index j shows the DOF of the limb in Cartesian coordinate and t is the time. By taking the first and second derivative of the explained path and setting them to zero for start and end of motion we can conclude:

$$\begin{aligned} &\mathrm{P}_{ij}(0) = F_{0ij},\ \mathrm{P}'_{ij}(0) = 0, \Rightarrow \mathrm{a}_{5ij} = 0,\ \mathrm{P}''_{ij}(0) = 0, \Rightarrow \mathrm{a}_{4ij} = 0, \\ &\mathrm{P}_{ij}(1) = F_{1ij}, \Rightarrow \Delta\mathrm{P}_{ij} = F_{1ij} - F_{0ij},\ \mathrm{P}'_{ij}(1) = 0,\ \mathrm{P}''_{ij}(1) = 0. \end{aligned} \tag{8}$$

Thus, by solving the set of simultaneous coupled equations, the unknown coefficients can be extracted:

$$\begin{bmatrix} \mathrm{a}_{1ij} \\ \mathrm{a}_{2ij} \\ \mathrm{a}_{3ij} \end{bmatrix} = \left(\begin{bmatrix} 1 & 1 & 1 \\ 5 & 4 & 3 \\ 20 & 12 & 6 \end{bmatrix} \right)^{-1} \times \begin{bmatrix} \Delta P_{ij} \\ 0 \\ 0 \end{bmatrix}. \tag{9}$$

Now it is enough to determine ΔP_{ij} for each step using geometrical relations to complete the path planning of the system. It should be also considered that the axis of coordinates rotates in each step since the direction of local axis x is always along the line which connects the two supporting grippers that are locked to the truss branches. For example, for the second step, the axis rotates around z equals to 90° and this value is about 143° for the third step. Thus, ΔP_{ij} for the first four steps (preconfiguring

https://doi.org/10.1017/S0263574719001814 Published online by Cambridge University Press

to pull up) can be calculated according to distances between scaffold profiles (Eq. (10)). Each one is a vector of size three including two planar translational movements of the manipulator and one rotational motion, through which the corresponding joint space motion and required motors' torque can also be extracted using the explained kinematics and kinetics of the robot. The final step of locomotion phase which is the fifth step is to pull the whole robot up to its initial home position configuration at the new height after increasing the altitude of the robot.

$$\Delta P_i = \begin{bmatrix} \Delta x_i \\ \Delta y_i \\ \Delta \phi_{Zi} \end{bmatrix}, \overset{i=1:4}{\rightarrow} \Delta P_1 = \begin{bmatrix} 0.2 \\ 0.2 - 0.4.\sin(\frac{\pi}{3}) \\ -a\tan\left(\dfrac{0.2}{0.4 - 0.4.\sin(\frac{\pi}{3})}\right) \end{bmatrix};$$

$$\Delta P_2 = \begin{bmatrix} 0.1 \\ \Delta P_y = 0 \\ \dfrac{\pi}{3} - a\tan\left(\dfrac{0.2}{0.4 - 0.4.\sin(\frac{\pi}{3})}\right) \end{bmatrix}; \Delta P_3 = \begin{bmatrix} 0.1\sin\left(a\tan(\frac{4}{3}) + \dfrac{\pi}{2}\right) \\ 0.1\cos\left(a\tan(\frac{4}{3}) + \dfrac{\pi}{2}\right) \\ \dfrac{\pi}{3} - a\tan\left(2\sin(\frac{\pi}{3}) - 1\right) \end{bmatrix}; \tag{10}$$

$$\Delta P_4 = \begin{bmatrix} -0.2 \\ 0.4\sin(\frac{\pi}{3}) - 0.2 \\ a\tan\left(\dfrac{0.2}{0.4 - 0.4\sin(\frac{\pi}{3})}\right) \end{bmatrix}.$$

It should be noticed that the final replacement introduced above, is according to the joint space of the robot while the first four steps were described according to the workspace of the robot. Note that the paths in Eq. (10) are in the workspace for the robot's end point displacement and rotation of the robot's end point and are defined by ΔP_{ij}; but here for the fifth step, the path is defined in joint space for the parallel part to pull the robot up to its initial configuration. It can be shown that the active joints in the parallel part should have the following movement to restore the robot to its initial configuration with the locked serial part. Actually, the serial free arm's joints should have zero motion in the fifth step:

$$\Delta P_{D.O.F} = \begin{bmatrix} \Delta d_1 \\ \Delta d_2 \\ \Delta \gamma \end{bmatrix} = \begin{bmatrix} \dfrac{-0.1}{\cos\left(a\tan\left(2\sin(\frac{\pi}{3}) - 1\right)\right)} + 0.2, & \dfrac{-0.1}{\cos\left(a\tan\left(2\sin(\frac{\pi}{3}) - 1\right)\right)} + 0.2, \\ \dfrac{\pi}{6} - a\tan\left(\dfrac{1}{2\sin(\frac{\pi}{3}) - 1}\right) \end{bmatrix}^T. \tag{11}$$

Thus, according to the above explanation, the path planning of locomotion phase consisting of increasing the altitude of the robot and returning to the home configuration can be covered through five steps, and this cycle needs to be repeated while the robot reaches to the desired altitude of manipulation. The above-desired displacement of the workspace regulation can be converted easily to the desired joint space displacement of the actuators using kinematics relations of Eq. (4) through which the algorithm of switching the actuators can be planned.

https://doi.org/10.1017/S0263574719001814 Published online by Cambridge University Press

$$\begin{bmatrix} \Delta P_x \\ \Delta P_y \\ \Delta \phi_z \end{bmatrix} = J \begin{bmatrix} \Delta d_1 & \Delta d_2 & \Delta \gamma & \Delta \theta_3 & \Delta d_3 \end{bmatrix}^T . \tag{12}$$

Thus, switching algorithm of active motors covering the mentioned steps for the climbing process of each cycle can be programmed as Eq. (13), and this pattern will be repeated for rest of the cycles. Here indexes *a, b, c, d, e,* and *f* according to Fig. 2 are all of the joints of the robot which can be either passive or active joints in specific motion step of the robot. In other words, according to the above algorithm, the mentioned joints (*a, b, c, d, e*, and *f*) should be switched between active and passive states. Indexes *a, b*, and *c* are related to prismatic joints (jacks), while *d, e*, and *f*, are related to rotary joints (rotary motors).

$$\begin{aligned}
&Switch_Cases(1:5)\\
&Case1; \quad \begin{bmatrix} \dot{q}_a \\ \dot{q}_b \\ \dot{q}_d \\ \dot{q}_f \\ \dot{q}_c \end{bmatrix} = J_1^{-1} \begin{bmatrix} V_{x1} \\ V_{y1} \\ \omega_{z1} \end{bmatrix}; \quad Case2; \quad \begin{bmatrix} \dot{q}_b \\ \dot{q}_c \\ \dot{q}_e \\ \dot{q}_d \\ \dot{q}_a \end{bmatrix} = J_2^{-1} \begin{bmatrix} V_{x2} \\ V_{y2} \\ \omega_{z2} \end{bmatrix}; \quad Case3; \quad \begin{bmatrix} \dot{q}_c \\ \dot{q}_a \\ \dot{q}_f \\ \dot{q}_e \\ \dot{q}_b \end{bmatrix} = J_3^{-1} \begin{bmatrix} V_{x3} \\ V_{y3} \\ \omega_{z3} \end{bmatrix};\\
&Case4; \quad \begin{bmatrix} \dot{q}_a \\ \dot{q}_b \\ \dot{q}_d \\ \dot{q}_f \\ \dot{q}_c \end{bmatrix} = J_4^{-1} \begin{bmatrix} V_{x4} \\ V_{y4} \\ \omega_{z4} \end{bmatrix}; \quad Case5(\text{PullingUp}); \quad \begin{bmatrix} \dot{q}_a \\ \dot{q}_b \\ \dot{q}_d \\ \dot{q}_f \\ \dot{q}_c \end{bmatrix} = J_5^{-1} \begin{bmatrix} V_{x5} \\ V_{y5} \\ \omega_{z5} \end{bmatrix}.
\end{aligned} \tag{13}$$

Then the corresponding required motors' torques are:

$$\begin{aligned}
&\tau_i = D(q_i)\ddot{q}_i + C(q_i, \dot{q}_i)\dot{q}_i + g(q_i);\\
&Switch_cases_i = 1:5;\\
&Case,\ i=1; \quad Case,\ i=2 \quad Case,\ i=3\\
&q_1 = \begin{bmatrix} q_a \\ q_b \\ q_d \\ q_f \\ q_c \end{bmatrix}, \tau_1 = \begin{bmatrix} F_a \\ F_b \\ \tau_d \\ \tau_f \\ F_c \end{bmatrix}; q_2 = \begin{bmatrix} q_b \\ q_c \\ q_d \\ q_e \\ q_a \end{bmatrix}, \tau_2 = \begin{bmatrix} F_b \\ F_c \\ \tau_d \\ \tau_e \\ F_a \end{bmatrix}; q_3 = \begin{bmatrix} q_a \\ q_c \\ q_d \\ q_e \\ q_b \end{bmatrix}, \tau_3 = \begin{bmatrix} F_a \\ F_c \\ \tau_d \\ \tau_e \\ F_b \end{bmatrix};\\
&Case,\ i=4; \quad Case,\ i=5(\text{PullingUp})\\
&q_4 = \begin{bmatrix} q_a \\ q_b \\ q_d \\ q_f \\ q_c \end{bmatrix}, \tau_4 = \begin{bmatrix} F_a \\ F_b \\ \tau_d \\ \tau_f \\ F_c \end{bmatrix}; q_5 = \begin{bmatrix} q_a \\ q_b \\ q_d \\ q_f \\ q_c \end{bmatrix}, \tau_5 = \begin{bmatrix} F_a \\ F_b \\ \tau_d \\ \tau_f \\ F_c \end{bmatrix}.
\end{aligned} \tag{14}$$

where the indexes of τ_i (F_a, F_b, F_c, τ_d, τ_e, and τ_f) shows the required motor torque related to the corresponding joint of the robot according to Fig. 2. It could be observed that for each step of motion, one joint is treated as the passive joint (joint *e* in step one, joint *f* in step two, joint *f* in step three, joint *e* in step four, and joint *e* in step five are treated as passive joints). The overall locomotion algorithm

https://doi.org/10.1017/S0263574719001814 Published online by Cambridge University Press

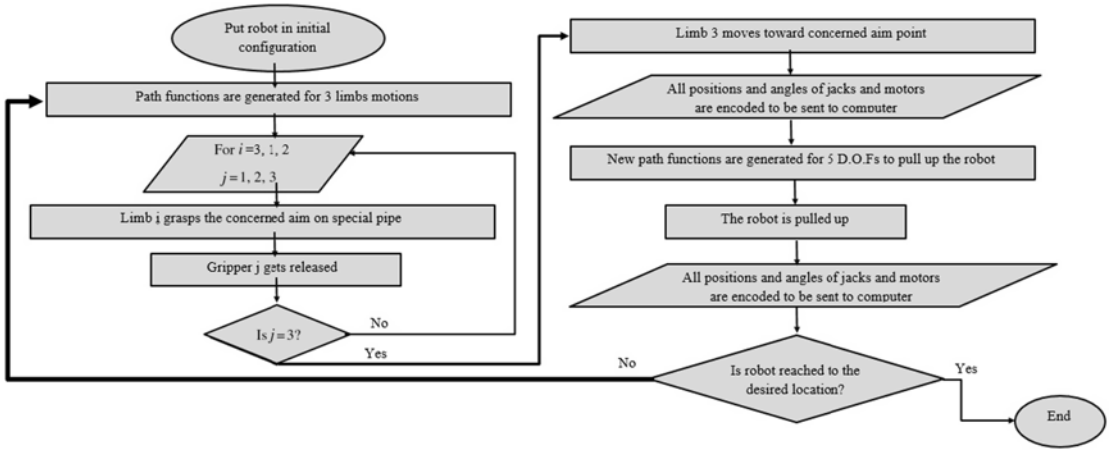

Fig. 4. Overall locomotion algorithm for the climbing phase of the robot.

of the mentioned switching control of the robot for the climbing phase is demonstrated as a flowchart in Fig. 4.

5. Manipulation Phase Control

For the climbing stage, no external manipulating force is implemented on the robot and thus, the external disturbances can be neglected. Therefore, a fast and practical nonlinear controller is preferred for climbing the robot through the branches. In the inverse dynamics, it was seen that the robot can be controlled using CTM with the force vector of Eq. (14). However, for the manipulation process, a proper nonlinear controller is required to neutralize the destructive effect of external force disturbances. Two model-based controlling strategies are employed in this paper to control the manipulation process, namely, simple FBL and adaptive force controller. It will be shown that the latter case is significantly more efficient since the actuators of the robot can be adapted according to the external implemented forces due to the manipulation process. This adaptation to external forces increases the robustness of the system in the presence of disturbances which is ignored in the CTM control.

5.1. Feedback linearization

In order to increase the robustness of the robot during its manipulation phase, global linearization is performed here using FBL by which the movement of the robot can be controlled using the PID method. Here the derivative and proportional gains of the controller K_v and K_p can be defined according to pole placement strategy. The final controlling input according to FBL can be explained as following, where d denotes the desired value of the parameter:[34]

$$\tau = \{D(\mathrm{q})\ddot{q}_d + C(\mathrm{q}, \dot{q})\dot{q} + g(\mathrm{q})\}_{(\text{FeedForward\_ Control})} + \left\{D(\mathrm{q})(-K_v\dot{e} - K_p e)\right\}_{(\text{Feedback\_ Control})}. \quad (15)$$

It can be seen from Eq. (15) that the input control of the FBL method is a combination of feedforward control and feedback one in which the feedforward term is derived from inverse dynamics of the system, while the feedback term is the product of the linear equation of error and inertia matrix of the robot. Thus, one can conclude that the control method is model based and has the advantages of analytic controllers.

5.2. Adaptive control of the robot

Here again, it is considered that the robot has reached the desired altitude through the truss, and it is supposed to conduct a special operation task such as painting, cleaning, welding, etc. In all of the mentioned operational tasks, an external force or torque will be exerted from the environment to the robot's end point such as the torques exerted from welding, riveting, etc. These external forces exerted on the end effector of the robot manipulator which is located on the top of the free arm (the point $P_{.man}$ shown in Fig. 3). The mentioned external forces or torques should be neutralized by the active joints of the robot including the motors and jacks which are the robot's main DOFs.

https://doi.org/10.1017/S0263574719001814 Published online by Cambridge University Press

Thus, the calculated amount of the force or torque of the joints which were previously evaluated through the inverse dynamics in Eq. (5) needs to be improved, so that not only the desired movement of the end effector should be achieved but also the environmental force or torque exerted as a result of the operational task could be supported. Using virtual work principle for the end effector during its manipulation, one can infer the following joint space generalized forces ($\tau_{(5*1)}$) which needs to be added to the CTM controlling signal to overcome the mentioned operational task:

$$\tau_{(5\times1)} = J^T_{(5\times3)}\ F_{(3\times1)};\quad F = \begin{bmatrix} F_x & F_y & M_z \end{bmatrix}^T. \tag{16}$$

where F is the vector of external forces along with the Cartesian coordinates (F_x, F_y) and external torque around z (M_z) and J is the Jacobian matrix of the system. Thus, the final required torques of the motors can be calculated as:

$$\tau = D(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) + J^T F. \tag{17}$$

However, the above-mentioned method works just in cases for which the external forces are exactly predetermined during the time. The fact is that in most cases, just the estimation of these forces are known, and in such cases, employing the above controlling torque results in significant errors or even instability of the system. In the literature, a complete study of different remedies was performed and it was seen that the best solution to bypass the mentioned challenge is designing a proper adaptive control for which the controller is able to adapt the system to external conditions which are the actual implemented external forces (force vector F in Eq. (16)). The adaptation procedure is accomplished by estimating these external forces within a logical period of time that is short enough to prevent the errors from increasing over a definite bound. This solution is proposed and employed in this paper. The employed adaptive force controller is model reference based and its stability is guaranteed by Lyapunov theory. To meet this goal, it is first required to rewrite the dynamic equation of the robot as a multiplication of a known matrix (Y) by an unknown force vector ($\tilde{a}$) for which its initial estimated value is supposed to be improved by the aid of adaptive controller:[35]

$$\tilde{D}(\mathrm{q})\ddot{q}_r + \tilde{C}(\mathrm{q}, \dot{q})\dot{q}_r + \tilde{g}(\mathrm{q}) + J^T(\mathrm{q})\tilde{F} = Y(q, \dot{q}, \dot{q}_r, \ddot{q}_r)\tilde{a}. \tag{18}$$

Here q_r is the virtual reference trajectory of the joints. In the above, $\tilde{a}$ is the unknown parameter vector which affects the robot. Note that here it is supposed that the end point force is undetermined. For the known matrix Y, we introduce $\tau_{FreeEnd}$ as Eq. (19) which declares the input during the time in which no external force acts on the end point of the robot:

$$\tau_{FreeEnd} = \tilde{D}(\mathrm{q})\ddot{q}_r + \tilde{C}(\mathrm{q}, \dot{q})\dot{q}_r + \tilde{g}(\mathrm{q}). \tag{19}$$

Thus, Y and a in Eq. (18) can be presented as:

$$Y(q, \dot{q}, \dot{q}_r, \ddot{q}_r) = \begin{bmatrix} \tau_{FreeEnd(5\times1)} & J^T_{(5\times3)} \end{bmatrix}_{(5\times4)};\quad a = \begin{bmatrix} 1 & F_x & F_y & M_z \end{bmatrix}^T. \tag{20}$$

To control the system, the joint states including their positions and velocities should be controlled simultaneously. q is the actual value of the joint states and q_d is its desired value. Thus, $\tilde{q}$ is the error of this value and the error of its derivation should also converge to zero.

Therefore, a sliding surface (S) can be introduced to decrease the error of both of these parameters simultaneously.

$$S = \dot{\tilde{q}} + \Lambda\tilde{q};\ S = \begin{bmatrix} s_1 & s_2 & s_3 & s_4 & s_5 \end{bmatrix}^T. \tag{21}$$

where Λ is a positive definite matrix and according to the above sliding surface, it is obvious that if $S = 0$, then the error will converge to zero value. Now q_r as the reference trajectory in Eq. (20) can be introduced as below:

$$\dot{q}_r = \dot{q}_d - \Lambda\tilde{q}, \Rightarrow \ddot{q}_r = \ddot{q}_d - \Lambda\dot{\tilde{q}}, \rightarrow S = \dot{\tilde{q}}_r = \dot{q} - \dot{q}_r = \dot{\tilde{q}} + \Lambda\tilde{q}. \tag{22}$$

It can be seen that if $S = 0$, then both errors and their derivatives will converge to zero. In order to guarantee the convergence of S toward zero while providing the stability of the system, the Lyapunov theory is employed. The Lyapunov candidate and its derivative are defined as:

https://doi.org/10.1017/S0263574719001814 Published online by Cambridge University Press

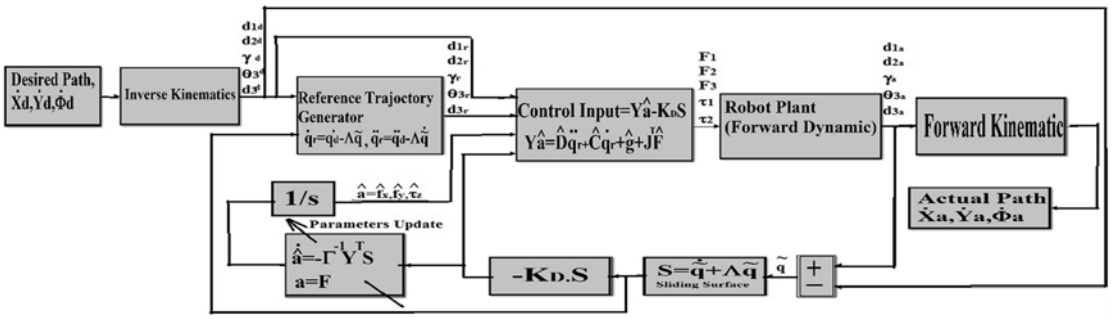

Fig. 5. Overall controlling scheme of the designed climbing robot.

$$V(\mathrm{t}) = \frac{1}{2}S^T DS + \frac{1}{2}\tilde{a}^T \Gamma \tilde{a} \rightarrow \dot{V}(t) = S^T D\dot{S} + S^T \dot{D}S + \tilde{a}^T \Gamma \dot{\tilde{a}},$$
$$\dot{D} - 2C = 0, \rightarrow Skew - Symmetric \rightarrow \dot{V}(t) = S^T D\dot{S} + \tilde{a}^T \Gamma \dot{\tilde{a}}. \tag{23}$$

where Γ is estimation gain which is a positive definite matrix. Derivatives are calculated using the skew-symmetric principal. The derivative of S is as follow:

$$\dot{S} = \ddot{q} - \ddot{q}_r. \tag{24}$$

Substituting the above values in Eq. (19) results in the following controlling torque:

$$\tau_c = \hat{D}(\mathrm{q})\ddot{q}_r + \hat{C}(\mathrm{q}, \dot{q})\dot{q}_r + \hat{g}(\mathrm{q}) + J^T \hat{F} - K_D S = Y\hat{a} - K_D S. \tag{25}$$

In the above controlling signal, the feedback of the sliding surface is also employed to strengthen the stability and robustness of the controller. Moreover, positive definite K_D should be defined in a way that the robustness of the system can be properly provided against the external disturbances. Considering the Lyapunov function, the derivation of this function will be:

$$\dot{V}(\mathrm{t}) = -S^T K_D S + \tilde{a}^T \left(\Gamma \dot{\tilde{a}} + Y^T S\right). \tag{26}$$

We know that the derivation of the Lyapunov function should be negative to ensure the stability of the system. Since the first term of the above function is trivially negative, the negativity of the second term and boundedness of the second derivation of Lyapunov function should be investigated to guarantee the stability of the system according to Barbalat lemma. As mentioned, the variation of the exerted forces on the free arm's end point of the robot is small enough and so it can be concluded that the adaptive algorithm and the estimation of the implemented forces can be extracted by equalizing the second term to zero.

$$\tilde{a}^T \left(\Gamma \dot{\tilde{a}} + Y^T S\right) = 0, \Rightarrow \Gamma \dot{\tilde{a}} + Y^T S = 0; \dot{\tilde{a}} \approx \dot{\tilde{a}}, \Rightarrow \dot{\tilde{a}} = -\Gamma^{-1} Y^T S, \Rightarrow \dot{V}(t) = -S^T K_D S << 0. \tag{27}$$

Therefore, derivation of Lyapunov candidate is negative and it can be proved that the second derivation of that is bounded, so the system stability is guaranteed. It can be seen that the derivation of the force estimation is gained and through time integration, the force itself can be evaluated. The overall controlling flowchart is summarized as Fig. 5.

The desired workspace trajectory of the robot is used in inverse kinematics to achieve its relevant desired joint space trajectory. This path is fed in the designed adaptive controller in the presence of force or parametric uncertainties to calculate the required generalized forces as the control inputs. The estimated force then will be employed as the input of the controller which updates the controller by the algorithm of the adaptation to calculate the required controlling signal and its related gains.

6. Simulation Study

In order to verify the applicability of the designed robot for climbing the trusses and also the efficiency of the proposed adaptive controller toward neutralizing the destructive effect of the disturbances, some analytic and comparative simulations are conducted in MATLAB-Simulink and their

https://doi.org/10.1017/S0263574719001814 Published online by Cambridge University Press

Table I. Geometrical and kinetic parameters of the modeled robot.

Row	Title	Parameter	Value	Unit
1	Cylinder's length	L_1	0.1	m
2	Piston's length	L_2	0.1	m
3	Distance between two fixed grippers	L_e	0.4	m
4	The length of the triangle	B	0.2	m
5	Mass of cylinders and pistons	m_i	2	Kg
6	Inertia of cylinders and pistons	I_i	0.0016	Kg.m^2
7	Mass of triangle component	M_G	4	Kg
8	Inertia of triangle component	I_G	0.003	Kg.m^2

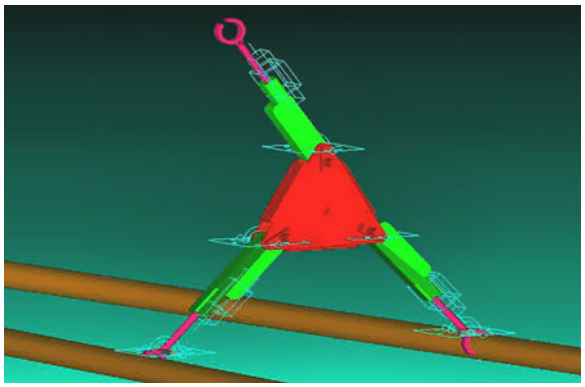

Fig. 6. Climbing robot modeled in MSC-ADAMS.

validity is verified by the aid of MSC-ADAMS simulation. The simulation study is again performed for two phases of the climbing mode and manipulation mode.

6.1. Manipulation phase: Modeling simulation and verification

In order to check the correctness of the modeling and verify its efficiency, kinematics and kinetics of the robot are modeled in MATLAB-Simulink using the parameters of Table I and the corresponding modeling results are compared with MSC-ADAMS, Fig. 6:

Similar rotary and translational joints are modeled in dynamic and kinematics simulator of MSC-ADAMS through which the climbing and manipulation process can be accomplished, considering the required switching between the active and passive joints. Here the friction of the joints is ignored and the robot is modeled in the field of earth gravity. The verification simulation is done for 5 s for the locomotion phase. Thus, two jacks are locked to two of the branches and the corresponding grippers of these two jacks are modeled here as two simple rotary joints. The geometrical and kinetic parameters of the modeled robot are as Table I.

In our previous work,[36] by the same authors the correctness of kinematics and kinetics of the robot in the phase of manipulation were verified by comparing the direct and inverse dynamics results and also comparing the profiles with the results of MSC-ADAMS. It will be seen that good compatibility can be observed through the mentioned comparisons which proves the correctness of modeling. Here the locomotion phase is also modeled and motion planning is performed in the joint space of the robot through which a considerable save of energy is realized at the torque of the joints. Also, the adaptive force controller is designed and implemented on the robot to compensate for the destructive effects of manipulation external forces acting on the end point of the free arm to increase the accuracy of the robot manipulation.

6.2. Control verification

Here it is supposed that the robot is reached to a specific altitude of a structure and is expected to perform an operational task. In this phase, two legs are locked to the branches by two grippers and the third leg can perform the desired manipulation by the aid of the parallel subsection motion. The

https://doi.org/10.1017/S0263574719001814 Published online by Cambridge University Press

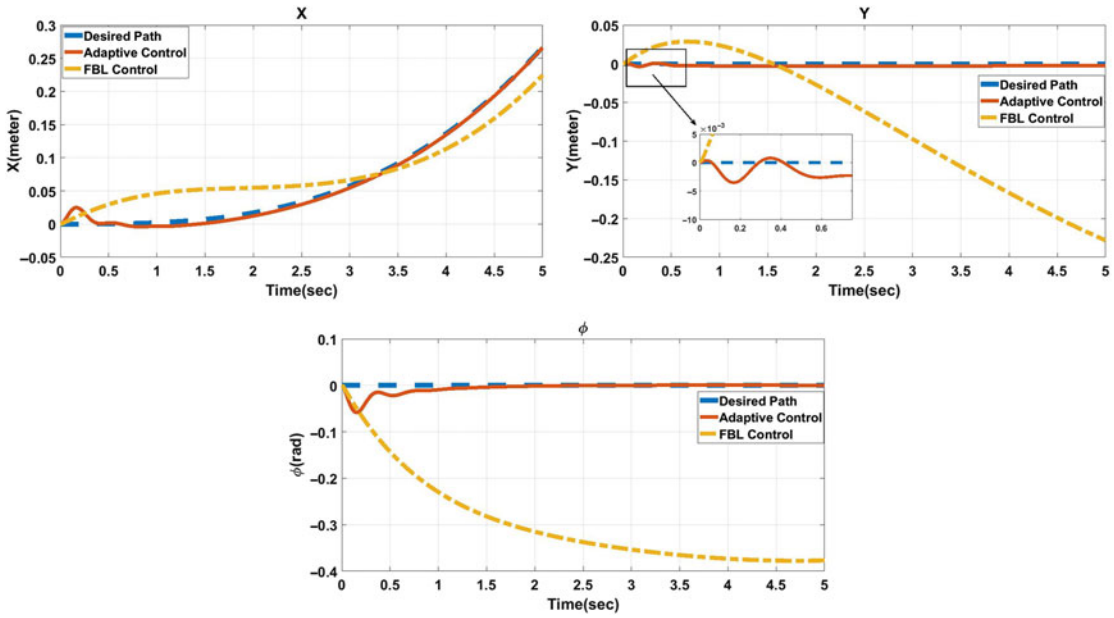

Fig. 7. Comparison of the tracked path of the robot's end point workspace in the presence of disturbance between feedback linearization (FBL) and adaptive controller in manipulating phase.

stability of the robot during the manipulation is controlled here by the aid of the mentioned designed adaptive controller and its results are compared with a simple FBL method. The desired path of the manipulator is a motion of the robot's end point along the X-axis with no rotation of manipulator's arm, so that the free arm does not rotate during the manipulation process. The manipulator path and the initial states of the active joints are as Eq. (28):

$$\begin{aligned} &V_x = 0.0064\mathrm{t}^2(\mathrm{m/s});\ V_y = 0(\mathrm{m/s});\ \omega_z = 0(\mathrm{rad/s}),\\ &\text{initial states:}\\ &\mathrm{d}_1(0) = 0(\mathrm{m});\ \mathrm{d}_2(0) = 0(\mathrm{m});\ \gamma(0) = \tfrac{2\pi}{3}(\mathrm{rad});\ \theta_3(0) = \tfrac{\pi}{2}(\mathrm{rad});\ \mathrm{d}_3(0) = 0(\mathrm{m}). \end{aligned} \tag{28}$$

While the implemented forces and torque on the free arm's end point of the robot are supposed to be as a function of time during the operation, and the amounts of these implemented forces are not exactly predetermined:

$$F_x = 10\sin\left(\frac{t}{3}\right)(\mathrm{N});\ F_y = 20\sin\left(\frac{t}{5}\right)(\mathrm{N});\ M_z = 30(\mathrm{N.M}). \tag{29}$$

The tracked path in the presence of the mentioned implemented forces and its comparison between FBL control and adaptive control is shown as Fig. 7.

As can be seen, great tracking is performed by the adaptive controller rather than the FBL controlling method. This is contributed to the fact that, for the adaptive case, the correct dynamics of the system in the presence of operational forces can be improved by the estimation of external forces. It can be seen in Fig. 8, by using the adaptive controller the external forces along X and Y-axes and the external torque around the Z-axis can be predicted (estimated) correctly after about 1 s. Control inputs are as Fig. 9. As can be seen in Fig. 9 for the adaptive controller, an effort is made to predict the actual operational force which results in 30% increase in the generalized force in some periods of scenario.

6.3. Simulation of climbing phase

Here the proposed robot with the mentioned kinematics and kinetics of Sections 3 and 4 is simulated for the climbing mission with the specifications of Table I. The planned path for the robot in order to climb a truss with a specific pattern is according to Section 4 and thus, the desired path of the robot is based on Eqs. (10) and (11). According to the extracted kinematics formulation, the required joint space movement of the motors and jacks can be calculated as below for a single cycle of a

https://doi.org/10.1017/S0263574719001814 Published online by Cambridge University Press

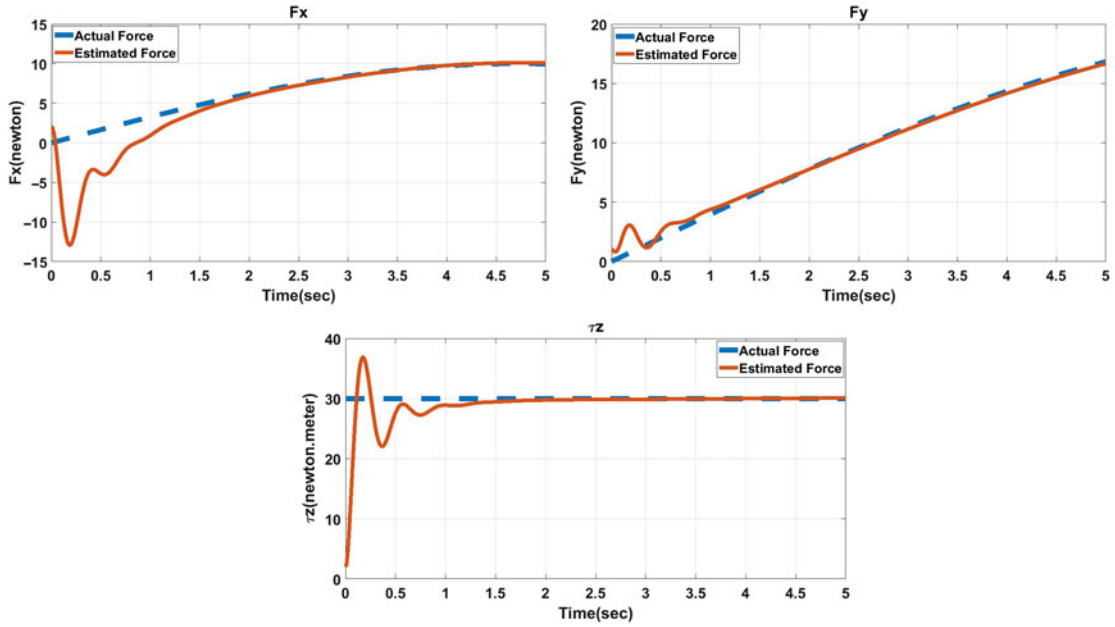

Fig. 8. Estimation of the actual implemented external forces by the aid of an adaptive controller in the manipulating phase.

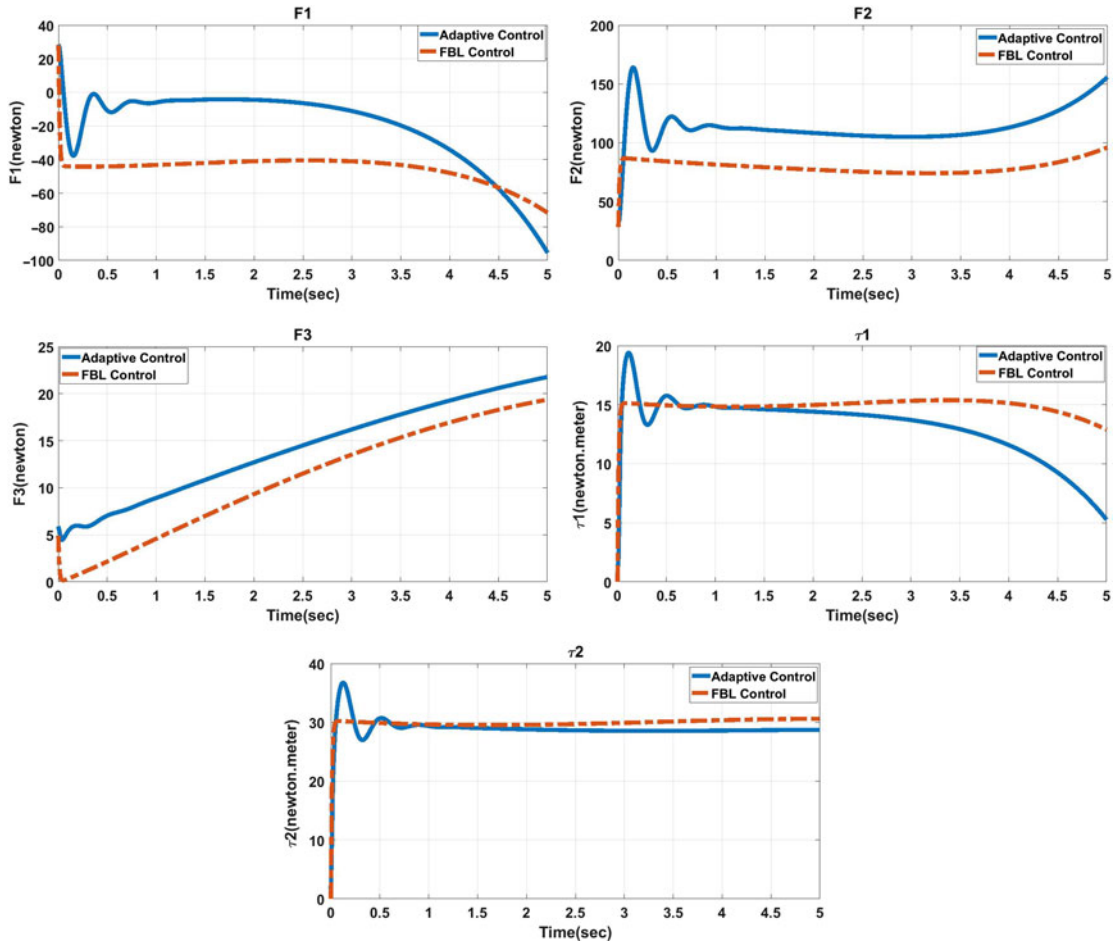

Fig. 9. Controlling input and its comparison between FBL and adaptive controller in manipulating phase.

https://doi.org/10.1017/S0263574719001814 Published online by Cambridge University Press

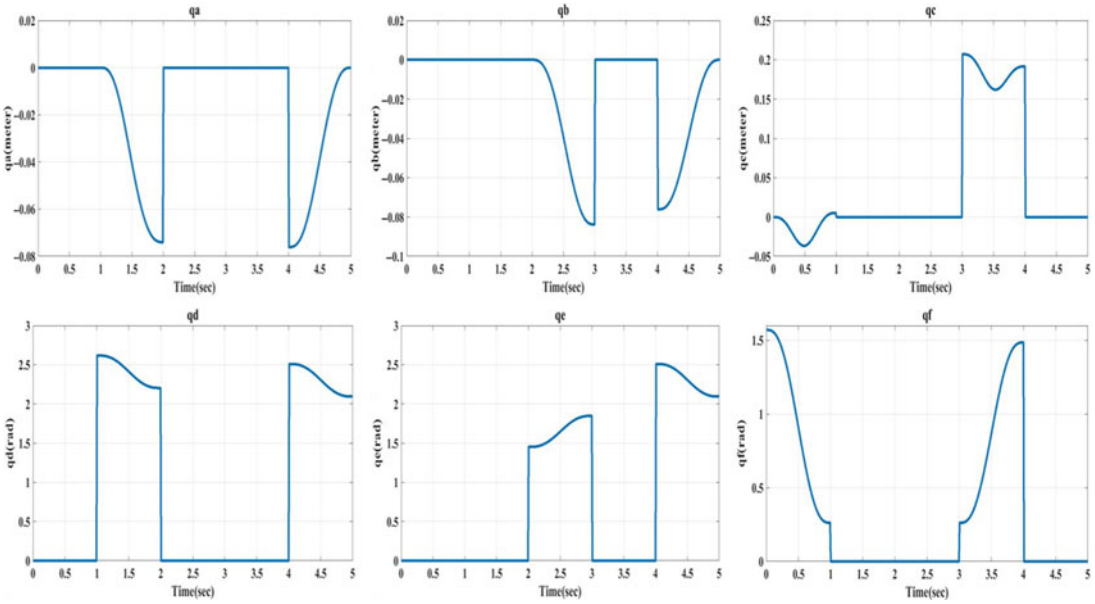

Fig. 10. Required joint space movements of the robot to accomplish the steps (gaits) of a single climbing cycle in locomotion phase.

climbing process consisting of its related five steps. It can be seen that as explained, the locomotion process consists of several steps during which in each step some joints are active, while the rest are passive. In each step, two prismatic joints and one revolute joint located on the triangular basis should be set as the active joints in the parallel submechanism of the robot. Choosing the active revolute joint in parallel submechanism which is the connection between the limb and the robot chassis is up to the designer which can be done according to the desired path and energy optimization considerations. Note that the two joints located on the free arm are treated as the active joints in each step. Hence, the algorithm of switching the active motors needs to be conducted according to the explained path planning procedure of Section 4. The comparison of the required motion of the joint space obtained from inverse kinematics of the robot which was studied for manipulation phase between the MATLAB and ADAMS is also valid for climbing phase since the same formulation with a specified switching pattern is employed here. It can be seen that as expected a specific pattern is iterated harmonically during five steps since the branches of the truss are supposed to be passed with a unique pattern.

Figure 10 shows the joint space paths for the climbing phase. The motion of the robot is according to Fig. 2. During the steps 1–4, the free arm moves toward the new branch while the robot chassis will be pulled up to get its initial configuration at step five.

Now it is desired to evaluate the required motors' torques and jacks' forces of the robot to realize the mentioned path. Since the active joints switch alternatively in each step, according to Eq. (14), the actuators work and stop frequently, and this pattern will be repeated within five steps for a single cycle. The required controlling input for each actuator during a complete cycle of climbing phase and its comparison with MSC-ADAMS is shown as Fig. 11.

Again here the validity of dynamic modeling of the climbing phase of the robot is proved since the result profiles of MATLAB and ADMAS are the same with good accuracy. Also, the switching process of the climbing phase can be again observed here, in which in each step, always one joint torque is zero. According to these simulations, it can be seen that the designed robot can successfully cover the required maneuver for climbing through a predetermined pattern in a truss-based structure.

In Figs. 10 and 11 motions and generalized forces for joints *a, b, c, d, e*, and *f* for complete 5-s cycle locomotion are obtained which are shown in Fig. 2. As can be seen, each step takes 1 s and thus, the overall cycle motion can be accomplished within 5 s. It should be noticed that during some steps, some actuator's torques are zero which means that in these steps the mentioned motors are passive and act just like a free joint with no incoming force. Alternatively, the joints with positive or negative generalized forces are active joints. As mentioned, choosing the active joints in each step

https://doi.org/10.1017/S0263574719001814 Published online by Cambridge University Press

Table II. The maximum torques and forces related to joints a:f for each step.

	Step one (0–1 s)	**Step two (1–2 s)**	**Step three (1–2 s)**	**Step four (3–4 s)**	**Step five (4–5 s)**	**Manipulation (CTM)**
F_a	37 (N)	−4.28 (N)	−4.28 (N)	51 (N)	82.3 (N)	29 (N)
F_b	49.3 (N)	51.37 (N)	51.37 (N)	76.5 (N)	81 (N)	63.5 (N)
F_c	5 (N)	−31.78 (N)	−31.78 (N)	4.91 (N)	5 (N)	4.94 (N)
τ_d	3.5 (N.m)	−1.94 (N.m)	−1.94 (N.m)	4 (N.m)	1.24 (N.m)	−6.4 (N.m)
τ_e	0 (N.m)	−4.73 (N.m)	−4.73 (N.m)	0 (N.m)	0 (N.m)	0 (N.m)
τ_f	7 (N.m)	0 (N.m)	0 (N.m)	7.86 (N.m)	−0.8 (N.m)	0.069 (N.m)

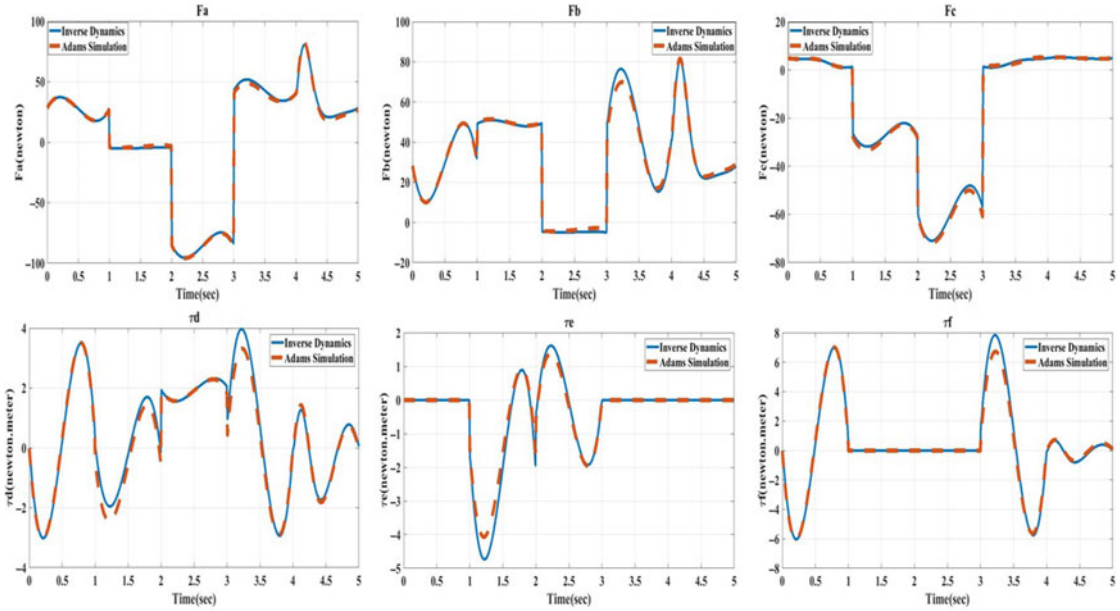

Fig. 11. The required controlling input for each actuator during a complete cycle of climbing phase and its comparison with ADAMS in locomotion phase.

is up to the designer, and this selection can be performed subject to the optimization of the desired objective function which leads in an optimal path planning. This phenomenon can be observed in Fig. 11. In each step, the maximum torque or force for all joints are shown in Table II, and for the passive joint, zero torque is attributed.

Note that the values related to the manipulation phase are contributed to the simulation with no engaged disturbance. It can be seen here that the critical joints of the robot are jack (a) and motor (f), which should be considered as the criteria of motor selection with a good safety factor in order to prevent the climbing failure.

7. Conclusion and Discussion

In this paper, a grip-based climbing robot was designed and proposed which can climb through the trusses and infrastructures and decrease the risk of operation by a human. The mechanism of the robot was designed and its mobility and maneuverability were optimized. The robot is a hybrid mechanism consisting of a triangular shape chassis and its locomotion is controlled by the aid of totally three rotational motors and three prismatic jacks. Considering the redundancy of the robot and the existence of parallel submechanism, the optimum active joints were selected which needs to be switched alternatively for covering the planned path of locomotion. As a result, the robot is able to work in two phases of locomotion and manipulation. Related kinematic and kinetic model of the robot were extracted for both of two mentioned phases through which controlling procedure of the robot can be developed. It was explained that in locomotion phase the robot is supposed to reach a definite altitude by a proper pattern of movement within the truss branches. Thus, for this phase, the same modeling is

https://doi.org/10.1017/S0263574719001814 Published online by Cambridge University Press

valid while path planning needs to be performed to guide the robot to the operating point. It was seen that the planned path is able to do the explained mission with an iterating scenario including five steps in a single cycle, while repeating the mentioned steps provides a higher climbing level by about 0.1 m. After reaching the operating point, the robot is supposed to perform a specific operational task. It was explained that since the robot is supposed to perform an operational task and the external implemented force on the end effector cannot be determined exactly, uncertainty needs to be considered for dynamic matrices of the CTM controlling signal. Thus, an adaptive controller was designed and implemented on the robot to neutralize the effect of the mentioned uncertainties and realize the force and position accuracy at the same time. The results were compared with a simple FBL controller. The correctness of modeling, efficiency of the designed robot toward climbing the trusses, and also the robustness of the designed controller for realizing the mentioned climbing and operational task were verified by conducting some comparative and analytic simulations in MATLAB. The correctness of modeling and simulations were also proved by comparing the results of MATLAB-Simulink with MSC-ADAMS and observing the acceptable compatibility of the kinematic and kinetic results with more than 80% accuracy. However, it was seen that since the robot is among the climbing ones, a high amount of kinetic force is required to overcome gravity. Thus, the optimum active joints were explored to optimize energy consumption since the proposed robot is redundant. It can be seen that selecting the proper active joints can decrease the force effort. Moreover, it was seen that the designed adaptive controller can estimate the actual implemented operational forces and increase the required improving force of the joints by about 30% within 1 s in order to prevent the robot from instability which is not possible using a simple FBL method. Finally, it was shown that iterating specified steps of manipulative movement with a proper path planning and suitable switching process can cover the climbing procedure or locomotion phase during a cycle. Therefore, it can be concluded that the designed robot together with the proposed controller is able to provide a climbing procedure by which any operational task can be achieved through the trusses and infrastructures with high accuracy.

Acknowledgment

The authors would like to thank Iran National Science Foundation for his financial supports to the research project No. 96005846.

Conflict of interest

The authors declare that they have no conflict of interest.

References

1. C. Balaguer, A. Gimenez and A. Jardón, "Climbing robots' mobility for inspection and maintenance of 3D complex environments," *Auton. Robots.* **18**(2), 157–169 (2005).
2. Y. Guan, H. Zhu, W. Wu, X. Zhou, L. Jiang, C. Cai, L. Zhang and H. Zhang, "A modular biped wall-climbing robot with high mobility and manipulating function," *IEEE/ASME Trans. Mech.* **18**(6) 1787–1798 (2013).
3. M. Abderrahim, C. Balaguer, A. Giménez, J. M. Pastor and V. Padron, "ROMA: A Climbing Robot for Inspection Operations," *Proceedings* of the *1999 IEEE International Conference on Robotics and Automation*, Michigan, USA (1999) pp. 2303–2308.
4. M. Tavakoli, A. Marjovi, L. Marques and A. T. de Almeida, "3DCLIMBER: A Climbing Robot for Inspection of 3D Human Made Structures," *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2008*, Nice, France (2008) pp. 4130–4135.
5. S.-K. Yun and D. Rus, "Optimal self assembly of modular manipulators with active and passive modules," *Auton. Robots* **31**(2–3), 183 (2011).
6. G. C. Haynes, A. Khripin, G. Lynch, J. Amory, A. Saunders, A. A. Rizzi and D. E. Koditschek, "Rapid Pole Climbing with a Quadrupedal Robot," *IEEE International Conference on Robotics and Automation, ICRA 2009*, Kobe, Japan (2009) pp. 2767–2772.
7. M. Almonacid, R. J. Saltaren, R. Aracil and O. Reinoso, "Motion planning of a climbing parallel robot," *IEEE Trans. Robot. Autom.* **19**(3), 485–489 (2003).
8. J.-C. Fauroux and J. Morillon, "Design of a Climbing Robot for Cylindro-Conic Poles Based on Rolling Self-Locking," *Mobile Robotics: Solutions and Challenges,* World Scientific (2010) pp. 447–454.
9. Y. Guan, L. Jiang, H. Zhu, W. Wu, X. Zhou, H. Zhang and X. Zhang, "Climbot: A bio-inspired modular biped climbing robot—system development, climbing gaits, and experiments," *J. Mech. Robot.* **8**(2) (2016) 021026.
10. M. Tavakoli, M. R. Zakerzadeh, G. Vossoughi and S. Bagheri, "A hybrid pole climbing and manipulating robot with minimum DOFs for construction and service applications," *Indust. Robot: Int. J.* **32**(2), 171–178 (2005).

11. A. Baghani, M. N. Ahmadabadi and A. Harati, "Kinematics Modeling of a Wheel-Based Pole Climbing Robot (UT-PCR)," *Proceedings of the 2005 IEEE International Conference on Robotics and Automation, ICRA 2005*, Barcelona, Spain (2005) pp. 2099–2104.
12. A. Bhole, S. H. Turlapati, V. Rajashekhar, J. Dixit, S. V. Shah and K. M. Krishna, "Design of a robust stair-climbing compliant modular robot to tackle overhang on stairs," *Robotica* **37**(3), 428–444 (2019).
13. C. Sunada, D. Argaez, S. Dubowsky and C. Mavroidis, "A Coordinated Jacobian Transpose Control for Mobile Multi-Limbed Robotic Systems," *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, San Diego, CA, USA (1994) pp. 1910–1915.
14. D. M. Bevly, S. Farritor and S. Dubowsky, "Action module planning and its application to an experimental climbing robot," *Proceedings* of the *IEEE International Conference on Robotics and Automation, ICRA 2000*, San Francisco, CA, USA (2000) pp. 4009–4014.
15. M. Korayem, A. Shafei and H. Shafei, "Dynamic modeling of nonholonomic wheeled mobile manipulators with elastic joints using recursive Gibbs–Appell formulation," *Scientia Iranica* **19**(4), 1092–1104 (2012).
16. L. Beji and M. Pascal, "The kinematics and the full minimal dynamic model of a 6-DOF parallel robot manipulator," *Nonlin. Dynam.* **18**(4), 339–356 (1999).
17. Y. Li and S. Staicu, "Inverse dynamics of a 3-PRC parallel kinematic machine," *Nonlin. Dynam.* **67**(2), 1031–1041 (2012).
18. M. H. Abedinnasab and G. Vossoughi, "Analysis of a 6-DOF redundantly actuated 4-legged parallel mechanism," *Nonlin. Dynam.* **58**(4), 611 (2009).
19. M. H. Korayem, A. Nikoobin and V. Azimirad, "Maximum load carrying capacity of mobile manipulators: Optimal control approach," *Robotica* **27**(1), 147–159 (2009).
20. M. Korayem and A. Nikoobin, "Maximum payload path planning for redundant manipulator using indirect solution of optimal control problem," *Int. J. Adv. Manuf. Technol.* **44**(7–8), 725 (2009).
21. M. Korayem, M. Irani and S. R. Nekoo, "Load maximization of flexible joint mechanical manipulator using nonlinear optimal controller," *Acta Astronautica* **69**(7–8), 458–469 (2011).
22. M. H. Korayem, H. R. Nohooji and A. Nikoobin, "Path planning of mobile elastic robotic arms by indirect approach of optimal control," *Int. J. Adv. Robot. Syst.* **8**(1), 10 (2011).
23. C. G. Atkeson, C. H. An and J. M. Hollerbach, "Estimation of inertial parameters of manipulator loads and links," *Int. J. Robot. Res.* **5**(3), 101–119 (1986).
24. J.-J. E. Slotine and W. Li, "On the adaptive control of robot manipulators," *Int. J. Robot. Res.* **6**(3), 49–59 (1987).
25. W.-W. Shang, S. Cong and Y. Ge, "Adaptive computed torque control for a parallel manipulator with redundant actuation," *Robotica* **30**(3), 457–466 (2012).
26. T.-C. Yang, J. C. Yang, and P. Kudva, "Load-adaptive control of a single-link flexible manipulator," *IEEE Trans. Syst. Man Cybern.* **22**(1), 85–91 (1992).
27. R. Hayat, M. Leibold and M. Buss, "Addressing control implementation issues in robotic systems using adaptive control," *Robotica* **38**(1), 171–184 (2020).
28. S. Dubowsky, C. Sunada and C. Mavroidis, "Coordinated motion and force control of multi-limbed robotic systems," *Auton. Robots.* **6**(1), 7–20 (1999).
29. D. Bevly, S. Dubowsky and C. Mavroidis, "A simplified Cartesian-computed torque controller for highly geared systems and its application to an experimental climbing robot," *J. Dyn. Syst. Measur. Control*, **122**(1), 27–32 (2000).
30. S. Aoi and K. Tsuchiya, "Locomotion control of a biped robot using nonlinear oscillators," *Auton. Robots* **19**(3), 219–232 (2005).
31. S. Dong, "Gravity and inertial load adaptive control of wall-climbing robot," *J. Eng.* **2019**(13), 442–446 (2019).
32. J.-S. Moon and J. Bae, "Gait optimization and energetics of ballistic walking for an underactuated biped with knees," *Nonlin. Dynam.* **85**(3), 1533–1546 (2016).
33. J.-P. Merlet, *Parallel Robots* (Springer Science & Business Media, 2006).
34. R. M. Murray, *A Mathematical Introduction to Robotic Manipulation* (CRC Press, 2017).
35. J.-J.E. Slotine and W. Li, *Applied Nonlinear Control* (Prentice Hall, Englewood Cliffs, NJ, 1991).
36. V. Boomeri, S. Pourebrahim and H. Tourajizadeh, "Kinematic and Dynamic Modeling of an Infrastructure Hybrid Climbing Robot, " *2017 IEEE 4th International Conference on Knowledge-Based Engineering and Innovation (KBEI)*, Tehran, Iran (2017) pp. 0834–0842.