

Xingbo Wang
Xiaotao Wang
Zhongpeng Zhang

<https://doi.org/10.21278/TOF.42205>
ISSN 1333-1124
eISSN 1849-1391

DYNAMICAL MODELLING AND A DECENTRALIZED ADAPTIVE CONTROLLER FOR A 12-TETRAHEDRAL ROLLING ROBOT

Summary

The 12-tetrahedral robot is an addressable reconfigurable technology (ART)-based variable geometry truss mechanism with twenty-six extensible struts and nine nodes arranged in a tetrahedral mesh. The robot has the capability of reconfiguring shape and dimension for environment sensing requirements, which makes it suitable for space exploration and environmental perception. In this paper, we have derived a dynamics model and presented a decentralized adaptive controller for a 12-tetrahedral robot. First, the robot is divided into the node and the strut subsystems, and the kinetic and the potential energy are calculated for the two subsystems. Then, the dynamics model is achieved by applying the Lagrangian formalism on the total energy of the robot. Since the dynamics is too complicated for implementing model-based controllers, a two-layer controller is presented to control the robot, in which the planning layer determines gait and trajectory of the robot, and the executive layer adopts the decentralized adaptive control strategy and consists of twenty-six strut controllers. Each strut controller regulates the movement of the corresponding strut without information exchange with other struts. Co-simulations based on ADAMS and Matlab have been conducted to verify the feasibility and effectiveness of the proposed controller.

Key words: 12-tetrahedral robot; environment sensing; dynamical modelling; decentralized adaptive controller

1. Introduction

For the purpose of exploring complex terrains on the Earth, Mars and the Moon, researchers have developed a great variety of mobile robots. Wheeled robots and legged robots are the most common autonomous mechanisms used for terrestrial exploration [1]. In general, wheeled mobile robots can traverse relatively flat terrain [2], but they move over rough terrain with difficulty. Legged robots can walk autonomously on uneven and heterogeneous terrain, but they may be instable when moving fast [3]. Crawler robots can traverse irregular environment, while they are often cumbersome and have poor flexibility [4]. Spherical robots take symmetric structures which allow them to move omnidirectionally. They can traverse rough terrain but may be instable when passing sloped terrain [5].

Several approaches have been proposed to improve the robot's capability of traversing various terrains. One such approach is to combine legged, wheeled, crawler and other types of robots into a hybrid autonomous robot, e.g., a wheeled-legged robot [6]. Another approach is to develop modular robots which can change their configuration according to the task requirements, such as reconfigurable robots and metamorphic robots. Reconfigurable robots built from structural modules can reconfigure themselves to autonomously attain different configurations for different tasks and environments but they are constrained by the dimensions of the basic building blocks. Metamorphic robots, based on the metamorphic theory, can adapt their configurations and degrees of freedom to different functional requirements [7].

The above mentioned mobile robots can overcome general obstacles but they walk on rough terrain or uneven ground, such as rough slopes, large gullies and narrow caves, with difficulty. The tetrahedral robot, which is an addressable reconfigurable technology (ART)-based variable geometry truss mechanism, consists of nodes and extensible struts arranged in a tetrahedral mesh. The robot can reconfigure the dimension and the shape, which makes it very suitable for exploration in irregular and complex environments in space.

Since 2004 three prototypes of the tetrahedral robot have been proposed and designed by NASA [8]. The first prototype was 1-tetrahedral robot consisting of four nodes and six struts with an extension ratio of about 2:1. The robot can move the centre of gravity and achieve a tumble through the coordination of all struts. The second generation was a 12-tetrahedral robot with twenty-six telescoping strut featured nested square tubes, within which pulleys were embedded. These pulleys were driven by an electric motor with a planetary gearhead and a worm gear. The robot is too cumbersome for the struts to support its motion. The third generation was also a 12-tetrahedral robot, whose struts were made of a new material and it had nested screws within the exoskeleton, which greatly reduced the weight of the robot. The struts can achieve a 5:1 extension ratio with the expectation of climbing a 40 degree slope.

Although several prototypes have been developed, the progress in modelling and control of the 12-tetrahedral robot, particularly in dynamic modelling, is slowing down. This is mainly because of the complex and highly nonlinear kinematics and dynamics associated with this class of robots. In [9], the authors analysed the kinematics of different motion phases of a 1-tetrahedral robot and presented the rolling critical condition. In [10], the authors proposed the direct and inverse kinematics for the 12-tetrahedral robot and presented a motion planning method for a robot based on the proposed kinematics. In [11-13], the authors planned gaits for 4-, 6-, and 8-tetrahedral robots based on a simulating walker model. In [14], the authors developed a gait planning method for a 12-tetrahedral robot, and proposed three kinds of gait planning procedures using simplified geometrical computation. In [15, 16], the authors proposed a motion planning strategy and the toppling condition for a single robot based on its kinematics.

In this paper, we first derived a dynamical model using Lagrangian formalism for a 12-tetrahedral robot. The robot can be divided into a node and strut subsystems, and the kinetic and potential energy are calculated for the two individual subsystems. Then, the dynamics is achieved by applying the Lagrange approach to the total energy of the robot system. Since the dynamics of the robot is a complex, nonlinear, and multi-input-multi-output system, we present a controlling system consisting of a planning layer in which a main controller plans the gait and the trajectory of the robot, and an executive layer which adopts the decentralized adaptive strategy and consists of twenty-six strut controllers. The decentralized adaptive strut controller regulates the corresponding strut based on the commands of the main controller and the measurements of that strut without information exchange with other strut controllers. Lastly, simulations based on ADAMS and Matlab were conducted to verify the performance of the controller for the 12-tetrahedral robot.

The remainder of the paper is organized as follows. In Section 2 we present the model and the inverse kinematics model of the 12-tetrahedral robot. In this section we also derive the Jacobian matrix for the robot. In Section 3 we derive a dynamic model of the robot using Lagrangian formulism. Then, the decentralized adaptive strategy is adopted to implement the control system in the robot in Section 4. Simulations are conducted in Section 5. Finally, conclusion is given in Section 6.

2. Mechanism and kinematics of the 12-tetrahedral robot

The 12-tetrahedral robot is a redundant parallel manipulator with nine nodes and twenty-six struts. In this section, we provide a brief overview of the mechanisms, the kinematics and the Jacobian upon which the dynamic model of the robot is based. In the following sections, we first describe the mechanism of the robot, then present the inverse kinematics and finally derive the robot Jacobian.

2.1 Mechanism of the 12-tetrahedral robot

The 12-tetrahedral robot is initialized to be a cube, with one strut along each edge of the cube, one diagonal strut along each facet, and eight struts from the central node to other external nodes, as shown in Fig. 1. Each strut is a nested-screw structure that has three segments, with two initial lengths of about 213 cm. The strut can contract or extend through appropriate actuation. The total length of the strut when fully extended is approximately 1,055 cm, which gives the strut a 5:1 extension ratio. All of the struts can collaborate to drive the robot rolling or walking. In this robot, the central node accommodates payloads, such as telemetry and communication modules, and the external nodes serve as landing points and accommodate sensors.

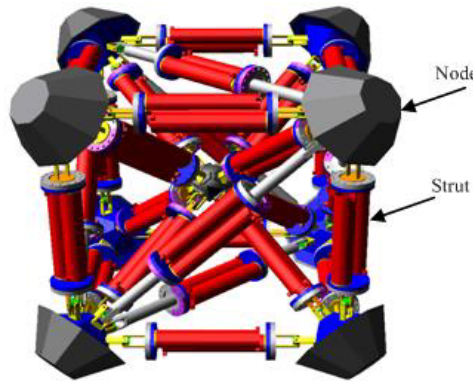


Fig. 1 12-tetrahedral rolling robot

2.2 Kinematics of the 12-tetrahedral robot

To facilitate the kinematic analysis of the 12-tetrahedral robot, a reduced model is given in Fig. 2, where N_0 is the central node, and $N_i (i=1, \dots, 8)$ the external nodes. A set of frames is attached to the model. Frame B , located at the centre of N_1 , denotes the reference frame, and frame $M_i (i=0, 1, \dots, 8)$, fixed at the centre of N_i , denotes a local frame. Let $p_{ij} = [x_{ij} \ y_{ij} \ z_{ij}]^T$ be the position of the hinge which is fixed at N_i and connected with N_j through a strut, expressed in frame M_i , and $P_{ij} = [X_{ij} \ Y_{ij} \ Z_{ij}]^T$ be its position expressed in frame B . Since each hinge remains unchanged relative to its fixed node, p_{ij} is a constant vector. The rotation of M_i with respect to B is given as follows:

$${}_{M_i}^B \mathbf{R}_{XYZ} = \mathbf{R}_Z(\gamma_i) \mathbf{R}_Y(\beta_i) \mathbf{R}_X(\alpha_i) = \begin{bmatrix} c\beta_i c\gamma_i & s\alpha_i s\beta_i c\gamma_i - c\alpha_i s\gamma_i & c\alpha_i s\beta_i c\gamma_i + s\alpha_i s\gamma_i \\ c\beta_i s\gamma_i & s\alpha_i s\beta_i s\gamma_i + c\alpha_i c\gamma_i & c\alpha_i s\beta_i s\gamma_i - s\alpha_i c\gamma_i \\ -s\beta_i & c\beta_i s\alpha_i & c\alpha_i c\beta_i \end{bmatrix} \quad (1)$$

where $s\theta := \sin \theta$ and $c\theta := \cos \theta$, $R_x(\theta)$, $R_y(\theta)$ and $R_z(\theta)$ are the elementary rotations of frame M_i about the x -, y - and z -axes of frame B through an angle θ . $(\alpha_i \ \beta_i \ \gamma_i)$ are the X - Y - Z fixed angles of frame M_i relative to frame B , where α_i is the yaw rotation about the x -axis of frame B , β_i is the pitch rotation about the y -axis, and γ_i is the roll rotation about the z -axis.

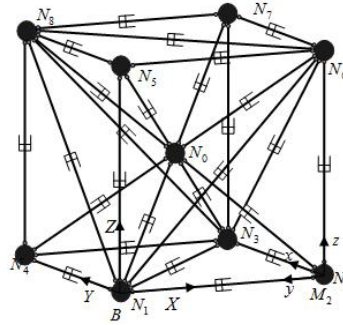


Fig. 2 A simplified model of 12-tetrahedral robot and definition of coordinate frames

Let $P_i = [x_i \ y_i \ z_i]^T$ be the position of the original frame M_i , then the pose of N_i with respect to B can be represented as $q_i = [x_i \ y_i \ z_i \ \alpha_i \ \beta_i \ \gamma_i]^T$. Given P_i and P_j , the strut from N_i to N_j can be represented as a vector

$$L_{ij} = P_j - P_i = {}_{M_j}^B R p_{ji} + P_j - {}_{M_i}^B R p_{ij} - P_i \quad (2)$$

with the length

$$l_{ij} = \sqrt{(X_{ji} - X_{ij})^2 + (Y_{ji} - Y_{ij})^2 + (Z_{ji} - Z_{ij})^2} \quad (3)$$

where i and $j (i < j)$ are the indices of N_i and N_j , respectively. Therefore, twenty-six explicit functions are needed to formulate the relations between the strut lengths and the node positions.

2.3 Jacobian calculation

The Jacobian reveals the relation between the strut velocities and the node velocities, and the relation between the forces on the nodes and the torque exerted by the struts. Let Q be the vector of all node poses, $Q = [q_0^T \ q_1^T \ \dots \ q_8^T]^T$, and L be the vector of all strut lengths, $L = [l_{01} \ l_{02} \ \dots \ l_{78}]^T$. Taking the time derivative of equations (2) and (3) yields the Jacobian

$$\dot{L} = J_1 J_2 \dot{Q} \quad (4)$$

where J_1 is a 26×156 matrix,

$$J_1 = \begin{bmatrix} -u_{01}^T & u_{01}^T & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & -u_{78}^T & u_{78}^T \end{bmatrix}_{26 \times 156} \quad (5)$$

with u_{ij} the unit vector defined along L_{ij} , $u_{ij} = \frac{L_{ij}}{l_{ij}} = u_{ji} (i < j)$, and J_2 is a 156×54 matrix,

$$J_2 = \begin{bmatrix} T_{01} & 0 & \cdots & 0 & 0 \\ 0 & T_{10} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & T_{78} & 0 \\ 0 & 0 & \cdots & 0 & T_{87} \end{bmatrix}_{26 \times 156} \quad (6)$$

with $T_{ij} = [I_{3 \times 3} \quad R_Z(\gamma)R_Y(X)S(e_x)R_X(\alpha)p_{ij} \quad R_Z(\gamma)S(Y)R_Y(\beta)R_X(\alpha)p_{ij} \quad S(Z)R_Z(\gamma)R_Y(\beta)R_X(\alpha)p_{ij}]_{3 \times 6}$. In the above equation, $I_{3 \times 3}$ is a 3×3 identity matrix, and $S(s)$ is the skew-symmetric matrix to the vector $s = [s_x \quad s_y \quad s_z]^T$, given by

$$S(s) = \begin{bmatrix} 0 & -s_z & s_y \\ s_z & 0 & -s_x \\ -s_y & s_x & 0 \end{bmatrix}. \quad (7)$$

X , Y and Z are the unit vectors in frame B .

Therefore, the Jacobian $J = J_1 J_2$ is a 26×54 matrix, which gives the relation between the extending or contracting velocities of the struts and the velocities of the nodes.

3. Dynamical modelling of the 12-tetrahedral robot

The dynamic model of the 12-tetrahedral robot with multiple nodes and struts is derived by using the Lagrange equation. First, the robot is divided into the node subsystem and the strut subsystem, and the kinetic and the potential energy is derived for the two individual subsystems. Lastly, the dynamic model is achieved by applying the Lagrangian formalism to the total energy equation of the robot.

3.1 Kinetic and potential energy of the nodes

Kinetic energy K_{ni} of the node N_i can be divided into translational energy, K_{mi} , and rotational energy, K_{ri} . K_{mi} is the kinetic energy of the linear velocity of N_i ,

$$K_{mi} = \frac{1}{2} m_{ni} (\dot{X}_i^2 + \dot{Y}_i^2 + \dot{Z}_i^2) \quad (8)$$

where m_{ni} is the mass of N_i , and $(\dot{X}_i, \dot{Y}_i, \dot{Z}_i)$ is the linear velocity. K_{ri} is the kinetic energy of N_i due to the rotation around its centre of mass,

$$K_{ri} = \frac{1}{2} \Omega_{mni}^T I_{mni} \Omega_{mni} \quad (9)$$

where I_{mni} is the inertia tensor of N_i in frame M_i , $\Omega_{mni} = R^T \Omega_{bni}$ is the angular velocity of N_i in frame M_i . In the above equation, $\Omega_{bni} = [\Omega_x \quad \Omega_y \quad \Omega_z]^T$ is the angular velocity of N_i expressed in frame B [17],

$$\Omega_{bni} = E_{XYZ} \dot{\Theta}_{XYZ} = \begin{bmatrix} c\beta_i c\gamma_i & -s\gamma_i & 0 \\ c\beta_i s\gamma_i & c\gamma_i & 0 \\ -s\beta_i & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\alpha}_i \\ \dot{\beta}_i \\ \dot{\gamma}_i \end{bmatrix} \quad (10)$$

where E_{XYZ} depends on the frame in which the orientation is represented. Substituting (10) into (9) gives the rotational energy of N_i as follows:

$$K_{ni} = \frac{1}{2} \left(R^T E_{XYZ} \dot{\Theta}_{XYZ} I \right)^T I_{mni} R^T E_{XYZ} \dot{\Theta}_{XYZ} = \frac{1}{2} \begin{bmatrix} \dot{\alpha}_i & \dot{\beta}_i & \dot{\gamma}_i \end{bmatrix} M_{mi} \begin{bmatrix} \dot{\alpha}_i \\ \dot{\beta}_i \\ \dot{\gamma}_i \end{bmatrix} \quad (11)$$

Thus, the total kinetic energy of N_i is represented as follows:

$$K_{ni} = \frac{1}{2} \dot{q}_i^T M_{ni} \dot{q}_i = \frac{1}{2} \dot{q}_i^T \begin{bmatrix} M_{mi} & \mathbf{0} \\ \mathbf{0} & M_{mi} \end{bmatrix} \dot{q}_i \quad (12)$$

where $M_{mi} = m_{ni} I_{3 \times 3}$ and $M_{mi} = E_{XYZ}^T R I_{mni} R^T E_{XYZ}$.

Result 1: The kinetic energy of 9 nodes is represented as follows:

$$K_{node} = \sum_{i=0}^8 K_{ni} = \frac{1}{2} \dot{Q}_i^T M_n \dot{Q}_i \quad (13)$$

where $M_n = \text{diag} \{ M_{n0}, \dots, M_{n8} \}$.

Result 2: The potential energy of 9 nodes is represented as follows:

$$P_{node} = \sum_{i=0}^8 P_{ni} = \sum_{i=0}^8 [0 \ 0 \ m_{ni} g \ 0 \ 0 \ 0] \cdot q_i = g \sum_{i=0}^8 m_{ni} z_i \quad (14)$$

where g is gravitational acceleration.

3.2 Kinetic and potential energy of the struts

The strut takes a nested screw structure which consists of three segments and is driven by two DC motors anchored in the middle of the strut to make the screws on both sides synchronously extend or contract. Because of the symmetry of the strut, its centre of mass will always coincide with its geometric centre. To simplify the computation of the kinetic energy of the strut, each strut is considered as a moving mass point.

Given P_{ij} and P_{ji} , the position of the centre of mass of the strut L_{ij} expressed in the reference frame can be given as follows:

$$\overline{BG}_{ij} = \frac{1}{2} (P_{ij} + P_{ji}) = \frac{1}{2} (P_i + {}_{M_j}^B R p_{ji} + P_j + {}_{M_i}^B R p_{ij}). \quad (15)$$

The velocity of the centre of mass of the strut is then given by time derivation of equation (15)

$$V_{Gij} = \frac{d\overline{BG}_{ij}}{dt} = \frac{1}{2} (V_{ji} + V_{ij}) \quad (16)$$

Therefore, the kinetic energy of L_{ij} is given by

$$K_{ij} = \frac{1}{2} m_{ij} V_{Gij}^T V_{Gij} \quad (17)$$

where m_{ij} is the mass of strut L_{ij} .

Result 3: The kinetic energy of twenty-six struts is represented as

$$K_{strut} = \sum_{i,j} K_{ij} = \frac{1}{2} \dot{Q}^T M_{strut} \dot{Q} \quad (18)$$

where M_{strut} and $H = \text{diag} \{ h_{01}, h_{02}, \dots, h_{78} \}$ with $h_{ij} = m_{ij} \begin{bmatrix} I_{3 \times 3} & I_{3 \times 3} \\ I_{3 \times 3} & I_{3 \times 3} \end{bmatrix}$.

Result 4: The potential energy of the struts is given as follows:

$$\begin{aligned} P_{strut} &= \sum_{i,j} P_{ij} = \sum_{i,j} m_{ij} g (\overline{BG}_{ij} \cdot Z) \\ &= \frac{1}{2} \sum_{i,j} m_{ij} g (z_i - x_{ij} s \beta_i + y_{ij} s \alpha_i c \beta_i + z_{ij} c \alpha_i s \beta_i + z_j - x_{ji} s \beta_j + y_{ji} s \alpha_j c \beta_j + z_{ji} c \alpha_j s \beta_j) \end{aligned} \quad (19)$$

3.3 Dynamic equation in the Cartesian space

The dynamic equation of the 12-tetrahedral robot is then described by the Lagrangian equation,

$$M(Q)\ddot{Q} + V_m(Q, \dot{Q})\dot{Q} + G(Q) + F_c = J^T(Q)F \quad (20)$$

where $M(Q)$, $V_m(Q, \dot{Q})$, and $G(Q)$ are the inertia matrix, the Coriolis and the centrifugal force, and gravity terms, respectively, F_c represents the contact forces between the nodes and the ground, J is the inverse Jacobian of the robot, and $F = [f_{01} \ \cdots \ f_{ij} \ \cdots \ f_{78}]^T$, with f_{ij} the driving force exerted at strut L_{ij} along the u_{ij} direction. $M(Q)$, $V_m(Q, \dot{Q})$, and $G(Q)$ can be written in the summation of two matrices as follows:

$$M(Q) = M_{node} + M_{strut} \quad (21)$$

$$V_m(Q, \dot{Q}) = V_{m,node} + V_{m,strut} \quad (22)$$

$$G(Q) = G_{node} + G_{strut} \quad (23)$$

From the expressions of the kinetic and the potential energy of the nodes and the struts the above matrices are obtained as follows:

(1) The mass matrix of the nodes: M_{node} is directly obtained from equation (12),

$$M_{node} = \text{diag} \{M_{n0}, \dots, M_{ni}, \dots, M_{n8}\} \quad (24)$$

where $\text{diag}\{\}$ denotes the diagonal matrix of a vector.

(2) The centrifugal and the Coriolis forces of the nodes: The centrifugal and the Coriolis torques of N_i can be derived from the mass M_{ni} of the node using the Christoffel symbols [18],

$$V_{mni}(q_i, \dot{q}_i) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & V_{44} & V_{45} & V_{46} \\ 0 & 0 & 0 & V_{54} & V_{55} & V_{56} \\ 0 & 0 & 0 & V_{64} & V_{65} & V_{66} \end{bmatrix} \quad (25)$$

where the (k,j)th element of $V_{mni}(q_i, \dot{q}_i)$ can be calculated as follows:

$$V_{kj} = \sum_{l=1}^6 C_{lyk} \dot{q}_{il} = \sum_{l=1}^6 \frac{1}{2} \left\{ \frac{\partial m_{kj}}{\partial q_l} + \frac{\partial m_{kl}}{\partial q_j} + \frac{\partial m_{ij}}{\partial q_k} \right\} \dot{q}_{il} \quad (26)$$

with m_{kj} the (k,j)th element M_{ni} , and q_{il} is the l th element of q_i .

Therefore, the centrifugal and the Coriolis terms V_{node} can be achieved as follows:

$$V_{node} = \text{diag} \{V_{mn0}, \dots, V_{mni}, \dots, V_{mn8}\} \quad (27)$$

(3) **The gravity of the nodes:** G_{node} is given as

$$G_{node} = [G_{n0}^T \quad \cdots \quad G_{ni}^T \quad \cdots \quad G_{n8}^T]^T \quad (28)$$

where $G_{ni} = [0 \quad 0 \quad m_{ni}g \quad 0 \quad 0 \quad 0]^T$ is directly obtained from (14).

(4) **The mass matrix of the struts:** M_{strut} can be obtained directly from equation (18),

$$M_{strut} = \frac{1}{4} J_2^T H J_2 \quad (29)$$

(5) **The centrifugal and the Coriolis forces of the struts:** Since the effects of the centrifugal and the Coriolis torques are negligible and can be omitted in almost all applications [19], in the paper the derivation of V_{strut} is neglected and assumed to be zero.

(6) **The gravity of the struts:** G_{strut} is obtained directly from the potential energy equation (19)

$$G_{strut} = \frac{\partial P_{strut}}{\partial Q} = [G_{n0}^T \quad \cdots \quad G_{n8}^T]^T \quad (30)$$

where G_{n8} is given by

$$G_{n8} = \sum_{j=0}^8 \frac{\partial}{\partial q_i} P_{ij} = \frac{1}{2} \sum_{j=0}^8 m_{ij} g \begin{bmatrix} 0 \\ 0 \\ 1 \\ y_{ij} c \alpha_i c \beta_i - z_{ij} s \alpha_i c \beta_i \\ -x_{ij} c \alpha_i - y_{ij} s \alpha_i s \beta_i - z_{ij} c \alpha_i s \beta_i \\ 0 \end{bmatrix}. \quad (31)$$

3.4 The external contact model of the robot

When the robot falls, certain external nodes will collide with the ground and their velocities change greatly in a short period of time. Hence, their accelerations are rapid and so are the contact forces [20, 21]. It is assumed that the positions of the contact nodes remain unchanged during collision. Therefore, the impact function, which considers the contact procedure as a linear spring-damper system, can be adopted to model the contact between the nodes and the ground. The contact force [22, 23] of $N_i (i=1, \dots, 8)$ with the ground can be represented as

$$\begin{cases} \mathbf{F}_{ci} = \mathbf{0} & (\delta_i^z \leq 0) \\ \mathbf{F}_{ci} = \mathbf{K} \delta_i + \mathbf{C} \dot{\delta}_i & (\delta_i^z > 0) \end{cases} \quad (32)$$

where $\delta_i = [\delta_i^x \quad \delta_i^y \quad \delta_i^z]^T$ is the depth of N_i penetrating into the ground, $\mathbf{K} = \text{diag}\{k^x, k^y, k^z\}$ is the contact stiffness and $\mathbf{C} = \text{diag}\{c^x, c^y, c^z\}$ is the damping coefficient. Therefore, the contact force \mathbf{F}_{ci} between N_i and the ground is proportional to the displacement δ_i and the velocity $\dot{\delta}_i$, and it will be zero when the node leaves the ground. Then, the total contact force on the nodes of the robot is obtained as

$$\mathbf{F}_c = [\mathbf{0} \quad \mathbf{0} \quad \mathbf{F}_{c1}^T \quad \mathbf{0} \quad \cdots \quad \mathbf{F}_{ci}^T \quad \mathbf{0} \quad \cdots \quad \mathbf{F}_{c8}^T \quad \mathbf{0}]^T \quad (33)$$

where $\mathbf{0}$ is a 1×3 vector.

Substituting the above terms, the Jacobian and the contact force into (20) gives the dynamic equation of the motion of a 12-tetrahedron robot, which has the same form as that of a general parallel robot.

From equation (20), it is easy to see that all elements of $M(Q)$, $V_m(Q, \dot{Q})$, and $G(Q)$ are complex functions of the position vector Q and the velocity vector \dot{Q} , and $F(t)$ is a 26×1 force vector applied to the struts. Since there is an interplay between the nodes and the struts and $M(Q)$ is not a diagonal matrix, there are inertial couplings between the nodes and between the struts. Furthermore, the dynamic parameters are difficult to be identified due to uncertainty or other factors, such as frictional forces. The development of high-performance model-based control algorithms for a 12-tetrahedral robot is a challenging problem. In addition, the centralized controller needs to process all sensing information and compute feedback or feedforward signals for each strut actuator, which significantly increases the computational burden for a robot with twenty-six driving struts. As the number of tetrahedrons increases, the number of struts will increase and the control system needs to be correspondingly redesigned [24].

4. The control system of the 12-tetrahedral robot

As mentioned previously, the dynamic model of a 12-tetrahedral robot is too complex to be used in designing and implementing a controller in the robot. The main focus of this section is to develop a decentralized adaptive controller, where each strut is controlled independently of other struts.

4.1 The control structure of the 12-tetrahedral robot

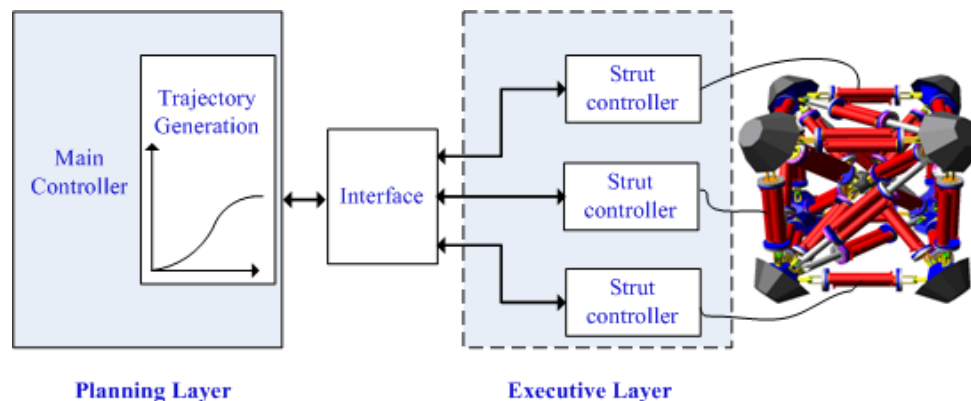


Fig. 3 Control system of 12-tetrahedral robot

The control system of the 12-tetrahedral robot consists of a planning layer, in which a main controller performs planning tasks, and an executive layer of twenty-six strut controllers, each of which independently regulates the motion of the corresponding strut, as illustrated in Fig. 3. The main controller determines the gait and the trajectory of the robot based on its current gait and task requirement and sends the planned trajectories to the strut controllers. The strut controller extracts information (e.g., the length and the velocity of the strut) from range sensors located on the strut and generates control signals to regulate the strut to follow the given trajectory.

The gait planning problem of the robot has been developed in [10], where the motion of the robot is divided into several steps and only certain struts are regulated to achieve the motion in each step. In this paper, we only focus on the executive layer and detail the implementation of the strut controller.

4.2 The strut controller

The objective of the strut controller is to provide force $F(t)$ applied to the struts in a sense that the displacement $D(t)$ of the struts tracks the reference (planned) trajectory $D_r(t)$ as closely as possible, where $D_r(t)$ is a function of time. In this section the decentralized adaptive control strategy presented in [25] is adopted to implement the strut controller, where the centrifugal and the Coriolis forces, the gravity of the struts, the coupling effects among the struts, and the unmodelled friction are regarded as disturbances on the struts. According to the decentralized control architecture, each strut controller can be viewed as an individual subsystem which independently regulates the length of the corresponding strut without information exchange with other subsystems.

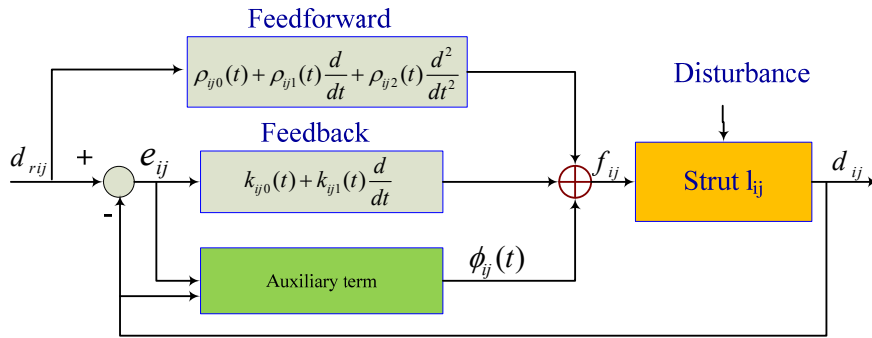


Fig. 4 Structure of strut controller

The structure of the strut controller is depicted in Fig. 4. Driving force $f_{ij}(t)$ only regulates the actual trajectory $d_{ij}(t)$ of strut L_{ij} to track the desired trajectory $d_{rij}(t)$ as closely as possible. The adaptive controller of L_{ij} is given as follows:

$$f_{ij}(t) = \phi_{ij}(t) + [k_{ij0}(t)e_{ij}(t) + k_{ij1}(t)\dot{e}_{ij}(t)] + [\rho_{ij0}(t)d_{rij}(t) + \rho_{ij1}(t)\dot{d}_{rij}(t) + \rho_{ij2}(t)\ddot{d}_{rij}(t)] \quad (34)$$

where $e_{ij}(t) = d_{rij}(t) - d_{ij}(t)$ is the tracking error, $\phi_{ij}(t)$ is an auxiliary term used to improve the tracking performance and to compensate for the disturbance, $k_{ij}(t) = k_{ij0}(t)e_{ij}(t) + k_{ij1}(t)\dot{e}_{ij}(t)$ is an adaptive position-velocity feedback controller, and $\rho_{ij}(t) = \rho_{ij0}(t)d_{rij}(t) + \rho_{ij1}(t)\dot{d}_{rij}(t) + \rho_{ij2}(t)\ddot{d}_{rij}(t)$ is an adaptive position-velocity-acceleration feedforward controller.

The above adaptive control law is implemented based on the weighted error (we only consider the strut controller for L_{ij} with subscript ij omitted for brevity)

$$r(t) = w_p e(t) + w_v \dot{e}(t) \quad (35)$$

Then, the auxiliary term is computed as follows:

$$\phi(t) = \phi(0) + a \int_0^t r(t) dt + br(t). \quad (36)$$

The feedback controller is given as follows:

$$k(t) = k_0(t)e(t) + k_1(t)\dot{e}(t) \quad (37)$$

where $k_0(t) = k(0) + \int_0^t \lambda_0 r(t)e(t) dt + \eta_0 r(t)e(t)$ and $k_1(t) = k_1(0) + \int_0^t \lambda_1 r(t)\dot{e}(t) dt + \eta_1 r(t)\dot{e}(t)$.

The feedforward controller is given by

$$\rho(t) = \rho_0(t)d_r(t) + \rho_1(t)\dot{d}_r(t) + \rho_2(t)\ddot{d}_r(t) \quad (38)$$

where $\rho_0(t) = \rho_0(0) + \int_0^t \zeta_0 r(t) d_r(t) dt + \xi_0 r(t) d_r(t)$, $\rho_1(t) = \rho_1(0) + \int_0^t \zeta_1 r(t) \dot{d}_r(t) dt + \xi_1 r(t) \dot{d}_r(t)$ and $\rho_2(t) = \rho_2(0) + \int_0^t \zeta_2 r(t) \ddot{d}_r(t) dt + \xi_2 r(t) \ddot{d}_r(t)$. In the above equations, $\{a, \lambda_0, \lambda_1, \zeta_0, \zeta_1, \zeta_2\}$ are positive scalar gain parameters, $\{b, \eta_0, \eta_1, \xi_0, \xi_1, \xi_2\}$ are zeros or positive scalar gain parameters, $\{w_p, w_v\}$ are positive weighting coefficients.

From the above equations, the strut controller is a model-free controller which is based entirely on the actual trajectory $d_{ij}(t)$ and the desired trajectory $d_{rj}(t)$, rather than the reference dynamic model of the robot. The time varying characteristics of the robot dynamics and the nonlinear coupling between the struts can be dealt with adaptively through adjusting the gains of the strut controllers according to the reference and the real trajectories. In addition, the decentralized architecture allocates computing burden to all of the strut controllers, which greatly reduces the computation time and complexity of the controller.

5. Simulation results

In order to test and verify the presented decentralized adaptive controller for the 12-tetrahedral robot, co-simulations of MATLAB and ADAMS are conducted.

5.1 Model setup of the robot

The model of the robot which ignores the motor, screws, etc., is built using the Pro/E software and then is imported into the ADAMS software. Therefore, the model is mainly made of nodes, struts and universal joints fixed on the nodes. The physical properties and other specifications of the model are summarized in Table 1.

Table 1 Main dimensions of the 12-tetrahedral robot

Parameter	Value	Unit
Mass of N_0	0.90	kg
Rotational inertia of N_0 (I_{xx}, I_{yy}, I_{zz})	(703.81, 705.08, 5515.16)	kg·mm ²
Mass of $N_1 \sim N_8$	1.41	kg
Rotational inertia of $N_1 \sim N_8$ (I_{xx}, I_{yy}, I_{zz})	(6316.94, 5916.40, 5515.16)	kg·mm ²
Mass of central links	0.84	kg
Length of central link	294	mm
Mass of border strut	0.90	kg
Length of border strut	330	mm
Mass of diagonal strut	1.33	kg
Length of diagonal strut	568	mm

In [10], we presented a motion planning approach for a 12-tetrahedral robot, including gait planning and trajectory planning. During the movement of the 12-tetrahedral robot according to the given gait, only certain nodes and struts are involved in every phase of the gait. In this section, we only consider the control action of the first two phases of the gait planning, where only struts l_{06} , l_{07} , l_{16} , l_{56} , l_{68} , and l_{78} are regulated to move N_6 and N_7 . Therefore, only six controllers are needed to construct these struts. The block diagram of each strut controller is presented in Fig. 5.

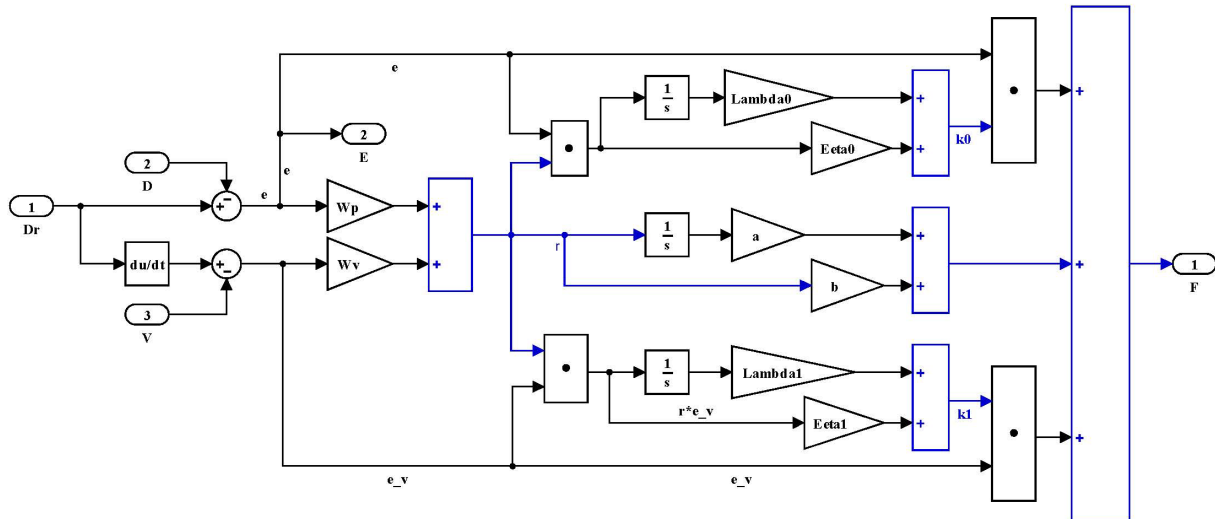


Fig. 5 Block diagram of individual strut controller

In the strut controllers, the feedforward terms are omitted and the auxiliary and the feedback terms are retained to reduce the computation time while the system performance is guaranteed.

The parameters in the strut controller shown in Fig. 5 were adjusted to be $w_p = 40, w_v = 1, a = 3, b = 2, \lambda_0 = 5, \eta_0 = 0.2, \lambda_1 = 0, \eta_1 = 0$. The gains in the adaptive law were initialized to be zero and the above parameters present better results achieved through many tests and optimizations, but are not optimal.

5.2 Simulation results and analysis

Because of the symmetry of the robot we only present the simulation results for struts $l_{07}, l_{56},$ and l_{68} for the space limit and without the loss of generality.

Figs. 6-8 show actual and desired length trajectories as well as tracking errors for struts $l_{07}, l_{56},$ and l_{68} . It can be seen from these figures that the true trajectories can accurately track their corresponding reference trajectories. Although there are couplings between the struts, the adaptive controller can reduce the coupling effects by tuning the parameters. And the maximum tracking errors of all struts will be limited to -1mm to 1mm within 1-2s.

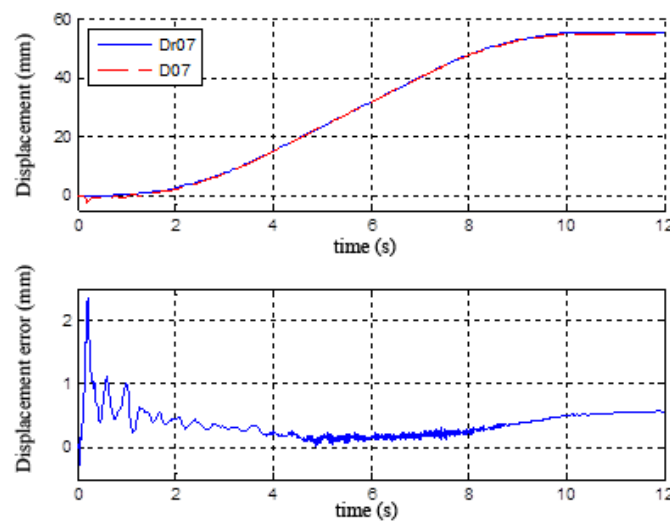


Fig. 6 Desired and actual displacements of link l_{07} and their errors

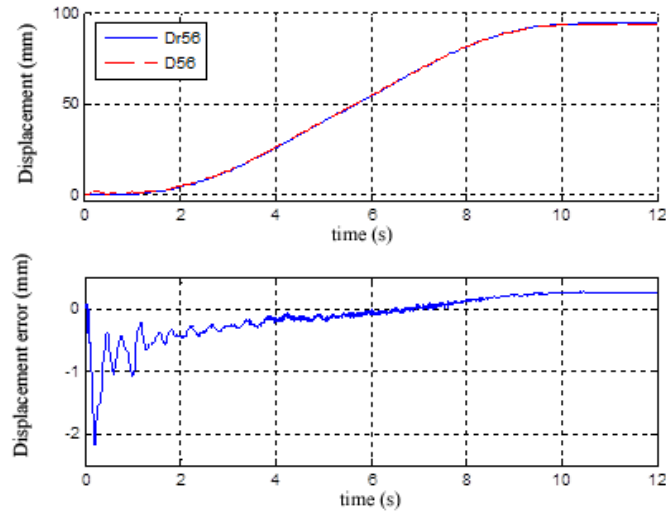


Fig. 7 Desired and actual displacements of link l_{56} and their errors

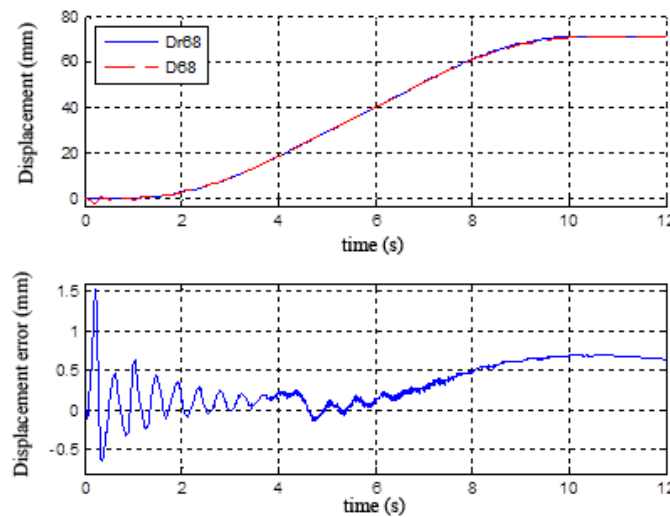


Fig. 8 Desired and actual displacements of link l_{68} and their errors

Figs. 9-11 show actual and desired velocities as well as tracking errors for struts l_{07} , l_{56} , and l_{68} . It can be seen from these figures that the true velocities can accurately track their corresponding reference velocities and that the tracking error will be within a limited range.

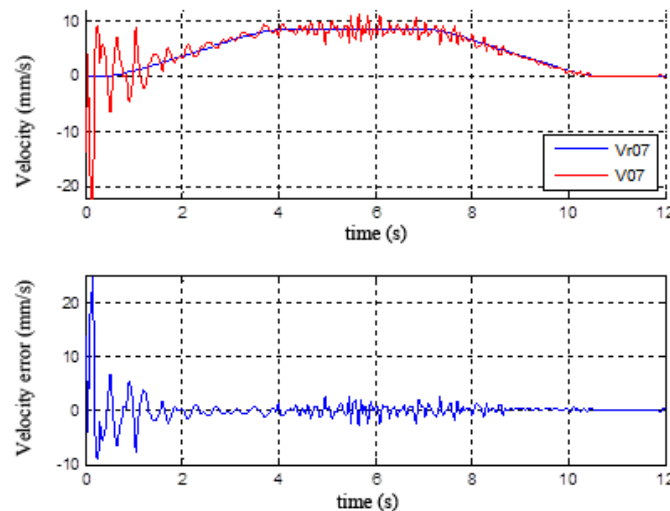


Fig. 9 Desired and actual velocities of link l_{07} and their errors

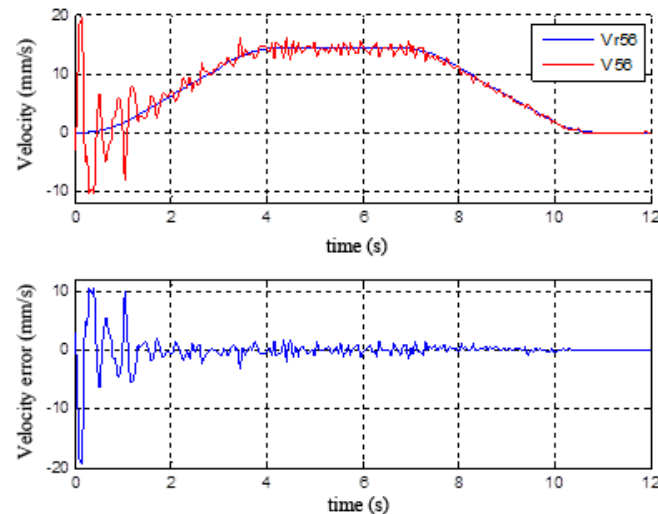


Fig. 10 Desired and actual velocities of link l_{56} and their errors

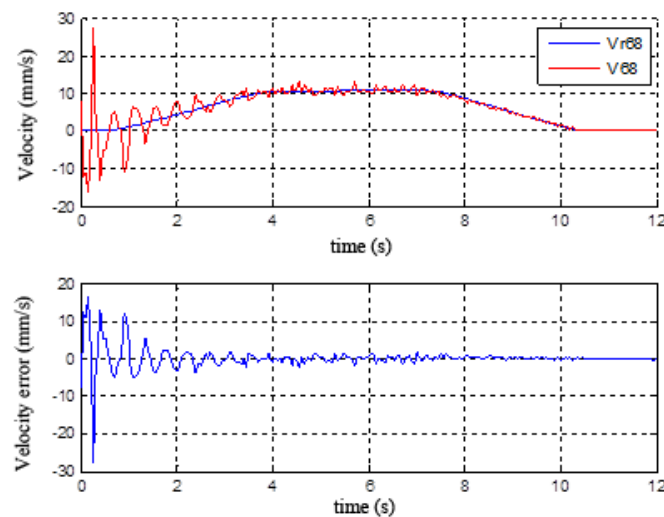


Fig. 11 Desired and actual velocities of link l_{68} and their errors

6. Conclusion

In this paper, we have derived a dynamic model and presented a controller for the 12-tetrahedral robot. Since the robot is a complicated manipulator with multiple struts and multiple nodes, it is difficult to use the obtained dynamic equation in implementing a controller in the robot. Therefore, a controller which consists of a planning layer and an executive layer, is designed for controlling the robot. In the planning layer, the main controller generates gait and trajectory and transmits commands to the strut controllers. The executive layer is designed based on the decentralized adaptive control strategy and consists of twenty-six strut controllers, each of which regulates the movement of the corresponding strut only using the commands from the main controller and the measurements from the sensors on that strut. The simulations have been conducted using ADAMS and Matlab to verify the feasibility and effectiveness of the proposed controller.

REFERENCES

- [1] L. Bruzzone, G. Quaglia: *Review article: locomotion systems for ground mobile robots in unstructured environments*. MECHANICAL SCIENCES, **3**, 49-62(2012). <https://doi.org/10.5194/ms-3-49-2012>

- [2] M. Kamel, Y. Zhang: *Developments and Challenges in Wheeled Mobile Robot Control*. The 10th International Conference on Intelligent Unmanned Systems (ICIUS), Montreal, QC, Canada 2014.
- [3] B. Rehman, M. Focchi, et al.: *Towards a multi-legged mobile manipulator*. IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden 2016: 318-324. <https://doi.org/10.1109/ICRA.2016.7487545>
- [4] Y. Okada, K. Nagatani, K. Yoshida, et al.: Shared autonomy system for tracked vehicles on rough terrain based on continuous three-dimensional terrain scanning. *J. Field Robot.*, **6**, 875-893(2011). <https://doi.org/10.1002/rob.20416>
- [5] J. Guo, G. Wu, and S. Guo: *Study on movement stability for the spherical amphibious robot*. In Proceedings of 2016 IEEE International Conference on Mechatronics and Automation, Harbin, China 2016: 55-60. <https://doi.org/10.1109/ICMA.2016.7558534>
- [6] A. Suzumura, Y. Fujimoto: *Optimal motion for obstacle stepping in a hybrid wheeled-legged hexapod*. IEEE Transactions on Industrial Electronics, **7**, 3648-3659(2014). <https://doi.org/10.1109/TIE.2013.2286071>
- [7] L. Zhang, S. Dai: *An overview of the development on reconfiguration of metamorphic mechanisms*. ASME/IFToMM International Conference on Reconfigurable Mechanisms and Robots, 2009: 8-12.
- [8] C. Steven, B. Matthew, et al.: *Tetrahedral Robotics for Space Exploration*. IEEE Aerospace and Electronic Systems Magazine, **6**, 22-30(2007).
- [9] L. Zhang, S. Bi: *Motion analysis and simulation of tetrahedral rolling robot*. Journal of Beijing University of Aeronautics and Astronautics, **4**, 415-420(2011).
- [10] X. Wang, X. Wang, Z. Zhang, Y. Zhao: *Motion Planning of Kinematically Redundant 12-tetrahedral Rolling Robot*. International Journal of Advanced Robotic Systems, **13**(2016). <https://doi.org/10.5772/62179>
- [11] M. Abrahantes, A. Silver, L. Wendt, D. Littio: *Construction and Control of a 4-Tetrahedron Walker Robot*. 40th Southeastern Symposium on System Theory, 2008: 343-346. <https://doi.org/10.1109/SSST.2008.4480251>
- [12] M. Abrahantes, L. Nelson, P. Doorn: *Modelling and Gait Design of a 6-Tetrahedron Walker Robot*. 42nd South Eastern Symposium on System Theory, 2010: 248-252.
- [13] M. Abrahantes, J. Dratz, C. Smits, and L. Nelson: *Modelling and Gait Design of a 8-Tetrahedron Walker Robot*. Models and Modeling Methodologies in Science and Engineering: MMMse, 2011: 241-245.
- [14] M. Abrahantes, A. Silver, L. Wendt: *Gait Design and Modelling of a 12-Tetrahedron Walker Robot*. 39th Southeastern Symposium on System Theory, Macon: Mercer University 2007: 21-25.
- [15] M. Izadi, M. Mahjoob M. Soheilypour, et al: *A motion planning for toppling-motion of a TET walker*. The 2nd International Conference on Computer and Automation Engineering (ICCAE), 2010: 34-39. <https://doi.org/10.1109/ICCAE.2010.5451369>
- [16] M. Izadi, M. Mahjoob M. Soheilypour: *Walking Gait of a Single-Tetrahedral Robot: Design, Modelling and Implementation*. Proceedings of the ASME 2010 10th Biennial Conference on Engineering Systems Design and Analysis, Istanbul, Turkey 2010: 1-5.
- [17] C. John: *Introduction to Robotics: Mechanics and Control*, Pearson Education Inc.: New Jersey, 2005.
- [18] B. Siciliano, O. Khatib: *Springer of robotics*. Springer-Verlag: Berlin Heidelberg, 2008. <https://doi.org/10.1007/978-3-540-30301-5>
- [19] G. Lebet, K. Liu, F. L. Lewis: *Dynamic analysis and control of a Stewart platform manipulator*. Journal of Robotic systems, **5**, 629-655(1993). <https://doi.org/10.1002/rob.4620100506>
- [20] S. Goyal, E. Pinson, F. Sinden: *Simulation of dynamics of interacting rigid bodies including friction I: General problem and contact model*. Engineering with computers, **3**, 162-174(1994). <https://doi.org/10.1007/BF01198742>
- [21] S. Goyal, E. Pinson, F. Sinden: *Simulation of dynamics of interacting rigid bodies including friction II: Software system design and implementation*. Engineering with computers, **3**, 175-195(1994). <https://doi.org/10.1007/BF01198743>
- [22] W. Lee, A. Sanderson: *Dynamic rolling locomotion and control of modular robots*. IEEE Transactions on Robotics and Automation, **1**, 32-41(2002). <https://doi.org/10.1109/70.988972>
- [23] W. Lee, A. Sanderson: *Dynamic rolling of modular robots*. Proceedings of the 2000 IEEE International Conference on Robotics and Automation, San Francisco, CA 2000: 2840-2846.

- [24] B. Solomon, P. Olivier: *Implementation of a Decentralized Adaptive Control System for the 12-TET Walker*. IEEE Thirty-Ninth Southeastern Symposium on System Theory (SSST'07), 2007: 102-103. <https://doi.org/10.1109/SSST.2007.352327>
- [25] H. Seraji: *Decentralized adaptive control of manipulators: theory, simulation, and experimentation*. IEEE Transactions on Robotics and Automation, 5(2): 183-201(1989). <https://doi.org/10.1109/70.88039>

Submitted: 26.12.2016

Accepted: 18.10.2017

Xingbo Wang
College of Automation, Nanjing
University of Posts and
Telecommunications, Nanjing 210023,
China;
Shanghai Key Laboratory of spacecraft
mechanism , Shanghai 201108, China
Xiaotao Wang
Zhongpeng Zhang
College of Astronautics, Nanjing
University of Aeronautics and
Astronautics, Nanjing 210016, China