Gait and Trajectory Rolling Planning and Control of Hexapod Robots for Disaster Rescue Applications

Article *in* Robotics and Autonomous Systems · June 2017

DOI: 10.1016/j.robot.2017.05.007

CITATIONS
13

READS
872

4 authors, including:

Hua Deng
Central South University
198 PUBLICATIONS **5,102** CITATIONS
SEE PROFILE

Guiyang Xin
The University of Edinburgh
20 PUBLICATIONS **115** CITATIONS
SEE PROFILE

Guoliang Zhong
Central South University
42 PUBLICATIONS **289** CITATIONS
SEE PROFILE

Robotics and Autonomous Systems 95 (2017) 13–24

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

Gait and trajectory rolling planning and control of hexapod robots for disaster rescue applications

Hua Deng [a,b], Guiyang Xin [a,b,c], Guoliang Zhong [a,b,*], Michael Mistry [c]

[a] School of Mechanical and Electrical Engineering, Central South University, Changsha 410083, Hunan, China
[b] State Key Laboratory of High-Performance Complex Manufacturing, Central South University, Changsha 410083, Hunan, China
[c] School of Informatics, The University of Edinburgh, Edinburgh EH8 9AB, United Kingdom

HIGHLIGHTS

- A complete control architecture based on gait and trajectory rolling planning method is proposed for hexapod robots.
- A method based on COG Jacobian is proposed to calculate joint motion depending on desired the robot's COG trajectory and end-point trajectories of each leg.
- Typical gaits are obtained according to environmental adaptability and ZMP stability margin.

ARTICLE INFO

Article history:
Received 12 January 2017
Received in revised form 14 April 2017
Accepted 22 May 2017

Keywords:
Disaster rescue robot
Rolling planning
Legged locomotion
Stability control
State estimation

ABSTRACT

Hexapod robots have stronger adaptability to dynamic unknown environments than wheeled or trucked ones due to their flexibility. In this paper, a novel control strategy based on rolling gait and trajectory planning, which enables hexapod robots to walk through dynamic environments, is proposed. The core point of this control strategy is to constantly change gait and trajectory according to different environments and tasks as well as stability state of robot. We established a gait library where different kinds of gaits are included. Zero moment point, which indicates the stability of robot, is estimated by a Kalman filter. According to this control strategy, a hierarchical control architecture consisting of a man–machine interface, a vision system, a gait and trajectory planner, a joint motion calculator, a joint servo controller, a compliance controller and a stability observer is presented. The control architecture is applied on a hexapod robot engaging in disaster rescue. Simulation and experimental results show the effectiveness of our control strategy.

1. Introduction

Legged robots have attracted the attention of researchers because of their superior adaptability to complex environments than their wheeled counterparts. Legged robots can be divided into two-legged humanoid robots, quadruped robots, hexapod robots and others with more legs. Among these items, hexapod robots can easily implement statically stable walking, and therefore have been investigated widely all over the world. The leg mechanisms of most existing hexapod robot prototypes are 3 Degrees-of-Freedom (DOFs) serial mechanisms including 3 links and 3 revolute joints, which is based on bionics principle. However, in our previous works [1–3], we proposed a novel hexapod robot called as PH-Robot with parallel leg mechanism so as to bring the advantage of parallel mechanisms to legged robots.

Compared with multi-legged robots, biped robots can achieve fast locomotion like what human beings do, but it is hard to keep balance. Even some very advanced biped robots, such as Atlas [4,5] and Asimo [6], still cannot run over rough terrain. Quadruped robots [7–9] might be a good choose for utility application of robotics because it cannot only move very fast but also have larger stability margin than humanoids. By contrast, hexapod robots have higher stability, but also have more complicated gait planning problem due to the increased number of legs. It can be demonstrated by creatures of nature. Children need time to learn upright walking because it is more difficult to keep balance by just two legs than when climbing using two legs and two arms. Most of mammals use four limbs to walk and run, whereas only human can walk upright. Insects always have six or more legs. So insects can traverse very rough terrain even can climb walls. But they need to

* Corresponding author at: School of Mechanical and Electrical Engineering, Central South University, Changsha 410083, Hunan, China.
E-mail address: zhong001985@csu.edu.cn (G. Zhong).

http://dx.doi.org/10.1016/j.robot.2017.05.007
0921-8890/

be lightweight in order to achieve fast locomotion. From this point, that is one reason for why only small features have six or more legs. As for humanoid robots, most of them depend on the control of zero moment point (ZMP) [10–12] to walk with balance. More research on this issue is implemented on quadruped robots [13–15]. Meanwhile a common dynamic model of inverted pendulum is applied for simplification due to the heavy body relative to their legs. By contrast, hexapod robot' legs make more contribution to the entire mass of robot than the body. Therefore, their method is not suitable for hexapod robots. Actually, hexapod robots have enough static stability margin to withstand most of external disturbance from the environment. Thus, we focus on gait planning and detection of ZMP rather than the planning and control of ZMP. Only in rare cases, we need to plan ZMP or center of mass (COG) trajectories and then track them using feedback control. We will show an example of carrying object using two adjacent legs to explain this special case in this paper.

In order to implement rolling planning of gait according to different missions and environmental conditions, different types of gait should be generated. Many studies have focused on the gait generation. As far as we know, it is hard for hexapod robots to execute dynamic gaits, such as trot or gallop which belongs to mammal gaits. The static gaits including tripod, quadruped, one-by-one types of gait as in [16] and other free gait in [17,18] are widely used in hexapod robots. Wang et al. [19] discussed three different gaits, i.e., insect-wave gait, mammal-kick gait and mixed gait. They still belong to static gaits. Asif et al. [20] proposed a method that is to increase the number of supporting legs according to the roughness of terrain in order to increase stability. Other researchers [21–24] focused on the analysis of fault tolerant gait as one or more legs are prevented from supporting and swing. Grzelczyk et al. [25–27] discussed the application of central pattern generators (CPGs) on hexapod robots to get stable and low energy consumption gaits. In this paper, we will give some typical gaits according to different tasks and environments, which establishes the gait library for rolling gait planning.

After gait planning, we need a trajectory generator to plan the motion of body and swing legs. On this issue, we must address the coupled kinematics of legs and body. In fact, we have derived the inverse and forward kinematics in our previous work [1]. We will present a new expression of kinematics that will lead to a solution of redundancy problem. On the other hand, the curve of trajectory function should be determined according to some rules like avoiding impact between feet and the ground. To avoid impact, the initial and final velocities and accelerations of the trajectory are chosen to be zero. Wang et al. [28] used the combination function of sine and cosine functions to satisfy the abovementioned condition. We have successfully applied a similar function called as cycloid curve to accomplish tripod gait in [1,3]. Rebula et al. [29] proposed a combined curve consisting of two linear segments and a parabolic segment. Kalakrishnan et al. [13] applied a series of quantic spline segments (fifth order polynomials) as the trajectories of COG and the endpoints of swing legs. Kolter et al. [14] used a very simple trajectory like a box as the foot trajectory. We will also employ fifth order polynomials which can provide convenience for rolling trajectory planning.

Another crucial aspect for legged robots is compliance control because we cannot get accurate information of ground via vision system. Asif et al. [30] proposed a control framework constituting a modified hybrid force–position controller to deal with the environmental disturbances. Mistry et al. [31] used model-based control to decrease the feedback gain of PD controller, and then decrease the stiffness of the whole controller in order to get compliant ability toward non-perceived obstacles and terrains. But it depends on precise dynamic model of robots, whereas dynamic modeling for legged robots is a difficult thing especially for our robot with parallel leg mechanisms. Another choice for getting compliant ability is to use elastic mechanical device to realize passive compliance. However, the price to pay is a typical degradation of robot performance. In fact, the elastic mechanical device applied in the passive compliance induces position errors, which reduces robot accuracy in tracking tasks, because of lightly damped vibrations and static deformation under gravity. That is why compliant robots like Baxter cannot achieve very accurate position control performance [32,33]. The solution to cope with the control issue of rigid manipulators interacting with the working environment is the active compliance strategy, which ranges from the concept of impedance control to the concept of hybrid position/force control. Especially, compared with hybrid position/force control, the impedance control has unique advantages in trajectory tracking studies because it is essentially a position control [3]. Furthermore, the principle of impedance control is to change the desired trajectory in Cartesian space after contacting ground.

In this study, a complete control architecture for hexapod robots applied to disaster rescue is presented based on seven subsystems. The main contributions of this paper can be stated as follows.

(1) A variety of gaits are proposed from different views from walking speed and stability margin to different tasks, and since gaits have large stability margin usually with slow walking speed, robots will choose different gaits according to the roughness of ground and assignment.
(2) A method based on COG Jacobian is proposed to calculate joint motion depending on the desired robot's COG trajectory and end-point trajectories of each leg.
(3) To address the problems associated with smooth contact with environment resulting from inaccuracy of LIDAR's detection, the present work adopts an impedance controller as compliance controller.
(4) In order to get more time for adjusting gait and trajectory, a stability observer based on ZMP is employed to predict stability margin of robots, which stems from the predictive ability of Kalman filter.

The rest of this paper is laid out as follows. In Section 2, we give a brief overview of the PH-Robot's hardware. Then, in Section 3, kinematics of PH-Robot is formulated. The proposed control architecture and detailed analysis of some parts of the control architecture are presented in Section 4. In Section 5, we present some experimental results. Finally, conclusions are presented in Section 6.

2. PH-Robot description

PH-Robot is very different with most popular hexapods due to its parallel leg mechanism as shown in Fig. 1. The purpose of using parallel mechanism is to improve payload capability, which has been analyzed in [2]. The robot stands approximately 820 cm in initial state (The height of robot depends on the length of linear actuators.) with a total mass of about 130 kg. There are 18 actuated degree of freedom excluding the one freedom of LIDAR. Each active joint is a prismatic joint corresponding to a linear actuator driven by servo motor with absolute encoder. An inertial measurement unit (IMU) mounted on the body can provide accurate 3-axis angle, angular rate and 3-axis acceleration data. Motion state will be estimated by the combination of IMU and encoders. There is a 6-axis force/torque sensor mounted on each foot to measure the contact force with ground. Data offered by all the above sensors is processed by an embedded controller CX2030 produced by Beckhoff. CX2030 can accomplish all the computation within 2 ms that is the servo cycle.

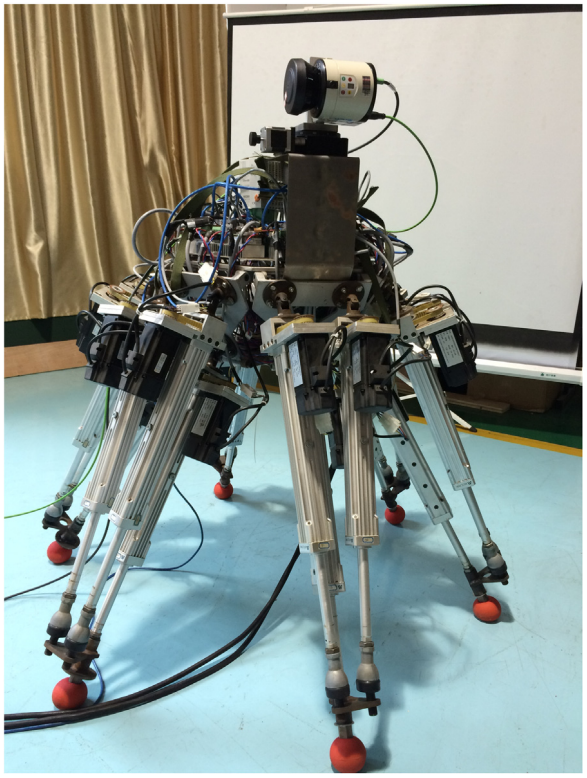

Fig. 1. Prototype of PH-Robot.

A Sick LMS111 planar LIDAR is mounted on a spindle that can rotate in order to realize 3D scanning. Since the 3D point cloud is very tremendous data, we use a super laptop to process the data from LIDAR instead of using CX2030. CX2030 can require what we need from that laptop via Ethernet communication.

3. Kinematics

As we know, legged robots are floating base systems with a main body and several leg modules [19,20]. Referring to Fig. 2, frame $C-xyz$ is a local frame fixed on the body. The position and orientation of the body with respect to global frame $B-XYZ$ can be described by a generalized coordinate $\mathbf{x}_0 = \begin{bmatrix}\mathbf{p}_0^T & \boldsymbol{\theta}_0^T\end{bmatrix}^T \in \mathbb{R}^{6\times 1}$ where $\mathbf{p}_0 \in \mathbb{R}^{3\times 1}$ is the position vector of the origin of body frame and $\boldsymbol{\theta}_0 \in \mathbb{R}^{3\times 1}$ is composed of pitch, yaw, and roll angles of the body. The position vector of the end point of ith leg with respect to global frame can be expressed as

$$\mathbf{p}_i = \mathbf{p}_0 + {}^B_C\mathbf{R}({}^C\mathbf{p}_{U_{1i}} + {}^C_0\mathbf{R}^0\mathbf{p}_{U_{1i}E_i}) \tag{1}$$

where ${}^C\mathbf{p}_{U_{1i}}$ is the position vector of the origin of frame $U_{1i}-x_0y_0z_0$ with respect to frame $C-xyz$, ${}^0\mathbf{p}_{U_{1i}E_i}$ is the position vector from point U_{1i} to the end point E_i with respect to frame $U_{1i}-x_0y_0z_0$, ${}^B_C\mathbf{R}$ and ${}^C_0\mathbf{R}$ are the rotation transformation matrices.

Differentiating Eq. (1) with respect to time gives

$$\dot{\mathbf{p}}_i = \dot{\mathbf{p}}_0 + \boldsymbol{\omega}_0 \times {}^B_C\mathbf{R}({}^C\mathbf{p}_{U_{1i}} + {}^C_0\mathbf{R}^0\mathbf{p}_{U_{1i}E_i}) + {}^B_C\mathbf{R}(\mathbf{0} + {}^C_0\mathbf{R}^0\mathbf{J}_i\dot{\mathbf{q}}_i) \tag{2}$$

where $\boldsymbol{\omega}_0$ means the rotating velocity of body frame with respect to global frame, ${}^0\mathbf{J}_i$ is the usual Jacobian matrix of the ith leg in terms of the local frame $U_{1i}-xyz$, $\mathbf{q}_i$ is the actuator variables or generalized coordinates of the ith leg, and hereafter, we will use ${}^B\mathbf{p}_i \triangleq {}^B_C\mathbf{R}({}^C\mathbf{p}_{U_{1i}} + {}^C_0\mathbf{R}^0\mathbf{p}_{U_{1i}E_i})$, ${}^B\mathbf{J}_i = {}^B_C\mathbf{R}{}^C_0\mathbf{R}^0\mathbf{J}_i$. Eq. (2) can be rewritten as

$$\dot{\mathbf{p}}_i = \dot{\mathbf{p}}_0 - S({}^B\mathbf{p}_i)\boldsymbol{\omega}_0 + {}^B\mathbf{J}_i\dot{\mathbf{q}}_i \tag{3}$$

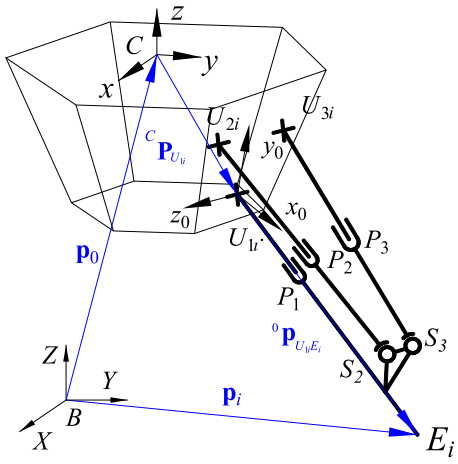

Fig. 2. Coordinate definition of ith leg.

where $S(\cdot)$ is the skew symmetric matrix. Eq. (3) expresses the coupled motion of body and actuators. Eq. (3) can also be expressed in matrix form as

$$\dot{\mathbf{p}}_i = \begin{bmatrix}\mathbf{I} & -S({}^B\mathbf{p}_i) & {}^B\mathbf{J}_i\end{bmatrix}\begin{bmatrix}\dot{\mathbf{x}}_0 \\ \dot{\mathbf{q}}_i\end{bmatrix}. \tag{4}$$

Then with the definition of $\mathbf{J}_i = \begin{bmatrix}\mathbf{I} & -S({}^B\mathbf{p}_i) & {}^B\mathbf{J}_i\end{bmatrix}$, Eq. (4) can be rewritten as,

$$\dot{\mathbf{p}}_i = \mathbf{J}_i\begin{bmatrix}\dot{\mathbf{x}}_0 \\ \dot{\mathbf{q}}_i\end{bmatrix}. \tag{5}$$

So far, one can use Eq. (5) to calculate forward kinematics easily. However, as for inverse kinematics, it is a redundant problem. We could take the pseudo-inverse of $\mathbf{J}_i$, and compute corresponding motion of generalized coordinates,

$$\begin{bmatrix}\dot{\mathbf{x}}_0 \\ \dot{\mathbf{q}}_i\end{bmatrix} = \mathbf{J}_i^+\dot{\mathbf{p}}_i \tag{6}$$

where the pseudo-inverse is defined as $\mathbf{J}_i^+ = \mathbf{J}^T(\mathbf{J}\mathbf{J}^T)^{-1}$ while $\mathbf{J}_i$ has full row rank. When $\mathbf{J}_i$ is not full row rank, the pseudo-inverse is computable via a singular value decomposition as shown in [34].

If a leg is a supporting leg, $\dot{\mathbf{p}}_i = 0$. Then Eq. (3) can be used to compute the motion of actuators according to body's motion as follows,

$$\dot{\mathbf{q}}_i = {}^B\mathbf{J}_i^{-1}(\dot{\mathbf{p}}_0 - S({}^B\mathbf{p}_i)\boldsymbol{\omega}_0). \tag{7}$$

4. Control architecture

The planning and control issue for a hexapod robot is to plan an optimal sequence of joint angles for 18 DOFs that move the robot to its desired position and precisely track the expected trajectory while maintaining stability. However, it is an intractable thing to find a perfect control strategy for a hexapod robot which needs to do so many tasks with complex environment conditions. Usually, researchers would like to establish a hierarchical control architecture to deal with the change of task and environment [13,14,20,30]. Based on the above research, we also present a hierarchical control architecture as shown in Fig. 3. What makes our hierarchical controller different is that we add more challenging gaits, use predicted state to monitor robot's stability and use impedance control to obtain compliance ability. Besides, we propose a method to compute joint trajectory according to planned COG trajectory based on COG Jacobian [35,36].

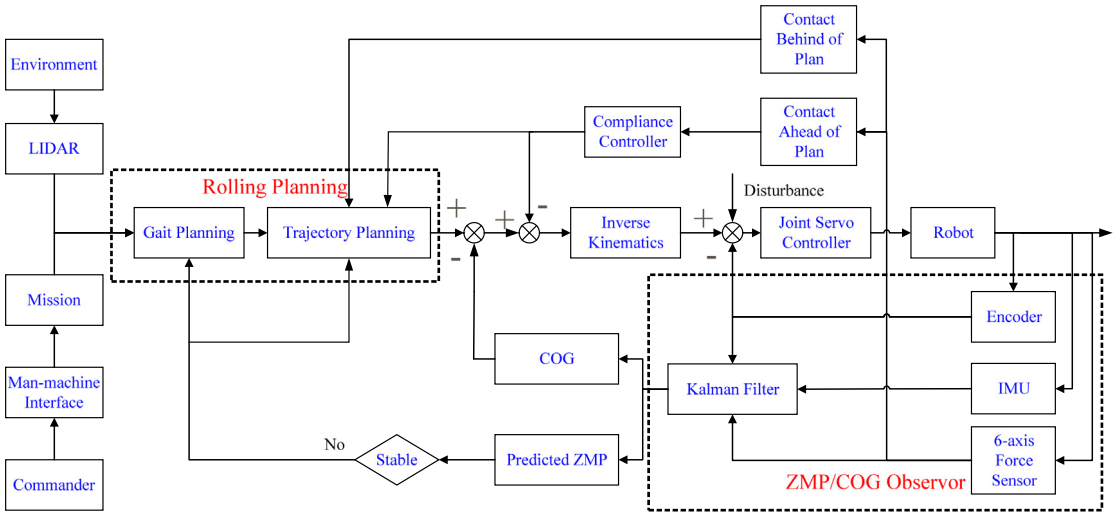

Fig. 3. Hierarchical control architecture.

The control architecture can be separated into seven parts, i.e., man–machine interface, vision system, gait and trajectory planner, joint motion calculator, joint servo controller, compliance controller and stability observer. This paper focuses on gait and trajectory planning, joint motion calculation, compliance control and stability observation. Man–machine interface and vision system will be discussed in the future work. Joint servo control, which tries to ensure the planned motion executed with accuracy and robustness to disturbance, has been studied in our previous work [37]. Gait planning level reserves a gait library to be used by robot according to different tasks, environments and stability state. Trajectory planner and inverse kinematics solver try to achieve the upcoming footsteps. In the following subsections, we will analyze each sub-system of our control approach in detail, highlighting the difference between it and past approaches.

4.1. Gait planning

As mentioned before, many types of gait have been proposed by researchers. We can classify them according to the type of support polygon or the number of supporting and swing legs as "3+3" alternating tripod(also known as tripod gait), "4+2" quadruped, and "5+1" one-by-one gait, as shown in Fig. 4. Tripod gait is considered as the fastest movement gait for hexapod robots as its duty factor β, which is defined as the fraction of cycle time in which a leg is in the supporting phase [38], is 1/2. By contrast, the duty factor β of quadruped and one-by-one gait is 2/3 and 5/6 respectively. However, we think sometimes duty factor β cannot be used as a criterion for walking speed. As shown in Fig. 4(a), (b), when the same movement distance of robot body is adopted for tripod gait and quadruped gait, the movement speeds of the two types of gait are identical. But it should be noted that if the body movement distance is the maximum, quadruped gait will cannot be realized because of the workspace limitation of leg 2 and 5. Therefore, if taking the limitation of legs' stride and motors' speed into consideration, tripod gait will be the fastest gait absolutely.

Another important aspect worthy to analyze is the stability margin of different gaits. Static stability margin is defined as the minimum distance from the projection on the ground of gravity center of robot to each edge of the support polygon [37], and the edge corresponding to the minimum distance is called as tipping over trended edge. According to that definition, it seems that the stability margins of the three gaits are identical. However, the number of tipping over trended edge for tripod gait is 3, whereas that is 2 and 1 for quadruped gait and one-by-one gait respectively. Therefore, we would like to employ one-by-one gait to traverse highly uneven terrain even if it has the lowest walking speed.

For hexapod robots, it should be noted that we do not have to plan the COG trajectory in order to keep stable walking, because all of these gaits have large static stability margin. That is very different from humanoid robots that must walk in quasi-dynamic gaits because of their small support polygon. This characteristic of hexapod robot will bring large convenience for trajectory planning and inverse kinematics, because it is very difficult to compute joint trajectory according to COG trajectory, whereas it is easy to compute joint motion according to the body's motion using Eq. (7). Although our control architecture includes COG trajectory planner, that is only used for several extreme cases like carrying object through two adjacent legs as shown in Section 4.1.2. Usually, we only plan the trajectories of body and swing legs when using these three normal gaits. Actually, the trajectories of COG and body are considered approximately identical in lots of literature [39–41] aimed at the locomotion of humanoid robots, because the mass of humanoid robot's legs can be ignored compared with the mass of body. Oppositely, that cannot be applied on our robot, because the legs of our robot are very heavy.

Besides those three normal gaits, we can use mixed gait to deal with more complicated cases. We will propose types of gait, i.e., climbing stairs gait and carrying object gait.

4.1.1. Climbing stairs gait

The climbing stairs gait is shown in Figs. 5 and 6. This gait is a combination of quadruped gait and one-by-one gait with four steps within a gait cycle. The former two steps belong to quadruped gait while the latter two steps belong to one-by-one gait. The reason for designing this gait sequence is that the projection of center of gravity is behind the line connecting foot 3 and 6 due to the inclination of stairs. In experimental cases, the sizes of stairs are measured by LIDAR. Trajectory planner will decide the stride and height of foot trajectory according the data from LIDAR data processor.

4.1.2. Carrying object gait

One of the purposes of our robot is to carry object using one or two legs rather than using an extra arm mounted on body as other legged robots do. That idea steams from insects that can use their one or two fore legs to catch food or other things. Fig. 7 shows an example of grabbing a bottle using an end-effector mounted on

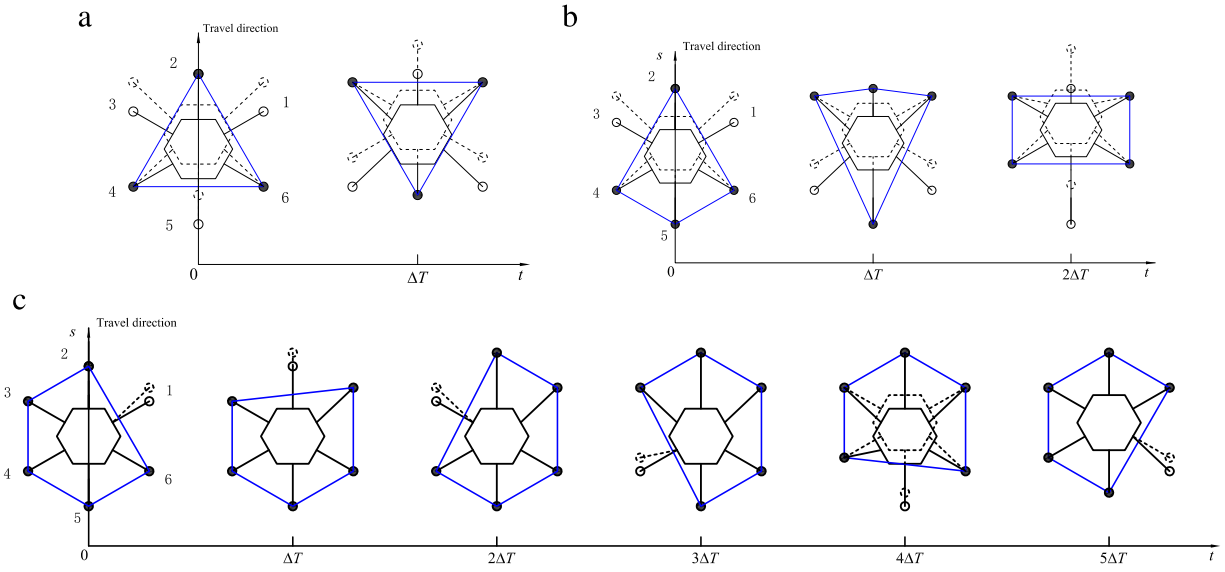

Fig. 4. Three typical gaits: (a) tripod gait, (b) quadruped gait, (c) one-by-one gait.

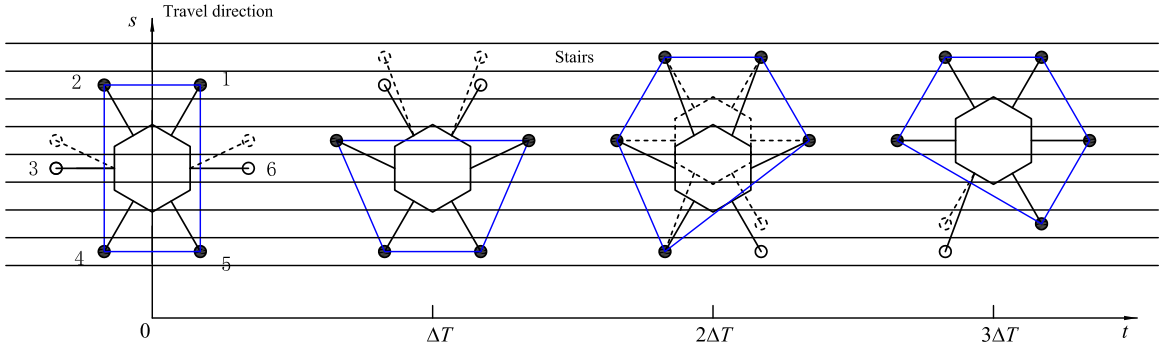

Fig. 5. Climbing stairs gait.

Fig. 6. An experiment of going up stairs.

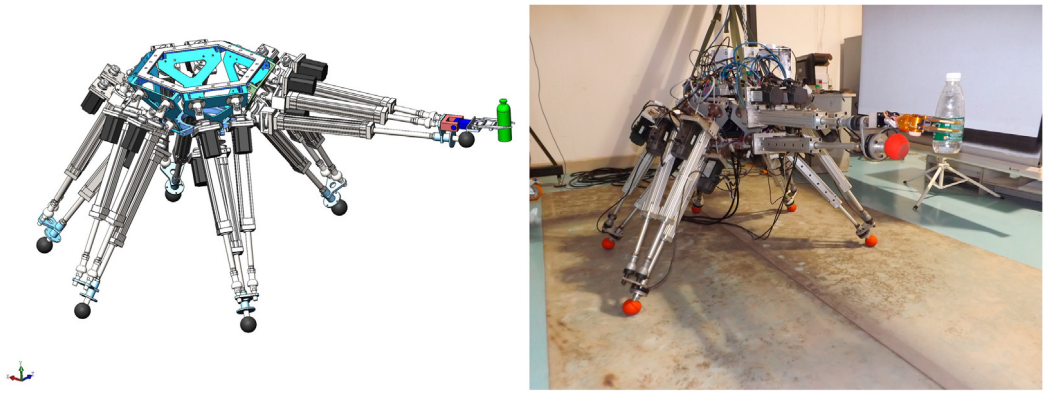

Fig. 7. Grab object using one leg.

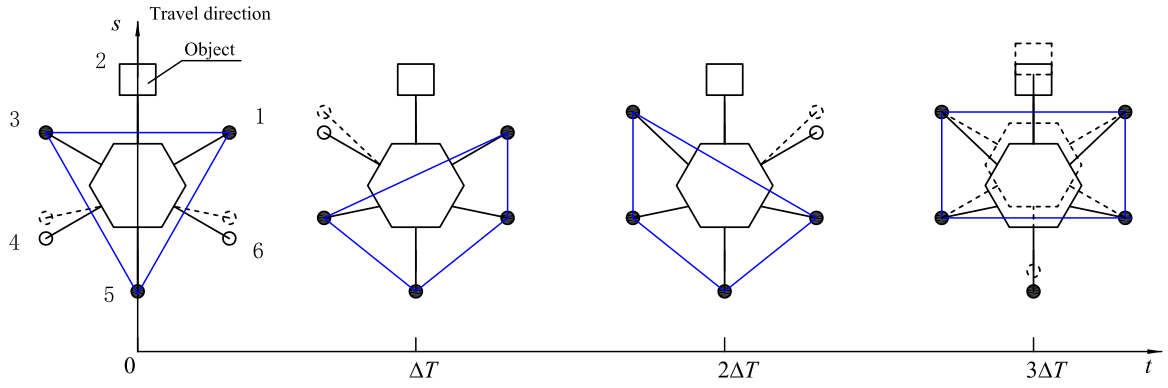

Fig. 8. Gait for walking with five legs and one leg grabbing an object.

one leg. It is easy for hexapod robot to grab and carry a small object using one leg. The gait for this case is depicted in Fig. 8. And we have implemented that gait successfully in experiment.

However, the gripper cannot grab some large objects like big boxes. Therefore, we propose to use two legs to grab large things as shown in Fig. 9. Obviously, it is difficult to lift off leg 3 or 6 when walking with the other four legs, because the projection of COG is out of the support triangle while only three legs remain as supporting legs. So, in this case, we must make planning of COG trajectory and track this trajectory strictly using close loop controller. The idea is to move COG into supporting triangle before lifting off leg 3 or 6 as shown in Fig. 10. However, there is no large static stability margin anymore because of the limited range of linear actuators as shown in Fig. 10(b). The problem for this case is not to design a gait and COG trajectory, but to implement this desired COG trajectory, i.e., how to calculate joint motion according to this desired COG trajectory. As mentioned before, the method treating COG motion as body motion, which has been widely used for humanoid robots, is not suitable for our robot. The COG position can be calculated by the following equation,

$$\mathbf{c} = \sum_{i=0}^{n} \sum_{k=1}^{n_i} \frac{m_{i,k}}{M} \mathbf{c}_{i,k} \tag{8}$$

where $\mathbf{c}_{i,k} \in \mathbb{R}^{3\times 1}$ denotes the position vector of center of mass of kth link in ith leg with respect to global frame, $i = 0$ means the COG of robot body, $m_{i,k}$ means the mass of kth link in ith leg, M means the total mass of the robot.

$$\begin{aligned} \mathbf{c} &= \frac{\sum_{i=0}^{n} m_i \mathbf{c}_i}{M} = \frac{\sum_{i=0}^{n} m_i (\mathbf{p}_0 + {}^{B}_{C}\mathbf{R}\,{}^{C}\mathbf{c}_i)}{M} \\ &= \mathbf{p}_0 + \sum_{i=0}^{n} {}^{B}_{C}\mathbf{R} \left(\sum_{k=1}^{n_i} \mu_{i,k}\, {}^{C}\mathbf{c}_{i,k} \right) \end{aligned} \tag{9}$$

where $\mathbf{c}_i$ is the position vector of center of mass of ith leg or body with respect to global frame, $\mu_{i,k} = m_{i,k}/M$. Then, differentiating Eq. (9) with respect to time, yields,

$$\begin{aligned} \dot{\mathbf{c}} &= \dot{\mathbf{p}}_0 + \sum_{i=0}^{n} {}^{B}_{C}\dot{\mathbf{R}} \left(\sum_{k=1}^{n_i} \mu_{i,k}\, {}^{C}\mathbf{c}_{i,k} \right) + \sum_{i=0}^{n} {}^{B}_{C}\mathbf{R} \left(\sum_{k=1}^{n_i} \mu_{i,k}\, {}^{C}\dot{\mathbf{c}}_{i,k} \right) \\ &= \dot{\mathbf{p}}_0 + \sum_{i=0}^{n} \left(\boldsymbol{\omega}_0 \times {}^{B}_{C}\mathbf{R} \right) \left(\sum_{k=1}^{n_i} \mu_{i,k}\, {}^{C}\mathbf{c}_{i,k} \right) \\ &\quad + \sum_{i=0}^{n} {}^{B}_{C}\mathbf{R} \left(\sum_{k=1}^{n_i} \mu_{i,k} \frac{\partial {}^{C}\mathbf{c}_{i,k}}{\partial \mathbf{q}_i} \dot{\mathbf{q}}_i \right) \\ &= \dot{\mathbf{p}}_0 + \boldsymbol{\omega}_0 \times (\mathbf{c} - \mathbf{p}_0) + \sum_{i=1}^{n} {}^{B}_{C}\mathbf{R}\, {}^{C}\mathbf{J}_{ci} \dot{\mathbf{q}}_i \end{aligned} \tag{10}$$

where ${}^{C}\mathbf{J}_{ci} = \sum_{k=1}^{n_i} \mu_{i,k} \frac{\partial \mathbf{c}_{i,k}}{\partial \mathbf{q}_i}$ is called as COG Jacobian of ith leg. Finally, we can get an equation with COG Jacobian of the full system as follows,

$$\dot{\mathbf{c}}_{fs}(\dot{\mathbf{c}},\ \dot{\mathbf{x}}_i) = \mathbf{J}_{fs} \dot{\mathbf{q}}_b \tag{11}$$

where $\mathbf{q}_b$ means the generalized coordinates of one of the supporting legs. Then, we can obtain the motion of one supporting leg,

$$\dot{\mathbf{q}}_b = \mathbf{J}_{fs}^{+} \dot{\mathbf{c}}_{fs}(\dot{\mathbf{c}},\ \dot{\mathbf{x}}_i). \tag{12}$$

With the definition of $\mathbf{X}_i = \begin{bmatrix} \mathbf{I} & -S({}^{B}\mathbf{p}_i) \end{bmatrix}$, Eq. (4) can be rewritten as

$$\dot{\mathbf{p}}_i = \mathbf{X}_i \dot{\mathbf{x}}_0 + {}^{B}\mathbf{J}_i \dot{\mathbf{q}}_i. \tag{13}$$

This yields,

$$\dot{\mathbf{x}}_0 = \mathbf{X}_i^{+} (\dot{\mathbf{p}}_i - {}^{B}\mathbf{J}_i \dot{\mathbf{q}}_i). \tag{14}$$

Obviously, for any other leg j, there is the following equation,

$$\dot{\mathbf{x}}_0 = \mathbf{X}_i^{+} (\dot{\mathbf{p}}_i - {}^{B}\mathbf{J}_i \dot{\mathbf{q}}_i) = \mathbf{X}_j^{+} (\dot{\mathbf{p}}_j - {}^{B}\mathbf{J}_j \dot{\mathbf{q}}_j). \tag{15}$$

Now, we can solve the motion of other legs based on $\dot{\mathbf{q}}_b$,

$$\dot{\mathbf{q}}_i = {}^{B}\mathbf{J}_i^{-1} (\dot{\mathbf{p}}_i + \mathbf{X}_i \mathbf{X}_b^{+} {}^{B}\mathbf{J}_b \dot{\mathbf{q}}_b). \tag{16}$$

So far, we get the motions of all of the active joints according to the motion of COG and the motion of end point of swing legs, i.e., $\dot{\mathbf{c}}$ and $\dot{\mathbf{p}}_i$. In other words, if we track these joints' motions obtained by Eqs. (12) and (16), the desired COG trajectory will be tracked very well. Actually, we have done this experiment successfully using this method.

4.2. Trajectory planning

In this section, we will address the issue of choosing functions, motion range, and time interval for trajectories of COG, body, and swing feet. Firstly, note that hexapod robot cannot keep a constant speed to walk because of the alternation of supporting phase and swing phase. That means the initial and end velocities of trajectories of COG, body and swing feet should be zero. In order to obtain a smooth motion for actuators, the initial and end accelerations of the planned trajectory are chosen to be zero as well. These are the boundary conditions for trajectory planning.

In our earlier works [1,3], we adopted cycloid function that satisfies the boundary conditions abovementioned. However, that function cannot be used for segmented function. Therefore, we choose a more flexible function that is fifth order polynomial as the trajectory function for the convenience of rolling planning. We can

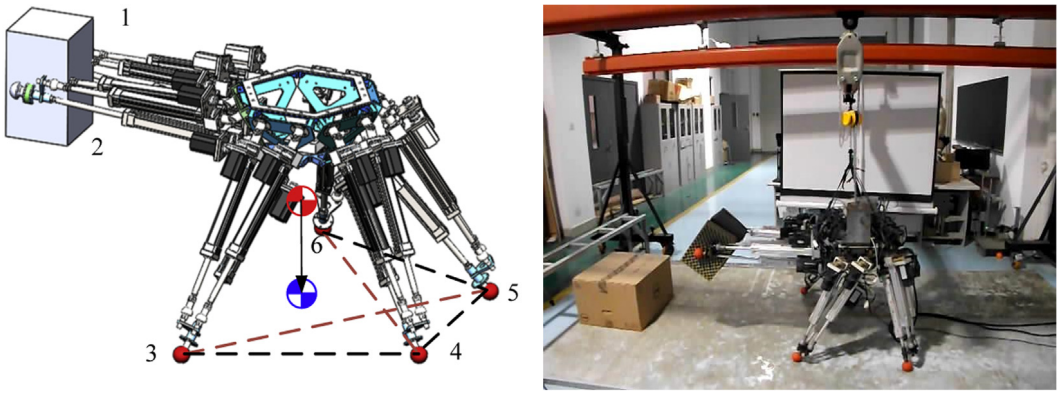

Fig. 9. Grab large object using two legs.

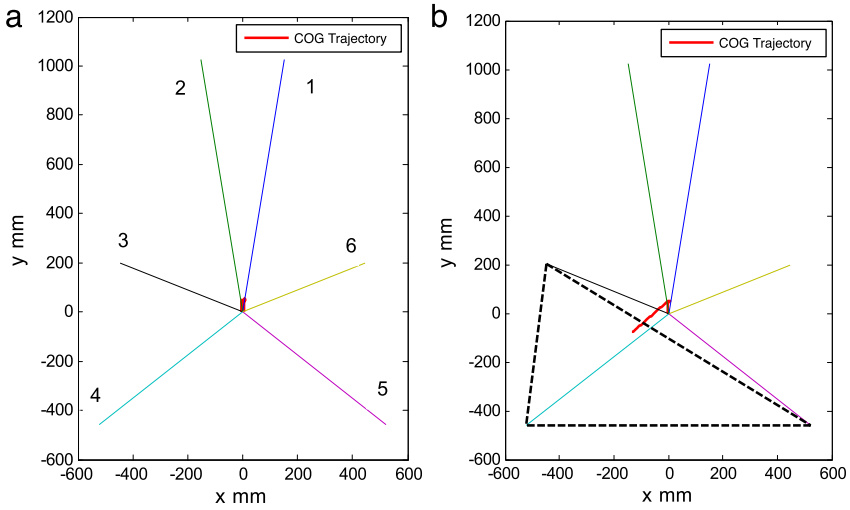

Fig. 10. Postures of PH-Robot for grabbing using two legs from up view: (a) initial posture, (b) desired COG trajectory before moving leg 6.

derive the formulation of that function according to the boundary conditions as follows,

$$q(t) = q_0 + \frac{10(q_{\Delta t} - q_0)}{\Delta t^3}t^3 + \frac{-15(q_{\Delta t} - q_0)}{\Delta t^4}t^4 + \frac{6(q_{\Delta t} - q_0)}{\Delta t^5}t^5 \quad t \in [0 \quad \Delta t] \tag{17}$$

where q_0 denotes the start value, $q_{\Delta t}$ denotes the end value, Δt means the time interval. The motions of 3-axis of a 3D trajectory in Cartesian coordinate system are defined by Eq. (17) respectively. As for foot trajectory, the height and stride constrained by the reachable workspace are determined by $q_{\Delta t} - q_0$. Moreover, Eq. (17) is also constrained by motor's maximum speed and torque.

Some researchers [42–44] try to find efficient optimization methods to solve this problem for real-time application. But we think it is not necessary to optimize the trajectory when walking. In fact, we can find some optimal trajectories offline under those constraints to be used as basic trajectories for rolling planning online. Compared with the velocity and torque limitation, we think the reachable workspace is the crucial constraint for online trajectory planning, because servo motors have overload ability while mechanisms never cannot reach outside reachable workspace. Therefore, it is necessary to ensure the end point of a trajectory satisfying the workspace constraint, and the time interval can be modified slightly according to the offline optimized trajectory.

4.3. Compliance control

The vision system, i.e., LIDAR sensor, cannot get absolutely precise map of the environment. Therefore, it is unavoidable that feet of swing legs contact with the ground before the end of planned trajectory, or feet still do not contact with ground after the end of planned trajectory. For the first case, we employ impedance control to realize smooth touch with environment. For the second case, we will use trajectory rolling plan method to get a new trajectory for this leg in order to get touch with ground. But we still do not know the distance between foot and ground. So, we must use impedance control to get contact with ground regardless of which cases.

We design an impedance controller based on position adjustment as the outer loop, where the mapping between the force error and position adjustment of the foot are established. Install a three-dimensional force sensor on the ankle to make a comparison between measured force $\mathbf{F}_{act}$ and desired force $\mathbf{F}_d$, then input the force error $\Delta\mathbf{F}$ into the impedance controller to generate a position adjustment value $\Delta\mathbf{X}$. Based on the impedance control theory we can obtain

$$\mathbf{M}_d\Delta\ddot{\mathbf{X}} + \mathbf{B}_d\Delta\dot{\mathbf{X}} + \mathbf{K}_d\Delta\mathbf{X} = \Delta\mathbf{F} \tag{18}$$

where $\Delta\mathbf{F} = \mathbf{F}_d - \mathbf{F}_{act}$, $\mathbf{M}_d$, $\mathbf{B}_d$, and $\mathbf{K}_d$ are inertia, damping, and elastic coefficient matrices, respectively. Moreover, all of them are diagonal matrices. $\Delta\mathbf{X}$ will be the feedback for foot trajectory and chance the desired trajectory in Cartesian space. In other words, impedance control is a kind of compliance control that

obtains compliance property through trajectory rolling planning, i.e., changing desired trajectory.

After Laplace transform, the mapping between the force error and position adjustment can be written as

$$\Delta \mathbf{X}(s) = \frac{\Delta \mathbf{F}(s)}{\mathbf{M}_d s^2 + \mathbf{B}_d s + \mathbf{K}_d}. \tag{19}$$

Then, by adding the position adjustment value $\Delta \mathbf{X}$ to the reference trajectory χ_d, we can obtain the new desired trajectory as follows

$$\hat{\chi}_d = \chi_d + \Delta \mathbf{X}. \tag{20}$$

4.4. Stability observer based on ZMP

Stability margin based ZMP has been widely used over the world although it is not a perfect characterization method for robots' stability. We dropped this method in our earlier work [35] because this method cannot include the effect for stability caused by the height of COG, i.e., stability margin will decrease with the increasing of COG height. But this is not a critical defect for a method of stability measurement. Additionally, ZMP is easier to be calculated through the measurement results of force sensors mounted on ankles. Based on these reasons, we will employ ZMP as stability measurement method in this work.

ZMP means zero moment point on the ground. There are two ways to calculate ZMP. One is to use state variables of the robot to calculate ZMP as follows,

$$x_{ZMP} = \frac{\sum_{i=1}^{n} x_i m_i (\ddot{z}_i + g_z) - \sum_{i=1}^{n} z_i m_i (\ddot{x}_i + g_x)}{\sum_{i=1}^{n} m_i (\ddot{z}_i + g_z)} \tag{21}$$

$$y_{ZMP} = \frac{\sum_{i=1}^{n} y_i m_i (\ddot{z}_i + g_z) - \sum_{i=1}^{n} z_i m_i (\ddot{y}_i + g_y)}{\sum_{i=1}^{n} m_i (\ddot{z}_i + g_z)} \tag{22}$$

where x_i, y_i, z_i and m_i are coordinates of center of mass with respect to global frame and mass for ith link. (g_x, g_y, g_z) is the vector of acceleration of gravity in terms of global frame. This method is more suitable for simulation because it is difficult to obtain these state variables in physical experiments. Usually, we would like to use measured forces of feet from ground to calculate ZMP as follows,

$$x_{ZMP} = \frac{\sum_{i=1}^{n} F_{zi} x_i}{\sum_{i=1}^{n} F_{zi}} \tag{23}$$

$$y_{ZMP} = \frac{\sum_{i=1}^{n} F_{zi} y_i}{\sum_{i=1}^{n} F_{zi}} \tag{24}$$

where x_i, y_i, F_{zi} are coordinates of contact point and force component of z axis with respect to global frame. Actually, ZMP is equal to center of pressure (COP) when it is within the supporting polygon. In order to use ZMP to indicate stability of robot, we need to define a type of stability margin using ZMP. The distances between ZMP and the edges of supporting polygon can represent stability margin numerically as shown in Fig. 11. Thus, the ZMP stability margin is defined and normalized as follows

$$\alpha_{ZMP} = \frac{\min(h_1, \ldots, h_n)}{H} \tag{25}$$

where H means the distance from origin to one of the edges, i.e., the distance in ideally initial status, h_i is the actual distance from ZMP to ith edge of supporting polygon. According to this definition, $\alpha_{ZMP} \in [0,\ 1]$.$\alpha_{ZMP} = 0$ means the robot is unstable and is going to tip over.

The Kalman filter is utilized to process the signal of force sensors. As we know, Kalman filter has five steps for one iteration.

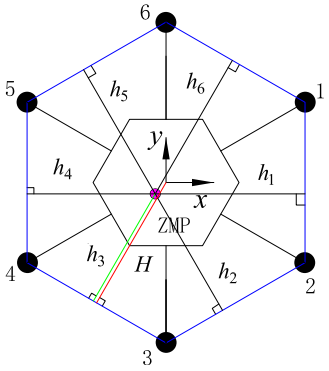

Fig. 11. Schematic diagram for ZMP stability margin.

Among these five steps, predicting step is to predict the state values of next sampling time. The equation for this step is as follows,

$$\mathbf{X}(k|k-1) = \mathbf{A}\mathbf{X}(k-1|k-1) \tag{26}$$

where $\mathbf{X}(k|k-1)$ means the predicted state values based on state values of time $k-1$, $\mathbf{X}(k-1|k-1)$ means the estimated state values of time $k-1$ by Kalman filter. We can use the predicted state values to calculate ZMP, and then can get the predicted ZMP stability margin. It is significant to know stable state of the robot ahead of one sample interval because we need time to response instability and adjust gait to keep balance. Therefore, the utilization of Kalman filter can obtain predicted stability as well as filtered signal.

5. Experiments and discussion

First of all, it should be noted that the computer vision system for our PH-Robot has not been exploited completely. We cannot test the robot in a real disaster scene since the robot needs vision system to recognize the environments to change gait and trajectory. Therefore, we have made assessment for different functionality of the proposed control architecture in different experiment with a certain scenario separately. For example, the assessment for climbing stairs gait was based on presetting information of the position and size of stairs. In this section, we will present two typical scenarios where one can test the compliance ability of our controller while the other one aims at the assessment of stability state observer under extreme condition in which the robot is pushed by human beings.

5.1. Traversing rough terrain

In this experiment, our robot gone through rough terrain with bricks successfully using the proposed control architecture. Although this is not a real disaster environment, it is good enough to test the abilities of trajectory planning, compliance control and stability observing. Fig. 12 shows the sequence of snapshots of the PH-Robot crossing rough terrain with several bricks. Tripod gait was employed as the default gait through the entire experiment. When one foot stepped on a brick, the originally planned trajectory has not been tracked to the end, and the compliance controller therefore is executed to stop the foot sticking the brick while the other two swing legs continue to track the rest of the original trajectories. Meanwhile, the ZMP observer did not predict the robot will lose stability due to the brick. Consequently, it did not change gait to keep balance. Fig. 13 shows the compliance result of one foot when stepping on a brick. These trajectories in Fig. 13 was calculated by forward kinematics based on the measurement

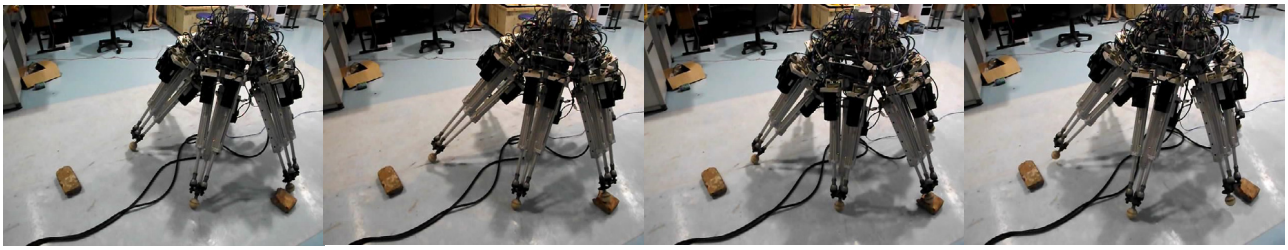

Fig. 12. Snapshots of the PH-Robot crossing rough terrain using the proposed controller.

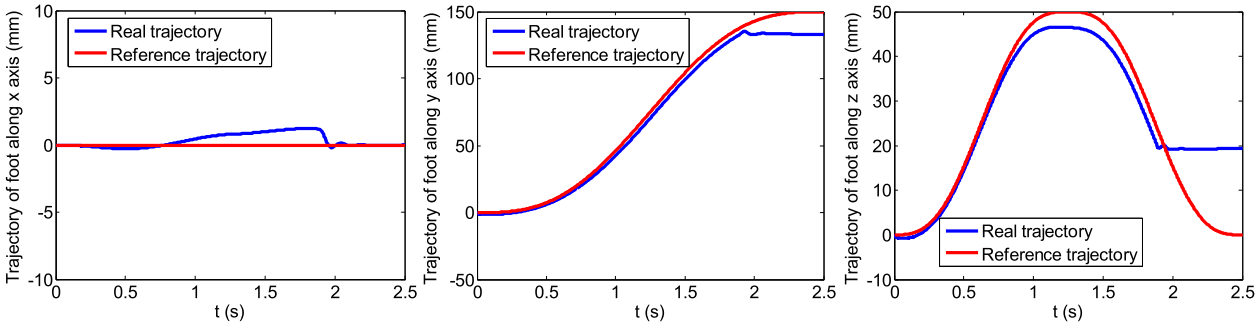

Fig. 13. Compliance control performance illustrated by foot trajectory.

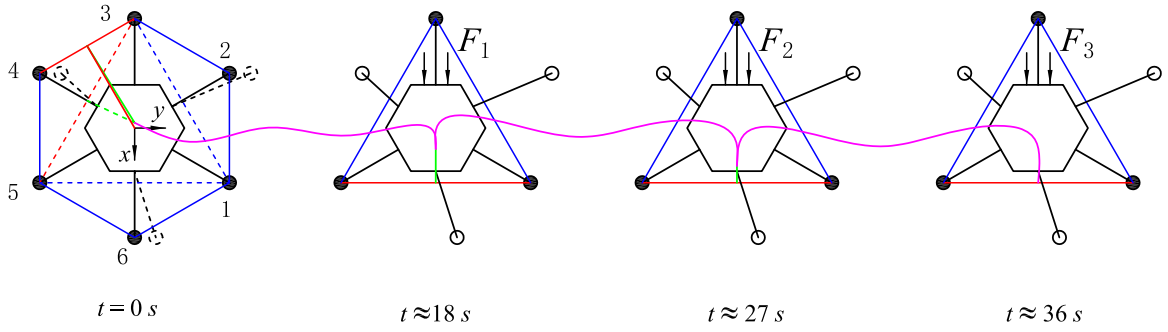

Fig. 14. Schematic diagram of pushing the robot three times while walking. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

results of motor encoders. The period of swing phase is 2.5 s. From Fig. 13, we can see that the trajectory of the foot almost did not change anymore after 1.8 s because the execution of compliance controller. The height of brick is about 20 mm. Hence, the displacement of foot along z axis is also about 20 mm as shown in the third graph in Fig. 13. Consequently, this experiment proves the effectiveness of the proposed method when applied to traversing uneven terrains. On the other hand, this experiment shows the advantage of hexapod robot in term of inherent stability compared with humanoid robot, and it also demonstrates the practicability on traveling rough terrain. It also should be noted that we did not use LIDAR to detect the environment in this experiment. The robot is blind and only depends on touch.

5.2. Pushing recovery

We also want to assess the resistance ability of our robot for external impact force. In this experiment, we pushed PH-Robot for three times with increasing magnitude of pushing force while the robot was walking using tripod gait as shown in Figs. 14 and 15. In Fig. 14, $\min(h_1, \ldots, h_n)$ is indicated by green line. The edge that most probably becomes the tipping over axis is with pink color. The claret line indicates the ZMP trajectory roughly. Obviously, ZMP stability margin will occur discontinuity while alternating

Fig. 15. Push the robot.

supporting phase and swing phase. We pushed the robot at 18 s, 27 s and 36 s for three times with pushing forces as $F_3 > F_2 > F_1$.

We can see that the force of foot 3 decreased when pushing the robot as shown in Fig. 16(a). And for the third pushing time, foot 3 leaved ground for a while, which means only two feet (i.e., foot 1

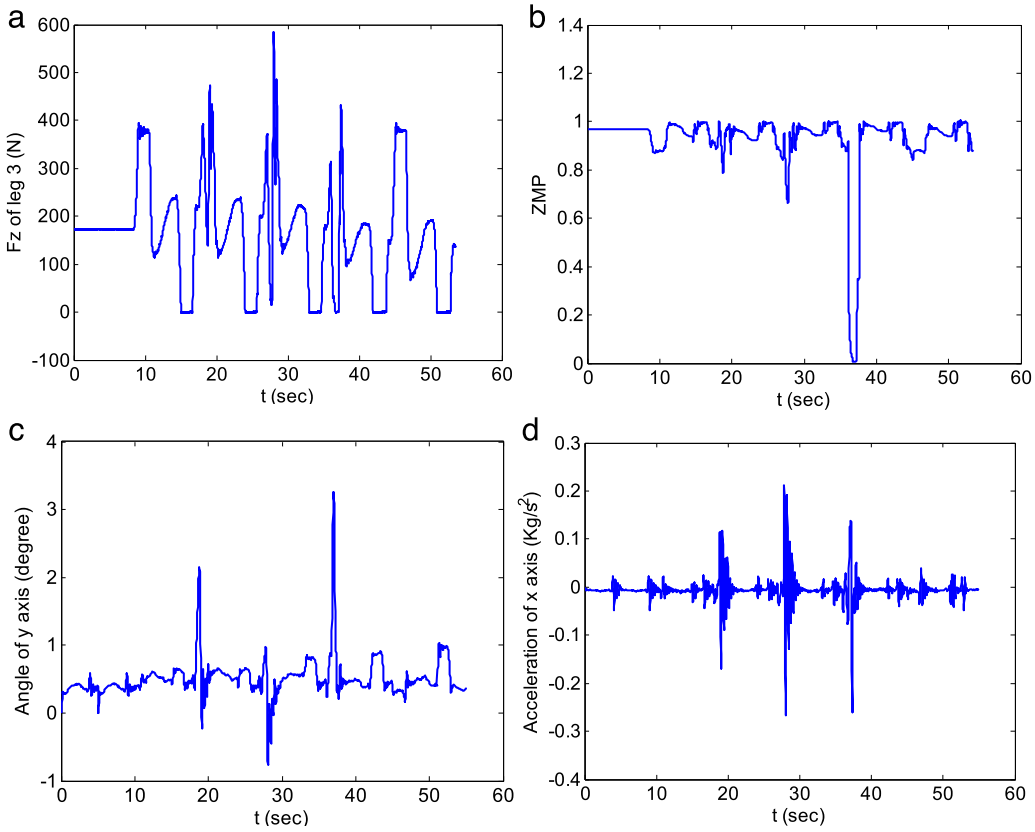

Fig. 16. Measured data in pushing experiment: (a) component of z axis of contact force for foot 3, (b) ZMP stability margin for this experiment, (c) rotation angle around tipping over axis, (d) acceleration of x axis of robot body.

and 5) still on ground. Therefore, the ZMP stability margin comes to zero during this period as shown in Fig. 16(b). We also can see ZMP stability margin decreased for the other two pushing times from Fig. 16(b). These results verify the effectiveness of the detection method for stability and sensitiveness of ZMP stability margin for external disturbance as well. By contrast, rotation angle and acceleration of robot body measured by IMU as shown in Fig. 16(c) and (d) cannot indicate the stable status of the robot precisely. But they can indicate the existence of external impact force as well.

It should be noted that we did not make any adjustment of gait and trajectory during this experiment. The robot still can recover stable after effect of impact forces. That is because our robot is so heavy and the structure is like a pyramid. So, as mentioned before, the robot has large static stability margin when walking using tripod gait. But we think it is safer to change gait or trajectory in order to avoid external force. Actually, we set a threshold value of ZMP stability margin for adjustment of gait and trajectory to get more stable status.

5.3. Discussion

Based on the above experiments and theoretical analysis, our control strategy is much easier to implement compared with some other existing methods proposed by other researchers. Firstly, that benefits from the huge inherent static stability of hexapod robots. Our controller does not need to use optimization method to get optimal trajectories or foothold position as those are done in [4,11]. Although there is a case, i.e., carrying objects using two legs as shown in Section 4.1.2, in which COG trajectory and joint trajectories should be carefully planned and calculated, it is still based on kinematic method instead of time-consuming optimization method. Secondly, essentially, our control architecture is a model-free method that also results in high online computation efficiency. An important aspect is that the impedance control does not depend on dynamic model whereas many works like [31,45,46] try to get a precise dynamic model to implement joint impedance control or Cartesian impedance control. In other words, they are trying to do joint torque control that is very dependent on the accuracy of dynamics. However, a generally acknowledged fact is that it is difficult to obtain an accurate dynamic model for multi-DOFs robots due to some uncertainties like friction forces. The advantage of model-based compliance control is that it does not need force sensors to detect external force, which is priceless when the external forces act on body or legs rather than feet. It cannot be denied that we also want to implement this kind of method on our PH-Robot. But the method proposed in this paper is more reliable and effective according to present situation because of the extreme complexity of dynamic equations for parallel mechanisms.

6. Conclusions

This paper addresses the issue of control strategy for a hexapod robot applied for disaster rescue in dynamic environments. The control strategy is included in a complicated control architecture. The core idea is to change gait and trajectory according to environment and robot status in order to keep balance and achieve assignment goals. Therefore, we have implemented various gaits for different situation. Among these gaits, there are not only simple gaits but also very complex gaits like carrying object gait that even needs COG trajectory planning and control. Compliance control and stability observation are other critical aspects for a walking robot. Of cause, for a practical disaster rescue robot, every part of the control architecture is very important. Therefore, this paper is much based on our previous works, such as inverse kinematics and joint servo control. These works make our robot traverse complicated terrain successfully. Experimental results can prove the effectiveness of our controller for hexapod robots. We acknowledge

that hexapod robots are easier to control compared with humanoid robots because of hexapods' structure with stability properties. However, it also implies that from the current situation, hexapod robots are more practical to be used in real disaster scene.

Future works will deal with perception of environment and the cooperation of autonomic control and remote control. Navigation and positioning are also significant coming works.

Acknowledgments

This study was supported by the National Basic Research Program (973) of China (Grant No. 2013CB035504); the National Natural Science Foundation of China (Grant No. 51405515); the Hunan Provincial Natural Science Foundation of China (Grant No. 2017JJ3381); the China Scholarship Council (Grant No. 201606370091).

References

[1] G. Xin, H. Deng, G. Zhong, H. Wang, Hierarchical kinematic modelling and optimal design of a novel hexapod robot with integrated limb mechanism, Int. J. Adv. Robot. Syst. 12 (2015) 123.
[2] G. Xin, H. Deng, G. Zhong, H. Wang, Dynamic analysis of a hexapod robot with parallel leg mechanisms for high payloads, in: Proceedings of 10th Asian Control Conference, ASCC, 2015, pp. 1–6.
[3] G. Zhong, H. Deng, G. Xin, H. Wang, Dynamic hybrid control of a hexapod walking robot: experimental verification, IEEE Trans. Ind. Electron. 63 (2016) 5001–5011.
[4] S. Kuindersma, R. Deits, M. Fallon, A. Valenzuela, H. Dai, F. Permenter, T. Koolen, P. Marion, R. Tedrake, Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot, Auton. Robots 3 (40) (2016) 429–455.
[5] T. Koolen, S. Bertrand, G. Thomas, T. de Boer, T. Wu, J. Smith, J. Englsberger, J. Pratt, Design of a momentum-based control framework and application to the humanoid robot atlas, Int. J. Humanoid Rob. 13 (2016) 1650007.
[6] Honda Motor Co., Ltd., 2013, ASIMO. http://asimo.honda.com/.
[7] M. Kalakrishnan, J. Buchli, P. Pastor, M. Mistry, S. Schaal, Learning, planning, and control for quadruped locomotion over challenging terrain, Int. J. Robot. Res. 30 (2011) 236–258.
[8] C. Semini, V. Barasuol, J. Goldsmith, M. Frigerio, M. Focchi, Y. Gao, D. Caldwell, Design of the hydraulically-actuated, torque-controlled quadruped robot HyQ2Max, IEEE/ASME Trans. Mechatronics 22 (2017) 635–646.
[9] M. Hutter, C. Gehring, M.A. Höpflinger, M. Blösch, R. Siegwart, Toward combining speed, efficiency, versatility, and robustness in an autonomous quadruped, IEEE Trans. Robot. 30 (2014) 1427–1440.
[10] S. Kagami, T. Kitagawa, K. Nishiwaki, T. Sugihara, M. Inaba, H. Inoue, A fast dynamically equilibrated walking trajectory generation method of humanoid robot, Auton. Robots 12 (2002) 71–82.
[11] R. Tedrake, S. Kuindersma, R. Deits, K. Miura, A closed-form solution for real-time ZMP gait generation and feedback stabilization, in: 2015 IEEE-RAS 15th International Conference on Humanoid Robots, Humanoids, 2015, pp. 936–940.
[12] S. Hong, Y. Oh, D. Kim, B.J. You, Real-time walking pattern generation method for humanoid robots by combining feedback and feedforward controller, IEEE Trans. Ind. Electron. 61 (2014) 355–364.
[13] M. Kalakrishnan, J. Buchli, P. Pastor, M. Mistry, S. Schaal, Fast, robust quadruped locomotion over challenging terrain, in: 2010 IEEE International Conference on Robotics and Automation, 2010, pp. 2665–2670.
[14] J.Z. Kolter, M.P. Rodgers, A.Y. Ng, A control architecture for quadruped locomotion over rough terrain, in: 2008 IEEE International Conference on Robotics and Automation, 2008, pp. 811–818.
[15] E. Garcia, P.G. de Santos, On the improvement of walking performance in natural environments by a compliant adaptive gait, IEEE Trans. Robot. 22 (2006) 1240–1253.
[16] B.-H. Lee, I.-K. Lee, The implementation of the gaits and body structure for hexapod robot, in: 2001 IEEE International Symposium on Industrial Electronics Proceedings, 2001, pp. 1959–1964.
[17] J.M. Porta, E. Celaya, Reactive free-gait generation to follow arbitrary trajectories with a hexapod robot, Robot. Auton. Syst. 47 (2004) 187–201.
[18] M.S. Erden, K. Leblebicioğlu, Free gait generation with reinforcement learning for a six-legged robot, Robot. Auton. Syst. 56 (2008) 199–212.
[19] Z. Wang, X. Ding, A. Rovetta, A. Giusti, Mobility analysis of the typical gait of a radial symmetrical six-legged robot, Mechatronics 21 (2011) 1133–1146.
[20] U. Asif, J. Iqbal, On the improvement of multi-legged locomotion over difficult terrains using a balance stabilization method, Int. J. Adv. Robot. Syst. 9 (2012) 1.
[21] T. Kawata, K. Kamiyama, M. Kojima, M. Horade, Y. Mae, T. Arai, Fault-tolerant adaptive gait generation for multi-limbed robot, in: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, 2016, pp. 3381–3386.
[22] J.M. Yang, J.H. Kim, A fault tolerant gait for a hexapod robot over uneven terrain, IEEE Trans. Syst. Man Cybern. Part B Cybern. 30 (2000) 172–180.
[23] S.K.K. Chu, G.K.H. Pang, Comparison between different model of hexapod robot in fault-tolerant gait, IEEE Trans. Syst. Man Cybern.- Part A: Syst. Humans 32 (2002) 752–756.
[24] U. Asif, Improving the navigability of a hexapod robot using a fault-tolerant adaptive gait, Int. J. Adv. Robot. Syst. 9 (2012) 34.
[25] D. Grzelczyk, B. Stańczyk, On the hexapod leg control with nonlinear stick-slip vibrations, Appl. Mech. Mater. (2015).
[26] D. Grzelczyk, B. Stańczyk, J. Awrejcewicz, Prototype, control system architecture and controlling of the hexapod legs with nonlinear stick–slip vibrations, Mechatronics 37 (2016) 63–78.
[27] D. Grzelczyk, B. Stanczyk, J. Awrejcewicz, Kinematics, dynamics and power consumption analysis of the hexapod robot during walking with tripod gait, Int. J. Struct. Stab. Dyn. 17 (2017) 1740010.
[28] Z.-Y. Wang, X.-L. Ding, A. Rovetta, Analysis of typical locomotion of a symmetric hexapod robot, Robotica 28 (2010) 893–907.
[29] J.R. Rebula, P.D. Neuhaus, B.V. Bonnlander, M.J. Johnson, J.E. Pratt, A controller for the littledog quadruped walking on rough terrain, in: Proceedings 2007 IEEE International Conference on Robotics and Automation, 2007, pp. 1467–1473.
[30] U. Asif, J. Iqbal, Motion planning using an impact-based hybrid control for trajectory generation in adaptive walking, Int. J. Adv. Robot. Syst. 8 (2011) 53.
[31] M. Mistry, J. Buchli, S. Schaal, Inverse dynamics control of floating base systems using orthogonal decomposition, in: 2010 IEEE International Conference on Robotics and Automation, 2010, pp. 3406–3412.
[32] D. Berio, S. Calinon, F.F. Leymarie, Learning dynamic graffiti strokes with a compliant robot, in: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, 2016, pp. 3981–3986.
[33] I.E. Makrini, C. Rodriguez-Guerrero, D. Lefeber, B. Vanderborght, The variable boundary layer sliding mode control: A safe and performant control for compliant joint manipulators, IEEE Robot. Autom. Lett. 2 (2017) 187–192.
[34] G. Strang, Linear Algebra and Its Applications, third ed., Harcourt Brace Jovanovich College Publishers, Orlando, 1988.
[35] T. Sugihara, Y. Nakamura, Whole-body cooperative balancing of humanoid robot using COG Jacobian, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2002, pp. 2575–2580.
[36] Y. Choi, D. Kim, B.-J. You, On the walking control for humanoid robot based on the kinematic resolution of CoM Jacobian with embedded motion, in: Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006, 2006, pp. 2655–2660.
[37] S. Long, G. Xin, H. Deng, G. Zhong, An improved force-angle stability margin for radial symmetrical hexapod robot subject to dynamic effects, Int. J. Adv. Robot. Syst. 12 (2015) 59.
[38] S.M. Song, B.S. Choi, The optimally stable ranges of 2n-legged wave gaits, IEEE Trans. Syst. Man Cybern. 20 (1990) 888–902.
[39] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, H. Hirukawa, Biped walking pattern generation by using preview control of zero-moment point, in: 2003 IEEE International Conference on Robotics and Automation, ICRA, 2003, pp. 1620–1626.
[40] J. Liu, M. Veloso, Online ZMP sampling search for biped walking planning, in: 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008, pp. 185–190.
[41] K. Erbatur, O. Kurt, Natural ZMP trajectories for biped robot reference generation, IEEE Trans. Ind. Electron. 56 (3) (2009) 835–845.
[42] R. Lampariello, D. Nguyen-Tuong, C. Castellini, G. Hirzinger, J. Peters, Trajectory planning for optimal robot catching in real-time, in: 2011 IEEE International Conference on Robotics and Automation, 2011, pp. 3719–3726.
[43] S. Macfarlane, E.A. Croft, Jerk-bounded manipulator trajectory planning: design for real-time applications, IEEE Trans. Robot. Autom. 19 (2003) 42–52.
[44] A. Gasparetto, V. Zanotto, A technique for time-jerk optimal planning of robot trajectories, Robot. Comput.-Integr. Manuf. 24 (2008) 415–426.
[45] A. Dietrich, K. Bussmann, F. Petit, P. Kotyczka, C. Ott, B. Lohmann, A. Albu-Schäffer, Whole-body impedance control of wheeled mobile manipulators, Auton. Robots 40 (2016) 505–517.
[46] B. Henze, M.A. Roa, C. Ott, Passivity-based whole-body balancing for torque-controlled humanoid robots in multi-contact scenarios, Int. J. Robot. Res. 35 (2016) 1522–1543.

Hua Deng received the B.E. degree in electrical engineering from Nanjing Aeronautical Institute, Nanjing, China, in 1983, the M.E. degree in electrical engineering from Northwestern Polytechnical University, Xi'an, China, in 1988, and the Ph.D. degree in manufacturing engineering from City University of Hong Kong, Kowloon, Hong Kong, in 2005.

He is currently a Professor with the School of Mechanical and Electrical Engineering, Central South University, Changsha, China. His research interests include intelligent control and learning, robotic dynamics and control, biomechatronics, and modeling and control of complex distributed parameter systems.

Guiyang Xin received the B.S. degree in mechanical engineering from China University of Geosciences, Wuhan, China, in 2012, and he is currently working toward the Ph.D. degree in mechanical and electronic engineering at Central South University, Changsha, China. Guiyang is also a joint Ph.D. student in robotics at School of Informatics, University of Edinburgh. His research interests include motion planning, compliance control, robotic dynamics, and mathematical modeling.

Guoliang Zhong received the B.S. degree in measurement and control technology and instrumentation from Hunan University of Science and Technology, Xiangtan, China, in 2007, the M.S. degree in mechanical design and theory from South China University of Technology, Guangzhou, China, in 2010, and the Ph.D. degree in human mechanical systems and design from Hokkaido University, Sapporo, Japan, in 2013.

He is an Assistant Professor with the State Key Laboratory of High-Performance Complex Manufacturing and the School of Mechanical and Electrical Engineering, Central South University. His research interests include mechatronic and robotic systems, intelligent modeling and control, nonlinear dynamics.

Michael Mistry is a Reader in robotics at School of Informatics, University of Edinburgh, where he is also a member of the Edinburgh Centre for Robotics. Michael is broadly interested in legged robotics, particularly in human motion and humanoid robotics. His research focuses on issues relevant to dexterous movement in legged robots, including redundancy resolution and inverse kinematics, operational space control and manipulation, stochastic optimal control, and internal model learning and control, particularly in environmental contact. Previously, Michael has been a Senior Lecturer in robotics at School of Computer Science, University of Birmingham, a postdoc at the Disney Research Lab at Carnegie Mellon University, a researcher at the ATR Computational Neuroscience Lab and a Ph.D. student in Stefan Schaals CLMC lab at the University of Southern California.