UDC 621.391:62.58:519.68
Preliminary communication
Received: 17.01.2008.

Trajectory planning of jumping over an obstacle for one-legged jumping robot

Zhaohong Xu, **Tiansheng Lu**, **Xuyang Wang** and **Fang Ling**
School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai 200240, CHINA
e-mail: zhaohongxu@sjtu.edu.cn

SUMMARY

For one-legged passive jumping robot, a trajectory planning strategy is developed to jump over an obstacle integrating three various dynamics among jumping process. Manipulability ellipsoids are effective tools to perform task space analysis and motion optimization of redundant manipulators. Jumping robot can be considered as a redundant manipulator with a load held at the end-effector. The concept of inertia matching ellipsoid and directional manipulability is extended to optimize the take-off posture of jumping robot, and the optimized results have been used to plan jumping trajectory. Aimed at the sensitivity of a trajectory to constraint conditions on point-to-point motion planning, the 6^{th} order polynomial function is proposed to plan jumping motion having a better robustness to the parameters change of constraint conditions than traditional 5^{th} order polynomial function. In order to lift the foot over the obstacle, correction functions are constructed under unchanged boundary constraint conditions. Furthermore, the body posture is controlled based on internal motion dynamics and steady-state consecutive jumping motion principle. A prototype model is designed, and the effectiveness of the proposed method is confirmed via simulations performed on parameters of designed prototype.

***Key words**: jumping robot, inertia matching, trajectory planning, robust polynomial, correction function.*

1. INTRODUCTION

Inspired by the agility of animal and human locomotion, the number of researchers studying and developing legged robots has been increasing at a rapid rate over the last few decades. In multi-legged robots, coordination of legs is complex. Different gait patterns are admissible in multi-legged systems depending on the number of legs, terrain, and speed of locomotion, particular gaits are most efficient. One of the motivations to study one-legged robots is to gain a good understanding of system dynamics extending it to human and animal locomotion. Therefore, a number of researchers have focused on single-legged systems [1].

One-legged robot has only one gait, viz. jumping. Despite the great potential of jumping machines, their control and trajectory planning are still issues. Li and Montgomery [2] proposed a closed-loop strategy that could optimally control the body orientation of a one-legged robot during flight phase using the internal motion of the leg. They showed that it was possible to control the orientation of one-legged hopping robot by using angular momentum constraint during its flight phase. Based on the same principle, Lapshin [3] analyzed the motion control problem of one-legged hopper in the flight phase based on linear springy telescopic leg. He investigated how the body and leg orientation could be changed during the flight phase by applying small perturbations. They have controlled the body attitude by using angular momentum only in flight phase, not in the whole jumping process. Especially, the angular momentum is discontinuous at the landing. Harbick and Sukhatme [4] described a model-based height controller for a hopping robot in

the sagittal plane with a pneumatically powered leg. They derived the desired leg-length setting to regulate apex hopping height using a PD controller. Ohashi and Ohnishi proposed [5] a method of controlling the hopping height by changing the leg length at bottom taking account of torque limits of motors. They described a way to estimate the actual thrust force. Using the estimated thrust force, command value of leg length in the landing phase is determined. Babic and Omrcen [6] performed vertical jump simulations using three different control algorithms including ZMP constraints and showed the effects of each algorithm on the vertical jump performance. Higashimori et al. [7] obtained various jumping patterns with respect to the torque limitation for a fixed mass of the robot by applying a genetic algorithm for determining torque assignment. With the increase of the torque limitation, double leg based jump, single-leg based jump, and spring-type jump are generated for achieving the largest jump height. Although these papers described some valid methods to obtain maximum jumping height and control jumping height based on stability or dynamics constraints, they didn't study the take-off phase. Take-off has a large influence to jumping performance because ground reaction force determines jumping trajectory. De Man et al. [8, 9] developed a control algorithm for one-legged hopping robot with a telescopic leg. They made it possible to change a number of objective locomotion parameters from one hop to another, thus allowing for locomotion on an irregular terrain. Similarly, Vermeulen [9, 10] developed a real-time applicable control algorithm for a planar one-legged robot for locomotion on an irregular terrain based on the choice of a number of objective locomotion parameters. Although their trajectory was only determined by objective locomotion parameters, they did not describe how to choose appropriate parameters in real-time. The concept of kinematic manipulability ellipsoids [11] was introduced by Yoshikawa as a measure of the capability of a manipulator for executing a specific task in a given configuration. Then he extended this concept to dynamic manipulability ellipsoids [12] as a measure of the manipulator capabilities when the arm dynamics is taken into account. A number of interesting extensions and applications of manipulability ellipsoids has appeared in robotic conceptual design [13], global task space optimization of redundant manipulators [14] and coordination control for multi-arm system [15] and so on during the last few years. Although some manipulability ellipsoids have been used to solve robotic task space design and posture optimization, there are some unsolved problems in trajectory planning with those manipulability ellipsoids.

This paper will study a trajectory planning of jumping over an obstacle for passive multi-joint jumping robot. The initial constraint conditions are obtained and optimized with inertia matching and directional manipulability. The ultimate constraint conditions are attained with obstacle parameters and dynamics. Because the highest power polynomial coefficient is most sensitive influencing the shape of a polynomial, the 6^{th} order polynomial function is constructed to track joint angle trajectory based on point-to-point motion planning. Its highest power coefficient is obtained with joint workspace constraints, and the other power coefficients are determined with the initial and ultimate constraint conditions. This method has a better robustness to the parameters change of constraint conditions than traditional 5^{th} order polynomial function. The paper is organized as follows. Section 2 describes the dynamics model. Section 3 introduces inertia matching and directional manipulability to optimize take-off posture. Section 4 contains the strategies of trajectory planning for jumping over an obstacle. Section 5 gives simulation results. Section 6 provides some conclusions.

2. MATHEMATICAL MODEL

Jumping process can be divided into three phases based on constraint conditions, viz., stance phase, flight phase, and landing impact phase. Each phase has a different dynamics equation because of their different constraint conditions. So the dynamics of jumping motion belongs to the dynamics of a various constraint system.

To study the conceptual features of a jumping or running machine, such as underactuated and nonholonomic features, our model is a multi-body underactuated system in sagittal plane (Figure 1). It consists of four segments, a massless foot, a lower leg, an upper leg and an upper body. The actuation of the robot consists of a passive ankle, an active knee and an active hip.

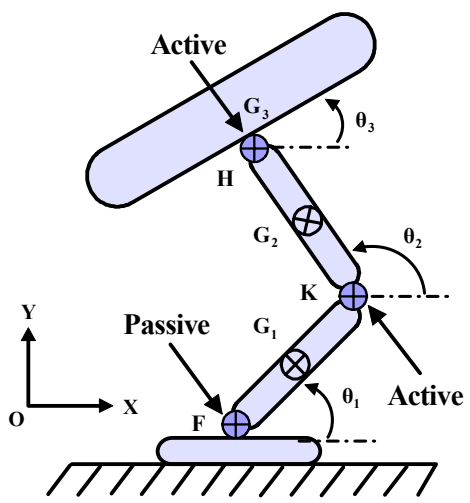

Fig. 1 One-legged underactuated jumping robot

Figure 1 depicts the robot geometry, where XOY is an inertia reference frame. The joint axis position of the connection between the upper body and the upper leg is referred to as the hip and is represented by point *H*. Analogously, the joint axis position of the connection between upper and lower leg is referred to as the knee and is represented by point *K*. The foot is called point *F*, and this point coincides with the ankle joint. Being a massless foot and underactuated ankle, the foot is considered as a point during flight.

The mass of the i^{th} rigid body is m_i, its length is l_i, and the moment of inertia around its center of mass G_i is I_i. The absolute angle between the horizontal axis and the i^{th} segment is θ_i, the direction of anticlockwise is positive. The location of the center of mass G_i of the leg is given by $\overline{FG_1} = \alpha_1 l_1$ and $\overline{KG_2} = \alpha_2 l_2$, where $0<\alpha_i<1$. The coordinates θ are the relative angles $(\theta_1, \theta_2)^T$, which describes the shape of the leg. The coordinates q are the relative angles $(\theta_1, \theta_2, \theta_3)^T$, which describes the shape of the robot. The robot's absolute position vector of COG $\boldsymbol{R}_G$ is specified by the Cartesian coordinates $(X_G, Y_G)^T$. The absolute position vector of foot $\boldsymbol{R}_F$ is specified by the Cartesian coordinates $(X_F, Y_F)^T$. During flight phase the vector of the generalized coordinates $\boldsymbol{q}_f$ can be denoted as $(\theta_1, \theta_2, \theta_3, X_F, Y_F)^T$, and during stance phase the vector of the generalized coordinates $\boldsymbol{q}_s$ can be denoted as $(\theta_1, \theta_2, \theta_3)^T$.

When the center of upper body G_3 does not coincide with the hip, the position of the global center of gravity G of the robot is a function of θ_1, θ_2 and θ_3. So the motion between leg motion and body rotation has a coupling. However, when G_3 coincides with the hip, the position of gravity G is a function only of θ_1 and θ_2. This condition leads to a decoupling between leg motion and body rotation. In that case the body rotation θ_3 can only be controlled by internal motion dynamics. In this paper, we study a model where G_3 coincides with the hip.

2.1 Dynamics equations during flight phase

The dynamics model can be derived in terms of the following generalized coordinates $\boldsymbol{q}_f$ by Lagrange equations:

$$\boldsymbol{D}_f(\boldsymbol{\theta})\ddot{\boldsymbol{q}}_f + \boldsymbol{H}_f\left(\boldsymbol{\theta},\dot{\boldsymbol{\theta}}\right)\dot{\boldsymbol{q}}_f + \boldsymbol{G}_f(\boldsymbol{\theta}) = \boldsymbol{B}_f\boldsymbol{\Gamma}_f \quad (1)$$

where $\boldsymbol{D}_f(\boldsymbol{\theta})$ is an inertia matrix, which is symmetric and positively defined, $\boldsymbol{H}_f\left(\boldsymbol{\theta},\dot{\boldsymbol{\theta}}\right)$ is a centrifugal matrix which contains the centrifugal acceleration and Coriolis terms, $\boldsymbol{G}_f(\boldsymbol{\theta})$ is a gravitational vector, $\boldsymbol{B}_f$ is a matrix $\begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \end{bmatrix}^T$, and $\boldsymbol{\Gamma}_f$ is external torque vector $\begin{bmatrix} \tau_K^f & \tau_H^f \end{bmatrix}^T$.

2.2 Dynamics equations during stance phase

During the stance phase, assuming a non-slippery rigid ground, the robot has three DOFs. Due to the fact that no foot torque is applied, the robot is still an underactuated mechanism, with one degree of underactuation. The equations of motion for the stance phase are found analogously as in flight phase. The Lagrange equations are now expressed in terms of the following generalized coordinates $\boldsymbol{q}_s$:

$$\boldsymbol{D}_s(\boldsymbol{\theta})\ddot{\boldsymbol{q}}_s + \boldsymbol{H}_s\left(\boldsymbol{\theta},\dot{\boldsymbol{\theta}}\right)\dot{\boldsymbol{q}}_s + \boldsymbol{G}_s(\boldsymbol{\theta}) = \boldsymbol{B}_s\boldsymbol{\Gamma}_s \quad (2)$$

3. INERTIA MATCHING

The concept of inertia matching [16] is widely used in the analysis of actuator and gear systems, primarily for selection of the optimum gear ratio based on the transmission performance between the torque produced at the actuator and the torque applied to the load. In this process, the performance of torque transmission is maximized by setting the optimal balance of inertial properties between the actuator system (including inertia of the rotor and shaft) and the load. The concept of inertia matching for jumping robot is proposed in this paper as a new index of the dynamic performance. The proposed inertia matching ellipsoid characterizes the dynamic torque-force transmission efficiency between joint actuators and a load held by the end-effector of a manipulator, encompassing a wide range of previous concepts.

3.1 Inertia matching for jumping robot

The concept of inertia matching can be extended to humanoid jumping robot as follows. Jumping robot can be considered as a redundant manipulator with a load held at the end-effector [17, 18]. Figure 2 shows the jumping robot and inertia matching ellipsoid (IME).

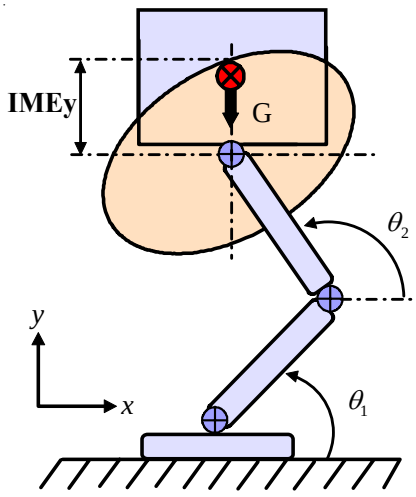

Fig. 2 Jumping robot and inertia matching ellipsoid (IME)

The motion and force equation analyzing the load held at the end-effector can be written as:

$$\boldsymbol{F}_e = m_l\left(\ddot{\boldsymbol{R}}_E + \boldsymbol{g}\right) \tag{3}$$

where $\boldsymbol{R}_E$ is the position vector of the end-effector, m_l is the mass inertia of load, $\boldsymbol{g} = (0,-g)^T$ is the gravity vector. The end-effector posture of manipulator $\boldsymbol{R}_E$ is related to the shape of the robot $\boldsymbol{\theta}$ as following:

$$\begin{cases} \boldsymbol{R}_E = \begin{bmatrix} X_E & Y_E \end{bmatrix}^T = \boldsymbol{E}_E(\boldsymbol{\theta}) \\ \dot{\boldsymbol{R}}_E = \boldsymbol{J}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} \\ \ddot{\boldsymbol{R}}_E = \boldsymbol{J}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \dot{\boldsymbol{J}}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} \end{cases} \tag{4}$$

where $\boldsymbol{J}(\boldsymbol{\theta})$ is the Jacobian matrix.

Jumping performance is affected by any motion control and jumping posture. Jumping posture affects not only the attitude during flight phase, but the angular momentum with respect to COG. When an external moment and force is applied to the jumping robot, with reference to Eq. (2) the dynamics equation for stance phase can be written as:

$$\boldsymbol{D}_s(\boldsymbol{\theta})\ddot{\boldsymbol{q}} + \boldsymbol{H}_s\left(\boldsymbol{\theta},\dot{\boldsymbol{\theta}}\right)\dot{\boldsymbol{q}} + \boldsymbol{G}_s(\boldsymbol{\theta}) + \boldsymbol{J}(\boldsymbol{\theta})^T\boldsymbol{F}_e = \boldsymbol{B}_s\boldsymbol{\Gamma}_s \tag{5}$$

where $\boldsymbol{F}_e$ is the external moment and force.

Substituting Eqs. (3) and (4) into Eq. (5), the torque matrix can be obtained by:

$$\begin{aligned} \boldsymbol{\Gamma}_s = \boldsymbol{B}_s^{\dagger}\Big[& \boldsymbol{D}_s(\boldsymbol{\theta})\boldsymbol{J}(\boldsymbol{\theta})^{\dagger}\left(\boldsymbol{F}_e - m_l\boldsymbol{g} - m_l\dot{\boldsymbol{J}}(\boldsymbol{\theta})\dot{\boldsymbol{q}}\right)\Big/m_l + \\ & + \boldsymbol{H}_s\left(\boldsymbol{\theta},\dot{\boldsymbol{\theta}}\right)\dot{\boldsymbol{q}} + \boldsymbol{G}_s(\boldsymbol{\theta}) + \boldsymbol{J}(\boldsymbol{\theta})^T\boldsymbol{F}_e\Big] = \\ & = \boldsymbol{Q}(\boldsymbol{\theta})\left(\boldsymbol{F}_e - \boldsymbol{F}_{bias}\right) \end{aligned} \tag{6}$$

where:

$$\begin{cases} \boldsymbol{Q}(\boldsymbol{\theta}) = \boldsymbol{B}_s^{\dagger}\left[\boldsymbol{J}(\boldsymbol{\theta})^T + \boldsymbol{D}_s(\boldsymbol{\theta})\boldsymbol{J}(\boldsymbol{\theta})^{\dagger}/m_l\right] \\ \boldsymbol{F}_{bias} = \boldsymbol{W}\left(\boldsymbol{J}(\boldsymbol{\theta})^T + \boldsymbol{D}_s(\boldsymbol{\theta})\boldsymbol{J}(\boldsymbol{\theta})^{\dagger}/m_l\right)^{\dagger} \times \\ \qquad \left[\boldsymbol{D}_s(\boldsymbol{\theta})\boldsymbol{J}(\boldsymbol{\theta})^{\dagger}\left(\boldsymbol{g} + \dot{\boldsymbol{J}}(\boldsymbol{\theta})\dot{\boldsymbol{q}}\right) - \boldsymbol{H}_s\left(\boldsymbol{\theta},\dot{\boldsymbol{\theta}}\right)\dot{\boldsymbol{q}} - \boldsymbol{G}_s(\boldsymbol{\theta})\right] \end{cases} \tag{7}$$

Here, $\boldsymbol{F}_{bias}$ is the bias force matrix of angular velocity and acceleration, $\boldsymbol{F}_e$–$\boldsymbol{F}_{bias}$ is the interia matching for jumping robot, and $\boldsymbol{J}(\boldsymbol{\theta})^{\dagger}$ is a pseudoinverse of the Jacobian matrix $\boldsymbol{J}(\boldsymbol{\theta})$. When the Jacobian matrix is a regular matrix, then $\boldsymbol{J}(\boldsymbol{\theta})^{\dagger} = \boldsymbol{J}(\boldsymbol{\theta})^{-1}$. In the case that Jacobian matrix is a rectangular matrix, then $\boldsymbol{J}(\boldsymbol{\theta})^{\dagger} = \boldsymbol{W}^{-1}\boldsymbol{J}(\boldsymbol{\theta})^T\left(\boldsymbol{J}(\boldsymbol{\theta})\boldsymbol{W}^{-1}\boldsymbol{J}(\boldsymbol{\theta})^T\right)^{-1}$, where **W** is a weight matrix.

3.2 Inertia matching manipulability

In Eq. (6), the coefficient matrix $\boldsymbol{Q}(\boldsymbol{\theta})$ indicates the moment or force transmission efficiency between the torque produced at the actuators and the force or moment applied to the load by the end-effector.

Based on the theory of singular value decomposition, $\boldsymbol{Q}(\boldsymbol{\theta})$ can be given by:

$$\boldsymbol{Q}(\boldsymbol{\theta}) = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T \tag{8}$$

where $\boldsymbol{U} \in \boldsymbol{R}^{m\times n}$ and $\boldsymbol{V} \in \boldsymbol{R}^{n\times n}$ are orthogonal matrices, $\boldsymbol{\Sigma} = diag(\sigma_1,\sigma_2,\cdots,\sigma_m) \in \boldsymbol{R}^{m\times n}$, and σ_i is a nonnegative singular value.

The manipulability measure ω of inertia matching $\boldsymbol{F}_e$–$\boldsymbol{F}_{bias}$ can be expressed as the product of σ_i, as:

$$\omega = \sigma_1 \cdot \sigma_2 \cdots \sigma_m \tag{9}$$

The principal axes are the product between the row vector $(u_1,...,u_m)$ of $\boldsymbol{U}$ and the singular value vector $(\sigma_1,...,\sigma_m)$. And moreover, the singular value vector $(\sigma_1,...,\sigma_m)$ shows the motion capability of the corresponding principal axis. The manipulability measure of inertia matching synthetically evaluates the isotropic flexibility of robot, and it measures the manipulability of manipulator as a whole.

3.3 Directional manipulability for inertia matching

Inertia matching is a vector with a value and direction. However, the inertia matching manipulability ω describes only the value not the direction. And moreover, jumping motion includes various jumping forms, such as vertical jumping or long jumping. In this paper, we introduce directional manipulability measure of inertia matching to analyze the jumping task and direction.

Assuming the moment and force vector applied to the center of load at end-effector is given by:

$$\boldsymbol{F}_e - \boldsymbol{F}_{bias} = A_{IM}\boldsymbol{P} \tag{10}$$

where A_{IM} is the scalar quantity form of inertia matching $\boldsymbol{F}_e$–$\boldsymbol{F}_{bias}$, $\boldsymbol{P}$ is the direction of the force in load at end-effector in Cartesian frame, and $\boldsymbol{P} = (\cos\beta_1,\cos\beta_2,\cdots,\cos\beta_n)^T \in \boldsymbol{R}^{n\times 1}$, where β_i is the angle between inertia matching and the positive horizontal axis.

Substituting Eq. (10) into Eq. (6), the following equation can be obtained:

$$\boldsymbol{\Gamma}_s = A_{IM}\boldsymbol{Q}(\boldsymbol{\theta})\boldsymbol{P} \tag{11}$$

Generally, the torque limits at each actuator in jumping robot are assumed to be symmetrical and constrained, viz. $-\tau_{i\,max} \le \tau_i \le \tau_{i\,max}$. The normalized joint torque $\tilde{\boldsymbol{\Gamma}}$ can be obtained using a conversion matrix $\boldsymbol{L} = diag(\tau_{1\,max},\cdots,\tau_{n\,max})$ as:

$$\tilde{\boldsymbol{\Gamma}} = \boldsymbol{L}^{-1}\boldsymbol{\Gamma} \tag{12}$$

Therefore, when a normalized torque with magnitude of 1 is produced, the inertia matching ellipsoid can be obtained as:

$$A_{IM}^2 \boldsymbol{P}^T \boldsymbol{Q}^T(\boldsymbol{\theta})\boldsymbol{L}^{-2}\boldsymbol{Q}(\boldsymbol{\theta})\boldsymbol{P} \le 1 \tag{13}$$

The directional manipulability of inertia matching can be given by:

$$DM_{IM} = A_{IM}^2 \le \left[\boldsymbol{P}^T \boldsymbol{Q}^T(\boldsymbol{\theta})\boldsymbol{L}^{-2}\boldsymbol{Q}(\boldsymbol{\theta})\boldsymbol{P}\right]^{-1} \tag{14}$$

The directional manipulability measure of inertia matching DM_{IM} reflects the manipulability in specified direction of robot.

3.4 The optimization of take-off posture

Analyzing the jumping robot as a whole, we can find that the velocity at take-off determines the jumping height, and the COG trajectory after take-off is a parabola. The take-off motion of leg can be regarded as a process from the initial posture $\boldsymbol{\theta}_o = \left(\theta_1^{td}, \theta_2^{td}\right)^T$ to the ultimate posture $\boldsymbol{\theta}_d = \left(\theta_1^{to}, \theta_2^{to}\right)^T$ through the harmonious movement of joints. The manipulability measure of inertia matching is a function of posture $\boldsymbol{\theta}$, and it reflects the moment/force transmission efficiency between the torque produced at the actuators and the force or moment applied to the load by the end-effector. The take-off posture is a main factor which affects jumping performance. When the force transmission efficiency of interior joints is maximal, the time integral of ground reaction force will reach the maximization. So the jumping height is maximal. We will discuss the optimal posture which make the robot reach the maximization of jumping performance applying the inertia matching and directional manipulability.

The posture optimization of jumping motion can be denoted by:

$$\begin{aligned} &max \quad \left[\boldsymbol{P}^T \boldsymbol{Q}^T(\boldsymbol{\theta})\boldsymbol{L}^{-2}\boldsymbol{Q}(\boldsymbol{\theta})\boldsymbol{P}\right]^{-1} \\ &s.\,t. \quad \boldsymbol{\theta}_o = f_{im}\left(\theta_{1o}, \theta_{2o}\right) \end{aligned} \tag{15}$$

where constraint conditionis $f_{im}\left(\theta_{1o}, \theta_{2o}\right)$ determined by the desired jumping angle. For example, if it wants to jump over an obstacle, the angle between $\overline{FG}$ and horizontal axis should satisfy a certain function.

4. TRAJECTORY PLANNING FOR JUMPING OVER OBSTACLE

Our algorithm for jumping over an obstacle is (Figure 3):

- The take-off postures, viz. θ_1^{to} and θ_2^{to}, is established by the optimization results with inertia matching and directional manipulability. The superscript ^{to} denotes the moment of take-off.
- The touch-down postures, viz. θ_1^{td} and θ_2^{td}, are given by desired input parameters. The superscript ^{td} denotes the moment of take-down. The velocity of the foot F at touch-down is given by input parameter k_i, it reflects the amount of kinetic energy loss during impact. If trajectory is a soft landing, k_i is zero.
- The trajectory planning satisfies dynamics constraints and boundary conditions.
- The motion of body is established by internal motion dynamics and steady-state consecutive jumping motion principle.

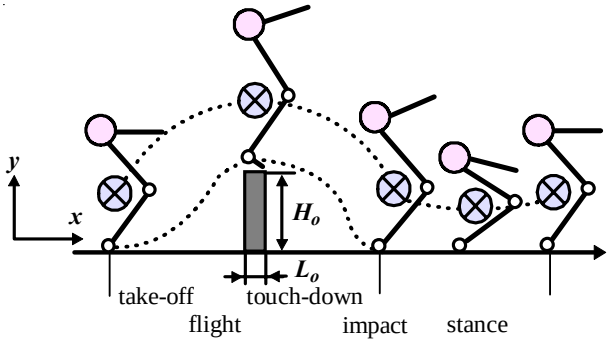

Fig. 3 Events and phases for jumping over obstacle

4.1 Trajectory during flight phase

4.1.1 General assumption and boundary conditions

Based on the kinematics of the robot, the relationship between the COG position and the foot F position can be expressed as:

$$\begin{bmatrix} \boldsymbol{R}_G \\ \dot{\boldsymbol{R}}_G \\ \ddot{\boldsymbol{R}}_G \end{bmatrix} - \begin{bmatrix} \boldsymbol{R}_F \\ \dot{\boldsymbol{R}}_F \\ \ddot{\boldsymbol{R}}_F \end{bmatrix} = \begin{bmatrix} \boldsymbol{E}_G(\boldsymbol{\theta}) \\ \boldsymbol{J}\dot{\boldsymbol{\theta}} \\ \dot{\boldsymbol{J}}\dot{\boldsymbol{\theta}} + \boldsymbol{J}\ddot{\boldsymbol{\theta}} \end{bmatrix} \tag{16}$$

where $\boldsymbol{R}_G = \left[\boldsymbol{X}_G, \boldsymbol{Y}_G\right]^T$, $\boldsymbol{R}_F = \left[\boldsymbol{X}_F, \boldsymbol{Y}_F\right]^T$.

The obstacle parameters are described by height H_o and length L_o. In order to let foot over the obstacle, we evaluate the jumping height at following (Figure 4):

$$\begin{cases} H_{jump} = \boldsymbol{E}_H\left(H_o, \boldsymbol{\theta}\right) \\ L_{jump} = X_F^{td} - X_F^{to} = \boldsymbol{E}_L\left(H_o, L_o, \boldsymbol{\theta}\right) \end{cases} \tag{17}$$

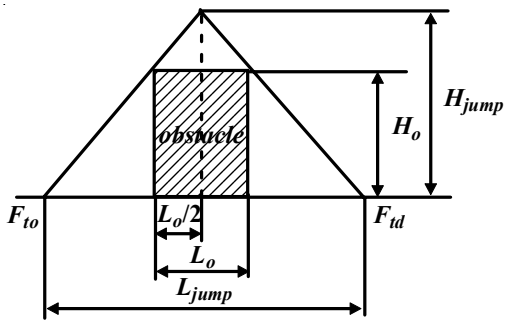

Fig. 4 Evaluation of jumping height and jumping length

Suppose that at take-off the foot does not slip. The velocity of the foot at touch-down determines the amount of kinetic energy loss during impact. Performing touch-down with an improper choice for the foot velocity at touch-down can cause high energy losses during impact. The acceleration of the foot at touch-down has an influence on the amplitude of the ground reaction force immediately after impact. The velocity and acceleration of the foot at touch-down will be defined here proportional to the velocity of the COG. So yields:

$$\begin{cases} \dot{X}_F^{to} = 0 & \dot{X}_F^{td} = k_1 \dot{X}_G^{td} \\ \dot{Y}_F^{to} = 0 & \dot{Y}_F^{td} = k_2 \dot{Y}_G^{td} \\ \ddot{X}_F^{to} = 0 & \ddot{X}_F^{td} = k_3 \ddot{X}_G^{td} \\ \ddot{Y}_F^{to} = 0 & \ddot{Y}_F^{td} = k_4 \ddot{Y}_G^{td} \end{cases} \tag{18}$$

During the ballistic flight phase the COG tracks a parabolic trajectory. The flight time can be expressed as following by the initial conditions:

$$\begin{cases} T^{fl} = \boldsymbol{E}_{T-H}\left(H_{jump}, \boldsymbol{R}_G^{tj}\right) \\ \dot{Y}_G^{to} = \boldsymbol{E}_{Y-H}\left(H_{jump}\right) \qquad (tj = to,td) \\ \dot{\boldsymbol{R}}_G^{td} = \boldsymbol{E}_{R-HL}\left(L_{jump}, H_{jump}, \boldsymbol{R}_G^{tj}, \boldsymbol{\theta}^{tj}\right) \end{cases} \tag{19}$$

Then we can get:

$$\begin{bmatrix} \dot{\boldsymbol{\theta}}^{tj} \\ \ddot{\boldsymbol{\theta}}^{tj} \end{bmatrix} = \boldsymbol{E}_{\theta-\theta}\left(\boldsymbol{\theta}\right)\begin{bmatrix} \boldsymbol{\theta}^{tj} \\ \dot{\boldsymbol{\theta}}^{tj} \end{bmatrix} + \boldsymbol{E}_{\theta-R}\left(\boldsymbol{\theta}\right)\begin{bmatrix} \boldsymbol{R}_G^{tj} \\ \dot{\boldsymbol{R}}_G^{tj} \end{bmatrix} \quad (tj = to,td) \tag{20}$$

4.1.2 Constructing the polynomial functions

Since now boundary conditions at take-off as well as at touch-down have been known for both the angles θ_i ($i=1,2$) and their first and second derivatives. Normally, a 5^{th} order polynomial tracking function can be established for θ_i under their position, velocity, and acceleration constraints of both initial point and ultimate point. It is the traditional point-to-point motion planning:

$$\tilde{\theta}_i^f = \sum_{j=0}^{5} a_i t^j \quad (i = 1,2) \tag{21}$$

There is a problem: Since the operational time is determined by initial and ultimate conditions such as flight, the polynomial is sensitive to the initial point or ultimate constraint conditions by using 5^{th} order polynomial to track trajectory (Figure 5). If one condition changes among the six constraint conditions, the trajectory would change very large. Moreover, the trajectory would not satisfy the angular work space.

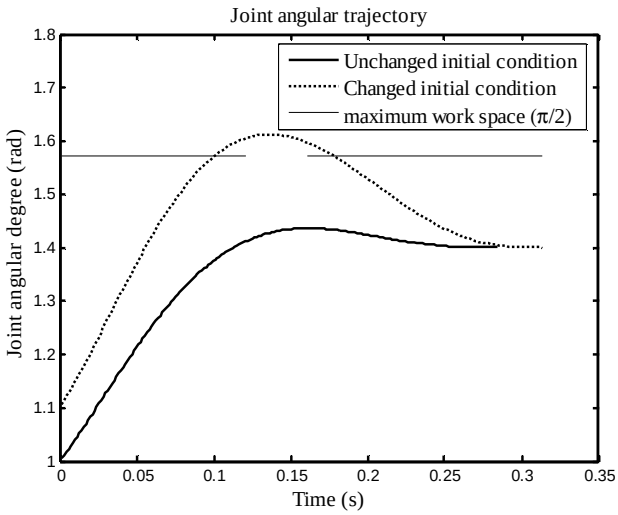

Fig. 5 Trajectory using 5^{th} order polynomial under before-change initial condition and after-change initial condition

Aimed at this problem, and in order to enhance the trajectory generation efficiency and to improve the trajectory robustness to constraint conditions, we propose 6^{th} order polynomial to track trajectory:

$$\tilde{\theta}_i^f = \sum_{j=0}^{6} a_i t^j \quad (i = 1,2) \tag{22}$$

The highest power coefficient a_6 is obtained by joint motion constraints, the other power coefficients are determined by the initial and ultimate constraint conditions. The trajectory can be ensured to satisfy the joint workspace through the optimization of the 6^{th} power coefficient because the highest power polynomial coefficient is the most sensitive and influencing the shape of a polynomial (Figure 6). If the highest power coefficient is ascertained, the trajectory polynomial will have a little influence with the change of the other lower power coefficients. Thus, this polynomial has a good robustness for the constraint conditions.

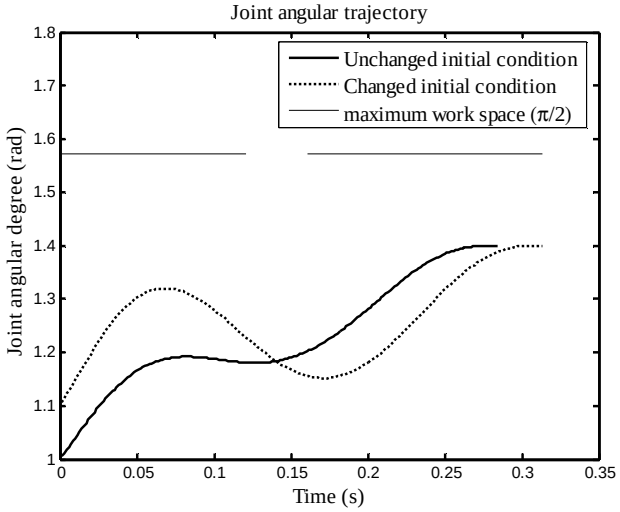

Fig. 6 Trajectory using 6th order polynomial under before-change initial condition and after-change initial condition

4.1.3 Constructing correction functions of controlling the foot to jump over the obstacle

The polynomial function θ is completely determined by the boundary points only. In order to let foot jump over the obstacle, a correction on the polynomial function is introduced. An intermediate point is added to the polynomial function to make sure that the foot reaches the height of obstacle at $t = t^* = \dfrac{\dot{Y}_G^{to}}{g}$, where t^* is the time step where G and foot reach their maximum height at the same time:

$$\begin{cases} \dot{Y}_G\left(t^*\right) = 0 \\ \dot{Y}_F\left(t^*\right) = 0 \end{cases} \tag{23}$$

This strategy is chosen here, since it results in an analytical solution for the correction functions. A correction functions $b_i\left(t\right)\left(i = 1,2\right)$ will be added, which do not change the boundary conditions of the polynomial functions:

$$\begin{cases} b_i\left(t\right) = K_i\, f_{co}\left(t\right) \\ f_{co}\left(t\right) = \left[1 - 3\left(t - t^*\right)\left(\dfrac{1}{t^*} - \dfrac{1}{T^{fl} - t^*}\right)\right]\dfrac{t^3\left(T^{fl} - t\right)^3}{\left(t^*\right)^3\left(T^{fl} - t^*\right)^3} \end{cases} \tag{24}$$

Based on parabolic motion of COG of the robot, K_i can be solved as the following set:

$$\begin{cases} Y_G\left(t^*\right) = Y_G^{to} + \dot{Y}_G^{to}\left(t^* - t_{to}\right) - \dfrac{g\left(t^* - t_{to}\right)^2}{2} \\ \dot{Y}_G\left(t^*\right) = 0 \end{cases} \tag{25}$$

Thus, the angular value θ_i^f can be devoted by the polynomial function $\tilde{\theta}_i^f$ and correction function $b_i(t)$ as following:

$$\theta_i^f = \tilde{\theta}_i^f + b_i\left(t\right) \tag{26}$$

4.1.4 Trajectory for upper body

Since our model has a decoupling between leg motion and body rotation, in that case the body rotation θ_3 can only be controlled by internal motion dynamics, such as angular momentum. The angular momentum with respect to COG can be calculated with the following general formula:

$$\mu_G^f = \sum_{i=1}^{3}\left(\overline{GG_i} \times m_i \overline{GG_i}' + I_i\dot{\theta}_i\right) \tag{27}$$

So we can obtain:

$$\begin{cases} \theta_3 = \theta_3^{to} + \dfrac{1}{I_3}\displaystyle\int_{t_{to}}^{t}\left[\mu_G^f - f_\mu\left(\theta,\dot{\theta}\right)\right]dt \\ f_\mu\left(\theta,\dot{\theta}\right) = \displaystyle\sum_{i=1}^{3}\left(\overline{GG_i} \times m_i \overline{GG_i}'\right) + \sum_{i=1}^{2}\left(I_i\dot{\theta}_i\right) \\ \Delta\theta_3^f = \dfrac{\mu_G^f T^f - A^f}{I_3} \\ A^f = \displaystyle\int_{t_{to}}^{t_{td}}\left[\sum_{i=1}^{3}\left(\overline{GG_i} \times m_i \overline{GG_i}'\right) + \sum_{i=1}^{2}\left(I_i\dot{\theta}_i\right)\right]dt \end{cases} \tag{28}$$

4.2 Trajectory during stance phase

Due to the constraints on the leg during stance demanding the foot should stay at a fixed position, the polynomial functions during stance phase can be constructed by using the results of impact phase and the results of the algorithm developed for flight phase:

- As a steady-state consecutive jumping, the final conditions during stance phase are equal to the initial conditions at take-off during flight phase;
- The initial conditions during stance phase are equal to the conditions after impact, and the conditions before impact are equal to the final conditions during flight phase respectively:

$$\begin{cases} \boldsymbol{\theta}^{so} = \boldsymbol{\theta}^{+} & \boldsymbol{\theta}^{sd} = \boldsymbol{\theta}^{to} & \boldsymbol{\theta}^{-} = \boldsymbol{\theta}^{td} \\ \dot{\boldsymbol{\theta}}^{so} = \dot{\boldsymbol{\theta}}^{+} & \dot{\boldsymbol{\theta}}^{sd} = \dot{\boldsymbol{\theta}}^{to} & \dot{\boldsymbol{\theta}}^{-} = \dot{\boldsymbol{\theta}}^{td} \\ \ddot{\boldsymbol{\theta}}^{so} = \ddot{\boldsymbol{\theta}}^{+} & \ddot{\boldsymbol{\theta}}^{sd} = \ddot{\boldsymbol{\theta}}^{to} & \ddot{\boldsymbol{\theta}}^{-} = \ddot{\boldsymbol{\theta}}^{td} \end{cases} \tag{29}$$

where the superscript ^{sd} denotes ultimate state configuration during stance phase, the superscript $^{+}$ denotes the state configuration after impact, and

the superscript $^-$ denotes the state configuration before impact.

4.2.1 General assumption and boundary conditions

At the landing, the foot of the jumper hits the ground. Let's assume the foot does not bounce back and does not slip, which means that it stays in contact with the ground. These are the assumptions corresponding to an inelastic impulsive impact. During this impact phase, although discontinuities in the velocity and acceleration state variables will occur, the configuration of the robot is assumed to stay unchanged, viz.:

$$\boldsymbol{\theta}^+ = \boldsymbol{\theta}^- \tag{30}$$

According to Zheng and Hemami [19] the discrete variation of the generalized velocities due to the inelastic impulsive impact with the ground can be calculated as follows:

$$\Delta\dot{\boldsymbol{q}} = \boldsymbol{D}_f^{-1}(\boldsymbol{\theta})\boldsymbol{J}^T\left(\boldsymbol{J}\boldsymbol{D}_f^{-1}(\boldsymbol{\theta})\boldsymbol{J}^T\right)^{-1}\Delta\overline{\boldsymbol{OF}} \tag{31}$$

with $\Delta\dot{\boldsymbol{q}} = \left(\Delta\dot{\boldsymbol{\theta}}, \Delta\dot{\theta}_3, \Delta\dot{X}_F, \Delta\dot{Y}_F\right)^T$ and $\Delta\overline{\boldsymbol{OF}} = \left(\Delta\dot{X}_F, \Delta\dot{Y}_F\right)^T$. From the Eq. (31), we can obtain:

$$\begin{bmatrix}\dot{\theta}_1^+\\ \dot{\theta}_2^+\\ \dot{\theta}_3^+\end{bmatrix} = \begin{bmatrix}\dot{\theta}_1^{td}\\ \dot{\theta}_2^{td}\\ \dot{\theta}_3^{td}\end{bmatrix} - \boldsymbol{D}_f^{-1}(\boldsymbol{\theta})\boldsymbol{J}^T\left(\boldsymbol{J}\boldsymbol{D}_f^{-1}(\boldsymbol{\theta})\boldsymbol{J}^T\right)^{-1}\begin{bmatrix}\dot{X}_F^{td}\\ \dot{Y}_F^{td}\end{bmatrix} \tag{32}$$

To calculate the angular accelerations $\ddot{\boldsymbol{\theta}}^+$ after the impact, the equation of motion for the stance phase can be used. The joint torques are considered to remain unchanged during the infinitesimal short time interval of the impact. Their values are those measured at the instance of landing impact:

$$\begin{cases}\tau_K^{so} = \tau_K^{td}\\ \tau_H^{so} = \tau_H^{td}\end{cases} \tag{33}$$

From the Eq. (33), we can obtain:

$$\ddot{\boldsymbol{\theta}}_s^+ = \left(\boldsymbol{D}_s^{so}\right)^{-1}\left(\boldsymbol{D}_f^{td}\ddot{\boldsymbol{q}}^{td} + \boldsymbol{H}_f^{td}\dot{\boldsymbol{q}}^{td} + \boldsymbol{G}_f^{td} - \boldsymbol{H}_s^{so}\dot{\boldsymbol{q}}^{so} - \boldsymbol{G}_s^{so}\right) \tag{34}$$

4.2.2 Solving stance time and constructing the polynomial functions

During flight phase, the angular momentum with respect to COG μ_G^f is conserved without external forces acting on the robot, and the angular momentum with respect to foot μ_F^f can be obtained by μ_G^f. During stance phase, there are external ground reaction forces on the foot, and the angular momentum with respect to foot μ_F^f can be obtained by integration over the stance time:

$$\begin{cases}\mu_F^f = \mu_G^f + \overline{FG}^f \times M\,\overline{FG}'^f\\ \mu_F^s = \mu_F^{td} + Mg\int_{t_{so}}^{t}\left(X_G - X_F\right)dt\end{cases} \tag{35}$$

The stance time T^s can be calculated by Eq. (35) as following:

$$Mg\int_{t_{st}^{to}}^{t_{st}^{td}} X_G dt = \overline{FG}^{to} \times M\,\overline{FG}'^{to} - \overline{FG}^{td} \times M\,\overline{FG}'^{td} \tag{36}$$

After obtaining six boundary conditions and stance time, in an analogue way as during flight, 5^{th} order polynomials can be calculated, which are the reference trajectories $\boldsymbol{\theta}$ for lower and upper leg respectively during stance phase.

4.2.3 Trajectory for upper body

Integrating Eq. (35) the following expression for the upper body rotation during stance phase can be obtained:

$$\begin{cases}\theta_3^s = \theta_3^+ + \dfrac{1}{I_3}\displaystyle\int_{tso}^{t}\left[\mu_G^{to} + \overline{FG}^{td} \times M\,\overline{FG}'^{td} - f_\mu\left(\boldsymbol{\theta},\dot{\boldsymbol{\theta}}\right) - Mg\int_0^{\varepsilon}\left(X_G - X_F\right)d\varepsilon\right]dt\\ \Delta\theta_3^s = \dfrac{1}{I_3}\left[\left(\mu_G^{to} + \overline{FG}^{td} \times M\,\overline{FG}'^{td}\right)T^s - A^s\right]\\ A^s = \displaystyle\int_{t_{so}}^{t}\left[f_\mu\left(\theta,\dot{\theta}\right) + Mg\int_0^{\varepsilon}\left(X_G - X_F\right)d\varepsilon\right]dt.\end{cases} \tag{37}$$

Considering a steady-state condition, the rotation during flight needs to be compensated by the rotation during stance:

$$\Delta\theta_3^{fl} + \Delta\theta_3^{st} = 0 \tag{38}$$

Introducing Eqs. (28) and (37) into Eq. (38) results in an expression which can be solved for μ_G^f:

$$\mu_G^f = \frac{A^f + A^s - B^s T^s}{T^f + T^s} \tag{39}$$

Thus, we can obtain the trajectory of upper body.

4.3 Flow chart

In order to clearly summarize the different steps of the strategy generating the trajectory, a flow chart is given in Figure 7.

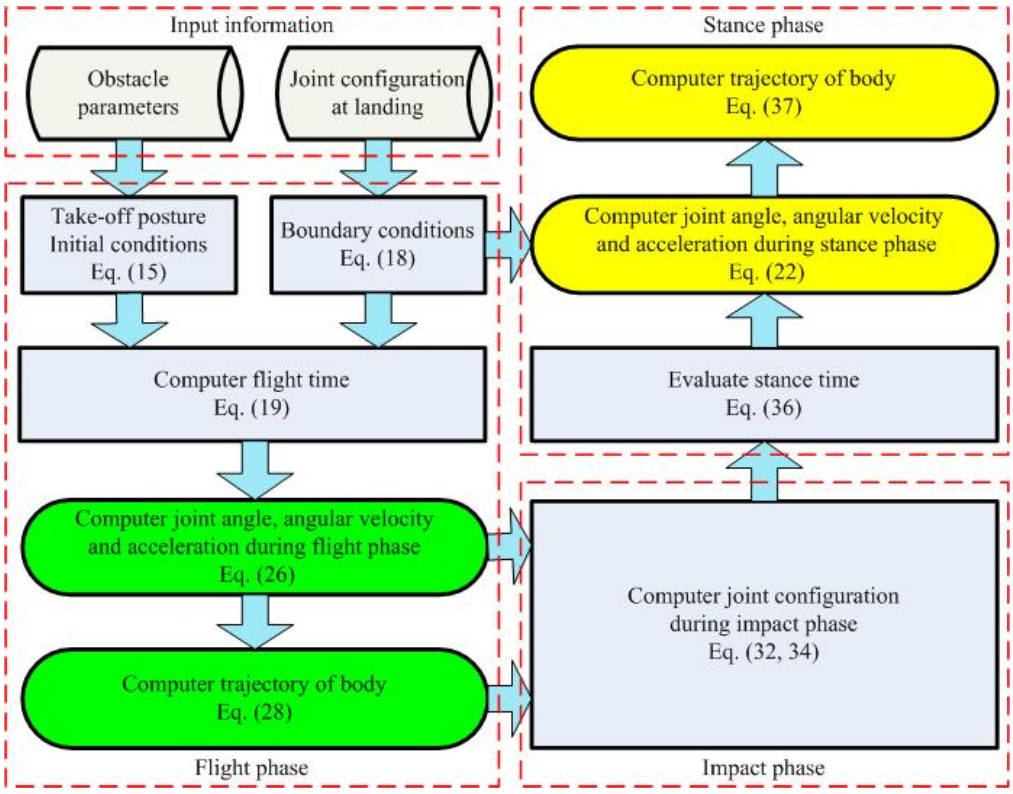

Fig. 7 Flow chart describing different steps in strategy

5. SIMULATION

To test the specified algorithm, a steady-state jumping pattern over an obstacle has been simulated. A jumping robot which has 3-DOF rotary joints and four rigid bodies (foot, lower leg, upper leg, upper body) is designed. Figure 8 shows the mechanical CAD model, and its inertia parameters are given in Table 1. The hip of robot is linked by a cylinder of a guidance device, and its motion is restricted to a planar jumping. This guidance device provides only lateral stabilization, and the robot can rotate around the axis in the hip and translate horizontally and vertically. Ankle joint is passive, and knee/hip joint is independently driven by servo motor. The type of servo motors is GWS S777, made in Taiwan, China. Its rated velocity is *0.12 s/60°* (*6.0V*), viz. *8.72 rad/s*, and its rated torque is *42 kgcm*. In this robot model, there are not any assistant elastic components, such as spring, damp, hydraulic or pneumatic actuators. The controller is an AVR system, and main control chip is Atmega128 from ATMEL company. The ATmega128 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. It provides four flexible Timer/Counters with compare modes and PWM, and PWM can directly drive servo motor.

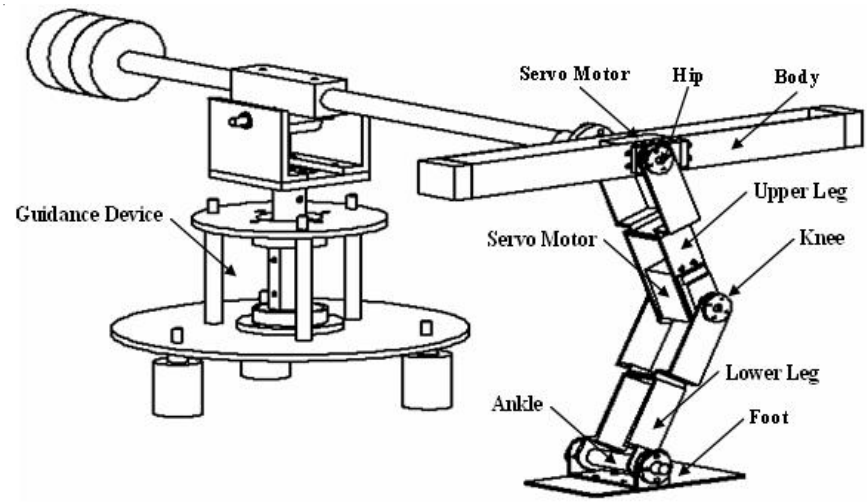

Fig. 8 Jumping robot

Table1 Inertial parameters of the jumping robot

i	*Length* L_i *(m)*	*Mass* m_i *(kg)*	*Moment of Inertia* I_i *(kg* m^2*)*	*Maximal Torque* τ_i *(kg cm)*
1	*0.34*	*0.178*	1.38×10^{-3}	*none*
2	*0.31*	*0.137*	2.18×10^{-3}	*42*
3	*0.67*	*0.851*	7.98×10^{-3}	*42*

The parameters of obstacle and postures at touch-down are the following:

$$H_o=0.012\ m;\quad L_o=0.01\ m;$$

$$\theta_1^{td}=1.89\ rad;\quad \theta_2^{td}=2.25\ rad$$

The following expatiates the simulation results. Figure 9 shows inertia matching under different take-off postures. Figure 10 shows distributing of ground reaction force under different take-off postures. The inertia matching and ground reaction force reach a maximization when θ_1 is equal to *1.01 rad.* Here, θ_2 can be obtained with the jumping angle. In this simulation, θ_1 is *1.89 rad* based on the size of the obstacle. In view of force transmission efficiency and the time integral of ground reaction force, jumping height is in direct proportion to the time integral of ground reaction force. When take-off posture is in the optimization, the time integral of ground reaction force and jumping height are maximal. When the angular angle of ankle increases from *0.52 rad* to *1.01 rad*, inertia matching gradually increases to the maximal *0.9 N*, and the ground reaction force also gradually increases to the maximum *17.8 N*. When the angular angle of ankle is on the increase, inertia matching rapidly reduces to zero, and the ground reaction force and jumping height also rapidly reduces. Thus inertia matching is in direct proportion to jumping height/ jumping performance, and it can be applied to analyze the jumping motion.

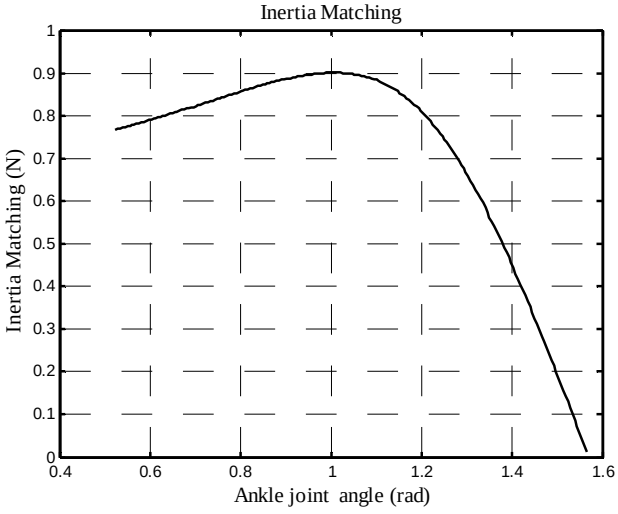

Fig. 9 Inertia matching under different postures

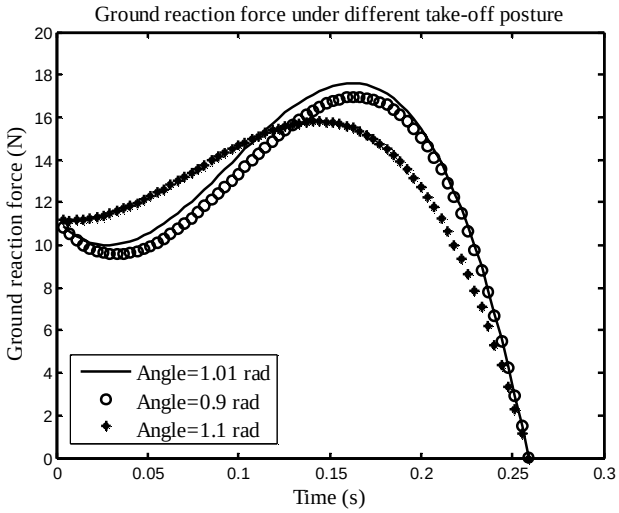

Fig. 10 Distributing of ground reaction force under different postures

The trajectories tracked by the actuators of the leg during the stance phase, which guarantee that the desired values for jumping over the obastcle are attained, cause a clockwise natural rotation of upper body. Therefore a counterclockwise rotation of upper body during the fight phase is suitable, since these both rotations can be compensated by each other. Suppose that a zero angular momentum would have been chosen during the flight phase, the upper body would rotate in the clockwise direction, due to the leg swing in the counterclockwise direction. In that case the global upper body rotation would drift during the consecutive jumping unless an actuator acted on it during the stance phase [8]. This situation can simply be avoided by choosing a positive value for the angular momentum during the flight phase. In this paper, angular momentum is *0.0185 kgm²/s* during the flight phase. The trajectory of angular momentum with respect to COG is shown in Figure 14 later.

The trajectory being tracked by the actuators for lower leg, upper leg and upper body are displayed in Figure 11. Using our proposed 6^{th} order polynomial functions, the joint angle trajectory are all under the joint workspace. Figure 12 gives the exerted torques by hip and knee actuator respectively. The peak value of both the hip torque and knee torque is smaller than the rate torque value *4.1 Nm* of servo motor. The peak value of the knee torque is significantly higher than hip's, being approximately *4 Nm* during the stance phase. If using a

passive element, e.g. a torsional spring, this peak value can be severely reduced. So, some hopping robot was designed to use some assistant elastic components, such as prototype OLIE in Vrije University Brussel, Belgium [20]. Figure 13 shows the vertical position of the foot using with/without correction function. We can find that the foot can jump over the designed obstacle after using correction function.

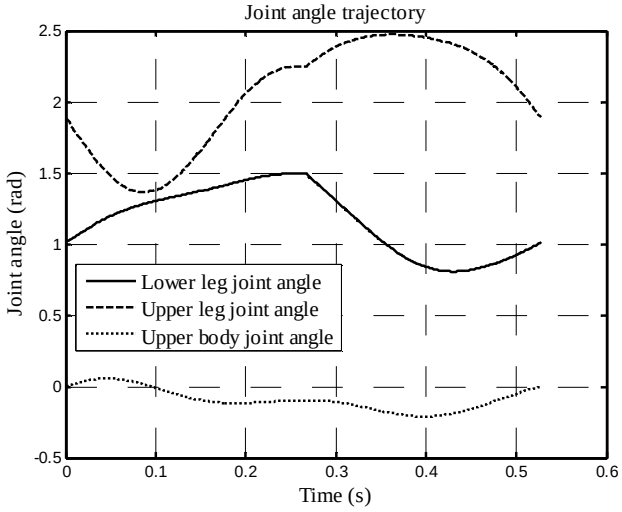

Fig. 11 Joint angle trajectory of lower leg, upper leg and body

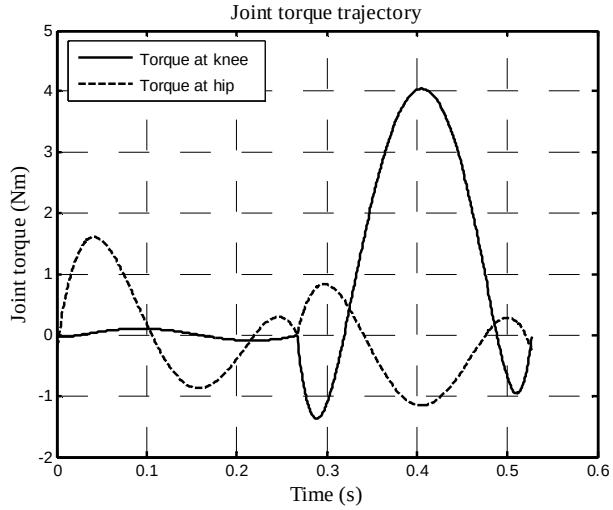

Fig. 12 Torque trajectory at knee and hip

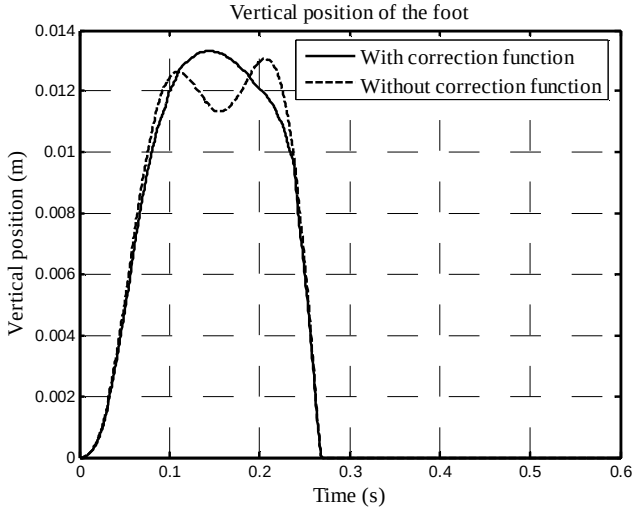

Fig. 13 Vertical position of the foot

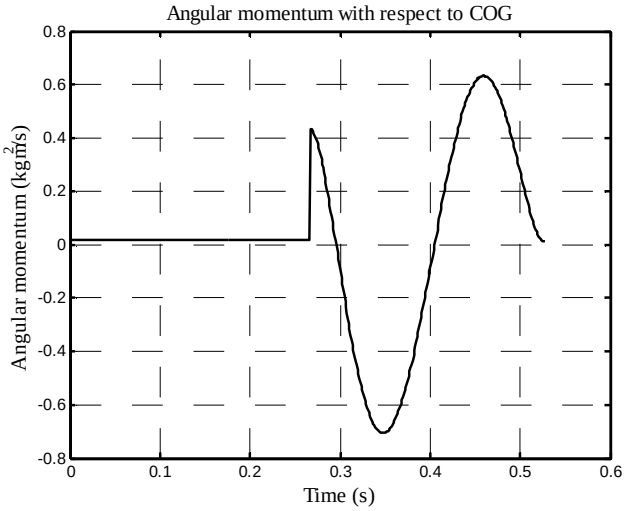

Fig. 14 Angular momentum trajectory with respect to COG

6. CONCLUSION

In this paper a trajectory generation strategy for one-legged passive jumping robot is developed. Due to the fact that the COG of the upper body is located at the hip joint, the motion of the robot's leg and its body are completely decoupled. The upper body rotation has no influence on the motion of the global COG of the robot. The algorithm is therefore built up by two independent tasks, being the control of lower leg and upper leg, and the control of the upper body motion.

Firstly, inertia matching and directional manipulability are applied to analyze jumping robot and to optimize take-off posture. The optimum results are used to plan jumping trajectory. Then aimed at the flight motion and point-to-point motion planning theory, 6^{th} order polynomials are proposed to track the joint trajectory. They have a better robustness to the changed constraint conditions than traditional 5^{th} order polynomials. This method improves the efficiency of trajectory generation. Furthermore, in order to lift the foot jump over the designed obstacle, a correction function is constructed under unchanged boundary constraint conditions. Finally, the body posture is planned with internal motion dynamics and steady-state consecutive jumping motion.

Jumping robot is a typical nonholonomic system, the control of drift during flight is a difficult issue. The development of jumping motion is to pursue an effective locomotive velocity. In our following work, we will focus on minimal control method, the natural passive dynamics for locomotion, and smaller control effort for energy loss compensation and stabilization.

7. REFERENCES

[1] A. Sayyad, B. Seth and P. Seshu, Single-legged hopping robotics research - A review, *Robotica*, Vol. 25, pp. 587–613, 2007.

[2] Z. Li and R. Montgomery, Dynamics and optimal control of a legged robot in flight phase, in: Proc. of IEEE Int. Conf. on Robotics and Automation, Cincinnati, pp. 1816-1821, 1990.

[3] V.V. Lapshin, Motion control of a legged machine in the supportless phase of hopping, *The International Journal of Robotics Research*, Vol. 10, No. 4, pp. 327-337, 1991.

[4] K. Harbick and G. Sukhatme, Controlling hopping height of a pneumatic monopod, in: Proc. of Int. Conf. on Robotics and Automation, Washington, pp. 3998- 4003, 2002.

[5] E. Ohashi and K. Ohnishi, Hopping height control for hopping robots, *Electrical Engineering in Japan*, Vol. 155, No. 1, pp. 64-71, 2006.

[6] J. Babic and D. Omrcen, Stability control of human inspired jumping robot, in: Proc. of 15th Int. Workshop on Robotics in Alpe-Adria-Danube Region, pp. 206-211, 2006.

[7] M. Higashimori, M. Harada, I. Ishii and M. Kaneko, Torque pattern generation towards the maximum jump Hhight, in: Proc. of IEEE Int. Conf. on Robotics and Automation, Orlando, pp. 1096-1101, 2006.

[8] H. De Man, D. Lefeber and J. Vermeulen, Design and control of a robot with one articulated leg for locomotion on irregular terrain, in: Proc. of 12th Symposium on Theory and Practice of Robots and Manipulators, New York, pp. 417- 424, 1998.

[9] J. Vermeulen, D. Lefeber and H. De Man, A control strategy for a robot with one articulated leg hopping on irregular terrain, in: Proc. of Int. Conf. on Climbing and Walking Robots, Madrid, pp. 399- 406, 2000.

[10] J. Vermeulen, D. Lefeber and B. Verrelst, Control of foot placement, forward velocity and body orientation of a one-legged hopping robot, *Robotica*, Vol. 21, pp. 45-57, 2003.

[11] T. Yoshikawa, Manipulability of robotic mechanisms, *The International Journal of Robotics Research*, Vol. 4, No. 2, pp. 3-9, 1985.

[12] T. Yoshikawa, Dynamic manipulability of robot manipulators, *Journal of Robotic Systems*, Vol. 2, No.1, pp.113-124, 1985.

[13] S.H. Lee, T.A. Lasky and S.A. Velinsky, Manipulability-based design and analysis of a hybrid manipulator for highway applications, *Mechanics Based Design of Structures and Machines*, Vol. 33, pp. 99-118, 2005.

[14] A. Bowling and O. Khatib, The dynamic capability equations: a new tool for analyzing robotic manipulator performance, *IEEE Transactions on Robotics*, Vol. 21, No. 1, pp. 115-123, 2005.

[15] C.Y. Kim and Y.S. Yoon, Task space dynamic analysis for multi-arm robot using isotropic velocity and acceleration radii, *Robotica*, Vol. 15, pp. 319-329, 1997.

[16] D.Z. Chen and L.W. Tsai, The generalized principle of inertia match for geared robotic mechanisms, in: Proc. of the IEEE Int. Conf. on Robotics and Automation, Sacramento, pp. 1282-1287, 1991.

[17] R. Kurazume and T. Hasegawa, A new index of serial-link manipulator performance combining dynamic manipulability and manipulating force ellisoids, *IEEE Transactions on Robotics*, Vol. 22, No. 5, pp. 1022-1028, 2006.

[18] Z.H. Xu, T.S. Lu and X.Y. Wang, Inertia matching manipulability and motion optimization for humanoid jumping robot, *Journal of Intelligent and Robotic Systems*, 2007. (submitted)

[19] Y.F. Zheng and H. Hemami, Mathematical modeling of a robot collision with its environment, *Journal of Robotic Systems*, Vol. 2, No. 3, pp. 289-307, 1985.

[20] M.D. Berkemeier and R.S. Fearing, Sliding and hopping gaits for the underactuated Acrobot, *IEEE Trans Robot Automation*, Vol. 14, No 4, pp. 629-634, 1998.

PLANIRANJE PUTANJE SKAKANJA JEDNONOŽNOG ROBOTA PREKO PREPREKE

SAŽETAK

Ovaj rad opisuje planiranje putanje skakanja jednonožnog pasivnog robota preko prepreke koristeći tri različite dinamike procesa preskakivanja. Podešavajući elipsoidi su učinkovito sredstvo kojim se može izvršiti zadana analiza prostora kao i optimizacija gibanja razgranatih manipulatora. Robot koji skače može se smatrati razgranatim manipulatorom koji drži opterećenje na krajevima izvršitelja. Ideja elipsoida izjednačavanja inercije kao i direktna mogućnost rukovanja proširena je na optimizaciju postavljanja u položaj za skok robota, a optimizirani rezultati koriste se za planiranje putanje skakanja. Uzimajući u obzir osjetljivost putanje na uvjete ograničenja planiranja gibanja od točke do točke, predlaže se polinom šestog stupnja za planiranje gibanja skoka, budući da je pogodniji za promjenu parametara uvjeta ograničenja od uobičajenog polinoma petog stupnja. Funkcije korekcije formirane su za konstantne rubne uvjete tako da robot može dignuti nogu preko prepreke. Nadalje, položaj tijela kontrolira se pomoću dinamike unutarnjeg gibanja i načela mirnog konsekutivnog skakanja. Konstruiran je model na kojem je potvrđena učinkovitost predložene metode simulacijama koje su obavljene na parametrima projektiranog modela.

***Ključne riječi**: skačući robot, izjednačavanje inercije, planiranje putanje, jaki polinom, funkcija korekcije.*