18th Mediterranean Conference on Control & Automation
Congress Palace Hotel, Marrakech, Morocco
June 23-25, 2010

Locomotion Control System Design for the LOCH Humanoid Robot

GuoQing Zhang, Ming Xie, HeJin Yang, Jing Li and XuePei Wu

Abstract— **Design of the locomotion control system for the LOCH robot is presented in this paper. Gait planning and control algorithm for uneven terrain is also considered. The LOCH robot is an adult-sized biped humanoid robot. It adopts the distributed control structure based on CAN bus, and uses a Linux operating system as the software platform. The architecture of both the hardware and software system is introduced. The emphasis is then put onto the biped planning and control. An on-line planner is designed on the basis of the inverted arm model. It generates walking gaits adaptively according to user inputs and floor flatness changes. In handling uneven floor, an imaginary foot approach is proposed to convert the problem into flat floor planning. Stability control and compliant landing control are also investigated. Both experiments and simulations are performed to show the effectiveness of the proposed design.**

I. Introduction

In the past few decades, significant progress has been achieved in the research and application of robotics. Today a great quantity of robots are in use for a variety of purposes, ranging from production line in factories to daily service at home. The mainstream forms of the existing robots include fixed-base manipulators, wheeled mobile robots and biped humanoids. Among them, the biped humanoid robots, because of their human-like mobility and appearance, have attracted much attention of both researchers and the public. However, biped humanoid robots come with the most difficulty in control design because of its complex mechatronic structure. For the development of human-level motion capability, several biped platforms have been built by various institutions. Successful ones include ASIMO[1] by HONDA, HRP[2], [3] series by AIST, WABIAN[4] by Waseda University, HUBO[5] by KAIST, etc.

Recently Nanyang Technological University has developed a low-cost biped humanoid robot, LOCH. It is an adult-sized humanoid robot, which measures 1.8m in height and weighs 80 kilograms. The LOCH robot has 12 degrees of freedom (DOF) in dual legs, and 2 DOF in the trunk. Regardless of the upper-body motion, up to 14 joints are involved in the biped walking control. Several kinds of sensors are installed onto the robot to sense the walking conditions, and a PC104 onboard handles the sensory information. Out of the consideration of low-cost requirement, a Linux operating system is chosen to provide basic systematic support. Issues about the mechanical design and the algorithm design have been described in [6], [7]. This paper is intended to present the design of its real-time control system for biped locomotion.

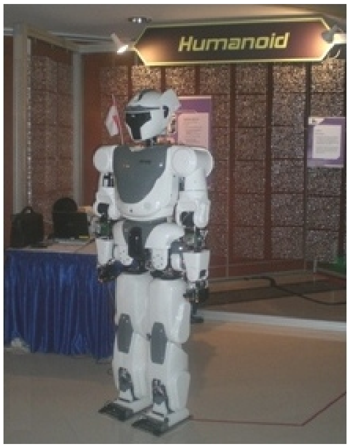

Fig. 1. The LOCH humanoid robot

In terms of biped locomotion, walking planning and stability control plays a critical role. Off-line gait planning with on-line control compensation is the most common fashion[8], [9], [11], [12]. ZMP based methods have been broadly accepted and expanded since its birth in the 1970s[9]. In dealing with uneven terrain, auxiliary ZMP control[2] and short cycle pattern generation method[3] were proposed. In our design, an adaptive on-line planner is designed for the generation of walking gaits. The planner changes gaits autonomously in response to user inputs and floor flatness changes. The stability controller and the landing controller are also designed to enhance walking stability, and to achieve compliant landing of leg swing.

The remainder of the paper is organized as follows. In Section II, the hardware architecture and the software architecture of the control system are presented. In Section III, the design of the adaptive on-line planner is given in detail. In Section IV and V, the stability controller and the landing controller are discussed. Section VI is the conclusion.

II. System Overview

A. Hardware Architecture

In terms of the hardware devices, the locomotion control system consists of a PC104, two force/torque sensors, two sole pressure sensors, one gyro, 14 joint controllers and a CAN bus.

PC104, which features the 1.6GHz CPU, 2GB memory and the compact size, provides sufficient computation power for signal processing and complex real-time calculations. On the PC104, a Linux OS, Mandriva, is installed. The reason that we choose the Linux system is to reduce the system

This work was supported by the Singapore government

The authors are with School of Mechanical and Aerospace Engineering, Nanyang Technological University. PWTC, 50 Nanyang Drive. Singapore, 637553 {zhanggq, mmxie, hjyang, lijing, xpwu}@ntu.edu.sg

cost, and to utilize the rich support resources available. Although strictly it is not a real-time operating system, the powerful PC104, together with a designed adaptive data buffer technology, makes an acceptable timing accuracy.

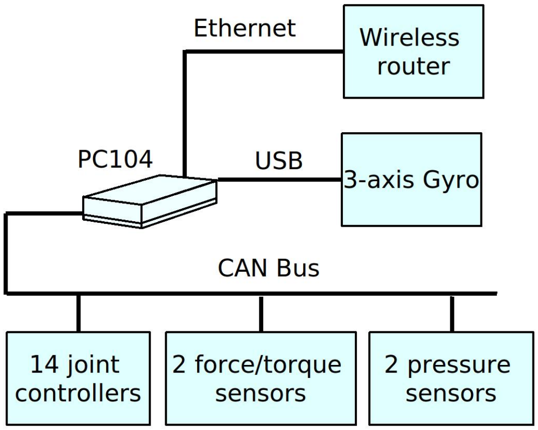

Fig. 2. Hardware architecture

The combination of the aforementioned sensors enables the LOCH robot to acquire information about the motion status and floor conditions. The information is essential for sensor-guided behaviors. The two force/torque sensors are installed on the robot ankles. During the walking process, Their outputs are used to evaluate the walking stability based on the ZMP concept. The pressure sensors are installed on the foot sole to detect the foot-floor contact condition. From the sensor outputs, the contact area of either foot is known. The gyro, which is installed on the trunk, offers information about the upper body attitude. The upper-body attitude and the foot-floor contact condition jointly indicate the terrain inclination. Such a deployment of sensors is primarily inspired by the analysis of human gaits.

The LOCH robot has 14 degrees of freedom (DOF) in the lower body: each leg has 6 DOF, and the waist has 2 DOF. These joint controllers, each actuating one joint, are deployed over the lower body. Such a distributed structure uses the CAN bus to pass commands and coordinate motion of joints. The CAN bus is selected because of its prominent reliability and routability.

B. Software Architecture

In terms of the functionality, the control system can be divided into four modules. Namely they are the planning and control module, the sensory module, the actuation module and the communication module. The software architecture is designed to have the same modular division.

The planning and control module is the heart of the control system. It plans the walking gait, and then converts the gait into a series of joint commands for each sampling instant. The gait planning is performed on-line according to the expected behavior and the detected floor conditions. For every walking cycle, the planning is conducted for four times: two for the dual single support phases (SSP) and the other two for the dual double support phases (DSP).

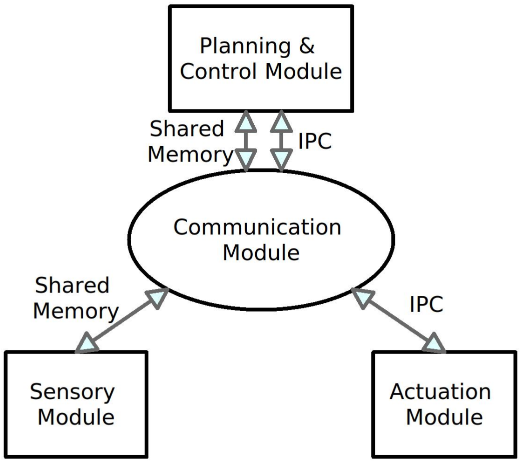

Fig. 3. Software architecture

The completion of one phase will trigger the planning of the following phase. The planned gait is amended by the stability controller and the landing controller, to achieve balanced walking and compliant landing of the swing leg. The amended gait is then passed to inverse kinematics for real-time joint control commands. The flowchart is shown in Fig 4.

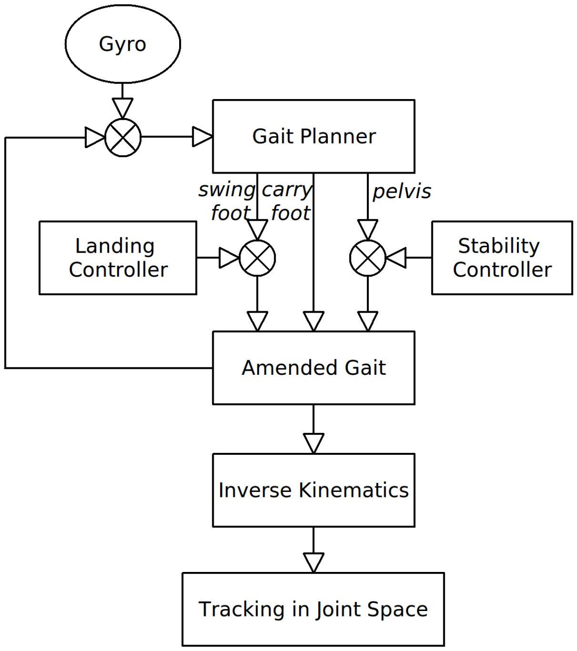

Fig. 4. Flowchart of the planning and control

The sensory module manages all the sensors onboard, performs necessary signal processing, and sends the sensory data to the walking and control module through interprocess communication. To ensure the data validity, the sensory module writes the sensory data into a specified shared memory segment at a higher frequency. Meanwhile the planning and control module retrieves these data periodically at a lower frequency. With such a mechanism, the overhead of the interprocess communication is minimized.

The actuation module runs the CAN bus and thereby handles the local joint controllers. It gets commands from the planning and control module through interprocess communication, and accordingly operates the CAN bus to inquire or change the motion status of joint controllers.

The communication module is the collection of the interprocess communication and the CAN bus communication. It monitors the communication status and maintains the communication infrastructure. Particularly it has an adaptive buffer that is designed to tolerate the timing error raised by the data exchange between the actuation module and the planning and control module. Data from the planning and control module are continuously fed into the buffer, and from the buffer the actuation module periodically retrieves data. If data amounts inside the buffer exceeds a threshold, which suggests the planning and control module is running at a higher frequency than the actuation module, the actuation module will slightly increase the running frequency of itself, and vice versa. This adaptive buffer conduces to compensating the timing error brought in by the OS scheduling.

III. Design of the On-line Gait Planner

A. Walking Phases and Inverted Arm Model

Biped walking is the combination of the placement of one leg and the movement of the rest of the body. It is a periodic phenomenon, and can be seen as the repetition of one walking cycle. In our design, one walking cycle is divided into four phases. The roles of the dual legs in each phase are listed in Table I. Planning of a certain phase is conducted on-line right at the beginning of this phase. Thus within one cycle, gait planning is conducted for four times.

TABLE I

Roles of dual legs in one cycle

phase No.	left leg role	right leg role	phase type
1	push	support	double support
2	swing	carry	single support
3	support	push	double support
4	carry	swing	single support

As an important property of biped walking, at least one foot is in contact with floor throughout the entire process. This foot on floor, under the assumption of no slip in between, is selected to be the base link of the leg chain. For instance, in Phase 1 and Phase 2, the right foot is the base link; While in Phase 3 and 4, the left foot is the base link. Moreover, change of base link is involved in the phase transition. The movement of the pelvis and the other foot is described relative to the base link. By doing this, walking robot can be represented by an inverted arm model. The rich knowledge of kinematics and dynamics of robot manipulators naturally becomes applicable. Meanwhile, this model enables the gait planning to be performed in the local co-ordinate frames, which brings benefits when the floor is not flat.

B. Gait Generation

The description of the walking gait involves three co-ordinate frames, denoted by $\{PV\}$, $\{LF\}$ and $\{RF\}$. $\{PV\}$ is set at the pelvis center with x-axis pointing to the front and y-axis pointing to the left side of the pelvis. $\{LF\}$ and $\{RF\}$ are respectively set at the left ankle and the right ankle, with the same definition of axis direction. With the above definitions, gait planning is converted to the design of trajectories for these three co-ordinate frames.

The clamped third order spline is selected for gait planning. For trajectories of double support phases, the position and the RPY angles of the start point and the target point, and the velocities thereof, must be specified. For trajectories of the single support phases, the position and RPY angles of an additional via point must also be designated.

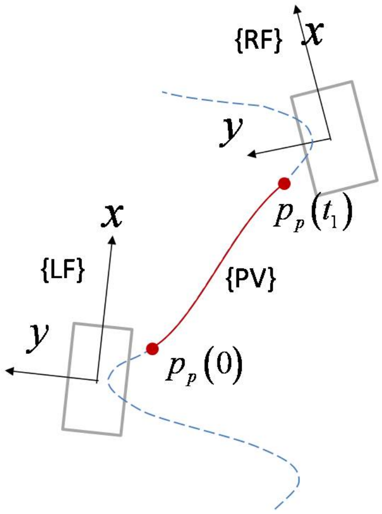

Fig. 5. Key points and the trajectories for Phase 1

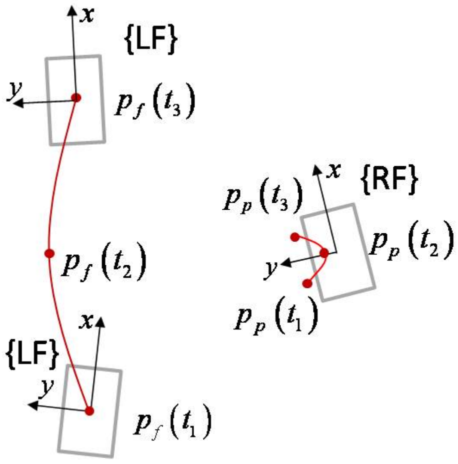

Fig. 6. Key points and the trajectories for Phase 2

Taking Phase 1 and 2 for example, the right foot is the base link, and trajectories of $\{LF\}$ and $\{PV\}$ are described relative to $\{RF\}$. The key points and the designed trajectories are shown in Fig 5 and Fig 6. $p_f(t)$ represents the foot trajectory, and $p_p(t)$ represents the pelvis trajectory. $p(t)$ is a 6×1 vector. Its first three rows denote the Cartesian coordinates relative to {RF}, and the last three rows represent the roll, pitch and yaw angles. t_1 is the DSP time and $t_3 - t_1$ is the SSP time.

$$p(t) = (x(t)\ y(t)\ z(t)\ \gamma(t)\ \beta(t)\ \alpha(t))^T \tag{1}$$

Given the walking parameters, such as the step length l_s, the step width w_s, the step height h_s, the pelvis height h_p, the lateral magnitude of pelvis m_p, etc, $p_f(t)$ and $p_p(t)$ can be parameterized. For example, at the instant $t = t_2$, a possible parameterization can be chosen as Eq 2. The parameterization relates the walking parameters to the trajectory interpolation.

$$\begin{aligned} p_f(t_2) &= (0\ w_s\ h_s\ 0\ 0\ 0)^T \\ p_p(t_2) &= (0\ m_p\ h_p\ 0\ 0\ 0)^T \end{aligned} \tag{2}$$

C. Evaluation of the Stability Margin

The designers have much freedom to choose walking parameters, such as the SSP time, the DSP time, the step height, etc. A basic requirement is that the resultant walking gait must ensure the walking stability. Based on the stability criterion, evaluation can be conducted on the walking parameters beforehand, which will provide much guidance in on-line parameter selection.

ZMP is a commonly used stability criterion. As long as the ZMP remains inside the supporting polygon, the walking gait is stable. Using the cart-table model, the ZMP is estimated from the movement of the center of mass (COM). Suppose the height of the COM is z_h, the coordinates of the COM is (x_{COM}, y_{COM}), then the coordinates of the ZMP satisfy[2]

$$x_{ZMP} = x_{COM} - \frac{z_h}{g}\ddot{x}_{COM} \tag{3}$$

$$y_{ZMP} = y_{COM} - \frac{z_h}{g}\ddot{y}_{COM} \tag{4}$$

In Phase 2 and 4, the supporting polygon is the outline of the carry foot. According to the inverted arm model, the expected location of ZMP is $(0,0)$. Therefore the cost function of the stability margin is selected to be Eq 5.

$$J_2 = J_4 = \int_{t_1}^{t_3} x_{ZMP}^2 + y_{ZMP}^2\, dt \tag{5}$$

In Phase 1 and 3, the supporting polygon is the minimal convex polygon covering the two feet on floor. The expected trajectory of the ZMP is considered to be the straight line connecting the centers of the dual feet. Here the squared distance of the actual ZMP to this straight line is selected as the cost function. E.g, in Phase 1, the right foot center is $(0,0)$, and suppose the center of the left foot is (x_L, y_L). Then the cost function has the form of Eq 6. Similarly, the cost function of Phase 3 has the form of Eq 7.

$$J_1 = \int_0^{t_1} \frac{(x_{ZMP}y_L - x_L y_{ZMP})^2}{x_L^2 + y_L^2}\, dt \tag{6}$$

$$J_3 = \int_0^{t_1} \frac{(x_{ZMP}y_R - x_R y_{ZMP})^2}{x_R^2 + y_R^2}\, dt \tag{7}$$

With the above cost functions, various optimization methods can be applied for optimal parameters[10]. The evaluation uses the cart-table model, and thus is a rough estimation of the walking stability. However, it gives a collection of allowed walking parameters beforehand. The collection will be refined though further experiments, and used as candidates of walking parameters for on-line gait planning.

D. Handling the Uneven Floor with an Imaginary Foot

Walking gait for flat floor is probable to cause the robot on uneven floor to tipover. One solution is to enhance the stability by real-time balance control and landing control. It is effective for the cases where the floor conditions gently change. A better solution is to make the gait planning adaptive to floor changes.

To realize such an adaptability, in the rear part of Phase 2 and 4, the ankle of the swing foot is to mimic a passive joint, and the foot landing in the vertical direction is to be terminated on firm foot-floor contact. The force/torque sensor installed on the ankle provides the feasibility of ankle passivity. Meanwhile, the pressure sensors mounted on the foot sole will report the firm foot-floor contact. A landing controller, which will be presented in the following section, is also designed for the control logic.

Following the completion of Phase 2 and 4, the flatness of the floor beneath the swing foot is estimated. First we give the convention to describe the spatial relationship between co-ordinate frames. The notation, ${}^{M}_{N}R$, refers to the rotation matrix of a frame {N} relative to another frame {M}. Due to the hardware configuration, the robot has the following properties.

1) ${}^{W}_{PV}R$ available. From the gyro equipped onto the trunk, the orientation of {PV}, ${}^{W}_{PV}R$, is available. {W} denotes the inertial global frame with x-axis pointing to the north, and y-axis pointing to the west. Apparently the x-y plane of {W} is the horizontal plane.
2) ${}^{PV}_{LF}R$ and ${}^{PV}_{RF}R$ available. This is a natural result of forward kinematics.

Taking Phase 2 for example, on completion, the orientation of the swing foot is calculated by

$${}^{W}_{LF}R = {}^{W}_{PV}R\, {}^{PV}_{LF}R \tag{8}$$

Set an auxiliary coordinate frame {LF'} onto the left ankle. It has the same origin and the same yaw angle as {LF}, but its x-y plane is parallel to that of {W}. {LF'} can be seen as an imaginary left foot stepping on the flat floor. At the beginning of the following Phase 3 and Phase 4, the gait planning is done relative to {LF'}, as if the robot is walking on the horizontal plane. Before execution, the trajectories are converted from {LF'} into the frame {LF}, and then sent to inverse kinematics.

Let $RPY(.)$ be the function that converts a rotation matrix into RPY angles, and $R(.)$ the function that converts RPY angles into a rotation matrix. Furthermore, suppose

$$RPY({}^{W}_{LF}R) = (\gamma,\, \beta,\, \alpha) \tag{9}$$

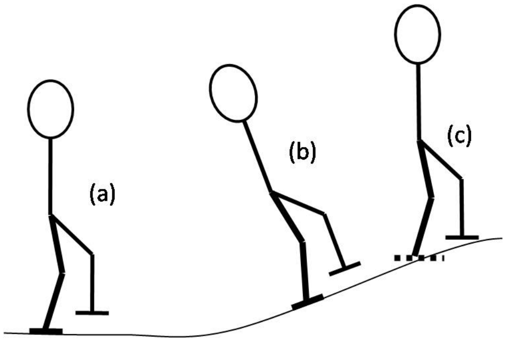

Fig. 7. Imaginary foot. (a) Planning relative to the real foot on flat floor; (b) Planning relative to the real foot on uneven floor;(c) Planning relative to the imaginary foot (the dashed) on uneven floor.

Then we have

$$ {}^{W}_{LF'}R = R(0,\ 0,\ \alpha) \tag{10} $$

$$ {}^{LF}_{LF'}R = {}^{W}_{LF}R^{-1}\ {}^{W}_{LF'}R = {}^{W}_{LF}R^{-1}\ R(0,\ 0,\ \alpha) \tag{11} $$

Since {LF'} and {LF} share the common origin, with ${}^{LF}_{LF'}R$, the trajectories of {PV} and {RF} that have been planned relative to the imaginary foot, {LF'}, can be easily transformed with respect to the real foot {LF}.

IV. Design of the Balance Controller

The application of the planned walking gaits cannot necessarily ensure the walking stability. The main reason is the difference between the cart-table model and the real robot dynamics. Feedback control is helpful to maintain and enhance the walking stability. Therefore ZMP-based balance controller is designed.

For single support phases, according to the inverted arm model, the carry foot is seen as the base link, and the hip is regarded as the end effector. If the carry leg imposes an active force $-F_a$ at the hip, the reaction force, F_a, can be used to counteract the tipping torque acting on the carry foot. This is a brief explanation of the control mechanism. For double support phases, a similar explanation can be given.

Suppose the desired ZMP position is denoted by (x^d_{ZMP}, y^d_{ZMP}), and the actual ZMP is located at (x_{ZMP}, y_{ZMP}), which is computable with force/torque outputs from the ankle. A proportional control law is designed.

$$ F^x_a = k_p(x^d_{ZMP} - x_{ZMP}) \tag{12} $$

$$ F^y_a = k_p(y^d_{ZMP} - y_{ZMP}) \tag{13} $$

Since the sampling frequency is high, the control force can be approximated by adjusting the pelvis position (p_x, p_y) in real-time. Accordingly the proportional control is replaced by a PI type control to eliminate the effect of approximation errors. PI type of controller is selected for its strong robustness, and more sophisticated controllers can be applicable if the dynamic model has sufficient precision[8].

$$ \Delta p_x(t) = -k_p(x^d_{ZMP} - x_{ZMP}) - k_i \int_0^t (x^d_{ZMP} - x_{ZMP})\,dt \tag{14} $$

$$ \Delta p_y(t) = -k_p(y^d_{ZMP} - y_{ZMP}) - k_i \int_0^t (y^d_{ZMP} - y_{ZMP})\,dt \tag{15} $$

V. Design of the Landing Controller

Unexpected foot-floor collision in the foot swing process is a serious threat to walking stability. To avoid such collision, the landing controller is designed to provide compliance to changes in floor inclination and height.

The landing controller is activated in the rear part of Phase 2 and 4. It takes full control of the two joints on the swing ankle in response to force/torque readings. With respect to the co-ordinate frame of the swing foot, the control law is described as Eq 16 and Eq 17, where (τ_x, τ_y) is torque readings from the force/torque sensor on the swing ankle.

$$ \Delta\gamma(t) = k^p_\gamma \tau_x + k^d_\gamma \frac{d}{dt}\tau_x \tag{16} $$

$$ \Delta\beta(t) = k^p_\beta \tau_y + k^d_\beta \frac{d}{dt}\tau_y \tag{17} $$

The walking gait is described relative to the carry foot, but $(\Delta\gamma, \Delta\beta)$ is relative to the swing foot itself. Therefore $(\Delta\gamma, \Delta\beta)$ must be converted into the co-ordinate frame on the carry foot before execution. Taking Phase 2 for example, relative to {RF}, suppose the current orientation of the swing foot is R_{LF}. The amended one, R^m_{LF}, is computed by Eq 18.

$$ R^m_{LF} = R_{LF}\ R(\Delta\gamma, \Delta\beta, 0) \tag{18} $$

The control law Eq 16 and Eq 17 enable the landing foot to adapt to changes in floor inclination. Meanwhile, for the changes in floor height, we give the following landing termination condition,

- *The foot landing is terminated if the pressure sensor on the foot sole report firm foot-floor contact.*

Another choice of termination condition is the force readings from the force/torque sensor. For an irregularly inclined floor, however, the ground reaction force has an uncertain direction. Thus it's more reliable to use the pressure sensors on the sole.

VI. Simulation and Experiments

Simulations and experiments are performed to verify the proposed design. Select the step length to be 0.10m and 0.30m respectively, and the SSP time and DSP time to be 0.9s and 0.7s. The planned pelvis trajectories for Phase 1 and 2 are illustrated with solid line in Fig 8 and Fig 9. The resultant ZMP trajectories are drawn with dashed line for Phase 1 and dotted line for Phase 2. The two rectangles in either of these two figures represent the robot feet.

Walking stability can be evaluated from Fig 8 and Fig 9. In Phase 1, the ZMP remains inside the supporting polygon that are spanned by the two feet. In Phase 2, ZMP is inside the outline of the right foot. According to the ZMP criterion, we conclude that the planned gaits are both stable gaits.

Further experiments are carried out on the LOCH robot, and the snapshots are shown in Fig 10. In the experiments the robot is commanded to walk forward for three steps. The moving frame in the snapshots is used to protect the big robot, and it didn't impose any force onto the walking robot.

In terms of uneven terrain, a simulation is also given as shown in Fig 11. A hump with the height of 1cm is placed

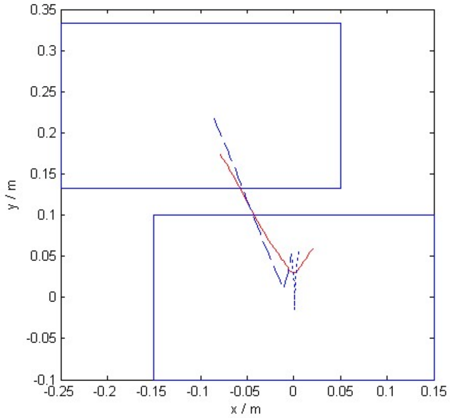

Fig. 8. The pelvis trajectory and the resultant ZMP trajectory when the step length is 0.10m.

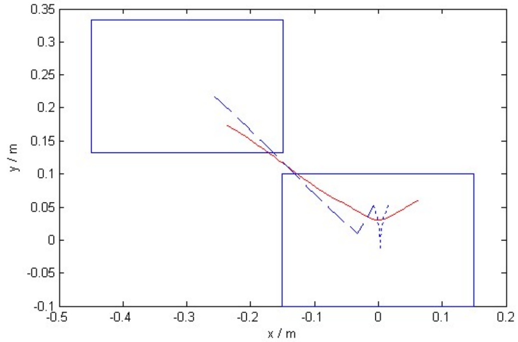

Fig. 9. The pelvis trajectory and the resultant ZMP trajectory when the step length is 0.30m.

on the floor, and its top surface is not flat either. In the snapshots, the red line represents the force vector obtained from the force/torque sensors on the ankle. And the pressure sensors on the sole change colors if touching the floor. The simulated robot walks over the hump stably with its gait autonomously adjusted. The experiments for uneven floor is in progress, and will be presented in future publications.

VII. Conclusions

This paper presents the control system design for the locomotion of LOCH robot. The system architecture, the gait planner and controllers are studied in detail. In dealing with the uneven terrain, an imaginary foot method is proposed. Finally, simulations and experiments are conducted to illustrate the effectiveness of the control design.

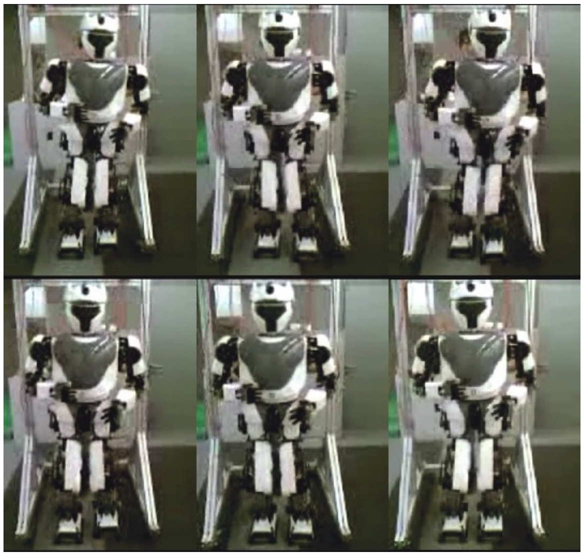

Fig. 10. Snapshots of biped walking performed on flat floor

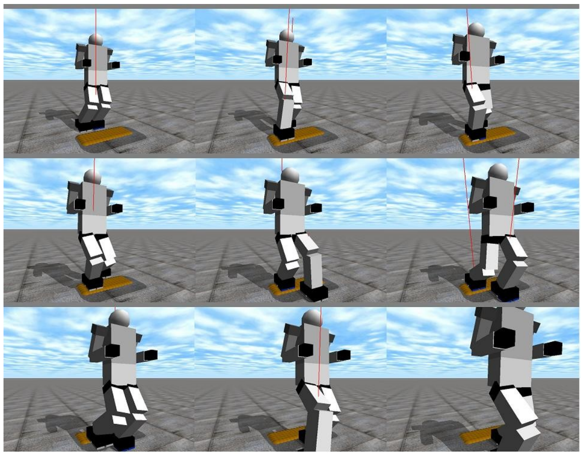

Fig. 11. Snapshots of simulation for uneven floor

References

[1] Y. Sakagami, R. Watanabe, C. Aoyama, et al, ‘The intelligent ASIMO: system overview and integration‘, *in Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 2002.*

[2] S. Kajita, M. Moriswa, K. Harada, et al, ‘Biped walking pattern generator allowing auxiliary ZMP control‘, *in Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 2006.*

[3] K. Nishiwaki and S. Kagami, ‘Walking control on uneven terrain with short cycle pattern generation‘, *in Proc. IEEE-RAS Int. Conf. on Humanoid Robots,2007*

[4] Y. Ogura, H. Aikawa, K. Shimomura, et al, ‘Development of a new humanoid robot WABIAN-2,‘ *in Proc. IEEE Int. Conf. Robotics and Automation, 2006*

[5] JY Kim, IW Park and JH Oh, ‘Walking control algorithm of biped humanoid robot on uneven and inclined floor‘, *Journal of Intelligent Robot System*, vol. 48, 2007, pp457-484.

[6] M. Xie, ZW Zhong, L. Zhang, et al, ‘A deterministic way of planning and controlling biped walking of LOCH humanoid robot‘,*Industrial Robot - An International Journal*, vol. 36, 2009, pp 317-325.

[7] GQ Zhang, M. Xie, H. Yin, et al, ‘Planning and control of biped walking along curved paths on unknown and uneven terrain‘, *Intelligent Robotics and Applications, LNAI.* 2009, Springer. pp 1032-1043.

[8] S. Czarnetzki, S. Kerner and O. Urbann, ‘Observer-based dynamic walking control for biped robots‘, *Robotics and Autonomous Systems*, vol. 57, 2009, pp 839-845.

[9] M. Vukobratovic, B. Borovac, ‘Zero-moment point- thirty five years of its life‘, *Int. Journal of Humanoid Robotics*, vol. 1, 2004, pp 157-173.

[10] GQ Zhang, M.Xie, B.Phuong, et al, ‘Optimization of Online Walking Gaits for the LOCH Humanoid Robot Using Genetic Algorithm‘, *to appear in the 13th CLAWAR, 2010.*

[11] Y. Hurmuzlu, F. Genot and B. Brogliato, ‘Modeling, stability and control of biped robots - a general framework‘, *Automatica*, vol. 40, 2004, pp 1647-1664.

[12] Q. Huang, K. Yokoi, S. Kajita, et al, ‘Planning walking patterns for a biped robot‘, *IEEE Trans. on Robotics and Automation*, vol. 17, 2001, pp 280-289.