

ODENS 2014 Team Description

Takuya Maruyama, Yasuhiro Masutani

Osaka Electro-Communication University
1130-70, Kiyotaki, Shijonawate, Osaka 575-0063, Japan

Abstract. In this paper, Team ODENS of Osaka Electro-Communication University is introduced. Especially methods of generating the gait and kick motion are described. The gait generation is based on a cart-table model and uses the preview control to follow desired ZMP. The kick motion is planned to smoothly follow walking to reduce the time lag.

1 Introduction

Team ODENS consists of members of Masutani Laboratory in Department of Computer Science, Faculty of Information Science and Arts, Osaka Electro-Communication University, Japan. ODENS has participated in RoboCup competition since RoboCup Japan Open 2007. ODENS participates the Small-Size Robot League, the SSL Humanoid (sub league of the Small-Size Robot League, competition of humanoid robots by using external camera) [1], and the 3D Simulation League at present. They have participated in the 3D Simulation League since 2009. The results of ODENS in the league, Japan Open were the 4th place in 2009, the 3rd place in 2010, the champion in 2011 and 2012, and the 3rd place in 2013. They also have participated in the world competition in the league since 2012.

In the Department of Computer Science, students belong to laboratories from the second semester of the second grade. In Masutani Laboratory, projects for RoboCup are themes for pre-seminar before regular graduation thesis. Moreover some students study RoboCup as also theme of graduation thesis.

An advantage of the ODENS simulation group is that there is a group working on the real humanoid robot in the same laboratory. Although they don't use the Nao, they can get hints and motivation by looking at real robots nearby.

In the following sections, the overview of the system of ODENS is introduced in Section 2. An online gait generation based on cart-table model is described in Section 3. A method of generating kick motion smoothly following walking motion is explained in Section 4.

2 Overview of system of ODENS

ODENS uses libbats2.0.1 as their base program, which is developed and released by the Little Green Bats[2]. Vision and sensor signals are processed by the functions of libbats. All their efforts are focused on developing robot motion and behavior decision.

2.1 Motion of robot

Motion generating is divided into two types. One is offline, for examples, getting-up. The other is online and base on the dynamic model, for example, walking and kicking.

The online motion generation is described in Section 3 and later.

2.2 Behavior decision

Each player has four roles, Attacker, Forward, Defender, and Goalkeeper. Attacker is the player closest to the ball. The other roles are determined by their uniform number. No.1 is Goalkeeper, No.2 to 6 are Defenders, No.7 to 11 are Forwards.

As shown in the Fig.1, the field is divided into 12 areas. Behavior of the player is determined by rule of every role based on the area where it is.

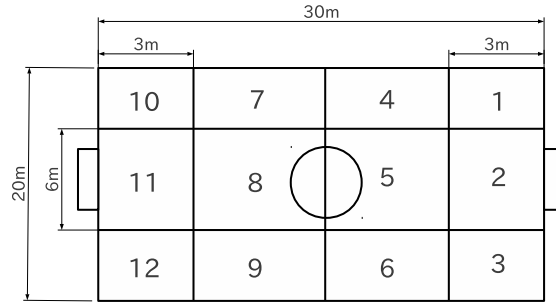


Fig. 1. Field divided into 12 areas

Behavior of each role is as follows;

Attacker

Attacker carries the ball toward the opponent goal. He uses dribbling and kicking.

Forward

Forward stays around the ball and becomes Attacker in case that former Attacker has missed the ball.

Defender

Defender stays own team's area interferes with the opponent in case that friend Forward has been broken.

Goal Keeper

Goalkeeper defends own team's goal staying in front of it.

3 Online gait generation

In the program of ODENS, gait is generated online. In case of offline gait generation, it is impossible to generate suitable gait for various situations. Online gait generation allows to adjust the motion to any situation.

3.1 Cart-table model

The gait generation is based on a cart-table model. As shown in Fig.2, this model consists of a running cart on a mass-less table. The position of the cart corresponds to COM of the robot and its mass is equal to the whole mass of the robot. The COM of the robot is assumed to be controlled keeping its height. ZMP equation for this simplified model is given in Eq.(1).

$$p(t) = x(t) - \frac{z_c}{g} \ddot{x}(t) \quad (1)$$

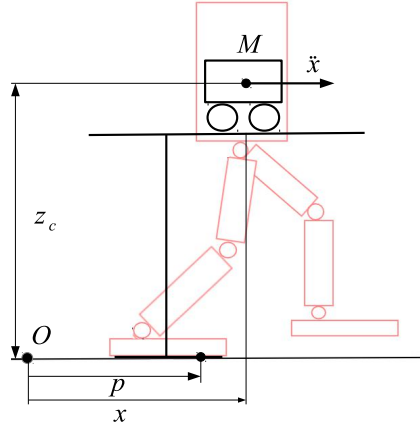


Fig. 2. Cart-table model

3.2 Preview control

By using the method of preview control, trajectory of COM is computed from the desired future ZMP. Considering a system equation by taking the jerk of the cart as the system input u as described in Eq.(2) and (3)

$$\frac{d}{dt} \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u \quad (2)$$

$$p = \begin{bmatrix} 1 & 0 & -z_c/g \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} \quad (3)$$

Next Eq.(2) and (3) are transformed to the discrete system (4) using the sampling period Δt .

$$\begin{cases} \mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{b}u_k \\ p_k = \mathbf{c}\mathbf{x}_k \end{cases} \quad (4)$$

where $\mathbf{x}_k$, u_k , p_k , $\mathbf{A}$, $\mathbf{b}$, and $\mathbf{c}$ are defined as follows;

$$\begin{aligned} \mathbf{x}_k &\equiv \begin{bmatrix} x(k\Delta t) \\ \dot{x}(k\Delta t) \\ \ddot{x}(k\Delta t) \end{bmatrix} \\ u_k &\equiv u(k\Delta t) \\ p_k &\equiv p(k\Delta t) \\ \mathbf{A} &\equiv \begin{bmatrix} 1 & \Delta t & \Delta t^2/2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix} \\ \mathbf{b} &\equiv \begin{bmatrix} \Delta t^3/6 \\ \Delta t^2/2 \\ \Delta t \end{bmatrix} \\ \mathbf{c} &\equiv [1 \ 0 \ -z_c/g] \end{aligned}$$

Based on the optimal control theory, the optimal controller that minimize an objective function is given by Eq.(5). The desired ZMP is given as $[p_{k+1}^t, \dots, p_{k+N}^t]^T$ in former N steps.

$$u_k = -\mathbf{K}\mathbf{x}_k + [f_1, f_2, \dots, f_N] \begin{bmatrix} p_{k+1}^t \\ \vdots \\ p_{k+N}^t \end{bmatrix} \quad (5)$$

where $\mathbf{K}$ and f_i are given as follows;

$$\begin{aligned} \mathbf{K} &\equiv \left[(R + \mathbf{b}^T \mathbf{P} \mathbf{b})^{-1} \mathbf{b}^T \mathbf{P} \mathbf{A} \right] \\ f_i &\equiv (R + \mathbf{b}^T \mathbf{P} \mathbf{b})^{-1} (\mathbf{A} - \mathbf{b} \mathbf{K})^{T(i-1)} \mathbf{c}^T Q \end{aligned} \quad (6)$$

Q and R are weights in the objective function. $\mathbf{P} \in R^{3 \times 3}$ is the solution of the Riccati equation (7).

$$\mathbf{P} = \mathbf{A}^T \mathbf{P} \mathbf{A} + \mathbf{c}^T Q \mathbf{c} - \mathbf{A}^T \mathbf{P} \mathbf{b} (R + \mathbf{b}^T \mathbf{P} \mathbf{b})^{-1} \mathbf{b}^T \mathbf{P} \mathbf{A} \quad (7)$$

3.3 Stride parameters

They use stride parameters as shown in Eq.(3) where $d_l[\text{m}]$ denotes the distance between left and right foot and $s_x[\text{m}]$, $s_y[\text{m}]$, $s_\theta[\text{deg}]$ denote strides in forward direction, lateral direction, and, rotational angle respectively. By using these parameters, the next landing position is represented with reference to the coordinate system of the current supporting foot in Eq.(8) in case of left foot reference and in Eq.(9) in case of right foot reference.

$$\begin{bmatrix} p_x \\ p_y \end{bmatrix} = \begin{bmatrix} \cos s_\theta & -\sin s_\theta \\ \sin s_\theta & \cos s_\theta \end{bmatrix} \begin{bmatrix} s_x \\ s_y - \frac{d_l}{2} \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{d_l}{2} \end{bmatrix} \quad (8)$$

$$\begin{bmatrix} p_x \\ p_y \end{bmatrix} = \begin{bmatrix} \cos s_\theta & -\sin s_\theta \\ \sin s_\theta & \cos s_\theta \end{bmatrix} \begin{bmatrix} s_x \\ s_y + \frac{d_l}{2} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{d_l}{2} \end{bmatrix} \quad (9)$$

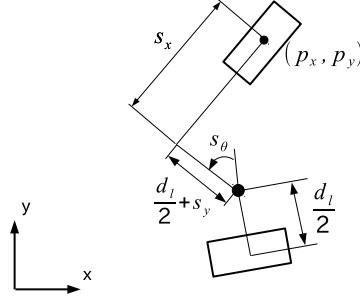


Fig. 3. Stride parameters

3.4 Gait generation

In order to generate the gait, first the trajectory of COM is computed by using the above method. Second angles of all joints in supporting leg are obtained by solving inverse kinematics of the COM. Finally, the position of foot of swing leg is determined then the angle of all joints in swing leg are obtained. Angular velocity command of each joint sent to the server is computed from the trajectory of its angle.

3.5 Evaluation

In order to confirm the gait generation, an experiment in the environment of the 3D Simulation League with parameters in Table 1 is carried out. The time sequences of the desired ZMP, the actual ZMP, and the position of COM of x and y directions are shown in Fig.4 and Fig.5 respectively. It is found that the actual ZMP well follows the desired ZMP.

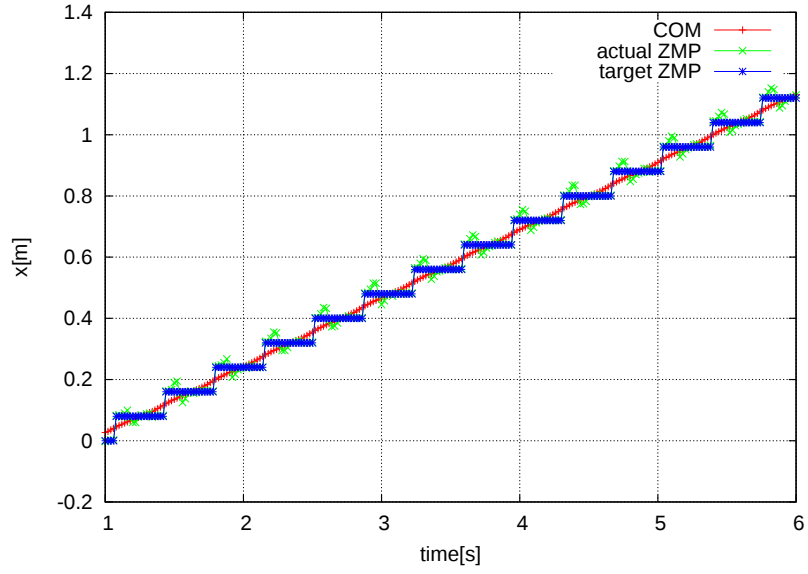
4 Omni-directional kicking

4.1 Kicking smoothly following walking

In the previous program of ODENS, the linear inverted pendulum model was employed to realize kick motion smoothly following walking. In 2014, since it is replaced with the

Table 1. Parameters for an example walking

parameter		value
control period	Δt	0.02[s]
number of preview steps	N	100
weight for error	Q	1000000
weight for input	R	0.001
height of COM	z_c	0.247[m]
walking period	T_s	0.36[s]
maximum stride in x	s_{xmax}	0.08[m]
maximum stride in y	s_{ymax}	0.07[m]
maximum stride in θ	$s_{\theta max}$	15[deg]
distance between feet	d_l	0.11[m]

**Fig. 4.** Trajectories of desired ZMP, actual ZMP, and COM (x direction)

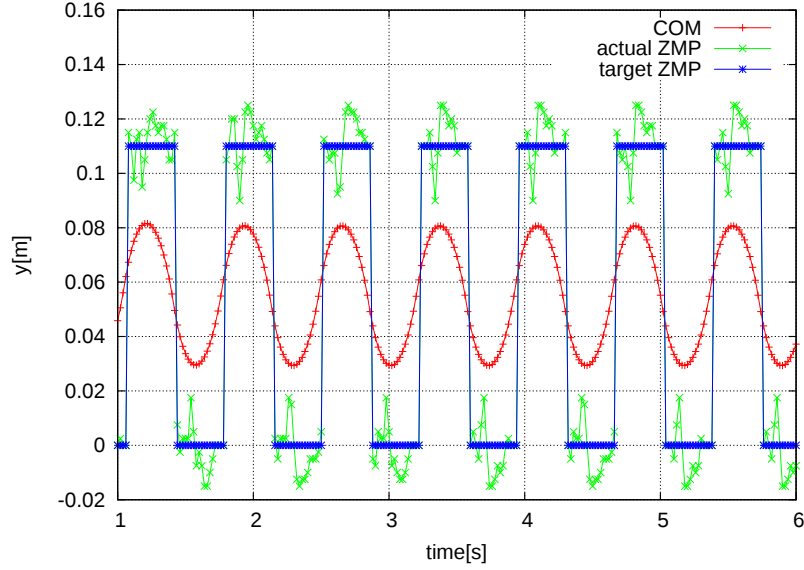


Fig. 5. Trajectories of desired ZMP, actual ZMP, and COM (y direction)

cart-table model, the distance of approach walk before kick is shorter and the restriction on design of kick motion is weaker. Just before the kick, the supporting leg becomes a pivot leg. On the other side, the swing leg becomes a kick leg. Motion of the pivot leg is moved so that the position of COM satisfies the requirement of the preview controller. Motion of the kick leg is designed in four phases as shown in Fig.6.

4.2 Switch of kicking

An omni-directional kicking is realized based on the method mentioned above. Let θ_t denote the desired direction of kicked ball. In case that $|\theta_t| \leq 90[\text{deg}]$, kick foot and kick type is switched as shown in Fig.7. In case that $|\theta_t| > 90[\text{deg}]$, the robot approaches ball via waypoint then uses inside kick as shown in Fig.8

4.3 Evaluation

In order to evaluate the proposed omni-directional kick, an experiment in the environment of the 3D Simulation League is carried out. In the initial state, the ball is placed at the origin and the robot is 1.5[m] away from the ball. Five trials for 36 directions are executed. The errors in kicked ball direction (RMS) and the distances of kicked ball (arithmetic mean) for all directions are shown in Fig.9 and Fig.10 respectively. The distance should be improved in the future work.

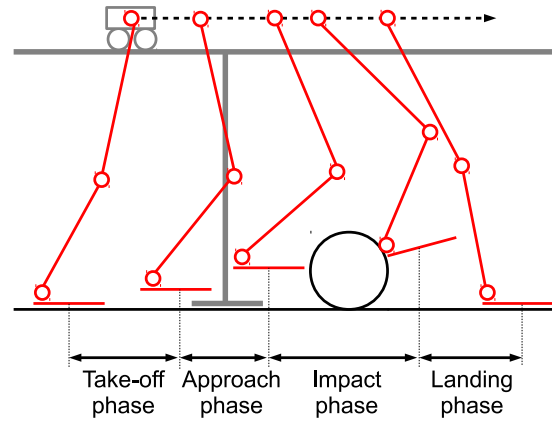


Fig. 6. 4 phases of kicking leg

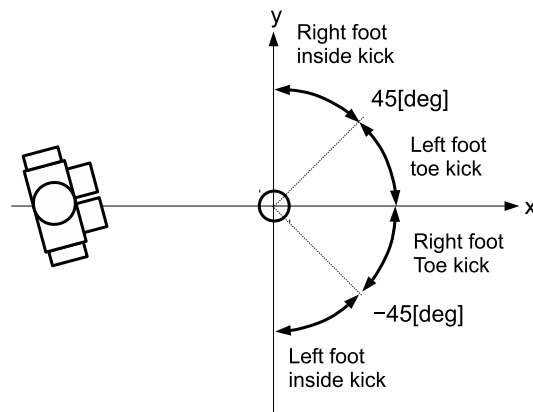


Fig. 7. Switch of kicking

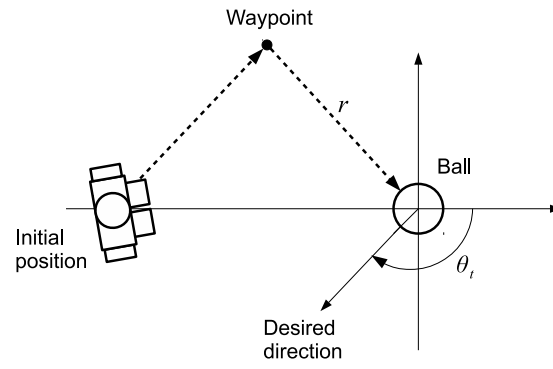


Fig. 8. Walk path for kick

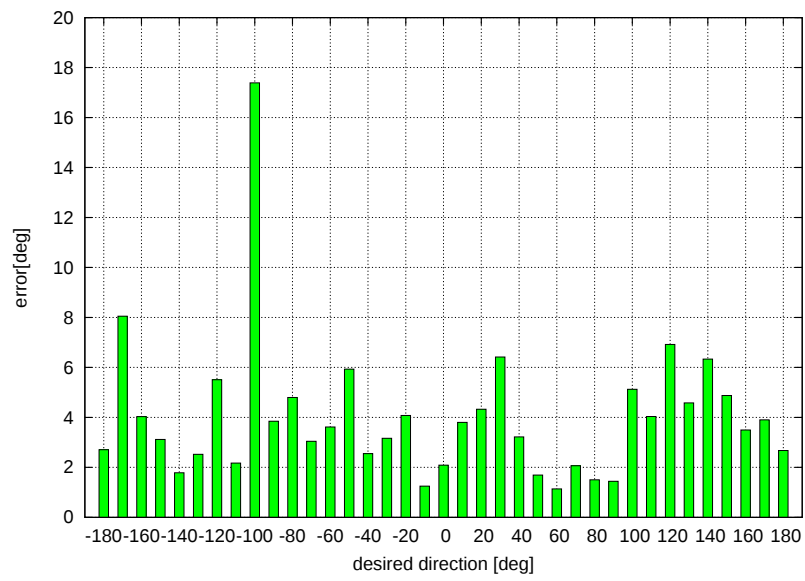


Fig. 9. Direction error of omni-directional kick (RMS)

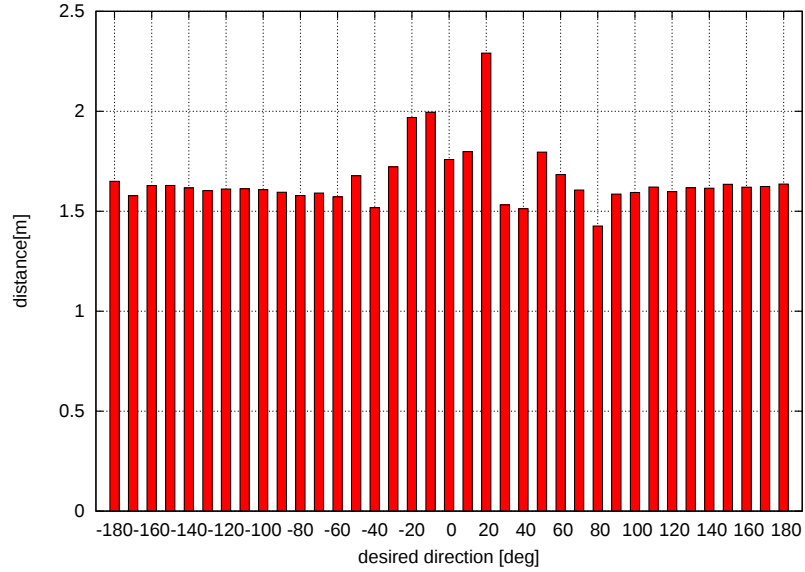


Fig. 10. Distance of omni-directional kick (arithmetic mean)

5 Conclusion

In this paper, control of individual robot of Team ODENS has been explained. By introducing dynamical model, generation of robot motion are achieved well.

The future works are a stabilization by using feedback of sensor information. Moreover, more flexible role assignment will be studied.

References

1. Tadashi Naruse, Yasuhiro Masutani, Noriaki Mitsunaga, Yasunori Nagasaka, Takashi Fujii, Masato Watanabe, Yukiko Nakagawa, and Osamu Naito, “SSL-Humanoid: RoboCupSoccer using humanoid robots under the global vision”, pp.60-71, RoboCup 2010: Robot Soccer World Cup XIV, Springer (2011).
2. littlegreenbats in Launchpad, <https://launchpad.net/littlegreenbats/>.
3. Shuuji Kajita, Humanoid Robot, Ohm-sha,(2005) (in Japanese).