

Soccer playing humanoid robots: Processing architecture, gait generation and vision system

Prahlad Vadakkepat*, Ng Buck Sin, Dip Goswami, Zhang Rui Xiang, Tan Li Yu

National University of Singapore, Electrical and Computer Engineering, 4 Engineering Drive 3, Singapore 117576, Singapore

ARTICLE INFO

Article history:

Available online 8 April 2009

Keywords:

Humanoid
Soccer playing robot
Processing architecture
Gait generation
Vision system

ABSTRACT

Research on humanoid robotics in Mechatronics and Automation (MA) Laboratory, Electrical and Computer Engineering (ECE), National University of Singapore (NUS) was started at the beginning of this decade. Various research prototypes for humanoid robots have been designed and are going through evolution over these years. These humanoids have been successfully participating in various robotic soccer competitions. In this paper, three major research and development aspects of the above humanoid research are discussed. The paper focuses on various practical and theoretical considerations involved in processing architecture, gait generation and vision systems.

© 2009 Elsevier B.V. All rights reserved.

1. Introduction

In the world we live in, many areas are created or configured for human access such as catwalks, tunnels, ladders or other restricted areas. Such areas can only be accessible by biped humanoids with a structure similar to humans. Recently, research on biped robots has attracted a lot of interest. The studies on biped robots are generally undertaken along two directions: (a) to elucidate the locomotion mechanism of humans from (robotics) engineering viewpoints, and (b) to impart intelligence for reproducing human-like activities.

1.1. FIRA and RoboCup

There are several international robotic soccer competitions for humanoid robots. RoboCup (www.robocup.org) and FIRA (www.fira.net) act as fuel to draw the worldwide attention for the research and development on humanoid robots. These competitions help to exchange various technologies among the research communities.

1.2. The NUS humanoid robot

The research on humanoid robots in MA Laboratory, ECE, NUS started in early this decade. The first generation of autonomous soccer playing humanoid robot, MANUS (Fig. 1(a)), was released in 2001. MANUS has 17 degrees-of-freedom (DOF) and is capable of performing various complex soccer playing tasks such as penalty striker and dynamic balancing. Over the years, research has been undertaken towards implementing several soccer playing tasks,

and modifications are made to improve the existing humanoid platform. The latest prototype, GENUS (Fig. 1(b)), was released in 2007. With 21 DOF GENUS is capable of many features required in soccer playing. It operates as either a striker or a goalkeeper. It can dive and recover from a fallen position.

The research and development on the humanoid robots in NUS is not confined to theoretical, simulation and experimental studies. Humanoid research is kept abreast by actively participating at various competitions such as FIRA and RoboCup to compete with the contemporary humanoid platforms. MANUS is the overall humanoid champion at FIRA RoboWorld Cup competition in the years 2003, 2005 and 2006. It has won first runner up in the FIRA RoboWorld Cup competition 2004. GENUS is the first runner up for the FIRA RoboWorld Cup competition 2007 in the robot dash category.

In this paper, the evolution of three major aspects of humanoid research is discussed: processing architecture, gait generation, and vision. Section 2 discusses related works reported in the literature to define the scope of the paper. The evolution of processing architectures over the years is discussed in Section 3. Various gait generation techniques, used in the humanoids, are explained in Section 4. Section 5 provides an insight into the vision system used to achieve better image data acquisition. The paper is concluded in Section 6.

2. Related works

2.1. Processing architecture

The processing architecture has a severe impact on real-time decision making and task execution performed by humanoid robots. Generally, the architecture comes under two categories: Multi-Processing and Multi-Tasking.

* Corresponding author. Tel.: +65 65 162296.

E-mail address: elepv@nus.edu.sg (P. Vadakkepat).

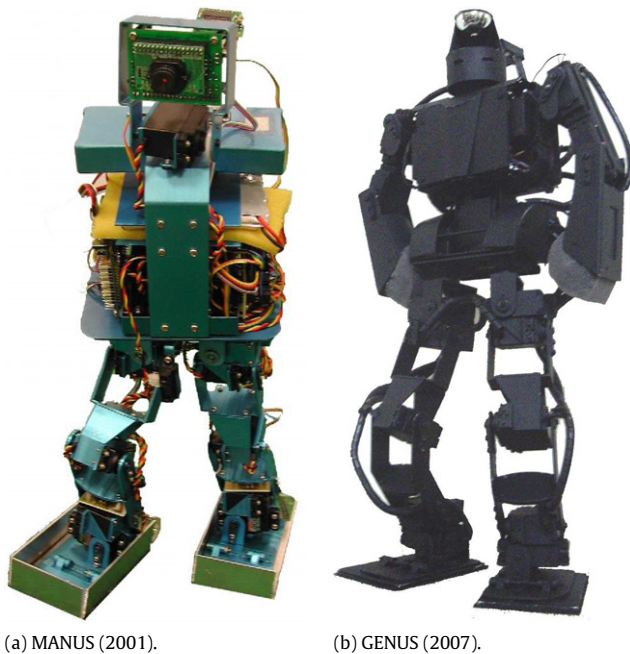


Fig. 1. The NUS humanoid robots.

Multi-Processing: In multi-processor architecture, more than one processor is utilized to handle the processing requirement for accomplishing a particular task. In humanoid robots, the requirement of parallel processing arises frequently. For example, information gathering from sensors, decision making based on them and gait generation for motion are expected to be pursued in parallel. The multi-processing architecture enhances the processing capabilities of such systems. However, use of multiple processors leads to increased complexity, space and resource overhead. Use of multi-processors in humanoid robots with height less than 60 cm is often restricted by its space and weight constraints. Moreover, multi-processor architecture needs a proper communication protocol for information exchange among different processing units.

Multi-Tasking: In single processor architecture, where task execution is often done sequentially, a certain task is executed after the completion of another. In the case of humanoids, different sensor data are very often read sequentially. The major disadvantage of such architecture lies in the bottleneck of sensors' feedback latency time and the resulting delay in decision making.

2.2. Gait generation

The motion of a humanoid robot comprises of time functions, trajectories, corresponding to angular positions and velocities of the robot's joints. Motion generated, as a combination of all these trajectories, is known as gait. The most noticeable difference between gait generation, in biped and multi-legged or wheeled locomotion, is the stability aspects. Generally, biped gaits are referred as stable when they are able to keep the structure upright. There are several analytical methods to verify the stability of certain biped gait. Zero-Moment-Point (ZMP) [1–7] is one of the most frequently used concepts in stability analysis. ZMP is defined as the point on the ground where the net moment of the inertial forces and the gravity forces have no component along the horizontal axes. Sufficient conditions for stable locomotion is to have the ZMP within the support polygon, the convex hull of the foot-support area, at all stages of the locomotion gait [8].

Based on the above analysis, biped gait generation is classified into three categories: trajectory-based, dynamic-based and the hybrid combination of trajectory-based and dynamic-based.

Trajectory-based Gait Generation. In Trajectory-based gait, biped robots use gaits that are generated from actual human walking. Several reported works on this method utilize additional control units to augment the reference joint trajectories [9]. A common form of augmentation is to use the Zero Moment Point (ZMP) to compute a desired stable foot placement [5]. Postural stability of bipeds are ensured by keeping the ZMP within the support polygon formed by the feet.

Central Pattern Generators (CPG) is another example of trajectory based approach. CPGs have been identified in most animals as being responsible for controlling many oscillatory activities like walking, swimming etc. They are mathematically formed by a set of coupled oscillators, some of which are excitatory and others inhibitory in nature. These oscillators, when configured properly, generate complete walking patterns. Many CPG models are proposed to explain the mechanism of motion pattern generation in humans. In [10], user supplied commands generate basic CPG. Cerebellar Model Articulation Controller (CMAC) [11, 12] neural network is used to generate joint angles for the proper balancing of a biped walking robot. Zheng [13] proposed a CPG which uses Van Der Pol oscillators to generate stable gait data for each joint.

Learning techniques such as neural network [14,15] and reinforcement learning [16] are used to generate walking gaits. Using feedback data and the learned parameters, the robot is able to control its stability when it is walking on uneven or inclined surface terrain.

A major drawback in using the trajectory-based control is the need for a large database to store the gaits required and very often the amount of motion data becomes insufficient to generate proper motion gaits. Nevertheless, trajectory-based control has achieved notable success in biped locomotion.

Dynamics-based Gait Generation. The dynamics-based control generates the joint time trajectories by solving inverse kinematics to maintain the physical stability of the humanoid based on system dynamics [17,18]. With an increase in DOF of the robot, it becomes computationally impractical to compute inverse kinematics. Moreover, such an approach often results in unnatural-looking motions and energy inefficiency. However, these approaches are suitable for off-line generation of joint trajectories.

The idea of dynamics-based control is modified and implemented by different approaches using natural dynamics in [19,20]. In these approaches, robots walk with natural gaits and energy efficiency. The advantage of dynamics-based control lies in the concept of using the passive effects of gravity and inertia to generate the gaits.

Hybrid approaches use a combination of both dynamics and trajectory based methods for gait generation. One such example is to use inverse kinematics to solve the joint angles. The inverse kinematics is solved and the solution is optimized, considering dynamics-based criterion such as ZMP to find out optimal gait [5].

2.3. Vision systems

In any intelligent system such as a humanoid robot, a vision system is necessary to provide information about the surrounding. A vision system may have either eye-to-hand configuration or eye-in-hand configuration. If a camera is fixed within the workspace, the vision configuration is called eye-to-hand or passive configuration. If a camera is mounted on the robotic system, it is an eye-in-hand or active configuration. In humanoid robots the camera is generally mounted on the robot's head and the vision system falls under the active-vision-system category.

In certain applications, it is desirable to extract three-dimensional information of the real-world from the corresponding two-dimensional image. Calibration is an obvious process to

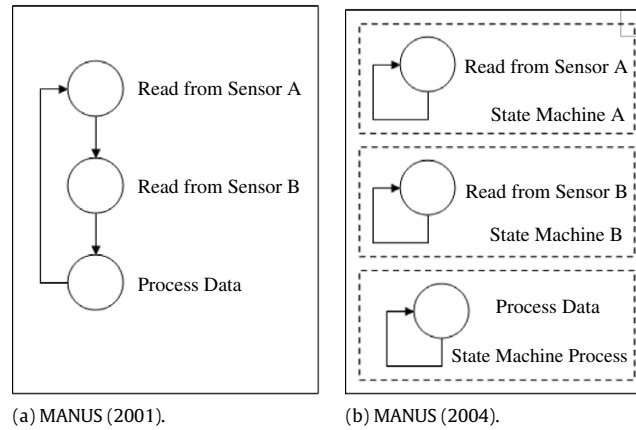


Fig. 2. The tasking architecture of the humanoid robots.

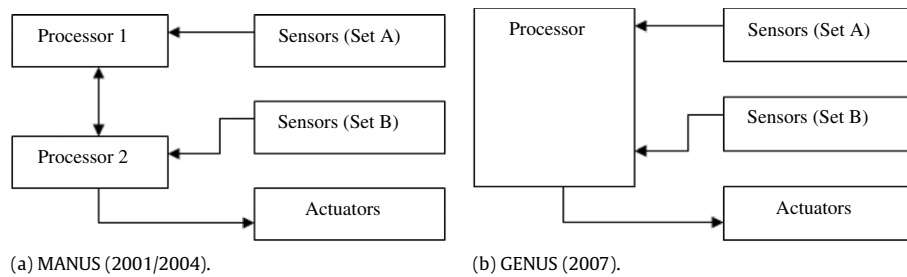


Fig. 3. The processor configuration of the humanoid robots.

address this issue. Calibration of any vision system involves parameter identification of two different natures: (a) Intrinsic parameters: parameters determining internal camera geometric and optical characteristics, (b) Extrinsic parameters: parameters to map the three-dimensional position and orientation of the real-world plane to the image-plane.

Vision-system calibration techniques can roughly be classified into two categories: photogrammetric calibration [21,22] and self-calibration [23,24]. The photogrammetric calibration methods use a known and well-structured object (called calibration reference) to determine the vision system parameters by matching the image-plane features to those on the calibration reference. Self-calibration uses the image of the environment instead of a calibration reference.

3. Processing architecture

The processing architecture in the humanoid robots such as MANUS and GENUS has been going through an evolution over the years. The computational requirement changes with time, bringing the processing architecture into consideration.

3.1. Dual processor configuration (MANUS-2001)

Two processing units (Motorola DSP56F807 MPU, 16 bit 80 MHz processor) are used in MANUS: high level processor and lower body processor. The two processors communicate through a Serial Peripheral Interface (SPI) and are connected to various sensors (Fig. 3).

The high level processor processes information from various sensors. After processing the information from the sensors and deciding on the appropriate action to be taken by the robot, the high level processor sends the respective action commands to be executed by the low level processor via the SPI.

The low level processor controls the actuators for appropriate gait generation. As real-time computation and gait updates are

required during task execution, certain sensors (tilt and force sensors) are connected to the low level processor to reduce the latency time for processing the information required for decision making.

The advantage of the multi-processor configuration in MANUS is the decentralized processing architecture with increased processing capabilities.

3.2. Dual processor configuration and virtual parallel machine architecture (MANUS-2004)

A noticeable feature of the Motorola DSP56F807 MPU processor is the multi-tasking programming paradigm called Virtual Parallel Machine Architecture (VPMA). Independent state machines can be built and executed in parallel in VPMA programming architecture. VPMA is a useful and effective technology for humanoid robots because of its need to integrate many different sensors and process sensor information in real-time (Fig. 2).

Unlike VPMA, the procedural programming architecture does not have parallel processing power. Procedural programs execute sequentially, i.e. each sensor is activated and read one-after-another which means the humanoid can only do single task at a time. Thus, reaction time of the robot is slower while using procedural programs.

Using VPMA, for every sensor on board, one state machine is built to control one sensor. The state machine is responsible for continuous monitoring and reading from that particular sensor. Thus, the main program is able to access the data of each sensor simultaneously during task execution, making the system response faster. For example, if there is a need to get information from the camera, the sequential approach halts the program and reads data from the camera. However, in a VPMA based architecture, the data retrieval is done by the dedicated state machine while the main program continues execution. With this architecture, a task program just needs to accomplish a particular task without consideration of organizing the sensor activation sequence. It acts as a library of real-time information stored in the database and

the main program simply needs to access the library for any information it needs about the environment.

3.3. Single processor configuration and virtual parallel machine architecture optimization (GENUS-2007)

In the dual processors configuration, in spite of higher processing capabilities, the need of communication between the two processors causes information latency. The speed with which information is shared is dependent on the communication protocols. The use of two processors also means a higher resource overhead and maintenance.

A single processor can be used to improve the utilization of the memory and computation power of the processor. The processing power is optimized by proper scheduling of the state machines for the sensors and gait generation by using the timer monitoring function embedded in the processor. By monitoring the timers, the state machine interval is kept consistent and any lapse in the state machine interval is immediately rescheduled to maintain the integrity of the timing.

Optimizing of the memory and computations are performed by using complexity analysis of the programs. Operational redundancy is reduced in the state machine for gait generation which is used for various computations of the trajectories. Memory pointers are dynamically repositioned to free memory banks that are not directly available in various part of the memory space to improve memory utilization.

The major advantages in using a single processor board are the elimination of inter-processor communication and the latency issues. In addition, a centralized control system provides better handling and processing of information among the state machines, leading to faster decision making and execution. Moreover, resource overhead is reduced without compromising the advantages of the multi-tasking feature of the VPMA paradigm.

4. Gait generation

4.1. Trajectory based gait generation utilizing ZMP criterion (MANUS-2001)

The robot's walking gait on a flat surface was first realized using a trajectory based gait generation algorithm. The walking gait is determined by the hip trajectory and the swing foot trajectory. The stability of the walking algorithm is characterized by the ZMP criterion.

The sagittal motion control algorithm includes the following steps:

- For each step, the desired final state, speed and position of hip and swing foot $(v_{xhe}, v_{zhe}), (v_{xfe}, v_{zfe}), (x_{he}, z_{he}), (x_{fe}, z_{fe})$ and step length L_s (Fig. 7) are specified.
- The desired θ_e (Fig. 8) is specified. The hip height adapting to the terrain is derived using the optimal hip height principle (4.1.9).
- The hip and foot trajectories are generated for the two parts of a single support period $(T_1$ and $T_p)$ (4.1.15).
- Change the parameters and repeat step i, ii, iii. The trajectory with satisfied ZMP trajectory is selected.
- The desired state at the beginning of a walking cycle is derived by Lemma 4.1.
- The double support phase drives the robot into the desired initial state.

Single and double support phases. A walking cycle can be divided into a single support phase and a double support phase. In Fig. 4, from point 1 to 2 is the single support phase, and from point 2 to 3 is the double support phase. In the single support phase, the support foot is stationary on the ground. The other foot swings from the back to the front. The hip also moves along a trajectory. The double support phase starts from the forward foot touching the ground and ends with the rear foot leaving the ground. During

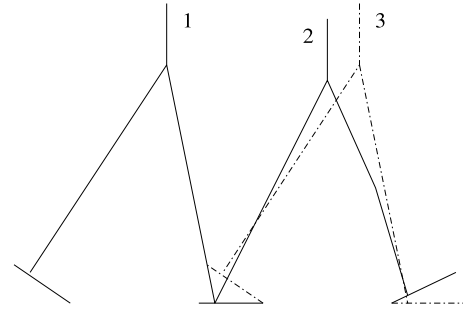


Fig. 4. Walking cycle: single and double support phase.

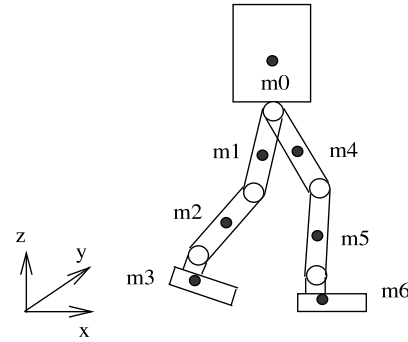


Fig. 5. Biped robot model.

the double support phase, the weight transfers from the rear foot to the forward foot. This phase is known as the weight acceptance phase [25].

To achieve continuous and dynamic walking, the transfer between single support phase and double support phase has to be smooth. At the beginning of the double support phase, the impact when the forward foot contacts the ground is very large and affects the walking stability. Force feedback control is used to resolve and minimize the reaction force to reduce instability during walking.

ZMP and ankle torque. The ZMP is the point on the ground where the sum of all the moments of the active forces equals zero. Under the assumption that no external force exists, the ZMP can be computed by [26]:

$$x_{zmp} = \frac{\sum_i m_i(\ddot{z}_i + g)x_i - \sum_i m_i\ddot{x}_i z_i - \sum_i I_{iy}\ddot{\theta}_{iy}}{\sum_i m_i(\ddot{z}_i + g)} \quad (4.1.1)$$

$$y_{zmp} = \frac{\sum_i m_i(\ddot{z}_i + g)y_i - \sum_i m_i\ddot{y}_i z_i - \sum_i I_{ix}\ddot{\theta}_{ix}}{\sum_i m_i(\ddot{z}_i + g)} \quad (4.1.2)$$

where $(x_{zmp}, y_{zmp}, 0)$ is the coordinate of the ZMP, (x_i, y_i, z_i) is the mass center of link i on a Cartesian coordinate system. m_i is the mass of link i , g is the gravitational acceleration. I_{ix} and I_{iy} are the inertia components, $\ddot{\theta}_{iy}$ and $\ddot{\theta}_{ix}$ are the angular speed around axis y and x about the center of mass of link i (Fig. 5).

Shown in Fig. 6, for the sagittal plane, the ZMP is derived directly through dividing the ankle torque by reaction force:

$$x_{zmp} = \frac{\tau_x}{\sum_i m_i(\ddot{z}_i + g)}. \quad (4.1.3)$$

Therefore, the ankle torque can be written as:

$$\tau_x = x_{zmp} \cdot \sum_i m_i(\ddot{z}_i + g). \quad (4.1.4)$$

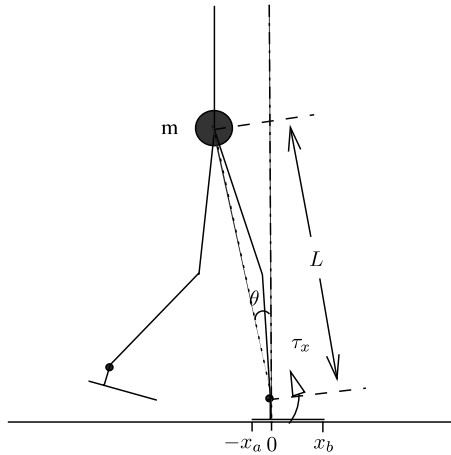


Fig. 6. Invert pendulum model to derive the ankle torque.

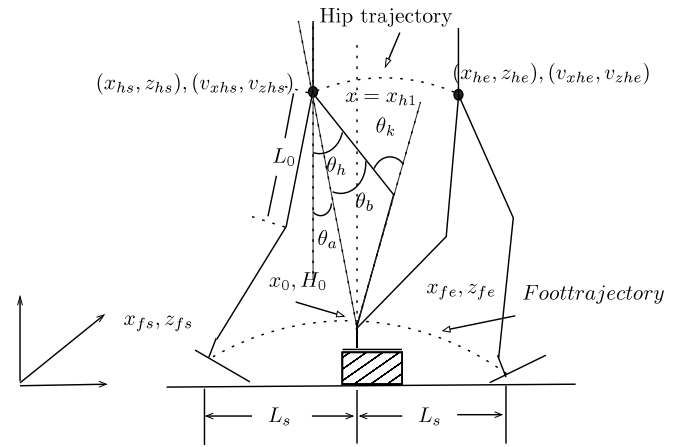


Fig. 7. Hip and foot trajectory.

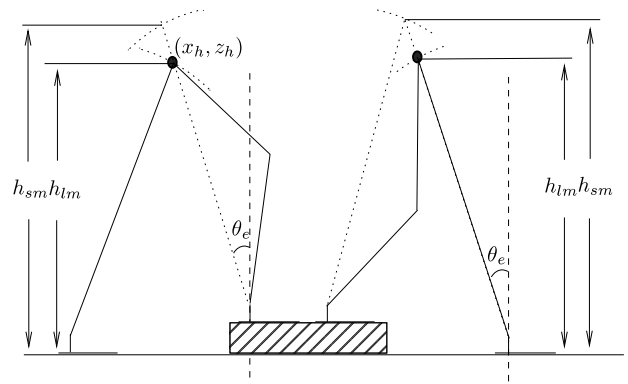


Fig. 8. Determine the height of hip.

For stable dynamic walking the range of the ZMP should be in the convex of the contact points between the foot and the ground. The range of x_{zmp} is defined as:

$$x_{zmp} \in \{S | S \in R, S \in (-x_a, x_b)\}. \quad (4.1.5)$$

Assuming the robot has a point mass at the hip and the supporting knee joint angle is kept constant, an inverted pendulum model is obtained. The length of leg is L , $\tau_{xa}(\theta)$ and $\tau_{xb}(\theta)$ are respectively the maximum and minimum ankle torques in the sagittal plane. Then the maximum ankle torque is:

$$\tau_{xa}(\theta) = m \left(g + \left(\frac{\tau_{xa}(\theta)}{Lm} - g \sin \theta \right) \sin \theta - \frac{v^2}{L} \cdot \cos \theta \right) x_a \quad (4.1.6)$$

where v is the velocity of the hip and $\frac{v^2}{L}$ is the centripetal acceleration. $\left(\frac{\tau_{xa}(\theta)}{Lm} - g \sin \theta \right) \sin \theta - \frac{v^2}{L} \cdot \cos \theta$ is the \ddot{z} in (4.1.4). The centripetal acceleration is much smaller than the other components so that it can be neglected in (4.1.6), and the ankle torque can be described as:

$$\tau_{xa}(\theta) = \frac{mgx_a(1 - \sin^2 \theta)}{1 - \frac{\sin \theta}{L} x_a}. \quad (4.1.7)$$

Similarly,

$$\tau_{xb}(\theta) = \frac{mgx_b(1 - \sin^2 \theta)}{1 + \frac{\sin \theta}{L} x_b}. \quad (4.1.8)$$

Hip trajectory for single support phase. The hip trajectory can be generated by a cubic polynomial if the initial and final states are known for single support phase Fig. 7. The initial state is known and the final state includes $[x_h, z_h]^T$ and $[v_{hx}, v_{hz}]^T$. The desired hip velocity $[v_{hx}, v_{hz}]^T$ is specified.

To adapt to various terrains, the supporting knee is bent accordingly to achieve the desired hip height z_h . For an instance, if the robot is walking up a stair, the legs should try to lift the hip. On the contrary, when it is descending a stair, it should try to lower the hip to enable the foot to reach the lower stair. If the step length L_s in Fig. 7 and angle θ_e in Fig. 8 are specified, the following rule is used to determine the hip height:

$$z_h = \min(\max(h_s), \max(h_l)), \quad (4.1.9)$$

where h_s and h_{sm} are respectively the possible and maximum hip heights for the support leg, h_l and h_{lm} are respectively the possible and maximum hip heights for the land leg at the beginning of double support phase (Fig. 8). This criterion provides a reasonable hip height no matter whether the next leg landing point is higher or lower than the current landing point. The hip position along x direction (x_h) can be derived as (4.1.15).

With the initial state known and the desired final state achieved, the hip trajectory for the single support phase can be generated by a cubic polynomial function. The initial and final constraints of the cubic trajectory for z direction $z_h(t)$ can be described as:

$$z_h(t) = \begin{cases} z_{hs} & \text{if } t = kT \\ z_{he} & \text{if } t = kT + T_s, \end{cases} \quad (4.1.10)$$

where T is the period for one step and T_s is the period for single support phase.

$$\dot{z}_h(t) = \begin{cases} v_{zhs} & \text{if } t = kT \\ v_{zhe} & \text{if } t = kT + T_s. \end{cases} \quad (4.1.11)$$

The cubic polynomial,

$$z_h(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3, \quad (4.1.12)$$

with four parameters a_0, a_1, a_2 and a_3 satisfies the initial and final constraints. Substituting (4.1.10) and (4.1.11) into (4.1.12), the four parameters can be derived and the cubic polynomial can be written as:

$$z_h(t) = z_{hs} + v_{zhs}(t - kT) + \frac{3(z_{he} - z_{hs}) - 2v_{zhs}T_s - v_{zhe}T_s}{T_s^2} (t - kT)^2 + \frac{2(z_{hs} - z_{he}) + (v_{zhs} + v_{zhe})T_s}{T_s^3} (t - kT)^3 \quad (4.1.13)$$

$$kT < t \leq kT + T_s$$

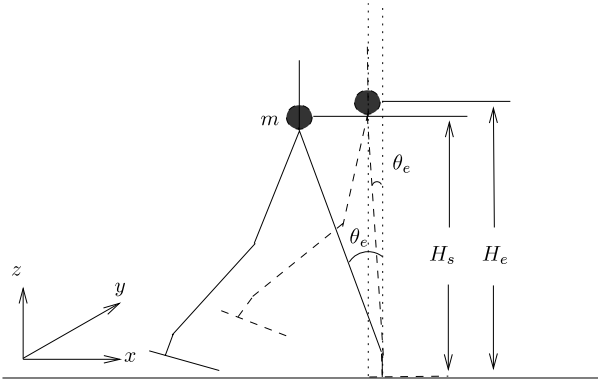


Fig. 9. From unstable to stable region.

$x_h(t)$ is divided into two parts: $x_h(kT)$ to $x_h(kT + T_1)$ and $x_h(kT + T_1)$ to $x_h(kT + T_p)$. The constraints for $x_h(t)$ are (Fig. 7):

$$\begin{cases} x_h(t) = x_{hs} & t = kT \\ x_h(t) = x_{h1} & t = kT + T_1 \\ x_h(t) = x_{he} & t = kT + T_p \\ \dot{x}_h(t) = v_{xhs} & t = kT \\ \dot{x}_h(t^-) = \dot{x}_h(t^+) & t = kT + T_1 \\ \dot{x}_h(t) = v_{xhe} & t = kT + T_p \\ \ddot{x}_h(t) = a_0 & t = kT, \end{cases} \quad (4.1.14)$$

where a_0 is to be specified to satisfy the initial acceleration. The trajectory is composed of two cubic polynomials and the trajectory satisfies constraints (4.1.14).

The cubic polynomial trajectory can be derived as:

$$x_h(t) = \begin{cases} x_{hs} + v_{xhs}(t - kT) + \frac{1}{2}a_0(t - kT)^2 \\ + \frac{(x_{h1} - x_{hs} - v_{xhs}T_1 - \frac{1}{2}a_0T_1^2)(t - kT)^3}{T_1^3} & kT < t \leq kT + T_1 \\ x_{h1} + v_{xh1}(t - kT - T_1) \\ + \frac{(3(x_{he} - x_{h1}) - 2v_{xh1}(T_2 - T_1))(t - kT - T_1)^2}{(T_2 - T_1)^2} \\ + \frac{(2(x_{h1} - x_{he}) + (v_{xh1} + v_{x2})(T_2 - T_1))(t - kT - T_1)^3}{(T_2 - T_1)^3} & kT + T_1 < t \leq kT + T_p \end{cases} \quad (4.1.15)$$

where v_{xh1} can be derived from the first part of (4.1.15).

Swing foot trajectory for single support phase. Cubic polynomial is used to generate the swing foot trajectory of a single support phase. At the starting point and ending point, the following position and speed constraints must be satisfied:

$$\begin{cases} x_f(t) = x_{fs} & t = kT \\ x_f(t) = x_{fe} & t = kT + T_s \\ z_f(t) = z_{fs} & t = kT \\ z_f(t) = z_{fe} & t = kT + T_s \end{cases} \quad (4.1.16)$$

$$\begin{cases} \dot{x}_f(t) = 0 & t = kT \\ \dot{x}_f(t) = 0 & t = kT + T_s \\ \dot{z}_f(t) = 0 & t = kT \\ \dot{z}_f(t) = 0 & t = kT + T_s. \end{cases} \quad (4.1.17)$$

Often, it is desirable for the biped to climb stairs or step over obstacles. Assume that the obstacle's height is H_o and the position is at x_o . In order to avoid colliding with an obstacle the height of the swing foot should be larger than H_o at $x = x_o$. The constraints than can be described as:

$$\begin{cases} z(t) = H_o & t = kT + T_o \\ \dot{z}(t) = 0 & t = kT + T_o. \end{cases} \quad (4.1.18)$$

Giving the start and end positions on x and z direction, and the height of the obstacle H_o , a smooth foot trajectory $f(t) = [x_f(t), z_f(t)]^T$ can be generated as:

$$x_f(t) = x_{fs} + 3(x_{fe} - x_{fs}) \cdot \frac{(t - kT)^2}{T_s^2} - 2(x_{fe} - x_{fs}) \cdot \frac{(t - kT)^3}{T_s^3} \quad kT \leq t \leq kT + T_s, \quad (4.1.19)$$

$$z_f(t) = \begin{cases} z_{fs} + 3(z_{fm} - z_{fs}) \cdot \frac{(t - kT)^2}{T_m^2} - 2(z_{fm} - z_{fs}) \cdot \frac{(t - kT)^3}{T_m^3} & kT \leq t \leq kT + T_m \\ z_{fm} + 3(z_{fe} - z_{fm}) \cdot \frac{(t - kT - T_m)^2}{(T_s - T_m)^2} - 2(z_{fe} - z_{fm}) \cdot \frac{(t - kT - T_m)^3}{(T_s - T_m)^3} & kT + T_m < t \leq kT + T_s. \end{cases} \quad (4.1.20)$$

The joint position of the hip and knee of the swing leg are derived by inverse kinematics. Suppose the hip and swing foot positions in the sagittal plane at time t are respectively, $h(t) = [x_h(t), z_h(t)]^T$ and $f(t) = [x_f(t), z_f(t)]^T$, then the inverse kinematics of the swing leg can be derived from Fig. 7 as follows:

$$q = \begin{bmatrix} \theta_a \\ \theta_b \end{bmatrix} = \begin{bmatrix} \sin^{-1} \frac{(x_f(t) - x_h(t))}{\sqrt{(x_f(t) - x_h(t))^2 + (z_h(t) - z_f(t))^2}} \\ \cos^{-1} \frac{(x_f(t) - x_h(t))}{2L_o} \end{bmatrix}. \quad (4.1.21)$$

As shown in Fig. 7, $\theta_h = \theta_a + \theta_b$ and $\theta_k = 2\theta_b$. θ_h and θ_k are the hip and knee joint angles respectively, and L_o is the length of the thigh and calf.

Double support phase control. The robot falls backward if the initial states $[x_{hs}, z_{hs}]^T$, $[v_{hxs}, v_{hzs}]^T$ are stable. The unstable period T_u is defined from the beginning of single support phase to the Center Of Gravity (COG) entering the support region (Fig. 9). Using the viewpoint of energy conservation, the following criterion can be derived:

Lemma 4.1. *When the rear foot leaves the ground, the kinetic energy should be large enough to drive the COG entering the range of the supporting region on the sagittal plane with limited control, otherwise the robot tips over. This can be represented by:*

$$\frac{1}{2}m(\dot{x}^2 + \dot{z}^2) + \int_{\theta_s}^{\theta_e} \tau_{\max}(\theta)d\theta \geq mg(H_e - H_s) \quad (4.1.22)$$

where τ_{\max} is the maximum ankle torque, I is the inertial component and $\dot{\theta}$ is the angular velocity of the support ankle joint. H_e and H_s are the initial and final hip heights for the unstable period, as shown in Fig. 9. It is assumed that the support knee joint angle is kept constant when the τ_{\max} is derived. Consequently this is only an approximation expression when the support knee joint angle changes.

The initial state of single support phase is the final state of the double support phase. In the double support phase, the ankle of the forward leg adapts to the terrain profile and the ankle of the rear leg is used to realize the state satisfying (4.1.22) by pushing the body forward and upward.

Implementing the trajectory based method with ZMP criterion on the humanoid robot, MANUS (2001) achieved a forward walking speed of 0.0480 m/s.

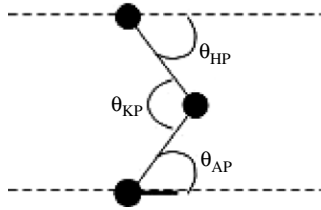


Fig. 10. Joint angles used for simple oscillator: θ_{HP} , θ_{KP} , θ_{AP} are the hip, knee and ankle pitch angles respectively.

4.2. Gait generation using simple oscillators (MANUS-2004)

The trajectory based method with ZMP criterion had shown viable results. However, the drawback of the method is the low walking speed of the robot.

A gait using simple oscillators, inspired by the CPG methods, has been realized. The simple oscillator method generates considerable speed during the walking of the robot. The method uses the hip, the knee and the ankle pitch joints (Fig. 10). The joints are actuated in an oscillatory fashion such that the DOFs at the identical locations of the leg move in opposite direction at any point of time. This creates a cycling motion where the humanoid appears to be hanging above ground. The gait generation method has three phases.

In the first phase, the hip, knee and ankle pitch motors are rotated with a ratio of 1:2:1. For instance, for a 1 degree rotation of the hip pitch motor, the knee pitch motor and ankle pitch motor rotate 2 degrees and 1 degree respectively. It results in the humanoid's leg bending and straightening up continuously.

At the initial move off stage, the first bending motion is made by only one leg. Subsequent motion involves the continuous bending of one leg and straightening of the other leg. The magnitude of angle of rotation is slowly increased until a stable oscillation of the humanoid body in the frontal plane is achieved. The condition for ending this phase is when the oscillation is large enough, such that each foot takes turn to act as the support foot while no weight is placed on the other foot. As a result, the robot continuously alternates between the single support phase and double support phase at the oscillating frequency of the gait generated (Fig. 11).

The second phase involves a point at which the humanoid starts moving. With the actuators rotating at the angles determined in the first phase, the humanoid is essentially marching on the spot when the second phase comes into play. When the second phase starts, the magnitude of angle of rotation for the hip pitch motor is increased. As such, the joints no longer rotate in 1:2:1 ratio. The angle of rotation for the hip pitch motor is increased until the point where the humanoid starts to move forward. At this point of time, the humanoid is moving forward with very small steps. The exact ratio of the angles of rotation for the hip, knee and ankle pitch motors are then calculated and recorded.

In the final phase of the dynamic gait generation, the angles of rotation for the motors are now increased in the ratio obtained in phase two. By increasing these angles of rotation, the step length of the humanoid is increased and the humanoid's movement speed

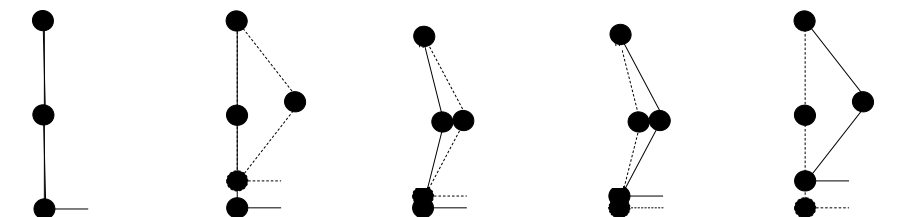


Fig. 11. Sequence of motion for first phase.

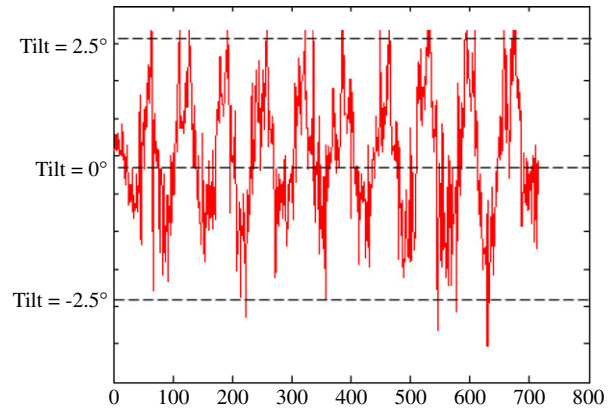


Fig. 12. Stable oscillations in the sagittal plane during walking. X-axis indicates time in seconds and Y-axis indicates the hip tilt in degrees.

is increased. As the humanoid is moving forward, the body will oscillate like an inverted pendulum in the sagittal plane. This oscillation increases as the step length is increased. The increment in the angles of rotation is stopped when the body begins to oscillate at angles more than 5 degrees in the sagittal plane. As the oscillation of the body in the sagittal plane increases, the stability of the humanoid decreases and the tendency to fall over is very high. With this final phase done, the dynamic gait generation for the humanoid is complete and the humanoid would achieve dynamic walking by rotating the actuators according to the final magnitudes of angles.

The dynamic gait so generated is successful and resulted in faster and stable biped locomotion. The readings of the angles of oscillation are obtained from the tilt sensor on the humanoid (Fig. 12). The readings of the force on each foot are obtained using the force sensors attached to the shoe of the humanoid. The period of oscillation is constant, which corresponds exactly to the frequency of force exerted on each foot.

Implementing the simple oscillator on the humanoid robot, MANUS (2001), the speed of walking during forward and backward dash achieved are 0.0706 m/s and 0.0545 m/s respectively.

4.3. Trajectory based gait generation using simple oscillator (GENUS-2007)

The method of generating dynamic gait motion is simple and effective. However, a drawback of the method is that no control efforts are used to consider the limitation of the actuators, mounted on the robot. As such, the simple oscillator method requires manual tuning and does not guarantee performance reliability.

Actuators employed on the humanoid robot do not provide feedback, and in cases where the actuator performance can be guaranteed, feedback is essentially not required. However, the simple oscillator method cannot provide performance consistency due to its linear trajectory nature and absence of feedback.

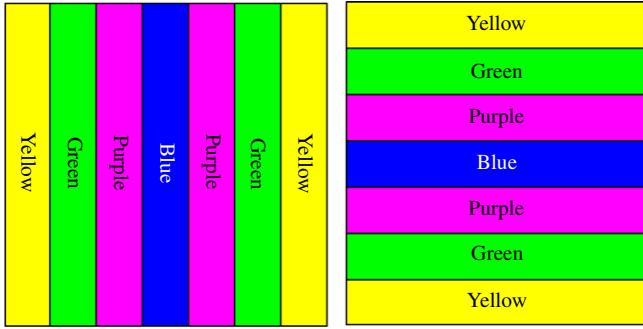


Fig. 13. Zones defined for the pan and tilt respectively.

Hence, a new method is proposed which combines the trajectory based method and the simple oscillator method.

Each actuator has a closed loop system for control. The assumption made on the actuators is that the performance is guaranteed without the need of feedback, as the actuator will always rotate to the set position. However, this assumption is not necessarily held true if:

- The actuators are operated outside the technical specifications and limits.
- The actuators are mechanically constrained or overloaded.
- The actuators are overstrained.

With the simple oscillator method, the angles of rotation of the actuators are increased in a linear fashion, resulting in linear trajectories. The method of linear trajectory planning is not recommended as it would mean that infinite acceleration and deceleration are required at the start and end of each motion or rotation. Linear trajectories lead to poor reliability performance.

In trajectory based gait generation using the ZMP criterion, it was shown that the cubic polynomial trajectory is effective in providing smooth motion planning for the joints. In addition, using the cubic polynomial trajectory, the speed and acceleration profile of the actuators can be effectively controlled. By doing so, the simple oscillation method is implemented without exceeding the limitation of the actuators. The gait generation method provided reliable performance, optimum gait and faster walking speed.

Implementing the trajectory-based simple oscillator, the humanoid robot GENUS (2007) achieved a forward speed of 0.0923 m/s and backward speed of 0.1000 m/s.

5. Vision

5.1. Pan-tilt vision using predictive algorithm (MANUS)

A necessary and effective functionality of the soccer playing humanoid robots is to locate the ball autonomously and fast for live tracking. Using the pan-tilt camera configuration, a simple but effective predictive algorithm is used to track the movements of the ball in real-time. The algorithm makes use of the information sent from the camera to the higher level processor. The humanoid starts scanning for the ball and employs the tracking algorithm once the ball is located in the image plane. The tracking algorithm has two stages. The first stage is active when the ball is within the image plane. During this stage, the higher level processor continuously alters the pan and tilt directions of the head to keep the ball at the center of the image plane. For example, if the coordinates of the ball fall at the top left hand corner of the image plane, the higher level processor instructs the head to pan left and tilt up. The ball is always kept within sight as long as the ball is not moving too quickly. The algorithm is simple yet effective.

The image plane of the camera is divided into certain zones (Fig. 13). When the ball is in the yellow zone of the image plane,

the pan or tilt or both, moves by 10 degrees. When the ball falls under the green zone, the corresponding rotation is 5 degrees, and similarly, 1 degree for the pink zone. The blue zone is the target area and the camera does not move when the ball is in the blue zone.

The second stage of the algorithm is employed when the camera fails to locate the ball. The higher level processor makes a prediction of the ball movement using information of the last image and head alignment. It moves the head in the direction where it predicts the ball's location. If the ball still cannot be located, the higher level processor commands the head to start scanning for the ball once more.

5.2. Omni-vision using genetic algorithm based calibration (GENUS)

The pan-tilt camera provides vision data to the robot. However, camera coverage is relatively small and the time required to scan and locate the ball, whenever the ball is lost, is considerably high. Omni vision uses a convex hyperbolic shaped mirror that reflects an image from the surrounding area. The main advantage of the omni-vision is that it provides wider vision coverage. However, this advantage comes at the cost of the image resolution or image deformation.

A concept of semi-omni vision is used by the GENUS (2007) humanoid. An semi-omni vision provides 180 degrees of view instead of the conventional omni-vision with 360 degrees of view. By compromising on the degrees of view, by half, the resolution doubles, as the camera only needs to see half the mirror. In omni-vision/semi omni-vision systems, the acquired image (Fig. 14) is distorted due to the convex nature of the mirror. As the real-world-plane is distorted in the image-plane, there is a need for calibration to map the image-plane back to the real-world-plane.

A square grid of 10 × 10 cm in real-plane placed on the flat ground is mapped based on the coordinates obtained from the image-plane. The y-axis indicates the sagittal-plane distance and the x-axis indicates the frontal-plane distance in the real world (Fig. 15).

Using a parabolic curve, the image-plane distance along the y-axis is mapped into the real-world-frontal-plane distance as per (5.2.2). For the ease of implementation and the consideration on the limitation of the processors, the parameters a, b and c are determined using the Least Square Error (LSE) method. This is done by minimizing the cost function (5.2.2).

$$y_r^i = a + by_{im}^i + c(y_{im}^i)^2 \tag{5.2.1}$$

$$\prod = \sum_{i=0}^n [y_r^i - a + by_{im}^i + c(y_{im}^i)^2]^2 \tag{5.2.2}$$

where y_{im}^i and y_r^i are the distances along the y-axis in image-plane and the corresponding real-world-sagittal-plane respectively, n is the total number of experimental data. However, using the LSE method for mapping, there are considerable errors between the real-world plane readings and the mapped image-plane points. Therefore, the LSE equation is modified as per (5.2.3) by adding an empirical function of frontal-plane distance.

$$\bar{y}_r^i = \hat{y}_r^i + \alpha(\hat{x}_r^i - \beta)^\gamma + \delta \tag{5.2.3}$$

where the estimated values \hat{y}_r^i and \hat{x}_r^i are calculated from the image reading as described earlier 15. \bar{y}_r^i is the modified estimation of the real-world-sagittal-plane distance. α , β , γ , δ are the parameters that are computed using the Genetic Algorithm (GA). GA determines the best mapping values of these unknown coefficients in (5.2.3).

The chromosome of real numbers, corresponding to different coefficients, is employed, and the fitness of the GA is computed

Fig. 14. Image acquired from implemented omni-vision.**Table 1**

GA parameters.

Chromosome length	4: [α , β , γ , δ]
Population size	100
Number of generations	100
Crossover rate	0.9
Mutation rate	0.03
α	–5 to 5
β	–2 to 2
γ	–10 to 10
δ	–1 to 1

Table 2

Optimal parameters.

α	β	γ	δ
3.8767	1.2459	8.4811	–0.9336

Fig. 15. Distortion of image produced by omni-vision.

by the inverse of the maximum error between the mapped image-plane reading and real-world-plane points.

Details of the GA parameters are tabulated in Tables 1 and 2.

Fitness Function:

$$F = \frac{1}{\max \prod_{i=0}^n \sqrt{(y_r^i - \bar{y}_r^i)^2 + (x_r^i - \bar{x}_r^i)^2}}. \quad (5.2.4)$$

An initial trial run with population sizes of 10 and 50 generations with a crossover rate of 0.8 and a mutation rate of 0.03 is initially set and adjusted subsequently.

The best fitness, achieved after varying the values of the various parameters, is approximately 0.5001. With this fitness, the maximum deviation of the actual reading from the mapped one

Fig. 16. Result of GA based calibration for the omni-camera: X-axis and Y-axis are the real-world frontal and sagittal plane distances in CM respectively. '+' indicates the actual locations and '•' indicates estimated locations.

is around 1.9996 cm with an average of 1.21 cm. Considering the ball-size, the playing field and the coverage of the camera, the error margin is considered small and the calibration results are satisfactory for soccer playing. Results of the calibration is shown in Fig. 16 where the y-axis indicates the sagittal-plane distance and the x-axis indicates the frontal-plane distance in the real-world-plane. The plus sign indicates the real-plane data and the dots are the computed real-plane data from the image coordinates.

6. Conclusion

Over the years, the NUS humanoid robots, MANUS and GENUS, have actively participated in the FIRA Roboworld Cup competition Hurosot category. In the competitions, the humanoids compete against each other in a number of events such as running, penalty kick and obstacle avoidance. In 2003, the MANUS attained its first championship title at the 8th FIRA Hurosot competition. In 2004, MANUS came in as the 1st runner-up and for the following two years, it regained and retained the overall championship title of the Hurosot competition. In 2007, GENUS managed to clinch the 1st runner up title in the robot dash category.

In this paper, the research and development of the humanoid robots in the Mechatronics and Automation Laboratory over the recent years is discussed. The evolution of the architecture from sequential to parallel processing has provided the gateway for

