17

Walking Gait Planning And Stability Control

Chenbo Yin, Jie Zhu and Haihan Xu
Nanjing University of Technology, School of Mechanical and Power Engineering

1. Introduction

Research on biped humanoid robots is currently one of the most exciting topics in the field of robotics and there are many ongoing projects. Because the walking of humanoid robot is complex dynamics inverse problem the pattern generation and dynamic simulation are extensive discussed. Many different models are proposed to simple the calculation. Many researches about the walking stability and pattern generation of biped robots are made using ZMP principle and other different methods.

Vukobratovic first proposed the concept of the ZMP (Zero Moment Point). Yoneda etc proposed another criterion of "Tumble Stability Criterion" for integrated locomotion and manipulation systems. Goswami proposed the FRI (Foot Rotation Indicator). As for the pushing manipulation, Harada researched the mechanics of the pushed object. Some researches mentioned that changes of angular momentum of biped robot play the key roles on the stability maintenance. However, there have been fewer researches on stability maintenance considering the reaction with external environment.

A loss of stability might result a potentially disastrous consequence for robot. Hence man has to track robot stability at every instant special under the external disturbance. For this purpose we need to evaluate quantity the danger extent of instability. Rotational equilibrium of the foot is therefore an important criterion for the evaluation and control of gait and postural stability in biped robots. In this paper by introducing a concept of fictitious zero-moment (FZMP), a method to maintain the whole body stability of robot under disturbance is presented.

2. Kinematics and dynamics of humanoid robot

Robot kinematics deals with several kinematic and kinetic considerations which are important in the control of robotic kinematics. In kinematic modeling of robots, we are interested in expressing end effector motions in terms of joint motions. This is the direct problem in robot kinematics. The inverse-kinematics problem is concerned with expressing joint motions in terms of end-effector motions. This latter problem is in general more complex. In robot dynamics (kinetics), the direct problem is the formulation of a model as a set of differential equations for robot response, with joint forces/torques as inputs. Such models are useful in simulations and dynamic evaluations of robots. The inverse-dynamics problem is concerned with the computation of joint forces/torques using a suitable robot model, with the knowledge of joint motions. The inverse problem in robot dynamics is

directly applicable to computed-torque control (also known as feed forward control), and also somewhat indirectly to the nonlinear feedback control method employed here.

2.1 Representation of position and orientation

2.1.1 Description of a position

Once a coordinate system is established we can locate any point in the universe with a 3×1 position vector. Because we will often define many coordinate systems in addition to the universe coordinate system, vectors must be tagged with information identifying which coordinate system they are defined within. In this book vectors are written with a leading superscript indicating the coordinate system to which they are referenced (unless it is clear from context), for example, ${}^{A}P$. This means that the components of ${}^{A}P$ have numerical values which indicated distances along the axes of {A}. Each of these distances along an axis can be thought of as the result of projecting the vector onto the corresponding axis.

Figure 2.1 pictorially represents a coordinate system, {A}, with three mutually orthogonal unit vectors with solid heads. A point ${}^{A}P$ is represented with a vector and can equivalently be thought of as a position in space, or simply as an ordered set of three numbers. Individual elements of a vector are given subscripts $x, y,$ and z :

$$ {}^{A}P = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \tag{1} $$

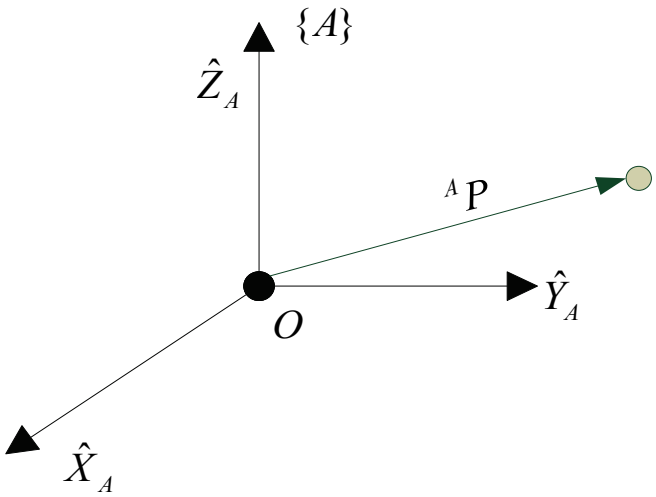

Fig. 1. Vector relative to frame example

In summary, we will describe the position of a point in space with a position vector. Other 3-tuple descriptions of the position of points, such as spherical or cylindrical coordinate representations are discussed in the exercises at the end of the chapter.

2.1.2 Description of an orientation

Often we will find it necessary not only to represent a point in space but also to describe the orientation of a body in space. For example, if vector ^{A}P in fig.2.2 locates the point directly between the fingertips of a manipulator's hand, the complete location of the hand is still not specified until its orientation is also given. Assuming that the manipulator has a sufficient number of joints the hand could be oriented arbitrarily while keeping the fingertips at the same position in space. In order to describe the orientation of a body we will attach a coordinate system to the body and then give a description of this coordinate system relative to the reference system. In Fig.2.2, coordinate system {B} has been attached to the body in a known way. A description of {B} relative to {A} now suffices to give the orientation of the body.

Thus, positions of points are described with vectors and orientations of bodies are described with an attached coordinate system. One way to describe the body-attached coordinate system, {B}, is to write the un it vectors of its three principal axes in terms of the coordinate system {A}.

We denote the unit vectors giving the principal directions of coordinate system {B} as $^{A}\hat{X}_{B}, {}^{B}\hat{Y}_{B}, and\ {}^{A}\hat{Z}_{B}$. When written in terms of coordinate system {A} they are called $\hat{X}_{B}, \hat{Y}_{B}, and\ \hat{Z}_{B}$. It will be convenient if we stack these three unit vectors together as the columns of a 3×3 matrix, in the order $^{A}\hat{X}_{B}, {}^{B}\hat{Y}_{B}, {}^{A}\hat{Z}_{B}$. We will call this matrix a rotation matrix, and because this particular rotation matrix describes {B} relative to {A}, we name it with the notation $^{A}_{B}R$. The choice of leading sub-and superscripts in the definition of rotation matrices will become clear in following sections.

$$ {}^{A}_{B}R = \begin{bmatrix} {}^{A}\hat{X}_{B} & {}^{A}\hat{Y}_{B} & {}^{A}\hat{Z}_{B} \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{2} $$

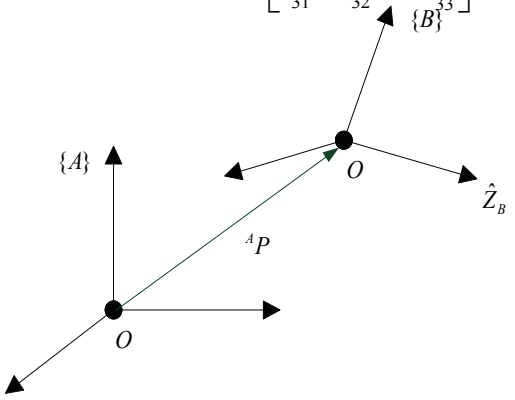

Fig. 2. locating an object in position and orientation

In summary, a set of three vectors may be used to specify an orientation. For convenience we will construct a 3×3 matrix which has these three vectors as its columns. Hence, whereas the position of a point is represented with a vector, the orientation of a body is represented with a matrix. In section 2.8 we will consider some other descriptions of orientation which require only three parameters.

We can give expressions for the scalars r_{ij} in (2.2) by nothing that the components of any vector are simply the projections of that vector onto the unit directions of its reference frame. Hence, each component of ${}^{A}_{B}R$ in (2.2) can be written as the dot product of a pair of unit vectors as

$$ {}^{A}_{B}R = \begin{bmatrix} {}^{A}\hat{X}_B & {}^{A}\hat{Y}_B & {}^{A}\hat{Z}_B \end{bmatrix} = \begin{bmatrix} \hat{X}_B \cdot \hat{X}_A & \hat{Y}_B \cdot \hat{X}_A & \hat{Z}_B \cdot \hat{X}_A \\ \hat{X}_B \cdot \hat{Y}_A & \hat{Y}_B \cdot \hat{Y}_A & \hat{Z}_B \cdot \hat{Y}_A \\ \hat{X}_B \cdot \hat{Z}_A & \hat{Y}_B \cdot \hat{Z}_A & \hat{Z}_B \cdot \hat{Z}_A \end{bmatrix} \tag{3} $$

For brevity we have omitted the leading superscripts in the rightmost matrix of (2.3). In fact the choice of frame in which to describe the unit vectors is arbitrary as long as it is the same for each pair being dotted. Since the dot product of two unit vectors yields the cosine of the angle between them, it is clear why the components of rotation matrices are often referred to as direction cosines.

Further inspection of (2.3) shows that the rows of the matrix are the unit vectors of {A} expressed in {B}; that is,

$$ {}^{A}_{B}R = \begin{bmatrix} {}^{A}\hat{X}_B & {}^{A}\hat{Y}_B & {}^{A}\hat{Z}_B \end{bmatrix} = \begin{bmatrix} {}^{A}\hat{X}^T_B \\ {}^{A}\hat{Y}^T_B \\ {}^{A}\hat{Z}^T_B \end{bmatrix} \tag{4} $$

Hence, ${}^{B}_{A}R$, the description of frame {A} relative to {B} is given by the transpose of (2.3); that is,

$$ {}^{B}_{A}R = {}^{B}_{A}R^T \tag{5} $$

This suggests that the inverse of a rotation matrix is equal to its transpose, a fact which can be easily verified as

$$ {}^{A}_{B}R^T \; {}^{A}_{B}R = \begin{bmatrix} {}^{A}\hat{X}^T_B \\ {}^{A}\hat{Y}^T_B \\ {}^{A}\hat{Z}^T_B \end{bmatrix} \begin{bmatrix} {}^{A}\hat{X}_B & {}^{A}\hat{X}_B & {}^{A}\hat{X}_B \end{bmatrix} = I_3 \tag{6} $$

Where I_3 is the 3×3 identity matrix. Hence,

$$ {}^{B}_{A}R = {}^{B}_{A}R^{-1} = {}^{B}_{A}R^T \tag{7} $$

Indeed from linear algebra we know that the inverse of a matrix with orthonormal columns is equal to its transpose. We have just shown this geometrically.

2.1.3 Description of a frame

The information needed to completely specify the whereabouts of the manipulator hand in Fig.2.2 is a position and an orientation. The point on the body whose position we describe could be chosen arbitrarily, however: For convenience, the point whose position we will

describe is chosen as the origin of the body-attached frame. The situation of a position and an orientation pair arises so often in robotics that we define an entity called a frame, which is a set of four vectors giving position and orientation information. For example, in Fig.2.2 one vector locates the fingertip position and three more describe its orientation. Equivalently, the description of a frame can be thought of as a position vector and a rotation matrix. Note that a frame is a coordinate system, where in addition to the orientation we give a position vector which locates its origin relative to some other embedding frame. For example, frame {B} is described by ${}^{A}_{B}R$ and ${}^{A}R_{BORG}$, where ${}^{A}R_{BORG}$ is the vector which locates the origin of the frame {B}:

$$\{B\} = \{ {}^{A}_{B}R,\ {}^{A}P_{BORG} \} \tag{8}$$

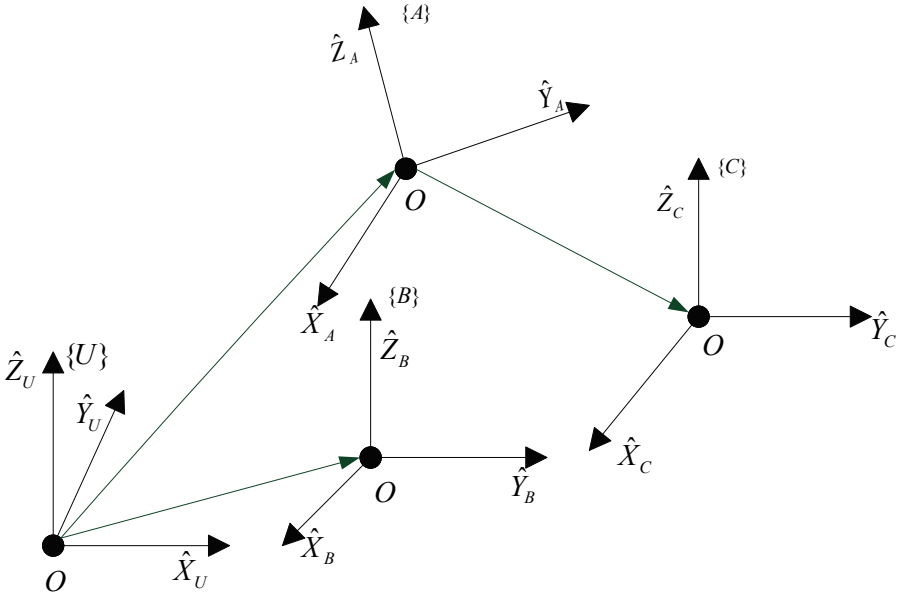

Fig. 3. Example of several frames

In Fig.2.3 there are three frames that are shown along with the universe coordinate system. Frames {A} and {B} are known relative to the universe coordinate system and frame {C} is known relative to frame {A}.

In Fig.2.3 we introduce a graphical representation of frames which is convenient in visualizing frames. A frame is depicted by three arrows representing unit vectors defining the principal axes of the frame. An arrow representing a vector is drawn from one origin to another. This vector represents the position of the origin at the head of the arrow in terms of the frame at the tail of the arrow. The direction of this locating arrow tells us, for example, in Fig.2.3, that {C} is known relative to {A} and not vice versa.

In summary, a frame can be used as description of one coordinate system relative to another. A frame encompasses the ideas of representing both position and orientation, and so may be thought of as a generalization of those two ideas. Position could be represented by a frame whose rotation matrix part is the identity matrix and whose position vector part locates the point being described. Likewise, an orientation could be represented with a frame. Whose position vector part was the zero vector.

2.2 Coordinate transformation

2.2.1 Changing descriptions from frame to frame

In a great many of the problems in robotics, we are concerned with expressing the same quantity in terms of various reference coordinate systems. The previous section having introduced descriptions of positions, orientations, and frames, we now consider the mathematics of mapping in order to change descriptions frame to frame.

Mappings involving translated frames

In Fig.2.4 we have a position defined by the vector ${}^{B}P$. We wish to express this point in space in terms of frame {A}, when {A} has the same orientation as {B}. In this case, {B} differs from {A} only by a translation which is given by ${}^{B}P_{BORG}$, a vector which locates the origin of {B} relative to {A}.

Because both vectors are defined relative to frames of the same orientation, we calculate the description of point P relative to {A}, ${}^{A}P$, by vector addition:

$$ {}^{A}P = {}^{B}P + {}^{A}P_{BORG} \tag{9}$$

Note that only in the special case of equivalent orientations may we add vectors which are defined in terms of different frames.

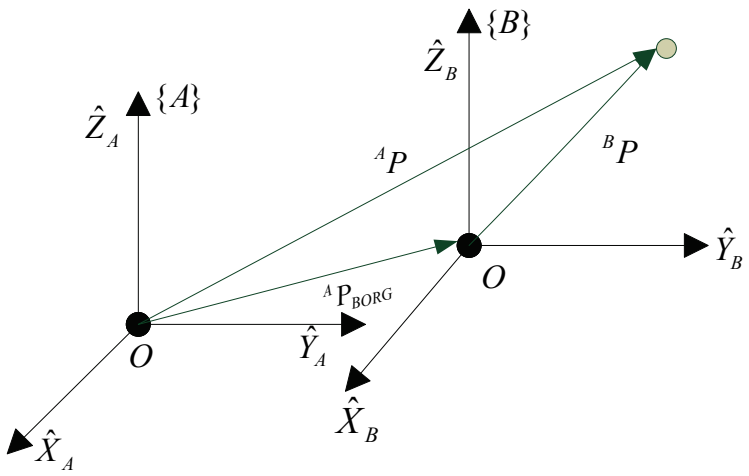

Fig. 4. Translational mapping

In this simple example we have illustrated mapping a vector from one frame to another. This idea of mapping, or changing the description from one frame to another, is an extremely important concept. The quantity itself (here, a point in space) is not changed; only its description in changed. This is illustrated in Fig.2.4, where the point described by ${}^{B}P$ is not translated, but remains the same, and instead we have computed a new description of the same point, but now with respect to system {A}.
We say that the vector ${}^{A}P_{BORG}$ defines this mapping, since all the information needed to perform the change in description is contained in ${}^{A}P_{BORG}$ (along with the knowledge that the frames had equivalent orientation).

Mappings involving rotated frames

Section 2.2 introduced the notion of describing an orientation by three unit vectors denoting the principal axes of a body-attached coordinate system. For convenience we stack these three unit vectors together as the columns of a 3×3 matrix. We will call this matrix a rotation matrix, and if this particular rotation matrix describes {B} relative to {A}, we name it with the notation ${}^{A}_{B}R$.
Note that by our definition, the columns of a rotation matrix all have unit magnitude, and further, these unit vectors are orthogonal. As we saw earlier, a consequence of this is that

$$ {}^{A}_{B}R = {}^{B}_{A}R^{-1} = {}^{B}_{A}R^{T} \tag{10} $$

Therefore, since the columns of ${}^{A}_{B}R$ are the unit vectors of {B} written in {A}, then the rows of ${}^{A}_{B}R$ are the unit vectors of {A} written, in {B}.
So a rotation matrix can be interpreted as a set of three column vectors or as a set of three row vectors as follows:

$$ {}^{A}_{B}R = \begin{bmatrix} {}^{A}\hat{X}_{B} & {}^{A}\hat{Y}_{B} & {}^{A}\hat{Z}_{B} \end{bmatrix} = \begin{bmatrix} {}^{B}\hat{X}^{T}_{A} \\ {}^{B}\hat{Y}^{T}_{A} \\ {}^{B}\hat{Z}^{T}_{A} \end{bmatrix} \tag{11} $$

As in Fig.2.5, the situation will arise often where we know the definition of a vector with respect to some frame, {B}, and we would like to know its definition with respect to another frame, {A}, where the origins of the two frames are coincident. This computation is possible when a description of the orientation of {B}, is known relative to {A}. This orientation is given by the rotation matrix ${}^{A}_{B}R$, whose columns are the unit vectors of {B} written in {A}.
In order to calculate ${}^{A}P$, we note that the components of any vector are simply the projections of that vector onto the unit directions of its frame. The projection is calculated with the vector dot product. Thus we see that the components of ${}^{A}P$ may be calculated as

$$
\begin{aligned}
{}^{A}p_{x} &= {}^{B}\hat{X}_{A} \cdot {}^{B}P, \\
{}^{A}p_{y} &= {}^{B}\hat{Y}_{A} \cdot {}^{B}P, \\
{}^{A}p_{z} &= {}^{B}\hat{Z}_{A} \cdot {}^{B}P.
\end{aligned}
\tag{12}
$$

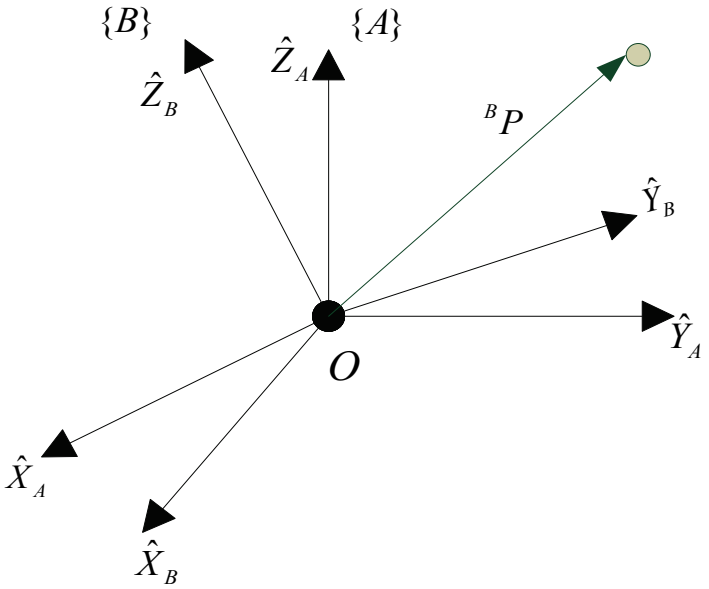

Fig. 5. rotating the description of a vector

In order to express (12) in terms of a rotation matrix multiplication, we note form (11) that the rows of ${}^{A}_{B}R$ are ${}^{B}\hat{X}_{A}$ ${}^{B}\hat{Y}_{A}$ and ${}^{B}\hat{Z}_{A}$. So (12) may be written compactly using a rotation matrix as

$$
{}^{A}P = {}^{A}_{B}R\,{}^{B}P \tag{13}
$$

Equation (13) implements a mapping—that is, it changes the description of a vector—from ${}^{B}P$, which description of the same point, but expressed relative to {A}.
We now see that out notation is of great help in keeping track of mappings and frames of reference. A helpful way of viewing the notation we have introduced is to imagine that leading subscripts cancel the leading superscripts of the following entity, for example the Bs in (13).

2.3 General rotation transformation

Mappings involving general frames

Very often we know the description of a vector with respect to some frame, {B}, and we would like to know its description with respect to another frame, {A}. We now consider the general case of mapping. Here the origin of frame {B} is not coincident with that of frame {A} but has a general vector offset. The vector that locates {B}'s origin is called ${}^{A}P_{BORG}$. Also {B} is rotated with respect to {A} as described by ${}^{A}_{B}R$. Given ${}^{B}P$, we wish to computer ${}^{A}P$, as in Fig.2.7.

We can first change ${}^{B}P$ to its description relative to an intermediate frame which has the same orientation as {A}, but whose origin is coincident with the origin of {B}. This is done by pre-multiplying by ${}^{A}_{B}R$ as in Section 2.3. We then account for the translation between origins by simple vector addition as in Section 2.3, yielding

$$ {}^{A}P = {}^{A}_{B}R\,{}^{B}P + {}^{A}P_{BORG} \tag{14} $$

Equation (2.17) describes a general transformation mapping of a vector from its description in one frame to a description in a second frame. Note the following interpretation of our notation as exemplified in (2.14): the B's cancel leaving all quantities as vectors written in terms of A, which may then be added.
The form of (2.14) is not as appealing as the conceptual form;

$$ {}^{A}P = {}^{A}_{B}T\,{}^{B}P \tag{15} $$

That is, we would like to think of a mapping from one frame to another as an operator in matrix form. This aids in writing compact equations as well as being conceptually clearer than (2.14). In order that we can write the mathematics given in (2.14) in the matrix operator form suggested by (2.15), we define a 4×4 matrix operator, and use 4×1 position vectors, so that (2.15) has the structure

$$ \begin{bmatrix} {}^{A}P \\ 1 \end{bmatrix} = \begin{bmatrix} {}^{A}_{B}R & {}^{A}P_{BORG} \\ 0\quad 0\quad 0 & 1 \end{bmatrix} \begin{bmatrix} {}^{B}P \\ 1 \end{bmatrix} \tag{16} $$

That is,
1□A "1"is added as the last element of the 4×1 vectors.
2□A row "$\begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$"is added as the last row of the 4×4 matrix.
We adopt the convention that a position vector is 3×1 or 4×1 depending on whether it appears multiplied by a 3×3 matrix or by a 4×4 matrix. It is readily seen that (2.16) implements

$$^{A}P = {}^{A}_{B}R\,{}^{B}P + {}^{A}P_{BORG}$$

$$1 = 1 \tag{17}$$

The 4×4 matrix in (2.16) is called a homogeneous transform. For our purposes it can be regarded purely as a construction used to cast the rotation and translation of the general transform into a single matrix form. In other fields of study it can be used to compute perspective and scaling operations (when the last row is other than"$\begin{bmatrix}0 & 0 & 0\end{bmatrix}$", or the rotation matrix is not orthonormal). The interested reader should see.

Often we will write equations like (2.15) without any notation indicating that this is a homogeneous representation, because it is obvious from context. Note that like while homogeneous transforms are useful in writing compact equations, a computer program to transform vectors would generally not use them because of time wasted multiplying ones and zeros. Thus, this representation is mainly for our convenience when thinking and writing equations down on paper.

Just as we used rotation matrices to specify an orientation, we will use transforms (usually in homogeneous representation) to specify a frame. Note that while we have introduced homogeneous transforms in the context of mappings, they also serve as descriptions of frames. The description of frame {B} relative to {A} is ${}^{A}_{B}T$.

2.4 Transformation matrix for links

Link description

A manipulator may be thought of as a set of bodies connected in a chain by joints. These bodies are called links. Joints form a connection between a neighboring pair of links. The term lower pair is used to describe the connection between a pair of bodies when the relative motion is characterized by tow surfaces sliding over one another.

Due to mechanical design considerations, manipulators are generally constructed from joints which exhibit just one degree of freedom. Most manipulators have revolute joints or have sliding joints called prismatic joints. In the rare case that a mechanism is built with a joint having n degrees of freedom, it can be modeled as n joints of one degree of freedom connected with n-1 links of zero length. Therefore, without loss of generality, we will consider only manipulators which have joints with a single degree of freedom.

The links are numbered starting from the immobile base of the arm, which might be called link 0. The first moving body is link 1, and so on, out to the free end of the arm, which is link n. In order to position an end—effector generally in 3-space, a minimum of six joints is required. Typical manipulators have five or six joints. Some robots may actually not be as simple as a single kinematic chain—they may have parallelogram linkages or other closed kinematic structures. We will consider one such manipulator later in this chapter.

A single link of a typical robot has many attributes which a mechanical designer had to consider during its design. These include the type of material used, the strength and stiffness of the link, the location and type of the joint bearings, the external shape, the weight and

inertia, etc. However, for the purposes of obtaining the kinematic equations of the mechanism, a link is considered only as a rigid body which defines the relationship between two neighboring joint axes of a manipulator. Joint axes are defined by lines in space. Joint axis i is defined by a line in space, or a vector direction, about which link i rotates relative to link i-1. It turns out that for kinematic purpose, a link can be specified with two numbers which define the relative location of the two axes in space.

For any two axes in 3-space there exists a well—defined measure of distance between them. This distance is measured along a line which is mutually perpendicular to both axes. This distance is measured along line which is mutually perpendicular to both axes. This mutual perpendicular always exists and is unique except when both axes are parallel, in which case there are many mutual perpendiculars of equal length. Figure 3.2 shows link i-1 and the mutually perpendicular line along which the link length, a_{i-1}, is measured. Another way to visualize the link parameter a_{i-1} is to imagine an expanding cylinder whose axis is the joint i-1 axis—when it just touches joint axis is the joint i-1 axis—when it just touches joint axis i the radius of the cylinder is equal to a_{i-1}.

The second parameter need to define the relative location of the two axes is called the link twist. If we imagine a plane whose normal is the mutually perpendicular line just constructed, we can project both axes i-1 and i onto this plane and measure the angle between them. This angle is measured from axis i-1 to axis i in the right-hand sense about a_{i-1}. We will use this definition of the twist of link i-1, a_{i-1}. In Fig.3.2, a_{i-1} is indicated as the angle between axis i-1 and axis i (the lines with the triple hash marks are parallel). In the case of intersecting axes, twist is measured in the plane containing both axes, but the sense of a_{i-1} is lost. In this special case, one is free to assign the sign of a_{i-1} arbitrarily.

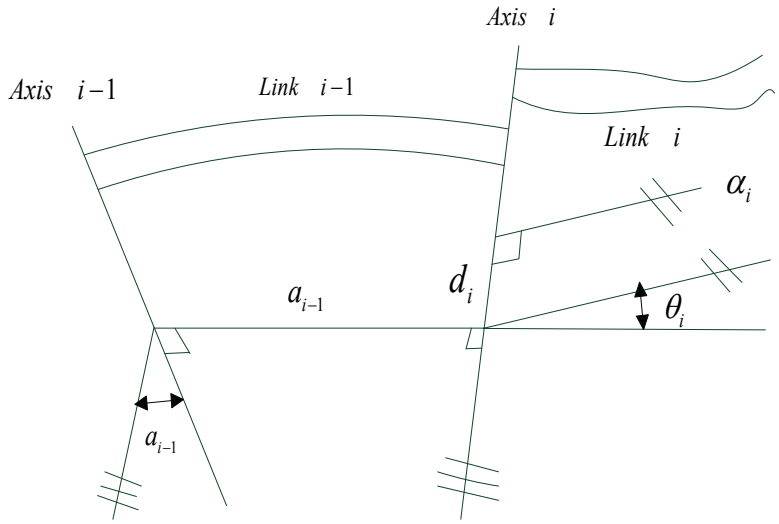

Fig. 6. The link offset, d, and the joint angle, θ, are two parameters which may be used to describe the nature of the connection between neighboring links.

2.5 Kinematics of robot

Robot kinematics is the study of the motion (kinematics) of robots. In a kinematic analysis the position, velocity and acceleration of all the links are calculated without considering the forces that cause this motion. The relationship between motion, and the associated forces and torques is studied in robot dynamics. One of the most active areas within robot kinematics is the screw theory.

Robot kinematics deals with aspects of redundancy, collision avoidance and singularity avoidance. While dealing with the kinematics used in the robots we deal each parts of the robot by assigning a frame of reference to it and hence robot with many parts may have many individual frames assigned to each movable parts. For simplicity we deal with the single manipulator arm of the robot. Each frames are named systematically with numbers, for example the immovable base part of the manipulator is numbered 0, and the first link joined to the base is numbered 1, and the next link 2 and similarly till n for the last nth link.

Robot kinematics is mainly of the following two types: forward kinematics and inverse kinematics. Forward kinematics is also known as direct kinematics. In forward kinematics, the length of each link and the angle of each join are given and we have to calculate the position of any point in the work volume of the robot. In inverse kinematics, the length of each link and position of the point in work volume is given and we have to calculate the angle of each joint.

Robot kinematics can be divided in serial manipulator kinematics, parallel manipulator kinematics, mobile robot kinematics and humanoid kinematics.

2.6 Reverse kinematics of robot

Direct kinematics consists in specifying the state vector of an articulated figure over time. This specification is usually done for a small set of “key-frames", while interpolation techniques are used to generate in-between positions. The main problems are the design of convenient key-frames, and the choice of adequate interpolation techniques. The latter problem, and in particular the way orientations can be represented and interpolated has been widely studied. Designing key positions is usually left onto the animator's hand, and the quality of resulting motions deeply depends on his skills. In many cases, available physical and biomechanical knowledge such as the characterization of motion phases for human walking, can help the animator to create relevant key-frames.

The exclusive use of direct kinematics makes it direct to add constraints to the motion, such as those specifying that the feet should not penetrate into the ground during the support phases. These constraints may be solved using inverse kinematic algorithms. Here, motion ΔX of the end link of a chain (ie. a foot) is specified by the animator in world coordinates. The system computes the variation $\Delta\theta$ of the state vector (ie. the orientations between intermediate links) that will meet the constraint. The relation between the _main task" ΔX and the angular displacements $\Delta\theta$ takes the form:

$$\Delta X = J\,\Delta\theta \tag{18}$$

where J is the Jacobian matrix of the system . J is not directly invertible, due to the direct dimensions of X and θ (ie. there is an infinity of angular positions at joints that lead to the same Cartesian position of a foot). So the most frequently used solution is:

$$\Delta\theta = J^{+}\Delta X + \alpha(I - J^{+}J)\Delta z \tag{19}$$

Where J^{+} is the pseudo-inverse of the Jacobian matrix J, α is a penalty constant, I is the identity matrix, and Δz is a constraint to minimize, called the secondary task. This secondary task is enforced on the null space of the main task. Thus, the second term does not affect the achievement of the main task, whatever the secondary task Δz is. Generally, Δz is used to account for joint angular limits or to minimize some energetic criteria.

3. Walking gait planning for humanoid robot

3.1 Walking pattern generation based on a inverted pendulum model

An inverted pendulum is a pendulum which has its mass above its pivot point. It is often implemented with the pivot point mounted on a cart that can move horizontally and may be called a cart and pole. Whereas a normal pendulum is stable when hanging downwards, an inverted pendulum is inherently unstable, and must be actively balanced in order to remain upright, either by applying a torque at the pivot point or by moving the pivot point horizontally as part of a feedback system.

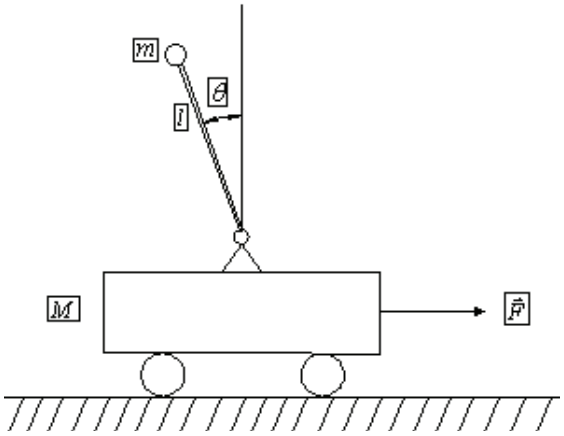

Fig. 7. a schematic drawing of the inverted pendulum on a cart. The rod is considered massless. The mass of the cart and the pointmass at the end of the rod are denoted by M and m. The rod has a length l.

The inverted pendulum is a classic problem in dynamics and control theory and widely used

as benchmark for testing control algorithms (PID controllers, neural networks, fuzzy control, genetic algorithms, etc). Variations on this problem include multiple links, allowing the motion of the cart to be commanded while maintaining the pendulum, and balancing the cart-pendulum system on a see-saw. The inverted pendulum is related to rocket or missile guidance, where thrust is actuated at the bottom of a tall vehicle. The understanding of a similar problem is built in the technology of Segway, a self-balancing transportation device. The largest implemented uses are on huge lifting cranes move the box accordingly so that it never swings or sways. It always stays perfectly positioned under the operator even when moving or stopping quickly.

Another way that an inverted pendulum may be stabilized, without any feedback or control mechanism, is by oscillating the support rapidly up and down. If the oscillation is sufficiently strong (in terms of its acceleration and amplitude) then the inverted pendulum can recover from perturbations in a strikingly counterintuitive manner. If the driving point moves in simple harmonic motion, the pendulum's motion is described by the Mathieu equation.

In practice, the inverted pendulum is frequently made of an aluminum strip, mounted on a ball-bearing pivot; the oscillatory force is conveniently applied with a jigsaw.

Equations of motion

Stationary pivot point

The equation of motion is similar to that for an uninverted pendulum except that the sign of the angular position as measured from the vertical unstable equilibrium position:

$$\ddot{\theta} - \frac{g}{l}\sin\theta = 0 \tag{20}$$

When added to both sides, it will have the same sign as the angular acceleration term:

$$\ddot{\theta} = \frac{g}{l}\sin\theta \tag{21}$$

Thus, the inverted pendulum will accelerate away from the vertical unstable equilibrium in the direction initially displaced, and the acceleration is inversely proportional to the length. Tall pendulums fall more slowly than short ones.

Pendulum on a cart

The equations of motion can be derived easily using Lagrange's equations. Referring to the drawing where x (t) is the position of the cart, $\theta(t)$ is the angle of the pendulum with respect to the vertical direction and the acting forces are gravity and an external force in the x-direction, the lagrangian L = T – V, where T is the kinetic energy in the system and V the potential energy, so the written out expression for L is:

$$L = \frac{1}{2}Mv_1^2 + \frac{1}{2}mv_2^2 - mgl\cos\theta \tag{22}$$

Where v_1 is the velocity of the cart and v_2 is the velocity of the point mass m.

v_1 and v_2 can be expressed in terms of X and θ by writing the velocity as the first derivative of the position:

$$v_1^2 = \dot{x}^2 \tag{23}$$

$$v_2^2 = \left(\frac{d}{dt}\left(l\cos\theta\right)\right)^2 + \left(\frac{d}{dt}\left(x + l\sin\theta\right)\right)^2 \tag{24}$$

Simplifying the expression for v_2 leads to:

$$v_2^2 = \dot{x}^2 + 2\dot{x}l\dot{\theta} + l^2\dot{\theta}^2 \tag{25}$$

The Lagrangian is now given by:

$$L = \frac{1}{2}\left(M+m\right)\dot{x}^2 + ml\dot{x}\dot{\theta}\cos\theta + \frac{1}{2}ml^2\dot{\theta}^2 - mgl\cos\theta \tag{26}$$

and the equations of motion are

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{x}} - \frac{\partial L}{\partial x} = F \tag{27}$$

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}} - \frac{\partial L}{\partial \theta} = 0 \tag{28}$$

Substituting L in these equations and simplifying leads to the equations that describe the motion of the inverted pendulum:

$$\left(M+m\right)\ddot{x} + ml\ddot{\theta}\cos\theta - ml\dot{\theta}^2\sin\theta = F \tag{29}$$

$$ml\left(-g\sin\theta + \ddot{x}\cos\theta + l\ddot{\theta}\right) = 0 \tag{30}$$

These equations are nonlinear, but since the goal of a control system would be to keep the pendulum upright the equations can be linearized around $\theta \approx 0$.

Pendulum with oscillatory base

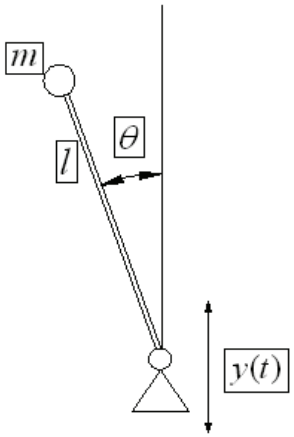

Fig. 8. a schematic drawing of the inverted pendulum on an oscillatory base. The rod is considered massless. The pointmass at the end of the road is demoted by m . The rod has a length l .

The equation of motion for a pendulum with an oscillatory base is derived the same way as with the pendulum on the cart, using the Lagrangian.
The position of the point mass is now given by:

$$\left(l\sin\theta, y + l\cos\theta\right) \tag{31}$$

And the velocity is found by taking the first derivative of the position:

$$v^2 = \dot{y}^2 - 2l\dot{\theta}\dot{y}\sin\theta + l^2\dot{\theta}^2 \tag{32}$$

The Lagrangian of this system can be written as:

$$L = \frac{1}{2}m\left(\dot{y}^2 - 2l\dot{\theta}\dot{y}\sin\theta + l^2\dot{\theta}^2\right) - mg(y + l\cos\theta) \tag{33}$$

and the equation of motion follows from:

$$\frac{d}{dt}\frac{\partial L}{\partial\dot{\theta}} - \frac{\partial L}{\partial\theta} = 0 \tag{34}$$

Resulting in:

$$l\ddot{\theta} - \ddot{y}\sin\theta = g\sin\theta \tag{35}$$

If y represents a simple harmonic motion, $y = a\sin\omega t$, the following differential equation is:

$$\ddot{\theta} - \frac{g}{l}\sin\theta = -\frac{a}{l}\omega^2 \sin\omega t \sin\theta \tag{36}$$

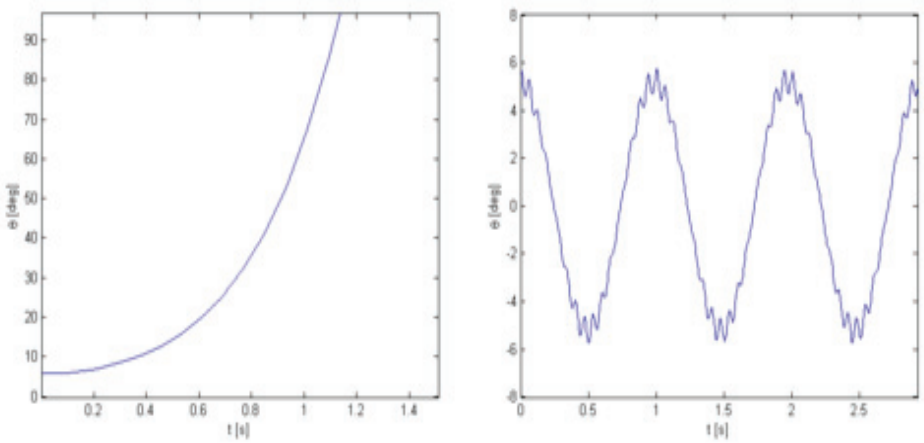

Fig. 9. Plots for the inverted pendulum on an oscillatory base. The first plot shows the response of the pendulum on a slow oscillation, the second the response on a fast oscillation

A solution for this equation will show that the pendulum stays upright for fast oscillations. The first plot shows that when y is a slow oscillation, the pendulum quickly falls over when disturbed from the upright position. The angle θ exceeds 90°after a short time, which means the pendulum has fallen on the ground.

If y is a fast oscillation the pendulum can be kept stable around the vertical position. The second plot shows that when disturbed from the vertical position, the pendulum now starts an oscillation around the vertical position ($\theta = 0$). The deviation from the vertical position stays small, and the pendulum doesn't fall over.

3.2 Gait planning of robot based on a seven-link model

In order to simplify research process we first discuss how to get ankle trajectory and hip trajectory. Then the knee trajectory could be got by kinematics. Here we take the left foot for example and the right foot is similar only with a delay of half cycle. The link model we used is shown in Figure 3.

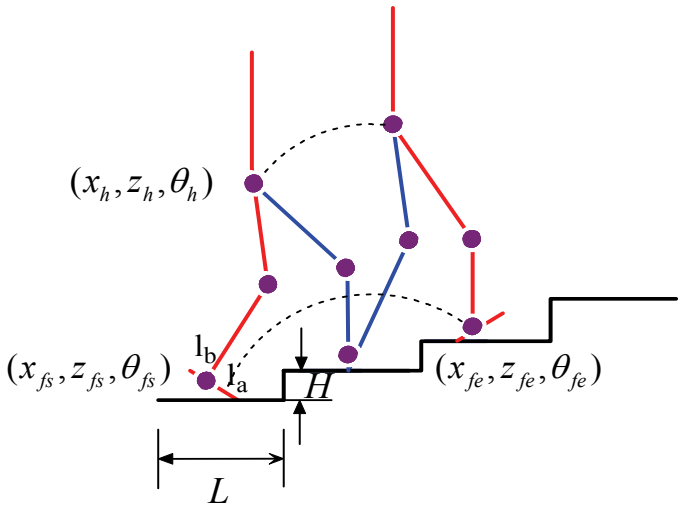

Fig. 10. The model of the humanoid robot going upstairs.

3.2.1 Gait Planning of Ankle

According to the walking procedure of human, we suppose that the walking cycle is T_c, $t = kT_c$ is the k th cycle begins with the moment when the left foot is just apart from the ground and ends with the left foot gets into contact with the ground; $kT_c < t \le kT_c + T_d$ is double support phase, during which the sole is rotated about toes, and the center of gravity moving forwards; the swing foot reaches the highest point when $t = kT_c + T_n$.

We get the key point $x_f(t), z_f(t)$ of ankle in plane XOZ as follows:

$$z_f = \begin{cases} z_{fs} + 2Hk + h_f & t = kT_c \\ z_{fs} + 2Hk + h_f \cos\theta_{fs} + l_a \left|\sin\theta_{fs}\right| & t = kT_c + T_d \\ z_{fs} + 2Hk + h_f + H & t = kT_c + T_n \\ z_{fs} + 2H(k+1) + h_f \cos\theta_{fe} + l_b \left|\sin\theta_{fe}\right| & t = (k+1)T_c \\ z_{fs} + 2H(k+1) + h_f & t = (k+1)T_c + T_d \\ z_{fs} + 2H(k+2) + h_f & t = (k+2)T_c \end{cases} \tag{37}$$

where l_a is the distance between tiptoe and the centre of gravity of sole; l_b is the distance between heel and the centre of gravity of sole; h_f is the height of heel and T_n is the time when the robot just walks through a step.

The key point of the angle between sole and ground can be denoted as follows:

$$\theta_f = \begin{cases} 0 & t = kT_c \\ \theta_{fs} & t = kT_c + T_d \\ 0 & t = kT_c + T_n \\ \theta_{fe} & t = (k+1)T_c \\ 0 & t = (k+1)T_c + T_d \end{cases} \tag{38}$$

Since the whole sole of the right foot is in contact with the ground at $t = kT_c$ and $t = (k+1)T_c + T_d$, the following derivative constraints must be satisfied.

$$\begin{cases} \dot{x}_f(kT_c) = 0 \\ \dot{x}_f((k+1)T_c + T_d) = 0 \end{cases} \tag{39}$$

$$\begin{cases} \dot{z}_f(kT_c) = 0 \\ \dot{z}_f((k+1)T_c + T_d) = 0 \end{cases} \tag{40}$$

$$\begin{cases} \dot{\theta}_f(kT_c) = 0 \\ \dot{\theta}_f((k+1)T_c + T_d) = 0 \end{cases} \tag{41}$$

3.2.2. Gait Planning of Hip

We assume that the robot is decelerated in double support phase and accelerated in single support phase and the acceleration in direction of x-axis and z-axis are a_{xh} and a_{zh} respectively. The distance between the hip and the ankle of supporting leg is x_s at the beginning of the double support phase and x_e at the end of the double support phase. The changes in the direction of z-axis are z_s and z_e at the beginning and end of the double support phase respectively. Then the trajectory of hip can be expressed like this:

$$x_h(t) = \begin{cases} x_h + 2Lk + x_s & t = kT_c \\ x_h + 2Lk + x_s + a_{xh}t & t = kT_c + T_d \\ x_h + 2Lk + L & t = kT_c + T_n \\ x_h + 2L(k+1) - x_e - a_{xh}t & t = (k+1)T_c \\ x_h + 2L(k+1) - x_e & t = (k+1)T_c + T_d \\ x_h + 2L(k+1) + x_s & t = (k+2)T_c \end{cases} \tag{42}$$

$$z_h(t) = \begin{cases} z_h + 2Hk + z_s & t = kT_c \\ z_h + 2Hk + z_s + a_{zh}t & t = kT_c + T_d \\ z_h + 2Hk + H & t = kT_c + T_n \\ z_h + 2H(k+1) - z_e - a_{zh}t & t = (k+1)T_c \\ z_h + 2H(k+1) - z_e & t = (k+1)T_c + T_d \\ z_h + 2H(k+1) + z_s & t = (k+2)T_c \end{cases} \quad (43)$$

It must satisfy the following constraints:

- The derivative constraints $\begin{cases} \dot{x}_h(kT_c) = \dot{x}_h(k+1)T_c \\ \ddot{x}_h(kT_c) = \ddot{x}_h(k+1)T_c \end{cases}$ and $\begin{cases} \dot{z}_h(kT_c) = \dot{z}_h(k+1)T_c \\ \ddot{z}_h(kT_c) = \ddot{z}_h(k+1)T_c \end{cases}$ must be satisfied.
- $z_h(t) \le h_{\max}$, $h_{\max}$ is the maximum height of hip; $h_{\max} = l_1 + l_2 + h_f$, l_1, l_2 are the length of thigh and shin respectively, h_f is the height of ankle.
- $z_h(t) \ge h_{\min}$ $h_{\min}$ is the minimum height of hip and it's value can be set according to the process of human walking.
- $\{[x_h(t) - x_a(t)]^2 + [z_h(t) - z_a(t)]^2\}^{1/2} \le l_1 + l_2$

4. Stability control of humanoid robot

4.1 ZMP and FZMP concept

Zero moment point was introduced in January 1968 by Miomir Vukobratović at The Third All-Union Congress of Theoretical and Applied Mechanics in Moscow. In the following works and papers that were produced between 1970 and 1972 it would then be called zero moment point and would be spread around the world.

The zero moment point is a very important concept in the motion planning for biped robots. Since they have only two points of contact with the floor and they are supposed to walk, "run" or "jump" (in the motion context), their motion has to be planned concerning the dynamical stability of their whole body. This is not an easy task, especially because the upper body of the robot (torso) has larger mass and inertia than the legs which are supposed to support and move the robot. This can be compared to the problem of balancing an inverted pendulum.

The trajectory of a walking robot is planned using the angular momentum equation to ensure that the generated joint trajectories guarantee the dynamical postural stability of the robot, which usually is quantified by the distance of the zero moment point in the boundaries of a predefined stability region. The position of the zero moment point is affected by the referred mass and inertia of the robot's torso, since its motion generally requires large ankle torques to maintain a satisfactory dynamical postural stability.

One approach to solve this problem consists in using small trunk motions to stabilize the posture of the robot. However, some new planning methods are being developed to define the trajectories of the legs' links in such a way that the torso of the robot is naturally steered in order to reduce the ankle torque needed to compensate its motion. If the trajectory planning

for the leg links is well succeeded, then the zero moment point won't move out of the predefined stability region and the motion of the robot will become smoother, mimicking a natural trajectory.
The resultant force of the inertia and gravity forces acting on a biped robot is expressed by the formula:

$$F^{gi} = mg - ma_G \tag{44}$$

Where m is the total mass of the robot, g is the acceleration of the gravity, G is the center of mass and a_G is the acceleration of the center of mass. The moment in any point X can be defined as:

$$M_X^{gi} = XG \times mg - XG \times ma_G - \dot{H}_G \tag{45}$$

where $\dot{H}_G$ is the rate of angular momentum at the center of mass. The Newton-Euler equations of the global motion of the biped robot can be written as:

$$F^c + mg = ma_G \tag{46}$$

$$M_X^c + XG \times mg = \dot{H}_G + XG \times ma_G \tag{47}$$

where F^c is the resultant of the contact forces at X and M_X^c is the moment related with contact forces about any point X.
The Newton-Euler equations can be rewritten as:

$$F^C + (mg - ma_G) = 0 \tag{48}$$

$$M_X^c + (XG \times mg - XG \times ma_G - \dot{H}_G) = 0 \tag{49}$$

So it's easier to see that we have:

$$F^c + F^{gi} = 0 \tag{50}$$

$$M_X^C + M_X^{gi} = 0 \tag{51}$$

These equations show that the biped robot is dynamically balanced if the contact forces and the inertia and gravity forces are strictly opposite.

If an axis Δ^{gi} is defined, where the moment is parallel to the normal vector *n* from the surface about every point of the axis, then the Zero Moment Point (ZMP) necessarily belongs to this axis, since it is by definition directed along the vector *n*. The ZMP will then be the intersection between the axis Δ^{gi} and the ground surface such that:

$$M_Z^{gi} = ZG \times mg - ZG \times ma_G - \dot{H}_G \tag{52}$$

with

$$M_Z^{gi} \times n = 0 \tag{53}$$

where *Z* represents the ZMP.
Because of the opposition between the gravity and inertia forces and the contact forces mentioned before, the *Z* point (ZMP) can be defined by:

$$PZ = \frac{n \times M_P^{gi}}{F^{gi} \cdot n} \tag{54}$$

where *P* is a point of the sole where is the normal projection of the ankle.
Fictitious zero moment point (FZMP) is an important expand of ZMP, it can be used in stability control. In order to evaluate dynamic stability, we use the ZMP principle. The ZMP is the point where the influence of all forces acting on the mechanism can be replaced by one single force. If the computed ZMP is the real ZMP, this means the computed ZMP inside the real support polygon, the biped robot can be stable. If the ZMP is not the real ZMP, this means the computed ZMP is on the boundary of the support polygon, the robot will fall down or have a trend of falling down. If the computed ZMP is outside the support polygon, then the robot will fall down and in this case, the computed ZMP is called fictitious ZMP. The link model of the humanoid robot is shown in Figure 3.4.

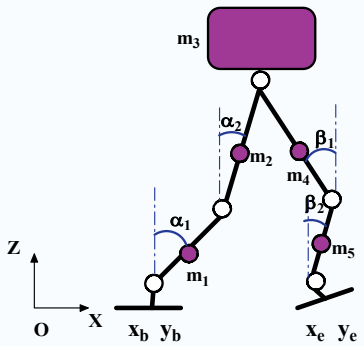

Fig. 11. The link model of the humanoid robot.

The projection of position vector of computed ZMP can be computed by the following equations:

$$x_{zmp} = \frac{\sum_{i=1}^{5} m_i(\ddot{z}_i + g)x_i - \sum_{i=1}^{5} m_i \ddot{x}_i z_i + \sum_{i=1}^{5} M_{iy}}{\sum_{i=1}^{5} m_i(\ddot{z}_i + g)} \tag{55}$$

$$y_{zmp} = \frac{\sum_{i=1}^{5} m_i(\ddot{z}_i + g)y_i - \sum_{i=1}^{5} m_i \ddot{y}_i z_i + \sum_{i=1}^{5} M_{ix}}{\sum_{i=1}^{5} m_i(\ddot{z}_i + g)} \tag{56}$$

where m_i is mass of every links, (x_i, y_i, z_i) is the coordinate of the mass center of the links, $(M_{ix}, M_{iy})^T$ is the moment vector.

If the ZMP is inside the support polygon and the minimum distance between the ZMP and the boundaries of support polygon is large, then the biped will be in high stable, and this distance is called the stability margin. We can know the situation of walking stability from the stability margin.

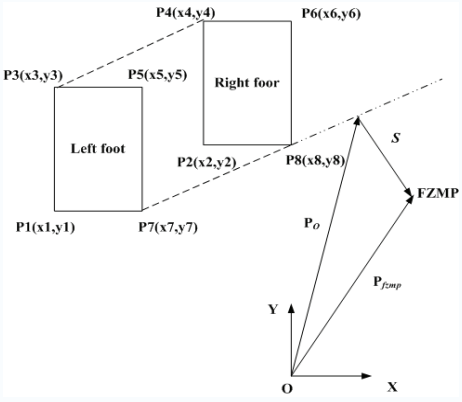

Fig. 12. The relationship between FZMP and support polygon.

As shown in Figure2, if the ZMP is outside the support polygon, i.e. FZMP, the norm of vector s represents the shortest distance between FZMP and the edges of the support polygon. This edge is called rotation edge. The direction of vector s is the rotation direction of the robot.

The importance of FZMP is:

- We can judge the falling down possibility by calculated the position of FZMP;
- According the position of FZMP, we can calculate the distance of rotate boundary and falling downing direction.

- When the robot in the stability situation, support polygon can be defined as the minimum distance between the boundaries of support polygon and the ZMP, this means robot stability margin; while in the instability situation, the minimum distance between the boundaries of support polygon and the ZMP, it is a measure of instability.

4.1.1 ZMP and stable walking

Apart from the realization of the relative motion of the mechanism's links, the most important task of a locomotion mechanism of humanoid robot during walking is to preserve its dynamic balance in contact with the ground. The foot relies freely on the upport and is realized via the friction force and vertical force of the ground reaction. The foot cannot be controlled directly but in an indirect way, by ensuring the appropriate dynamics of the mechanism above the foot. Thus, the overall indicator of the mechanism behavior is the point where the influence of all forces acting on the mechanism can be replaced by one single force. This point was termed the Zero-Moment Point (ZMP). ZMP is very important for humanoid robot as dynamic criterion of gait planning, stability and control. The ZMP principle can be generalized as follows.

- If ZMP is inside of the footprint of support foot in single support phase, or inside of support polygon in double support phase, then biped robot can keep its dynamic balance and the stable walking is possible.
- If ZMP is on the boundary of the footprint in single support phase or of support polygon in double support phase, then the robot will fall down or have a trend of falling down.
- If computed ZMP is outside of the footprint of support foot in single support phase, or without support polygon in double support phase, then the robot cannot be in the dynamic stable and will fall down. In this case, it should be called fictitious ZMP, shortly FZMP.

There are two different cases in which the ZMP plays a key role:

□(1) in determining the proper dynamics of the mechanism above the foot to ensure a desired ZMP position. This belongs to the task of gait synthesis.

□(2) in determining the ZMP position for the given mechanism motion. This refers to the gait control.

Biped walking is a periodic phenomenon. A complete walking cycle is composed of two phases: a double-support phase and a single-support phase. During the double-support phase, both feet are in contact with the ground. This phase begins with the heel of the forward foot touching the ground, and ends with the toe of the rear foot leaving the ground. During the single-support phase, one foot is stationary on the ground, the other foot swings from the rear to the front. The gait of walking robot can be generated by using ZMP principle.

4.1.2.FZMP and stability maintenance

For determination of dynamic equilibrium we have to consider the relationship between the computed position of reaction point P on the ground and the support polygon. If the position of point P is within the support polygon, the robot is in dynamic equilibrium. The computed position P is called traditional ZMP, if only one foot contacts with floor, the force acting at ZMP is real reaction force, if two feet contact with floor it is total force of all contact reaction forces. If the computed point P is located outside the support polygon, it can be called

as a fictitious ZMP (FZMP). In this case, the humanoid robot would start to rotate about the edge and the robot would lose the stability. The real acting point of ground reaction force would be located in the edge. The calculated position of the point P outside the support polygon represents only fictitious locations. The FZMP is very useful to deal with the stability maintenance and control in the emergency case. For the stable walking of humanoid robot the ZMP must be kept within the support polygon. To maintain regularly the mechanism dynamic stable at the moment of the occurrence of an external disturbance an emergency-coping strategy based on FZMP concept can be applied. The importance of the FZMP to deal with the stability control and maintenance is mentioned by several authors. But how to fully utilize its property should be further researched. In this paper the FZMP is efficiently used to deal with the stability maintenance of humanoid robot under disturbance.

4.2 The determination of support polygon and stability margin

4.2.1 The determination of support polygon

If only one foot contacts with floor, the above mentioned support polygon is the region of the foot. But if the two feet contact with the floor, the situation would be sometime complex. In the current related researches the support polygon used to be expressed simply with the graphs. It is not convenient in stability analysis and control. In this paper, we present a computerization expression of support polygon.

We assume that the shape of the foot is rectangle. Then two feet contain eight edges all together. The support polygons are composed of some edges of the above mentioned eight edges and other two new edges. We call all edges that constitute the support polygon as valid connection edges (VCE). The candidates of the VCE are all connection edges of the eight corner points on two feet. In figure 3 the corner points in left foot are denoted with P_1, P_3, P_5, P_7, the right foot P_2 P_4, P_6, P_8. The line L_{ij} through two point P_i and P_j can be expressed as follows:

$$y = a_{ij}x + b_{ij}$$

$$a_{ij} = \frac{y_i - y_j}{x_i - x_j} \quad b_{ij} = \frac{x_j y_i - x_i y_j}{x_j - x_i} \tag{57}$$

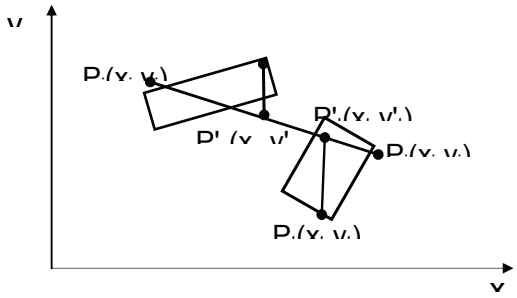

Fig. 13. The determination of support polygon

In order to determine whether the edge connecting point P_i and P_j is the VCE we have to consider the position relationship between L_{ij} and all eight corner points. If all eight corner points are the same side of the line L_{ij}, that is, satisfy (6), then edge E_{ij} according to line L_{ij} is the VCE. Otherwise E_{ij} is not VCE and should be ignored. Here, P_s, P_t is respectively the one of the eight corner points.

$$\begin{cases} y_s - y_s^{'} > 0 \\ y_t - y_t^{'} > 0 \end{cases} \text{ or } \begin{cases} y_s - y_s^{'} < 0 \\ y_t - y_t^{'} < 0 \end{cases} \quad (s,t = 1,...8.s \neq t) \tag{58}$$

Where $P_s^{'}$ and $P_t^{'}$ are the projection points of P_s and P_t on the

$$L_{ij}\ , y_s^{'} = a_{ij}x_s + b_{ij}\ , y_t^{'} = a_{ij}x_t + b_{ij}\ .$$

4.2.2 The Relationship Between FZMP and Support Polygon

We have to determine which is the rotation edge in all VCEs when robot lose stability. and in this case the distance from FZMP to the rotation edge can be calculated. In figure 4 the distance from FZMP to E_{ij} is expressed as follows:

$$\|\mathrm{s}\| = \min_{i,j=np} \|p_{FZMP} - p_{ij}\| \tag{59}$$

Where p_{ij} is the position vector of vertical point from FZMP to E_{ij} .

$$\begin{aligned} x_{P_{ij}} &= \frac{a_{ij}b_{ij} - a_{ij}y_{fzmp} + x_{fzmp}}{1 - a_{ij}^2} \\ y_{P_{ij}} &= \frac{b_{ij} - a_{ij}^2 y_{fzmp} + a_{ij}x_{fzmp}}{1 - a_{ij}^2} \end{aligned} \tag{60}$$

We denote the point p_o as the position vector of the vertical point that satisfies (7). That is

$$\mathrm{s} = p_{fzmp} - p_o \tag{61}$$

If we know the position of FZMP the rotation edge can be determined according to (7), the distance from FZMP to the rotation edge and the direction of losing stability can be calculated by (9).Those two parameters play the key role in maintaining the robot stability.

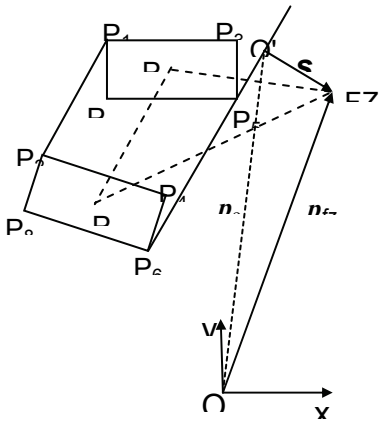

Fig. 14. The relationship between FZMP and support polygon

4.2.3 Control algorithm considering external environment

When the robot might lose stability because of the external disturbance, it must immediately react this situation and be controlled to keep in stable state.

The control approaches could be one of the several methods such as the movement of the upper body to change of the center of gravity, the enlargement or movement of the support polygon, the attachment of robot hand to the surrounding.

(1) The Enlargement of Support Polygon

We can enlarge the support polygon by modifying the prescribed landing position of the swing foot to maintain stability under disturbance. It is mentioned by some research, but it is not explained how to realize the enlargement of support polygon. It is not realistic to move parallel the rotation edge, which means to moving two feet at same time.

In figure 5 the moving foot should land the planed position expressed in dashed line if no external disturbance. But under disturbance the robot will be rotate about RB. In this case, the landing position should be changed to maintain the stability. The changed angle α_f^* of moving direction of the foot is determined by (10).

$$\alpha_f^* = \cos^{-1} \frac{e \cdot s}{\|e\| \cdot \|s\|} \tag{62}$$

Here e is the normal planed direction vector.

The foot moving distance l_f^* relative to planed landing position is determined by the formula (4.19)

$$l_f^* = \sin(\alpha^* - \beta)\left[\mathrm{l}\cos(\gamma - \alpha^* - \beta) - \mathrm{b}\right] + \mathrm{l}\sin(\gamma - \alpha^* - \beta) + \mathrm{a} \tag{63}$$

If the change of center of gravity due to the foot extra moving is ignored, then at new landing position the robot will be stable.

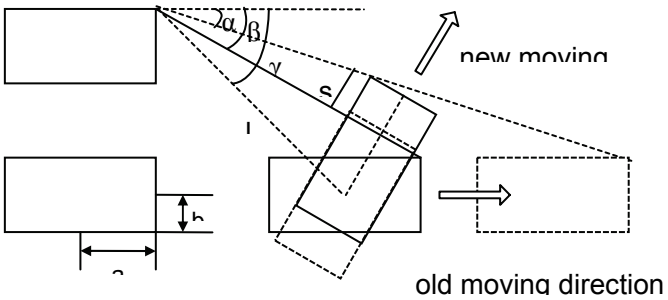

Fig. 15. The determination of foot landing position

(2) The Movement of Upper Body

We assume that the all links of upper body have same the displacement, the velocity and acceleration s_u, v_u, a_u respectively. The moving direction should be pointed to the FZMP. That is, $x_j = s_u \cos\alpha_f^*$, $y_j = s_u \sin\alpha_f^*$, $\dot{x}_j = v_u \cos\alpha_f^*$ $\dot{y}_j = v_u \sin\alpha_f^*$, $\ddot{x}_j = a_u \cos\alpha_f^*$, $\ddot{y}_j = a_u \sin\alpha_f^*$. We assume that $z_j = const, \ddot{z}_j = 0$ for all upper body j-th link. In this case, equation (3) is modified to equation (12).

$$\begin{aligned}
&\bar{x}_{zmp}\left(\sum_{i=1}^{m} Q_{iz} + M\left(\ddot{z}_G + g\right)\right) = \Delta\dot{H}_{G_Y} + \left(Mx_G + M_u s_u \cos\alpha_f^*\right)\left(\ddot{z}_G + g\right) \\
&- z_G\left(M\ddot{x}_G + M_u a_u \cos\alpha_f^*\right) - \tau_y - \sum_{i=1}^{m}\left(x_{Q_i} Q_{iz} - z_{Q_i} Q_{ix}\right) \\
&\bar{y}_{zmp}\left(\sum_{i=1}^{m} Q_{iz} + M\left(\ddot{z}_G + g\right)\right) = \Delta\dot{H}_{G_x} + \left(My_G + M_u s_u \sin\alpha_f^*\right)\left(\ddot{z}_G + g\right) \\
&- z_G\left(M\ddot{y}_G + M_u a_u \sin\alpha_f^*\right) - \tau_x - \sum_{i=1}^{m}\left(y_{Q_i} Q_{iz} - z_{Q_i} Q_{iy}\right)
\end{aligned} \tag{64}$$

$$\Delta H_{G_Y} = \sum_{i=1}^{n_1}\left(-\frac{M_u}{M} s_u \cos\alpha_f^* m_i \dot{z}_{G_i}\right) + \sum_{i=n_1+1}^{n_2}\left(\frac{M-M_u}{M} s_u m_i \cos\alpha_f^* \dot{z}_{G_i}\right)$$

$$\Delta H_{Gx} = \sum_{i=1}^{n_1}\left(-\frac{M_u}{M} s_u \sin\alpha_f^* m_i \dot{z}_{G_i}\right) + \sum_{i=n_1+1}^{n_2}\left(\frac{M-M_u}{M} s_u m_i \sin\alpha_f^* \dot{z}_{G_i}\right)$$

where M_u is the total mass of upper body. $\bar{x}_{zmp}$ and $\bar{y}_{zmp}$ are the x- and y-projector of pre-designed ZMP. From this equation s_u^* and a_u^* can be calculated.

(3) The Attachment of the Hand to the Surrounding

Through the arm movement to attach with the surrounding to ensure additional support

points the static equilibrium may be re-established and the dynamically balanced gait continued. This procedure of re-establishing dynamic equilibrium might be considered as a kind of total compliance procedure. The position of computed zero-moment point will be changed under the support reaction force as shown as following:

$$x'_c - x_c = \frac{u_{hy} + x_l Q_{lz} - z_l Q_{lx} + x_r Q_{rz} - z_r Q_{rx}}{\sum_{i=1}^{m} Q_{iz} + Q_{lz} + Q_{rz} + M\left(\ddot{z}_G + g\right)}$$

$$y'_c - y_c = \frac{u_{hx} + x_l Q_{lz} - z_l Q_{ly} + x_r Q_{rz} - z_r Q_{ry}}{\sum_{i=1}^{m} Q_{iz} + Q_{lz} + Q_{rz} + M\left(\ddot{z}_G + g\right)} \tag{65}$$

Here we assume $x_l = x_r = x_a$, $y_l = y_r = y_a$, $z_l = z_r = z_a$, $Q_{lx} = Q_{rx} = Q_x^*/2$, $Q_{lz} = Q_{rz} = Q_z^*/2$, $Q_{ly} = Q_{ry} = 0$, $\Delta x_{ch} = x'_c - x_c$, $\Delta y_{ch} = y'_c - y_c$, we can obtain:

$$Q_x^* = \frac{(\Delta y_{ch} - \Delta x_{ch}) x_a}{(\Delta y_{ch} - x_a) z_a} \left(\sum_{i=1}^{m} Q_{iz} + M(\ddot{z}_G + g \right)$$

$$Q_z^* = \frac{\Delta y_{ch}}{\Delta y_{ch} - x_a} \left(\sum_{i=1}^{m} Q_{iz} + M(\ddot{z}_G + g \right) \tag{66}$$

Because the hand can not only push but also pull the environment, the forces Q_x^* and Q_z^* can be positive or negative.

(4) The Optimization Control Strategy

The method above can be used to maintain the stability of the robot, but in some cases only one method is unrealistic because of the limitation of the time or foot stride etc.. we have to use the combination of the method above to maintain the stabilities of robot. In this case the stability maintenance can be considered the following dynamic optimization problem.

Objective function:

$$\begin{aligned} F(X) = & \min_{X \in R^m} (\lambda \max_t (\left\| p_c(X) - p_{zmp}^* \left(k_f X_f \right) \right\|_{p_c = p_{zmp}}) \\ & + \gamma \max_t (\left\| p_c(X) - p_o(X) \right\|_{p_c = p_{fzmp}})) \end{aligned} \tag{67}$$

Subject to

$$x_c = \frac{\dot{H}_{Gy} + \left(Mx_G + M_u k_u s_u \cos\alpha_f^*\right)\left(\ddot{z}_G + g\right) - \left(M\ddot{x}_G + M_u k_u a_u \cos\alpha_f^*\right) z_G}{\sum_{i=1}^{m} Q_{iz} + k_a Q_z + M\left(\ddot{z}_G + g\right)}$$

$$- \frac{\tau_y + \sum_{i=1}^{m}\left(x_{Q_i} Q_{iz} - z_{Q_i} Q_{ix}\right) + k_a\left(x_a Q_z - z_a Q_x\right)}{\sum_{i=1}^{m} Q_{iz} + k_a Q_z + M\left(\ddot{z}_G + g\right)}$$

$$y_c = \frac{\dot{H}_{Gx} + \left(My_G + M_u k_u s_u \sin\alpha_f^*\right)\left(\ddot{z}_G + g\right) - \left(M\ddot{y}_G + M_u k_u a_u \sin\alpha_f^*\right) z_G}{\sum_{i=1}^{m} Q_{iz} + k_a Q_z + M\left(\ddot{z}_G + g\right)}$$

$$- \frac{\tau_x + \sum_{i=1}^{m}\left(y_{Q_i} Q_{iz} - z_{Q_i} Q_{iy}\right) + k_a y_a Q_z}{\sum_{i=1}^{m} Q_{iz} + k_a Q_z + M\left(\ddot{z}_G + g\right)}$$

$$\begin{aligned} H_G &= \sum_{i=1}^{n} H_{G_i} + \sum_{i=1}^{n_1}\left(p_{G_i} - p_G - \frac{M_u}{M} k_u s_u I_u\right) \times m_i \dot{p}_{G_i} \\ &+ \sum_{i=n_1+1}^{n_2}\left(p_{G_i} - p_G + \frac{M - M_u}{M} k_u s_u I_u\right) \times m_i\left(\dot{p}_{G_i} + k_u v_u\right) \\ &= H_{G0} + \sum_{i=1}^{n_1}\left(-\frac{M_u}{M} k_u s_u I_u\right) \times m_i \dot{p}_{G_i} + \\ &\sum_{i=n_1+1}^{n_2}\left(\frac{M - M_u}{M} k_u s_u I_u\right) \times m_i\left(\dot{p}_{G_i} + k_u v_u\right) \\ &+ \sum_{i=n_1+1}^{n_2}\left(p_{G_i} - p_G\right) \times m_i\left(k_u v_u\right) \end{aligned}$$

$$X_{\min} \le X \le X_{\max}$$

$p_o(X)$ is determined by (7)

Where $X = \begin{pmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \end{pmatrix}^T = \left(X_f, X_u, X_a\right)^T$, $p_c(X) = \begin{pmatrix} x_c & y_c & 0 \end{pmatrix}^T$ is the optimization design variable. $X_f = \left(l_f, \alpha_f\right)^T$ is the movement vector of the foot, in which l_f, α_f is the moving distance and direction of the foot relative to planed landing position, respectively. $X_u = (s_u \quad v_u \quad a_u)^T$ is the movement parameter vector of upper body. $X_a = (Q_x, Q_z)^T$ is the hand support force parameters vector, and the force in y-direction

is ignored. λ, γ are the weight coefficients, but they should be set with different numerical value. λ should be chose a relative small value, which depends on the optimization demand. The value γ expresses the influence extent of FZMP on the objective function. It is very important to choice right γ value. If computed value p_c is within the support polygon, γ =0. If outside the support polygon, γ should be chose the large value as the punishment. $k = (k_u \quad k_f \quad k_a)$ is the choice coefficient vector in which k_u , k_f , k_a equal 0 or 1. For example, $k = (1,1,0)$ means that the extra movement of upper body and foot are considered and the hand attachment does not exist.

This is a parameter optimization problem that means the normal gait pattern of robot before the external disturbance is introduced is known. From (15) we present a hierarchy control strategy as shown in figure 4.4.

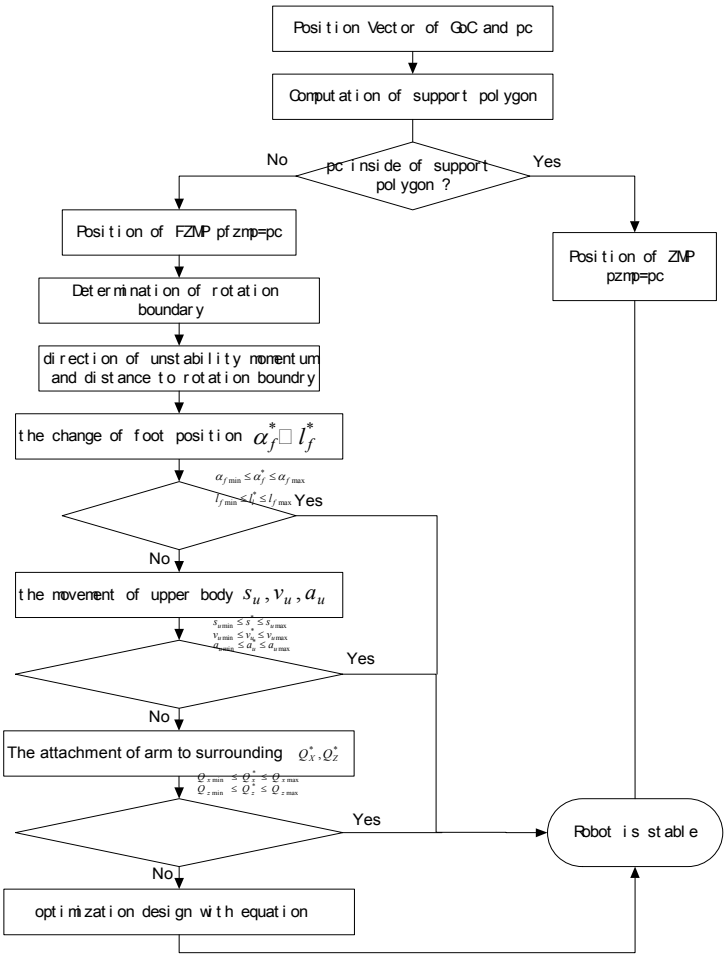

Fig. 16. The control strategy

4.3 Example:
With the method above we have constructed a simulator of humanoid robot by using dynamic analysis software package ADAMS. The total height of humanoid robot is 1650 mm. The walking speed is 2 km/h, the stride is 520 mm. The part of simulation parameters is shown in Table 1.

Components	Length(m)	Mass (kg)
Thigh	0.35	5.0
Shank	0.35	3.0
Ankle Height	0.10	0.5
Upper Arm	0.307	0.56
Lower Arm	0.241	0.58
Hand	0.178	20.0
Foot heel / Tiptoe	0.10 / 0.12	0.8

Table 1. Simulation Parameters

After building the model of humanoid robot we have made the several simulations considering the upper movement, enlargement of foot stride and their combination. In simulation the foot contact with the ground is emphasized. The figure 7 shows the normal walking process of humanoid robot.

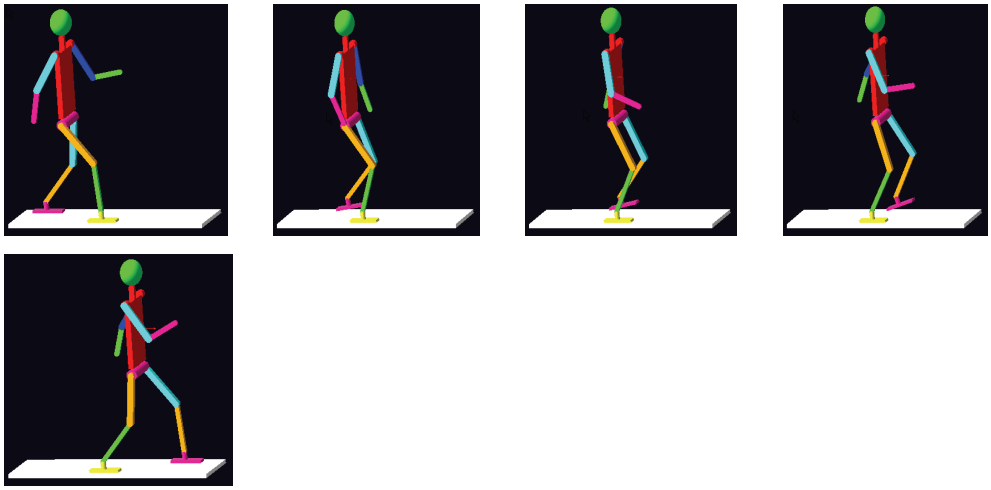

Fig. 17. The walking simulation

The x-coordinate and z-coordinate of center of gravity at normal walking and added upper body moving to maintain stability under external disturbance are shown in figure 8 and

figure 9The stable walking under the reaction of external force can be kept also by the change of foot stride. In figure 10 the foot stride should be enlarged 120 mm compared the normal stride, where the walking direction was not changed. The optimization process is shown in figure 11.The simulations show that the stability under external disturbance can be kept with the described strategy.

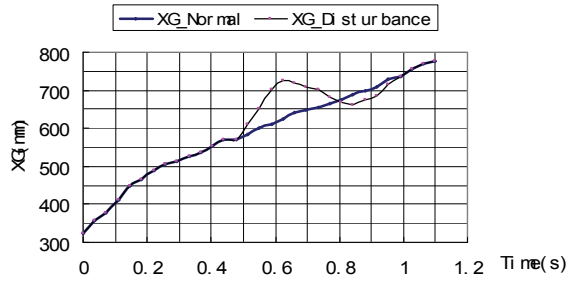

Fig. 18. The x-coordinate of center of gravity

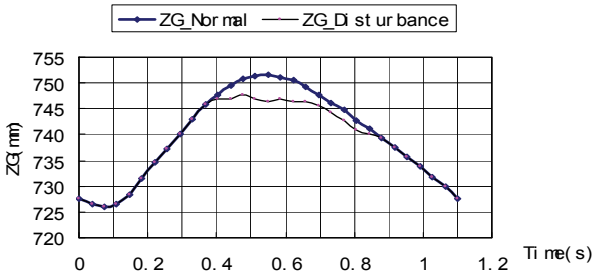

Fig. 19. The z-coordinate of center of gravity

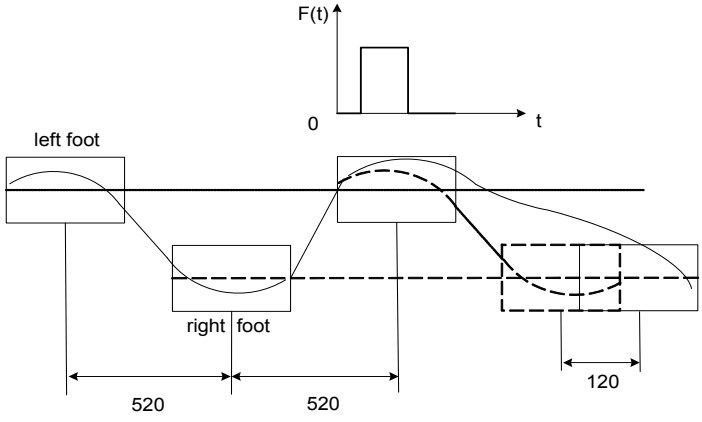

Fig. 20. The enlargement of foot stride (no change in direction)

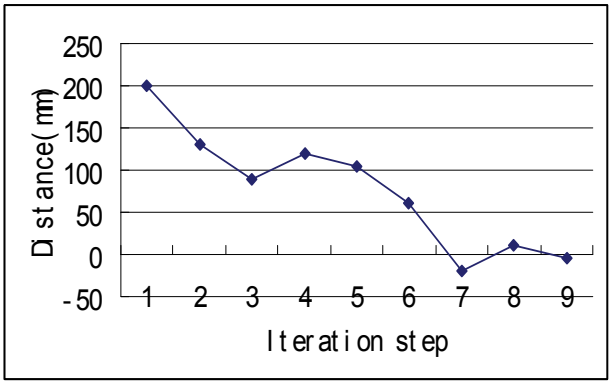

Fig. 21. The Distance between FZMP and center of support polygon

The stability maintenance is important issue in humanoid robot walking. The roles of FZMP are emphasized to maintain the robot stability. The support polygon is expressed with computerized form. According to the position of FZMP the rotation border can be automatic determined. The distance between the FZMP and the rotation border represents the strength and direction of losing the stability. The stability maintenance methods such as the movement of upper body, the change of foot landing position and the hand attachment with environment are discussed. The optimization control model considering different stability maintenance measures is proposed. The numerical simulation shows that the proposed method is effective and has the advantage of less calculation time.

5. Summary

The stability control is important issue in humanoid robot walking. A balance controller consisting of an off-line walk pattern planner and a real-time modification was proposed. If we can solve this problem, the robot can walk smoothly and adapt to unknown environments, and many functions will become true. Inverted pendulum model, ZMP and FZMP conception are effective methods for walking gait planning and the stability control.

6. References

K Kaneko, S Kajita, F Kanehiro, K Yokoi, K Fujiwara, H Hirukawa, T Kawasaki, M Hirata and T Isozumi, "Design of Advanced Leg Module for Humanoid Robotics Project of METI," Proceedings of the 2002 IEEE International Conference on Robotics & Automation, Washington, DC, May 2002

A Konno, "Design And Development of the Biped Prototype Robian," Proceedings of the 2002 IEEE International Conference on Robotics & Automation, Wasington,DC; May 2002.

F Pfeiffer, K Loeffer, M Gienger, "The Concept of Jogging JOHNNIE," Proceedings of the 2002

IEEE international conference on robotics & Automation, Washington, DC, May 2002

S Kajita, F Kanehiro, K Kaneko, K Fujiwara, K Harada, K Yokoi and H Hirukkawa, "Biped Walking Patern Generation By Using Preview Control of Zero-Moment Point," Proceedings of the 2003 IEEE International Conference on Robotics &Automation, Taipei, Taiwan, September 14-19,2003

R Stojic, C. Chevallereau, "On the Stability of Biped with Point Foot-Ground Contact," Proceedings of The 2000 IEEE International Conference on Robotics &Automation, San Francisco, CA, April 2000

Y Ogura, "Stretch Walking Pattern Generation for a Biped Humanoid Robot," Proceedings of 2003 International Conference in Intelligent Robots and Systems, Las Vegas, Nevada, October 2003.

M. Vukobratovic, "Biped Locomotion: Dynamics,Stability,Control And Application," Spring Verlag, Berlin,1990.

M Vukobratovic, B Borovac, "Zero-Moment Point — Thirty Five Years of Its Life," International Journal of Humanoid Robotics Vol. 1, No. 1 (2004), pp. 157–173

K Yoneda, S Hirose, "Tumble Stabilit Criterion of Integrated Locomotion and Manipulation," Proceedings of IEEE International Conference on Intelligent Robot and Systems, 1996, pp. 870-876

A Gowami, "Postural Stability of Biped Robots and the Foot-Rotation Indicator (FRI) Point," The International Journal of Robotics Research, vol.18, no. 6, June 1999, pp. 523-533

K Harada, S Kajita, K Kaneko and H Hirukawa, "Pushing Manipulation by Humanoid considering Two-kinds of ZMPs," Proceedings of the 2003 IEEE international conference on robotics & Automation, Taipei, Taiwan, September 14-19, 2003, pp. 1627-1632

A Goswami, V. Kallen, "Rate of changes of angular momentum and balance maintenance of biped robots," Proceedings of the 2004 IEEE international conference on robotics & Automation, New Orleans, LA, April 2004,pp. 3785-3790

S Kajita, F Kanehiro, K Kaneko etc., "Resolved Momentum Control: Humanoid Motion Planning Based on the Linear and Angular Momentum," Proceedings of IEEE International Conference on Intelligent Robot and Systems, Las Vegas, Nevada, October 2003, pp.1644-1650

Rüdiger Dillmann, Regine Becher and Peter Steinhaus, ARMAR II — A Learning and Cooperative Multimodal Humanoid Robot System, International Journal of Humanoid Robotics Vol. 1, No. 1 (2004) 143–155

Chenbo Yin□Qingmin Zhou and Le Xiao, walking stability of a humanoid robot based on fictitious zero-moment point, control, Automation, Robotics and Vision, 2006. ICARCV'06.9th International conference on, 5-8 Dec. 2006 On page(s): 1-6

Chenbo Yin, Albert Albers, Jens Ottnad and Pascal Häußler, stability maintenance of a humanoid robot under disturbance with fictitious zero-moment point, Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on Volume, Issue, 2-6 Aug. 2005 Page(s): 3149 - 3156

John J.Craig, Introduction to robotics: mechanics and control, Addison-Wesley Publishing

Company, 1989, 19-36.
Ming Tan, De Xu and Zengguang Hou ect., Advanced robot control(in Chinese), High education press, 2007.5, 20-56.