# BIPED ROBOTS: FROM INVERTED PENDULUM TO PROGRAMMING 12DOF DANCING POSTURES

Nevena Ackovska
Institute of Informatics, Faculty of Natural Sciences and Mathematics
Skopje, Macedonia

Stevo Bozinovski
Department of Mathematics and Computer Science,
South Carolina State University
Orangeburg, SC, USA

ABSTRACT

This paper points some aspects of biped robots, starting with the basic mechanical metaphor, the inverted pendulum system. We give the strategy of biped walking related to inverted pendulum. The second part of the paper describes our laboratory work with biped robots resembling humanoids, exhibiting some dancing poses. A 12 DOF biped is used. We propose a pseudocode of a general programming routine for a biped robot.

## I. INTRODUCTION

We tend to think that robotics is science that supports the notion of movement. If a robot does not move, it might be considered as a kind of Artificial Intelligence agent, but motion is the ultimate expected action of a robot. There are various aspects of robotics that are taken into consideration, with purpose of meaningful motion. Figure 1 [8] represents this point of view.

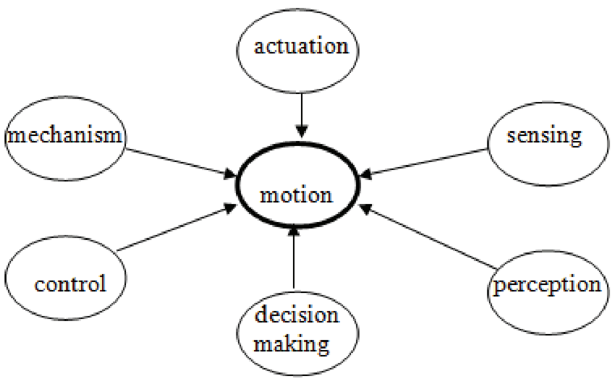



Figure 1: Motion centered robotics

The earlier stages of robotics were mostly considering robotics arms. In contemporary robotics due to the interest in humanoid robots, as well as other walking machines such as hexapods, robotic legs have been extensively studied, built, and software has been developed for their control. One kind of walking robot is a biped.

## II. INVERTED PENDULUM AS MODEL OF A BIPED ROBOT

Biped robot is an inherently unstable device and needs control at any time to maintain its upright posture. The simplest model of a biped dynamics is the inverted (or inverse) pendulum model. Early biped robots indeed used inverted pendulum to maintain stability of the walk [7]. The inverted pendulum model is shown in Figure 2.

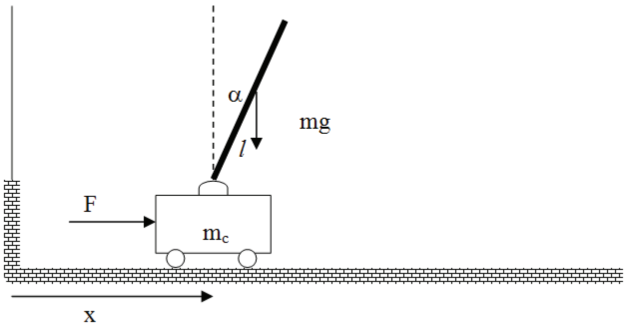



Figure 2. Inverted pendulum model

The system can be modeled with pair of differential equations (adopted from [6]):

$$\dot{\omega} = \frac{F + \mu_c \mathrm{sign}(v) + ml\omega^2 \sin\theta - M(mgl\sin\theta - \mu\omega)/(ml\cos\theta)}{l(m\cos\theta - 4M/(3\cos\theta))}$$

$$\dot{v} = \frac{F + \mu_c \mathrm{sign}(v) + ml\omega^2 \sin\theta - (3/4)(mg\sin\theta - \mu\omega/l)\cos\theta}{M - (3/4)m\cos^2\theta}$$

where $M$ is the total mass of the system (cart+pole), $m$ is mass of the pole, $g$ is acceleration due to gravity, $l$ is distance between the pivot and pole's center of mass, $\dot{\omega}$ is angular acceleration of the pole, $\dot{v}$ is linear acceleration of the cart, and $F$ is the applied force generated by the controller. The friction coefficients (between the cart and the ground $\mu_c$ , and pole and pivot, $\mu_c$) in some simulations and control tasks can be omitted.

The control of inverted pendulum has been studied extensively. Our previous work on the subject [2, 3, 4] has been related to learning a strategy (policy) how to control. The control strategy found is to focus on sign of angle and angular velocity. The control strategy found is

if sign(θ)sign(ω) >0 then sign(F)=sign(θ)
else do nothing

This strategy would keep the pendulum near (θ, ω) = (0,0).

A walking robot can be modeled several ways and one such model is shown in Figure 3 [5], [1]. From Figure 3 we can see that a simple biped robot can be modeled as a 5 degree of freedom (5DOF) kinematics system. It has 1 motor moving the upper body, and 2 motors for each leg. The center of mass (CM) of the robot is controlled by the movement of the body.
.

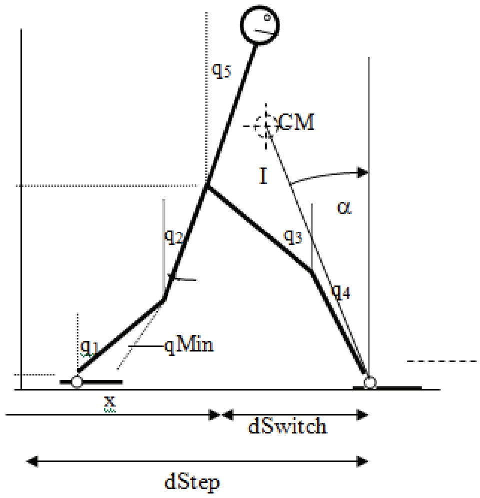


Figure 3. Model of a biped walking robot

The terminology of biped robot control is based on the concept of two phases of a walk: the single support phase and a double support phase [1]. During double support phase the robot is controllable, which is usually not the case of the single support phase (due to the lack of an actuated foot). During the double support phase there is a transition between one leg to another. The single support phase starts when the robot is about to lift one of its foot, especially the rear one. This is so called take-off configuration $C_{TO}$, which is modeled with three parameters: dStep, dSwitch, and qMin. In the configuration $C_{TO}$ we can consider the whole biped robot as an inverted pendulum The representing inverted pendulum is determined by calculating the center of gravity CM, the inertia momentum I of the whole system, and the pendulum angle α. A global control strategy for a biped robot needs to transform the initial configuration $C_0$ into the take off configuration $C_{T0}$ In the $C_{T0}$ configuration the robot will starting behaving as inverted pendulum, so the inverted pendulum dynamics can be used to control the swing leg movement.

The objective is to swing the leg forward in coordination with the body movement. To achieve the objective the walking parameters should be carefully chosen. If dStep is too long, in would produce unfavorable positions for actuators and the system might show undesirable oscillation. If dStep is too short it would produce slow locomotion. The angle qMin should be maintained in a value different than zero to avoid singularity in calculations. This parameter is directly related to the maximal leg extension which is reached by the rear leg in the take-off configuration. The starting angle of the inverted pendulum is directly related to the parameter dSwitch: the higher value for dSwith the longer path of the inverted pendulum, thus a higher initial velocity, and the higher actuator power is needed. So the motors should be chosen to be powerful enough to carry out the robot in the inverted pendulum, one legged phase.

The control principle of walking is can be viewed as inverted pendulum control: Obtain a single support walking phase in which the inverted pendulum moves clockwise i.e. obtain moving the robot along positive x-axis. To achieve that the initial angular velocity of the pendulum must be greater than the minimum angular velocity obtained by conservation energy principle

$$\frac{(ml^2 + I)\dot{\alpha}_0^{\,2}}{2} = mgl(1 - \cos\alpha_0)$$

where from

$$\overset{*}{\alpha}_0 = \sqrt{\frac{2mgl(1-\cos\alpha_0)}{ml^2 + I}}$$

where $m$, $l$ and $I$ are pendulum mass, length, and inertia momentum respectively, $g$ is the gravity acceleration. Once we have the angular velocity of the whole walking system, we can compute the velocities of each actuator using inverse kinematics.

## III. Leg Kinematicss

Biped model in Figure 3 is a basic mathematical model and does not include third dimension, the distance between the heap motors. To assure stability, a leg has at least three actuators, two at the heap and one at the knee. One of the heap motors assures movement forward and backward, and one assures side step, for stability. So a biped has usually 9 motors. Figure 4 shows the kinematics modeling.

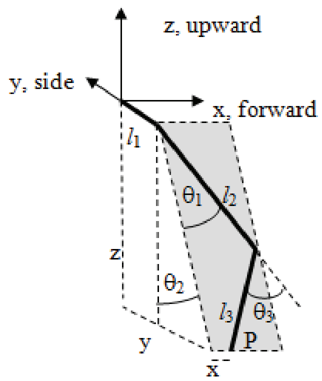


Figure 4: Kinematics of a leg

Let x, y, z define the position of the paw P. Direct kinematics $(x, y, z) = f(\theta_1, \theta_2, \theta_3)$ is modeled [1] as

$$x = l_3\sin\theta_3\cos\theta_1 + (l_2 + l_3\cos\theta_3)\cos\theta_2\sin\theta_1$$
$$y = l_1 + (l_2 + l_3\cos\theta_3)\sin\theta_2$$
$$z = l_3\sin\theta_3\sin\theta_1 + (l_2 + l_3\cos\theta_3)\cos\theta_2\cos\theta_1$$

where $l_1$, $l_2$, $l_3$ is length of the leg shoulder, upper limb, and lower limb respectively, and $\theta_1$, $\theta_2$, $\theta_3$ are actuator coordinates respectively.

The inverse kinematics $(\theta_1, \theta_2, \theta_3) = g(x, y, z)$ can be obtained as

$$\theta_3 = \pm \arccos\left(\frac{x^2 + z^2 + (y - l_1)^2 - (l_2^2 + l_3^2)}{2 l_2 l_3}\right)$$

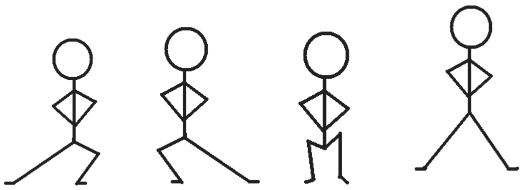
$$\theta_2 = \arcsin\left(\frac{y - l_1}{l_2 + l_3 \cos\theta_3}\right)$$

$$\theta_1 = \arctan\left(\frac{x}{-z}\right) + \arccos\left(\frac{(l_2 + l_3 \cos\theta_3)\cos\theta_2}{\sqrt{x^2 + z^2}}\right)$$

Note that there are two solutions for any reachable position (x, y, z) depending on the position of the knee actuator, being positive or negative.

## IV. NON-WALKING POSTURES OF A BIPED ROBOT

The leg kinematics is needed also when we program bipeds to execute non-walking routines, such as dancing, yoga, or posing routines. Some poses are given on Fig. 5.



A B C D

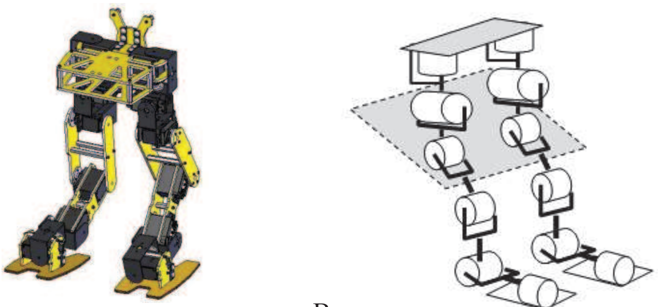
Figure 5. Some poses that do not require walking foot routine

In our lab work we focused on non-walking movements as shown in Figure 5.

## V. WORK WITH OUR LAB BIPED ROBOT

Here we give some details of our lab work. We have worked with Scout type biped, which is a 12 motors biped and resembles a humanoid robot (Figure 6A).

.



A B

Figure 6. Our lab biped robot

Each leg has 6 motors. Note that in the posture shown, the knees are positioned backwards. The robot carries its batteries in a compartment in front, so backward legs offer better stability. Figure 6B shows the motor placement and mutual orientation, so that the leg joints are defined. Here the robot is shown in its "sitting" position.

### A. *The motors*

The motors are crucial for bipod robot design. Low-weight and high-torque motors are needed for a work with bipeds and humanoids. The most important motors are the foot motors. We use the servo motors HS5645MG. It is a 3 pole ferrite motor, digitally controlled. It can be run by voltage range 4.8-6V, with speed 0.23sec/60deg and toque 10.3kg.cm at 4.8V and speed 0.16sec/60deg and torque 12.1kg.cm at 6V. It has weight of 60gr and dimensions 4.06x1.98x3.78cm. It is controlled by a pulse width control, positioning the motor in neutral position with pulse width 1.5 sec.

#### 1) *Servo motor control*

The servo motors are controlled in a closed loop. Classically the loop was part of the main program. Now there are separate servo controllers that allow the main program to control the motors as if the control is open-loop. The main computer (such as PC) sends a control sequence to a servo controller which then executes closed loop control to motors. In our case we use the SS32 servo controller that allows control of up to 32 servo motors. In Table 1 we give an example of assigning motors a communication channel (pin) of the servo controller.

Table 1. Assigning communication pin to motors

```
Left leg:  00: foot rotate, 01: ankle Y,
02: knee Y, 03: hip Y,04: hip X; hip rotate
Right leg: 16: foot rotate, 17: ankle Y;
18: knee Y, 19: hip Y, 20: hip X, hip rotate.
```

In the above table Y is side-to-side (lean) control, and X is front to back (walk) control.

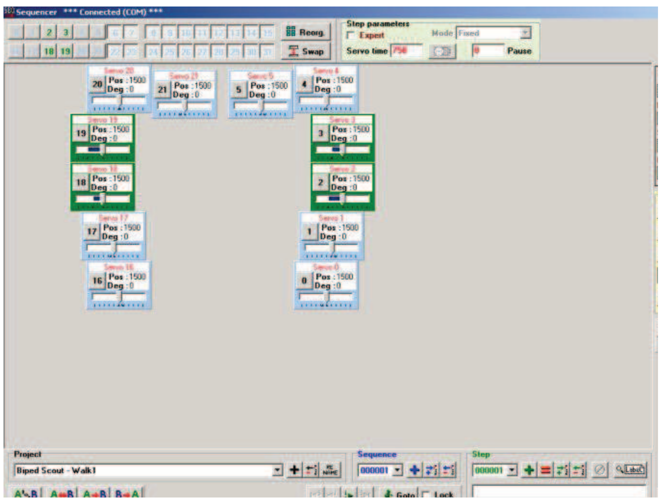


Figure 7. The software we use for biped motor adjustment

### *B. Visual programming software*

The positions of the motors should be figured out before the command is written. Usually it is a trial and error procedure of adjusting motors. There are visual programming tools that allow optimizing the position of the motors. We used the SS32 Sequencer software as tool for programming SS32 controller. Figure 7 shows an example of Sequencer visual programming.

## VI. A GENERAL ROUTINE FOR A 12DOF BIPED ROBOT

On the basis of our experience, here we give a pseudocode of a program that is rather general for biped robots. It can execute a walking routine, but also other routines. We use Basic like language to show the pseudocoide, given in Table 2.

Table 2. Walking routine pseudocode

```
Enable 12 servos: enableservo = 1111111100001111;
p0-p4, p8-p15
X – front to back (stride) ; Y - side to side
(lean) , Z - lift leg
Initial values of variables:
        Right leg: R_L,  R_X,  R_Y,  R_Z,  R_A
        Left: leg:  L_L,   L_X,  L_Y,  L_Z,  L_A
gosub Update

Main:
gosub Update several times for each leg with set of
variables: delay, X, Y, Z.
define
        motor position constrains, for example wzmax
= 5.5
        controller pins: for example RHR = p15;
right hip rotate
        motor centers, example RHRC = 132;  0  =
right , 255 = left
        position variables: for example RHRpos
        old position variables : for example RHRopos
        leg speed variables: for example RHRspeed
        error byte: for example RHLlow = Rerror.bit0
        motor variables: for example mH, mT0

Subroutine Update
compute motor values, for example : mX = f(R_X,
constrains)
gosub Calcpos
compute motor positions, example RHSpos = g(RHSC,
mT0, constrains)
compute motor values, example : mX = f(R_X,
constrains)
gosub Calcpos
set error byte to 00000000
If motor positions out of range then rise flag in
error byte
If motor positions in range
        define variable speedmod = delay/256
        motors speeds, e.g.: RHRspeed =
abs(RHRoposRHRpos)/speedmod
        gosub Updatepos
        move legs
        for example  hservo [RHR\RHRpos\RHRspeed  ;
right hip rotate]
        pause delay
endif
return

subroutine Updatepos
replace old positions with new ones: for example
RHRopos = RHRpos
return

subroutine Calcpos (actually inverse kinematics)
return
```

From the Table 2 we can observe that the software first defines the robot's initial position, and then defines constrains of the motors, followed by the update values for motors, and finally executes the command.

## VII. CONCLUSION

Biped robots are special class of inherently instable robots and could be modeled using the well known inverted pendulum model. Modern bipeds resemble humanoids and have often 12 motors. Here we have shown some of our work in programming non-walking sequence of postures (dancing). We also proposed a general purpose routine for biped robots, if form of a pseudocode.

## REFERENCES

[1] Appin Knowledge Solution, *Robotics*, Infinity Science Press, 2007

[2] S.-Bozinovski, "Inverted pendulum learning control" Adaptive Networks Group Memo, December 10, 1981, Computer and Information Science (COINS) Department, University of Massachusetts, Amherst, 1981

[3] S. Bozinovski. *Consequence Driven Systems: Teaching, Learning, and Self-Learning Agents,* Gocmar Press., Athol, MA, 1995

[4] S. Bozinovski S. "Crossbar Adaptive Array: The first connectionist network that solved the delayed reinforcement learning problem" in A. Dobnikar, N. Steele, D. Pearson, R. Albert (Eds.) Artificial Neural Networks and Genetic Algorithms, Springer Verlag, p. 320-325, 1999

[5] S. Bozinovski, *Robotics and Intelligent Manufacturing Systems*, (In Macedonian) Gocmar Press, 1997

[6] R. Cannon, *Dynamics of Physical Systems,* McGraw-Hill, 1967

[7] K. Kato, A. Takanishi, H. Jishikawa, I.. Kato, "The realization of quasi-dynamic walking by the biped walking machine". In A. Morecki, G. Bianchi, K. Kedzior (eds.) Proc Symp on Theory and Practice of Robots and Manipulators, p. 341-351, Warsaw, Polish Scientific, 1983

[8] M. Xie. *Fundamentals of Robotics: Linking perception to Action.* World Scientific, 2003