An Equilibrium Point based Model Unifying Movement Control in Humanoids

Xue Gu
Computer Science Department
University of Rochester
Rochester, NY 14627-0226, USA
Email: xgu@cs.rochester.edu

Dana H. Ballard
Department of Computer Science
University of Texas at Austin
Austin, TX 78712-0223, USA
Email: dana@cs.utexas.edu

***Abstract*— Despite all the dynamics methods effectively used in robotics control, few tackle the intricacies of the human musculoskeletal system itself. During movements, a huge amount of energy can be stored passively in the biomechanics of the muscle system. Controlling such a system in a way that takes advantage of the stored energy has lead to the Equilibrium-point hypothesis (EPH). In this paper, we propose a two-phase model based on the EPH. Our model is simple and general enough to be extended to various motions over all body parts. In the first phase, gradient descent is used to obtain one kinematics endpoint in joint space, given a task in Cartesian space. In the second phase where the movements are actually executed, we use damped springs to simulate muscles to drive the limb joints. The model is demonstrated by a humanoid doing walking, reaching, and grasping.**

I. Introduction

Humans and other primates can easily perform a wide variety of tasks without much knowledge about themselves and the environment. This contrasts with the current state of robotics: even for a robot to reach to a position with natural poses can be a research topic, much less for the robot to be as dexterous and intelligent as humans. The current state of the art for humanoids is not much of an improvement. For example, our research uses a virtual humanoid from Boston Dynamics Inc. Although the virtual guy can perform a repository of motions, it is not an adaptive and intelligent agent. The reason lies in the fact that the virtual guy is simply playing back the motion data captured from humans. When it encounters new environments or tasks, it does not have the ability to plan new movements. Even within the method of playing back captured motion data, mismatch and unrealistic movements are highly possible due to various reasons as sensor errors, calibration errors, and other metric difference between the virtual model and real humans.

The issue of motor control is also unresolved in human biology. One of the central questions of studying human movements is how the Central Nervous System (CNS) calculates the motor commands to drive the limb. One proposal, derived from robotics, is that the brain computes inverse dynamics solutions. In movement control, the task is usually described in Cartesian space, which is different from the actual space where the motor commands are executed. Therefore, a proper coordinate transformation is required to find the solution in the joint space given a task in Cartesian space, which is well known as inverse dynamics. This problem turns out to be quite difficult, because the musculoskeletal system typically has many more degrees of freedom (DOFs) than the task constraints at hand. Among the inverse dynamics methods, one approach is to study movement control as a formal optimization problem as exemplified in [18] [13] [22]. Some researchers tried to solve the same problem by adding constraints to the redundancy as in [21] [25] [19]. Most of those approaches can only be applied to simple robots with known geometry and in static environments. Few models are eligible to be used in robot systems as complicated as humans in dynamic environments. The inverse dynamics calculation for an anthropomorphic robot with more than 30 DOFs requires extremely high computation.

Contrary to the inverse dynamics force control model, Equilibrium Point Hypothesis is another theoretical framework used by a lot of researchers in human motor control. Feldman [6] [7] pioneered the EPH that limb movements could be achieved by shifting the limb postures represented as equilibrium points from one position to another. Researchers put forward the theory and proposed many more ’dialects’ of EPH [3] [4] [10]. The central idea of EPH spring models discriminates movement planning from execution. Motor planning is to program the movement tasks by choosing a succession of discrete equilibrium points(EPs). Once these points are chosen, in the execution phase the muscle spring system moves without further direction under CNS control.

Whether it is EPH or inverse dynamics that really controls human movements is a subject of controversy. Many researchers argue against the EPH by providing experimental evidence [12] [16]. Feldman and other researchers defended the EPH in various reference [5] [8] [9]. With all those debates, most of EPH researchers’ attention has been attracted towards proving the validity of the theory. Little research has been directed to study how humans choose those EPs for a given task. Less work is devoted to demonstrate how the simple EPH mechanism can be applied to control human motions. In this paper, we propose a two-phase control model based on the idea of the EPH. Given a task in Cartesian space, we first develop a motor simulation model to plan the EPs in joint space, which specifically addresses how the EPs are calculated to achieve a particular motor goal. During the movement execution, damped springs are used to simulate muscles to actually drive the movements. We demonstrate that the model is a general model that can unify the control of various motions, such as reaching, walking and hand movements.

The next section describes the details of the model in the context of a simple reaching task. Section III demonstrate the humanoid doing a diverse of complex motions. Finally we conclude the paper and discuss avenues for future work.

II. Model Description

Our model suggests that human movements can be planned in segments, and each segment has an equilibrium endpoint in joint configuration. Before the movements are initiated, the endpoint is calculated using the motor planning model elaborated below, and then used to set muscle lengths, modeled as damped springs' natural lengths, for movement execution. During movement planning, the lest amount of necessary EPs is calculated for a motor task. For example, in simple voluntary arm movements, only final EP is probably required. But in more complicated movements as obstacle avoidance, more than one EPs are necessary. Movements are generated by gradually shifting from one segment EP to the next.

A. Motor planning

Given a task in Cartesian space, the first step of our model is to get a kinematics solution in joint space. To do so, a recent suggestion has been to steer to the end point using gradient descent of an objective function that expresses variation of the distance between the current handtip position to the destination [23]. Although we were able to replicate their results, in our experience this method is delicate and very sensitive to its various parameter settings. Besides, their pure gradient descent method can have unrealistic trajectories as shown in Fig. 2. However, it serves as the starting point for our method which uses the gradient method, not to actually control the movement execution, but instead in simulation to generate end configurations in joint space that are then used by subsequent processing stages.

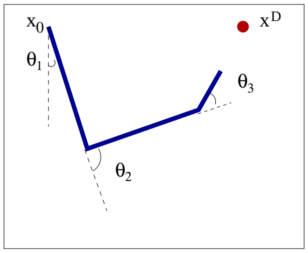

Fig. 1. Reaching task in a 2D space with a 3 DOF system.

Consider a three joint limb moving in a vertical plane as shown in Fig. 1. Let X, the task space, be the set of points that can be reached by the arm in the plane. Let Θ, the configuration space or joint space, be a subset of R^3, and it specifies all the possible postures. There exists a vector function f, which maps Θ onto X, i.e., every hand location $x \in X$ can be written as $f(\theta)$ for at least one $\theta \in \Theta$, and every θ maps to a certain $x \in X$. The objective function is defined as the distance from current end effector position $f(\theta)$ to the destination x^D as in Equ. (1).

$$r(\theta, x^D) = \sqrt{\sum_{i=1}^{2} (x_i^D - f_i(\theta))^2} \tag{1}$$

Bringing the hand to the destination consists of repeatedly computing gradient of the objective function and changing the posture a small amount for each step until the hand reaches the destination. The gradient function in Equ. (2) is a vector, and each component specifies how much to change each joint value to bring the hand closer to the target. β is a scaling factor to adjust the amount of changes of each time step. $J(f(\theta))$ is the Jacobian matrix.

$$d\theta = -\beta \frac{(x - f(\theta)) \times J(f(\theta))}{\sqrt{\sum_{i=1}^{2} (x_i^D - f_i(\theta))^2}} \tag{2}$$

We regenerate the results of [23] in the left subfigure of Fig. 2. The connected lines are the postures obtained by doing gradient descent to the objective function for each step, and the green curve gives the trajectory generated. There are at least two factors that make their method nonbiological. First, the trajectory goes through a big curve and follows a line going back to the destination, which is not what we observe in human reaching movements. Second, the end posture shown in bold blue lines does not fit with the human posture in the same task.

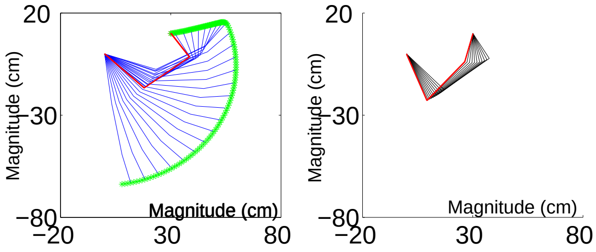

Fig. 2. End posture planning. A) left: gradients of the distance function for each time step. B) right: energy optimization steps to get the realistic end point.

To solve the final posture problem, we take the idea of energy minimization. The same idea has been studied by [1] [17], but different from the use of energy in our method. We minimize the potential energy during the planning phase, instead of the kinetic energy which will not be available until the movements begin. We assume that the potential energy is at least proportional to the energy required for the movements, even if it does not represent the exact energy consumption during the execution. We define another cost function given in Equ. (3) using Lagrange multiplier method for the constraint optimization problem. The goal is to minimize the first part, which is the potential energy, while keeping the second part, which constraints the end effector to the destination, to be zero. The right figure of Fig. 2 illustrates how the joint configurations gradually changing from the unrealistic one to the final biological end posture shown in red bold lines.

$$E(\theta) = Mgh(\theta) + \lambda(f(\theta) - x^D) \tag{3}$$

where, M is the total mass of the limb. $h(\theta)$ is the height of the limb's center of gravity (COG), and it is a function of the current limb configuration θ. g is the gravity parameter. The term of $Mgh(\theta)$ is the potential energy of the limb system. $f(\theta)$ is the current end effector position, and x^D is the destination. λ is the parameter for the constraint function in Lagrange multiplier method [2].

B. Spring model

Human movements are driven by the flexion and extension of muscles. Among the models to describe muscle properties, one widely-used model is proposed by Hill [14] [15]. A muscle produces two kinds of forces, elastic and viscous. The sum of those two forces composes the muscle's total force. The passive element of a muscle has an elastic property and can be simply modeled as a spring. The elastic force varies directly with the distance that is enlonged from its resting length. The formula is given in Equ. (4):

$$F_e = K \times (l - l_r) \tag{4}$$

where, F_e is the elastic force, l_r and l are the spring's resting length and actual length, respectively. K is called the spring constant or stiffness.

The molecular structure of a muscle causes it to display a property of viscosity, that is, the resistance to move. This resistive force is like a shock absorber [20]. If you push the piston sitting over a fluid in an encapsulated bucket, the shock absorber will resist by a tension of F_v that depends on the viscosity B of the fluid. The faster you push the piston, the stronger resistant force the fluid generates. The relation between the speed of pushing and the force can be written in Equ. (5)

$$F_v = B \times \dot{l} \tag{5}$$

where, $\dot{l}$ is the velocity of the spring length. Combining the two properties, the force produced by an active muscle is a function of both the muscle length and its rate of changing as represented in Equ. (6).

$$F = K \times (l - l_r) + B \times \dot{l} \tag{6}$$

where K and B are the spring stiffness and viscosity parameters respectively.

C. Movement execution

The dynamics system of the limb system in Fig. 1 can be usually expressed in Equ. (7).

$$\tau = I(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + G(\theta) \tag{7}$$

where τ is a 3×1 vector of torques applied to shoulder, elbow and wrist, $I(\theta)$ is a 3×3 matrix representing the kinetic energy, $C(\theta, \dot{\theta})$ is a 3×3 matrix of centrifugal and Coriolis effects, $G(\theta)$ is a 3×1 vector of gravitation. θ, $\dot{\theta}$, and $\ddot{\theta}$ are vectors of joint values, velocities and accelerations.

In inverse dynamics methods, for each time step, certain torques are computed to drive the limb to move along a planned trajectory represented as a series of joint values, velocities and accelerations. This involves a considerable amount of computation even in a simple 3 DOF system. In our model, the joint space solution obtained during the planning phase allows the ready determination of the spring natural lengths. As far as the spring length is not equal to the natural length, forces are generated to attract the joint configurations to the end point. Comparing to the inverse dynamics methods, springs take charge of the movement control once the natural lengths are set and generate proper torques to drive the limb to the destination.

We design the spring-muscle setting based on the human musculoskeletal system. Human muscles usually attach to the two segments of a joint to control the flexion or extension of the joint, and several muscles are used to control one degree of freedom in human. Instead of using all the biological muscles, we use an abstraction of the actual muscle system. One simplified spring setting is attached to the middle of the segments, and each spring controls both the flexion and extension for one degree of freedom. Fig. 3 shows the spring-muscle setting for a three joint limb moving in a 2D space. Each spring functions as in Equ. (6), where two parameters K and B can be used to adjust the movement performance. Other placements of the springs will also be feasible as far as all the degrees of freedom is controlled by at least one spring, and won't affect the movement performance. Spring placements can also follow the human anatomical muscle settings, and it will involve more computations in determining the spring natural lengths for the planned EPs. With other spring-muscle placements, the springs will have different natural lengths corresponding to the same EP and the parameters of K and B change as well to generate the same movements.

Our simplified spring-muscle model captures the main scope of movements generated by the human's biological muscle system. It can be viewed as the first step to functionally study the human movement control scheme. Once we solve how the simplified spring-muscle model is used to control various movements, how the biological muscles are combined to achieve the same movements as if only the abstract spring-muscle system is in control can be studied as a subproblem. In human system, the motor commands the brain directs to the motor system can be more complicated than the abstract spring-muscle settings of our model. Even if this is the case, we cannot deny that the brain might plan the movements in a higher and more abstract level. Then, the brain or spinal cord maps the functional patterns (abstract muscles) to the activation patterns of actual muscles.

The reaching movement is activated by setting the spring natural lengths corresponding to the EP. To avoid sudden

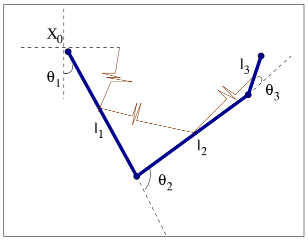

Fig. 3. Three damped springs controlling a 3 DOF system

movements when changing the spring natural lengths from the initial values l_{init} to the end value l_{EP}, we take several steps to gradually do the adjustments as in Equ. (8).

$$\begin{aligned} l(t) = l_{init} + \epsilon \times t \quad t = 1, 2, \cdots, T \\ \epsilon = \frac{(l_{EP} - l_{init})}{T} \end{aligned} \tag{8}$$

where, l_{init} and l_{EP} are the spring initial and end point natural lengths. $l(t)$ is the spring natural lengths setting at time step t. ϵ is the amount of changes applied to the spring natural lengths for each time step. T is the number of steps to gradually change the spring natural lengths setting from initial one to the EP. Usually, T is set to be shorter than the movement time.

III. Humanoid Simulation

We build a 33 DOF humanoid as in Fig. 4. Each leg has three joints: hip, knee and ankle, and each joint has one DOF. Each arm has six DOFs, three DOFs for the shoulder, one for the elbow and the other two for the wrist. Torso has one DOF for leaning forward or backward. Currently, the head is not involved in the movement control. The humanoid has one hand with 14 DOFs.

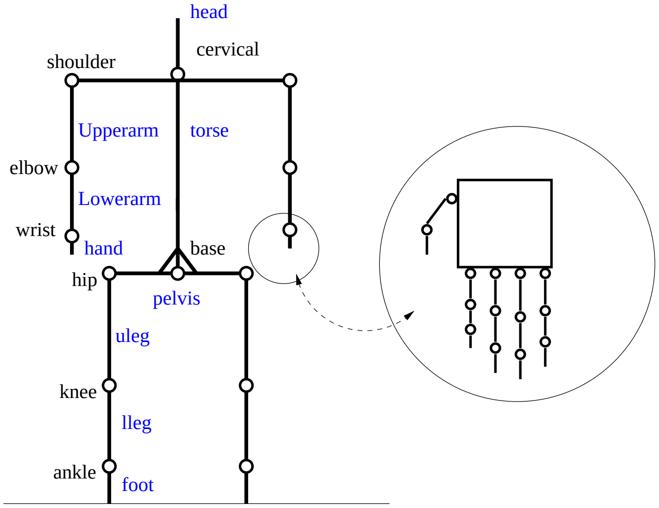

Fig. 4. Humanoid skeleton structure.

A. Reaching

This part shows the simulation results of the illustration example of 3 DOF limb reaching in a 2D space. The limb configuration and spring setting is given in Fig. 3. Fig. 5 shows four snapshots of the arm moving from the initial position to the destination. In the simulation, we use constant spring stiffness K and viscosity B during the process of reaching. Different sets of parameters will generate different reaching movements. Currently we manually adjust the K and B to simulate the realistic reaching. To further study how close our simulation is to human reaching, we captured some human movement data in the same task and compare them to the simulation results. Fig. 7 presents the reaching trajectory and speed profiles comparison. Solid lines are the captured human data, and the dashed lines are the model prediction. We get fairly good fitting of the two, and the model generates the tuning curves in speed profiles typically observed in human reaching movements.

Reaching with avoidance is always an important topic of research. In our model, reaching with obstacle avoidance for simple cases can be easily implemented by adding one more EP during planning. In the planar movements as in Fig. 6, the 3D via point is chosen beyond certain distance from the obstacle. Motor planning algorithm is used to calculate the EP in joint space with respect to the via point. We use a simple planar obstacle avoidance example to illustrate the flexibility of our model. For complex movements in 3D space, this algorithm does not guarantee to avoid the obstacle, because the other part of the arm might collide with the obstacle. Further study to apply the model to more complicate obstacle avoidance instances is on our future work list. Then movements are achieved by setting the two EPs consecutively. That is, for each time step, we monitor the difference between the current posture to the desired EP. When the current limb posture gets close enough to the first EP, the spring natural lengths corresponding to the next EP are set. The limb continues the movements from the first EP till the actual posture arrives at the second EP. This control mechanism works for movements involving multiple EPs.

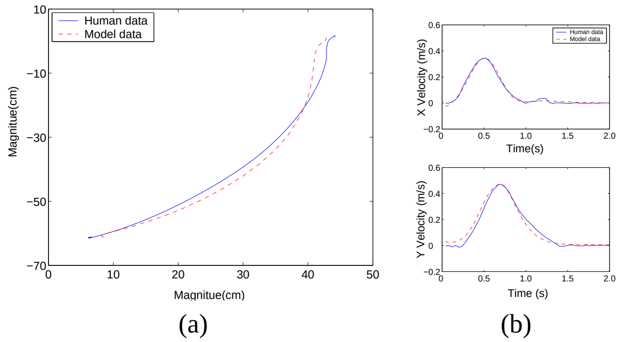

Fig. 7. Model simulation and human movement comparison. (a) end effector trajectory. (b) speed profiles.

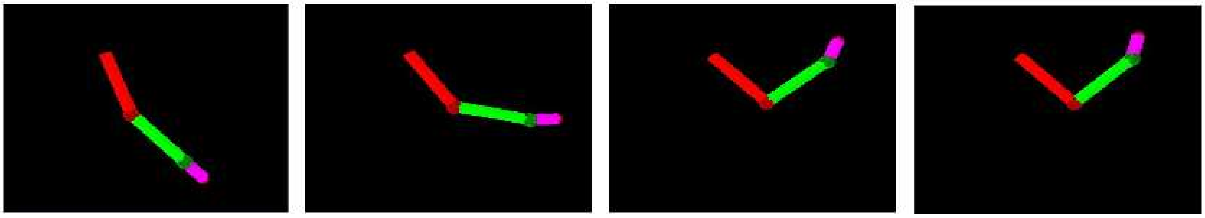

Fig. 5. Reaching movement executed.

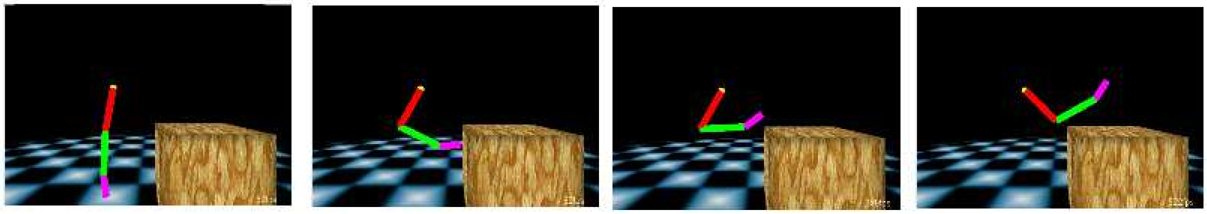

Fig. 6. Reaching movement while avoiding the desk.

B. Walking

Walking is an extensively studied research topic, and various algorithms and models are available. However, most of those models are specific to control only walking. This part explains a preliminary way to extend the general EP model to control walking.

In our model, walking can be grouped into two categories. The first one is regular walking on an even floor without turning or obstacles. This kind of walking scheme is quite consistent and does not require motor planning to calculate the specific position to put the foot on. Humans control the movements within the body coordinates. The second category involves walking with respect to goals or objects extrinsic to human body, such as obstacle avoidance, turning, stepping, walking on uneven floor etc. This kind of walking requires the motor planning phase to calculate the leg joint configurations corresponding to the Cartesian space goals. Actually the first category of walking can be viewed as a specific case of the second one, as the regular walking always involves the planning for the same EPs so that humans remember those EPs and can directly apply them. The general way to choose those EPs for the second category of walking is the motor planning algorithm introduced in section II-A, given the humanoids knows where the next step should be. Taking the obstacle avoidance in walking as an example, there exist several ways to calculate the leg configurations according to different situations of the obstacle. If the obstacle is a small box, one way is to plan the front foot onto the obstacle to step over it. The other way is to define a via point above the obstacle and plan the leg to move over the obstacle to a position in front of it. If the obstacle is huge, the robot has to walk around it. The planning of this situation involves specifying the foot steps along certain path bypassing the obstacle.

So far, we have implemented the first category of walking. One cycle of walking can be represented by four EPs as in Fig. 8. Those four EPs are extracted after analyzing the human walking data from Boston Dynamics Inc. Other EPs slightly different than what we choose are also possible, since people also have different walking styles. We choose the same simplified spring-muscle setting as in the reaching simulation. Each leg has three damped springs to control the contraction and extension of the hip, knee and ankle. Natural and realistic walking is generated by setting the spring natural lengths from one EP to the next.

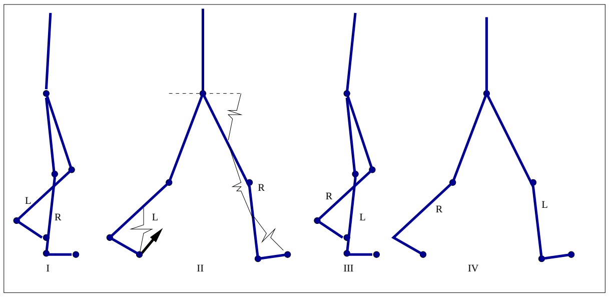

Fig. 8. Four EPs for one cycle of walking

During walking simulation, one technical difficulty is to keep the humanoid's balance along the walking direction. When the springs pull the robot's legs forward, the torso is always legged behind, since the body has no outsider force to provide the forward speed. The robot always ends up falling backwards. We solve this problem by instantly pushing the back foot on the ground. The counterforce from the ground will provide the body a forward speed. In the implementation, the pushing force is produced by setting the back ankle's spring natural length to a larger value for a short period, when the front foot gets contact with the ground. This way, the body's COG shifts from the back foot to the front. Another interesting

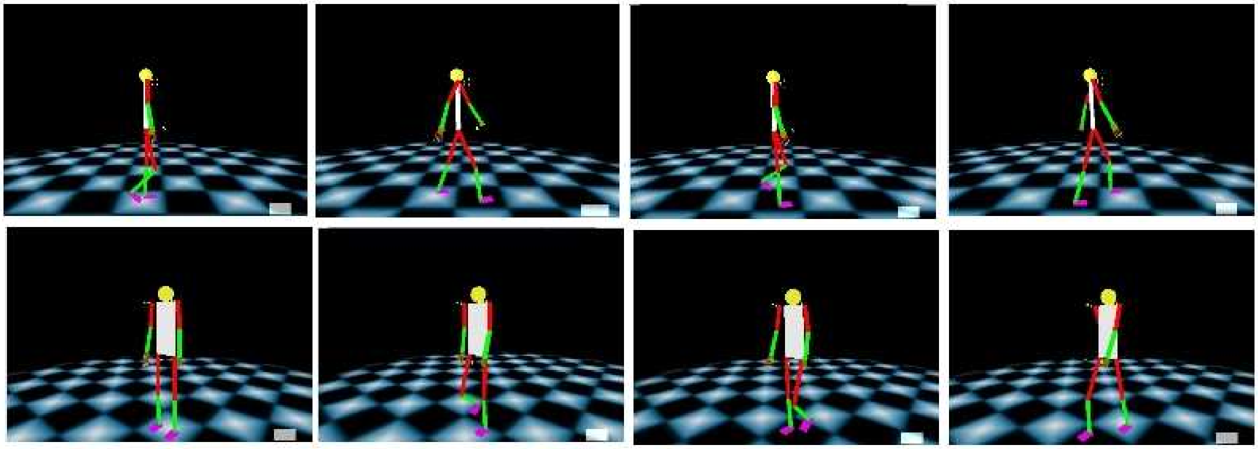

Fig. 9. Humanoid walking with coordinated whole body movements. The upper and lower figures show walking from different views.

phenomenon we discover is that when a larger force is set on the back foot so that two legs are pushed above the ground, the walking simulation can be turned into a running model. Even if we did not further study the details of turning the walking into running, it is obvious that this model is ready to control running motions. Fig. 9 shows the snapshots of the humanoid walking.

Till this point, we only implemented the first category of walking without disturbance from the outside, so that the balance is not a problem now. Walking on uneven floor, obstacle avoidance, turning, climbing steps etc. are our future planes. Those walking might introduce another challenging problem of dynamic balance. One possible solution is to redefine certain balance modules that can be activated under different situations. For example, small disturbance may be balanced by stronger spring stiffness. Sometimes a backward or forward stepping might be necessary to avoid falling down. When falling is not avoidable any more, the humanoids might want to minimize the hurt by hand reaction.

C. *Hand movements*

In this part, we demonstrate the model to control the hand grasping and manipulation of objects. We build a 14 DOF hand as in Fig. 4, where thumb has 2 DOFs and the rest of fingers each has 3 DOFs. Grasping based on the EP model is much easier than traditional robotics methods, where precise finger positions on the object need to be planned ahead of time. Fuentes [11] used different predefined grasping plans for different types of regular objects, and a genetic algorithm to plan the finger tip positions on an irregular object. In our case, grasping can be done by setting the finger EPs into the object, and the general motor planning algorithm is applied to calculate joint configuration. On one hand, this strategy can greatly simplify the planning computation compared to Fuentes' planning algorithms. On the other hand, it provides the necessary force and friction to manipulate the object when the finger tips get contact with the object. How much contact force to exert on the object can be adjusted by the stiffness K of the finger springs.

Fig. 10 shows the hand grasping a cylinder, and releasing it. This is a simple demonstration of the model applied to the hand doing grasping and manipulation. The hand is a simplified version of human hand. The goal of this part is to illustrate that our EP based model is general and simple enough to be applied to the hand control. To synthesize more complex hand and objects interactions, we need to build the hand closer to human hand by adding a heading rotation to each finger, and two more rotations to the thumb. Grasping and manipulation of sphere, cubic or other kinds of objects can be implemented easily based on this EP model. The grasping and manipulating of the cylinder is a simple demonstration which does not require much planning of the finger positions on the object but setting the EPs into the object. In other cases, more planning beyond setting the EPs inside the ojbect might be necessary. For example, if the hand grasps the cylinder from the top, fingers should evenly spread around the cylincer to get more stable manipulation. Comparing to the traditional precise finger position planning for rigid hand manipulation, the EP model gives more flexiblilty for the finger position planning because the spongy spring-muscle system is able to compromise some noise and errors during the manipulation.

IV. Conclusion and Future Work

A. *Conclusion*

We discuss the limitations of traditional inverse dynamics methods applying to an anthropomorphic robot to generate complex behaviors in a physical world. Based on equilibrium-point hypothesis, we develop a two-phase control model that is general enough to control a diversity of motions involving any parts of human body. Our major contribution is in presenting a novel human movement control model that can unify various both simple and complex human motions. Besides, our model are inspired from findings of human movement research, and can serve as a candidate to further study human movements.

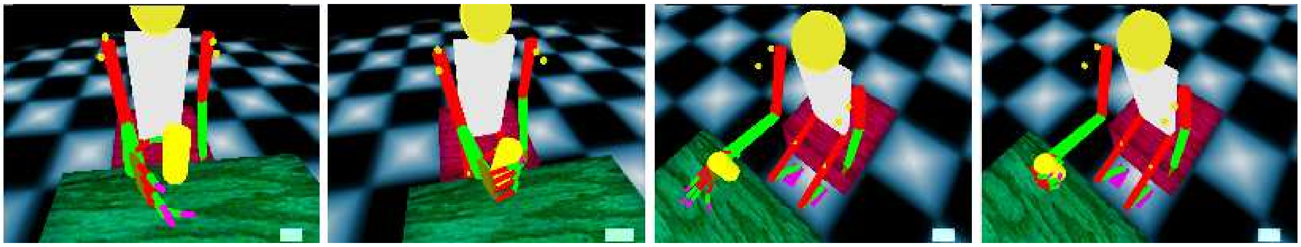

Fig. 10. Humanoid reaching and grasping the object.

Till this point, most of human control models are restricted to simple movements in 2D space within very small scope. To further study human motions, models that can accommodate large movements in 3D space is indispensable for a complete understanding of the human movement system. Thus our model can be viewed as a test-bed to human movement research, as well as a humanoid robot to demonstrate its capability.

From the aspect of both robotics and human movement modeling, one of the biggest advantage of the EP based model is their low computation requirement. For example, electromechanical systems that use direct servoing, such as ATR's SARCOS system, require rates as high as 10KHz to implement inverse control methods, while cortical neurons' signaling is typically in the range of 10 to 100 Hz. Thus any biologically plausible method of control should respect the low sampling rates available. The only computation of the EP model is to choose a succession of discrete EPs. Once these points are chosen, the muscle spring system moves without further guidance under the control.

B. Future work

Till this point, we only build a prototype of the control model, and scratch the surface of the research that can be done based on the model. With the graphics humanoid, we want to build motor movement units, such as protective stepping and arm reaction when falling down, different rising after a fall, balancing etc. Learning algorithms can be attempted to learn the mapping between the state space to the action control. Such algorithm can make use of constraint satisfaction techniques that exploit libraries of human movement data [24]. The humanoid will be studied how to autonomously react to dynamic environments. Another direction can be motor learning. Instead of the traditional motor learning of trajectories or dynamics, we can have robot learn the EPs from observing the teacher's behaviors. For example, for a mimic robot to repeat the demonstrator's reaching movement, the only parameters it has to learn are the end posture and spring parameters. This greatly reduces the amount of information and computation conveyed to the learner, and simplifies the process.

References

[1] R. McN. Alexander. A minimum energy cost hypothesis for human arm trajectories. *Biological Cybernetics*, 76:97–105, 1997.

[2] Dimitri P. Bertsekas. *Constrainted Optimization and Lagrange Multiplier Methods*. Academic Press, 1982.

[3] E. Bizzi, N. Accornero, W. Chapple, and H. Hogan. Arm trajectory formation. *Exp. Brain Res.*, 46:139–143, 1982.

[4] E. Bizzi, F. A. Mussa-Ivaldi, and S. Giszter. Computations underlying the execution of movement: a biological perspective. *Science*, 253:287–291, 1991.

[5] A. Feldman and M. Latash. Testing hypotheses and the advancement of science: recent attempts to falsify the equilibrium point hypothesis. *Exp. Brain Res.*, 161:91–103, 2005.

[6] A. G. Feldman. Functional tuning of the nervous system with control of movement or maintenance of a steady posture. ii. controllable parameters of the muscle. *Biophysics*, 11:565–578, 1966.

[7] A. G. Feldman. Functional tuning of the nervous system with control of movement or maintenance of a steady posture. iii. mechanographic analysis of execution by man of the simplest motor tasks. *Biophysics*, 11:766–775, 1966.

[8] A. G. Feldman. Once more on equilibrium-point hypothesis (λ model) for motor dontrol. *J. Motor Behavior*, 18:17–54, 1986.

[9] A. G. Feldman, D. J. Ostry, M. F. Levin, P. L. Gribble, and A. B. Mitnitski. Recent tests of the equilibrium-point hypothesis (λ model). *Motor Control*, 2:189–205, 1998.

[10] T. Flash. The control of hand equilibrium trajectories in multi-joint arm movements. *Biological Cybernetics*, 57:257–274, 1987.

[11] Olac Fuentes. *Behavior-Based Dextrous Manipulation: The Virtual Tool Approach*. PhD thesis, University of Rochester, 1997.

[12] H. Gomi and M. Kawato. Equilibrium-point control hypothesis examined by measured arm-stiffness during multi-joint movement. *Science*, 272:117–120, 1996.

[13] C. M. Harris and D. M. Wolpert. Signal-dependent noise determines motor planning. *Nature*, 394:780–784, 1998.

[14] A.V. Hill. The heat of shortening and the dynamic constants of muscle. *Proceedings of Royal Society B*, 126:136–195, 1938.

[15] A.V. Hill. The series elastic component of muscle. *Proceedings of Royal Society B*, 137:399–420, 1950.

[16] M. R. Hinder and T. E. Milner. The case for an internal dynamics model versus equlibrium point control in human movement. *Journal of Physiology*, 549(3):953–963, 2003.

[17] T. Kashima and Y. Isurugi. Trajectory formation based on physiological characteristics of skeletal muscles. *Biological Cybernetics*, 78:413–422, 1998.

[18] M. Kawato. Internal models for motor control and trajectory planning. *Current Opinion in Neurobiology*, 9:718–727, 1999.

[19] C. Phillips and N. Badler. Interactive behaviors for bipedal articulated figures. *ACM Computer Graphics*, 25(4):359–362, 1991.

[20] R. Shadmehr and M.A. Arbib. A mathematical analysis of the force-stiffness characteristics of muscles and the role of reflexes in control of a single joint system. *Biological Cybernetics*, 66:463–477, 1992.

[21] N. Sundar. Dexterous robotic hands: Kinematics and control. Master's thesis, Indian Institute of Technology, 1983.

[22] E. Todorov and M. Jordan. Optimal feedback control as a theory of motor coordination. *Nature Neuroscience*, 5(11):1226–1235, 2002.

[23] E. Torres and D. Zipser. Reaching to grasp with a multi-jointed arm. i. computational model. *Journal of Neurophysiology*, 88:2355–2367, 2002.

[24] Katsu Yamane, James J. Kuffner, and Jessica K. Hodgins. Synthesizing animations of human manipulation tasks. *ACM Transactions on Graphics*, 23(3):532–539, 2004.

[25] J. Zhao and N. Badler. Inverse kinematics positioning using nonlinear programming for highly articulated figures. *ACM Transactions on Graphics*, 13(4):313–336, 1994.