Dynamic Simulation for Zero-Gravity Activities

Norman I. Badler, Dimitris N. Metaxas, Gang Huang, Ambarish Goswami, Suejung Huh
University of Pennsylvania, Philadelphia, USA

Abstract:

Working and training for space activities is difficult in terrestrial environments. We approach this crucial aspect of space human factors through 3D computer graphics dynamics simulation of crewmembers, their tasks, and physics-based movement modeling. Such virtual crewmembers may be used to design tasks and analyze their physical workload to maximize success and safety without expensive physical mockups or partially realistic neutral-buoyancy tanks. Among the software tools we have developed are methods for fully articulated 3D human models and dynamic simulation. We are developing a fast recursive dynamics algorithm for dynamically simulating articulated 3D human models, which comprises kinematic chains -- serial, closed-loop, and tree-structure -- as well as the inertial properties of the segments. Motion planning is done by first solving the inverse kinematic problem to generate possible trajectories, and then by solving the resulting nonlinear optimal control problem. For example, the minimization of the torques during a simulation under certain constraints is usually applied and has its origin in the biomechanics literature. Examples of space activities shown are zero-gravity self orientation and ladder traversal. Energy expenditure is computed for the traversal task.

Keywords: dynamic simulation, 3D computer graphics simulation, articulated body, human model, optimal control, motion planning.

1. Introduction:

Working and training for space activities is difficult in terrestrial environments. There is a great demand for human factors regarding the dynamic ability of the human body in space. In computer graphics (CG) and virtual reality (VR) such human factors would be helpful to create realistic human animations, and design tasks to maximize success and safety without always having to resort to expensive physical mockups or partially realistic neutral-buoyancy tanks. The latter accommodates some of the 0-gravity experience but adds unrealistic dynamic components such as damping from the water viscosity and different masses for large manipulated objects.

There are many forward and inverse dynamics simulation techniques for simulating human behavior that have been implemented by CG researchers [4][9][11], but they are not computationally efficient and simulations cannot be done in real time. Some control techniques have also been used, such as feedback control to follow the desired trajectories generated by inverse kinematics [4] and the spacetime constraint method based on optimization theory [12]. However, they do not address explicitly either the torque optimization problem or dynamics that deal with the various topological configurations of the articulated body.

This paper first presents an efficient way to convert the geometry of an articulated body to a dynamic tree structured model, which is suitable for dynamic simulation and motion planning. In the dynamic tree, each joint has only one degree of freedom, which gives only one parent link and one child link. Each link has its own root joint, and up to three branch joints. The link can be a dummy link, which has no mass, length and inertia. The root of the dynamic tree usually is a base joint fixed to the reference frame. Each leaf of the dynamic tree must be link, though the link can be a dummy link.

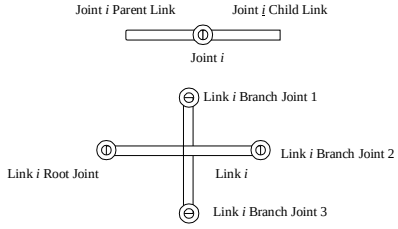


Figure 1. The Relationship between Joints and Links

Section 2 of the paper shows a simple type of function that we used for simulating the motions, and its properties for kinematic and dynamic simulation. All the dynamic simulations in this paper use this function, which we call kinematic motion function, to specify the motion of each joint or the body.

In Section 3 an efficient recursive dynamics method for the dynamic tree based on Featherstone's method [2] is studied and used to dynamically simulate astronaut self-orientation (Section 4) and ladder traversal in 0-gravity (Section 5). Furthermore, in Section 6 we present an optimal control algorithm based on this efficient recursive dynamics method and gradient method to find the local minimum joint-torque motion under specified constraints [3]. From the biomechanics point of view, the minimum joint-torque motion is the optimal motion under certain conditions for human body, for example the motion to do weight lifting in 10 seconds. Finally, in Section 7 we outline some energy expenditure models that can be evaluated based on the computed joint torques.

2. Dynamic Model for an Articulated Body:

The physically based simulation of human motion is obtained from kinematic and dynamic calculations. The efficiency of the computation depends on the algorithms used for these calculations.

An articulated body can be represented topologically as a tree whose links represent its major parts. Basically, this means there are no kinematic loops and that no part of the articulated body is entirely disconnected from the rest. An immobile link (or an appropriately chosen link if none of the links is immobile) is considered as the root of the dynamic tree, and the outermost links are its leaves. Two links connected by a joint are the parent and child links of the joint, depending on whether the link is more proximal or distal to the root. The joint is called the root joint of its child link. The joints with the same parent link are called the branch joints of the link. The base link is the link connecting the fixed point to the root joint of the dynamic tree. Every link except the base link has exactly one root joint, but may have up to three branch joints if modeling humans.

We represent the articulated body as a tree structure. Each joint has the following properties:

1) Tree structure properties: the index for the joint, the index for its parent and child links. Note the base link is defined as the null link, and the joint connected to the null link is joint 0. (See Fig. 2.)
2) Physical properties: the number of degrees freedom of the joint, the joint type[1], (such as XY-joint, ZY-joint, and XYZ-joint, etc.) the joint limits, the joint torque limits, the damping coefficients for the joint.
   Note: each joint has its own joint coordinate frame called joint private coordinate frame, which is different from the joint local coordinate frame defined in the tree structure, in which the joint rotation axis is always the z-axis. For example: in the ZY-joint private coordinate frame, the z-axis and the y-axis are joint rotation axes, and the x-axis is defined in the right-hand sense. In the joint local coordinate frame both rotation axes are defined as the z-axis. By rotating the first z-axis 90 degrees it coincides with the second z-axis. (See Fig. 2.)

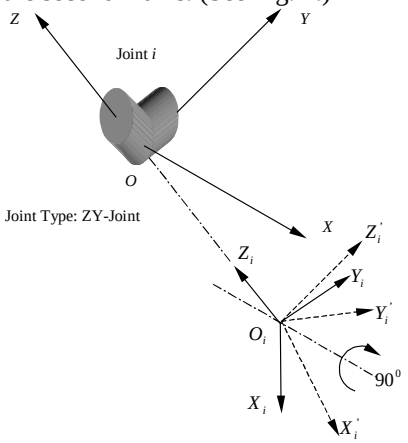


Figure 2. Joint Private Coordinate frame $O-XYZ$ for ZY-joint, Joint Local Coordinate frame $O_i - X_i Y_i Z_i$ for *o-z* the rotation axis, and $O_i - X_i' Y_i' Z_i'$ for *o-y* the rotation axis

Each link has the following geometric and dynamic properties:

1) Tree structure properties: the index for the link, the index for its root joint, the number of the branch joints, the index for each branch joint. Note a leaf link has no branch joint, we define it as the null joint.
2) Dynamic properties: the mass, the center of mass in the link's root joint coordinate frame, the principal inertial matrix, and the rotation matrix from the link's root joint coordinate frame to its principal coordinate frame.
3) Denavit-Hartenberg notation [12] parameters for each pair of the link's root

[1] We consider only multiple-degree-of-freedom joint with orthogonal rotation axis.

and branch joint j[2] (See Fig. 3): the length of the common normal a_j; the distance between the origin O_{i-1} and the point H_j; the angle α_j between the link's root joint axis and branch joint axis in the right-hand sense; the angle θ_j between the link's root joint coordinate frame x-axis and the common normal, which is the x-axis of the link's branch joint coordinate frame, measured about its root joint z-axis in the right-hand sense.

If a link's root joint is a multiple-degree-of-freedom joint, the link's root joint z-axis is the last rotation axis in the joint private coordinate frame. For example if the joint type is XY-joint, the y-axis in the joint private coordinate frame is the link's root joint z-axis we are using. In the same manner, if a link's branch joint is a multiple-degree-of-freedom joint, the link's branch joint axis is the first rotation axis in the joint private coordinate frame.

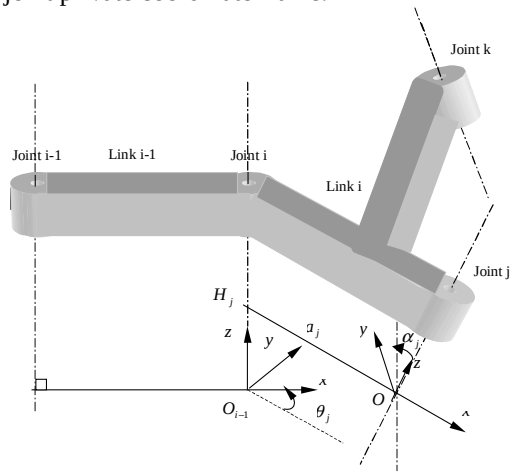


Figure 3. Denavit-Hartenberg Notation

The numbering scheme we use for the articulated body is that each link has the same index number as the link's root joint. The index of a link's root joint is less than the index of the link's leftmost branch joint. The index of a link's left branch joint is less than the index of the link's right branch joint. (The numbering on Fig. 4 follows this scheme.) Thus, we define the articulated body as a left-right tree-like structure.

[2] Strictly speaking, the pair is the single-degree-of-freedom root joint and the single-degree-of-freedom branch joint.

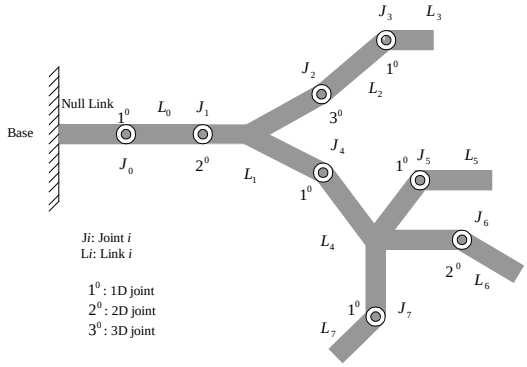


Figure 4. Numbering Scheme for Articulated Tree

We now present a method to efficiently represent an articulated tree using a dynamic tree representation. Multiple-degree-of-freedom joints can be synthesized from the appropriate number of single-degree-of-freedom joints. A moving articulated body could be treated as a dynamic tree structure by introducing a fictitious, unpowered joint between the root and some fixed point. If the root of the dynamic tree has complete motion freedom then the joint has six degrees of freedom.

The articulated body defined as an articulated tree structure is converted to a dynamic tree by representing every multiple-degree-of-freedom joint with the appropriate number of single-degree-of-freedom joints. Based on our method, we add dummy links into a multiple-degree-of-freedom joint, such that every joint in the dynamic tree is a single-degree-of-freedom joint. The dummy link is defined as the link with no length, no mass, and no inertia. Between the two consecutive generated single-degree-of-freedom joints there is a rotation matrix, which defines the transformation matrix between the two new joint local coordinates. Therefore a two-degree-of-freedom joint will generate two single-degree-of-freedom joints, such that the z-axes in both of the joint local coordinates are parallel to the original joint rotation axes. The numbering scheme for the dynamic tree is the same for the articulated tree. (See Fig. 5.) The joint *i* coordinate frame is attached to link *i*.

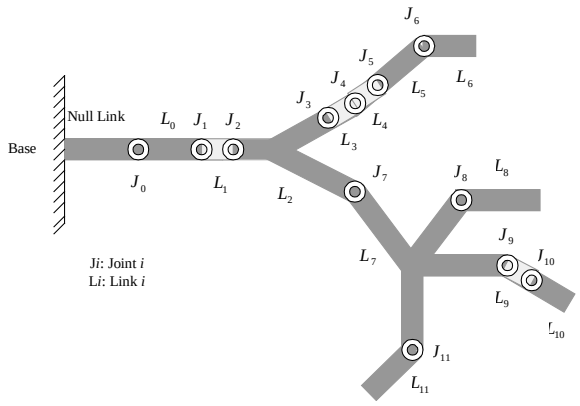


Figure 5. Numbering Scheme for Dynamic Tree, which is generated from Articulated tree in Fig. 4.

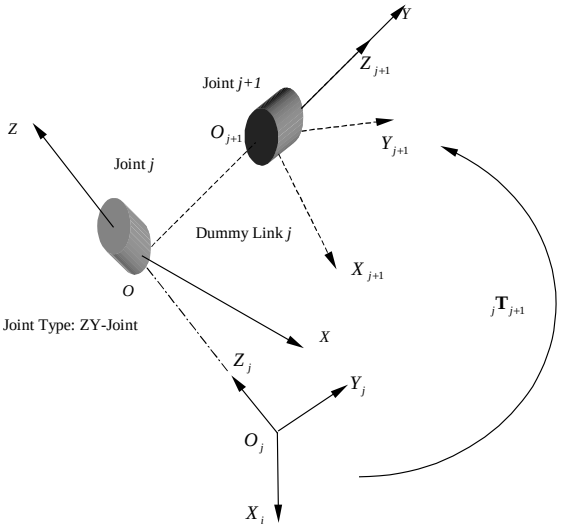


Figure 6. Two-degree-of-freedom ZY-joint converted to two single-degree-of-freedom joints and one dummy link. (Compare with Fig. 1.) If joint *i* in articulated tree can generate joint *j*, dummy link *j*, and joint *j+1* in dynamic tree, $_{j}\mathbf{T}_{j+1}$ is the transformation matrix between the joint *j* local coordinate frame and the joint *j+1* local coordinate frame.

3. Kinematic Motion Function:

In our simulation we would like to assess the feasibility of continuous motion. Therefore, we first need to generate these motions and then use inverse dynamics to compute the joint torques. The human motion simulation is carried out based on kinematic calculations, and the joint torques are obtained by inverse dynamics. The issue is what kind of function we should use to specify the motion of joints or bodies. There are an infinite number of functions we could choose for the motion simulation, for example linear functions and quadratic functions. However, most human motion starts with zero velocity and zero acceleration, and ends with zero velocity and zero acceleration as well. To satisfy these boundary conditions, we choose the simplest type of function. Its derivative (namely, velocity) function is in the form of:

$$\dot{x}(t) = C_0 (t - t_0)^2 (t_e - t)^2$$

where C_0 is a constant that can be calculated from the initial and ending positions, t_0 is the starting time, and t_e is the ending time.

For simplicity, we set $t_0 = 0$, $t_e = T$, and $x(0) = 0$, $x(T) = X$. The kinematic motion function is

$$\dot{x}(t) = C_0 t^2 (T - t)^2$$

This function obviously satisfies the velocity and acceleration constraints. Namely,

$$\dot{x}(0) = 0,\ \dot{x}(T) = 0,\ \ddot{x}(0) = 0,\ \ddot{x}(T) = 0.$$

We integrate the velocity function to obtain the motion function:

$$x(t) = C_0 \left(\frac{1}{3} T^2 - \frac{1}{2} T t + \frac{1}{5} t^2 \right) t^3$$

From $x(T) = X$, we get

$$C_0 = 30 \frac{X}{T^5}$$

So the function for the motion simulation is

$$x(t) = \frac{X}{T^5} \left(10T^2 - 15Tt + 6t^2 \right) t^3 \qquad (1)$$

The variable $x(t)$ can represent a position vector (the three components have different parameters of the kinematic motion function) or joint variable (for example $\theta(t)$ for a revolute joint).

Now we study a more complicated motion problem, in which the motion function should satisfy the constraint that the maximum speed during the motion should have an upper bound $V_{\max}$. First we solve the function for the motion simulation as above, and then check if $V_{\max} \geq \dot{x}(T/2) = \frac{15X}{8T}$ (since the kinematic motion function reaches its maximum at the middle of the motion). If it is satisfied, the function we calculated is the final function for the motion simulation. If it is not satisfied, we will change the motion to three phases. The first phase is the acceleration phase, in which the motion speeds up to $V_{\max}$ in t_2 seconds. The second phase is the constant speed phase, in which the motion keeps the speed at $V_{\max}$ for t_1 seconds. The third phase

is the deceleration phase, in which the motion slows down to zero velocity in t_2 seconds. Thus,

$$t_1 = 2\frac{X}{V_{\max}} - T,\ t_2 = T - \frac{X}{V_{\max}}$$

$$x(t) = \frac{V_{\max} t^3}{t_2^3}(t_2 - t/2),\ t \in [0, t_2)$$

$$x(t) = V_{\max}(t_2/2 + t),\ t \in [t_2, t_1 + t_2)$$

$$x(t) = \frac{1}{12}t^4 - \frac{1}{6}(T + t_1 + t_2)t^3 + \frac{1}{2}T(t_1 + t_2)t^2$$
$$-\frac{1}{6}(t_1 + t_2)^2(3T - t_1 - t_2)t$$
$$-\frac{1}{12}(t_1 + t_2)^3(t_1 + t_2 - 2T) + V_{\max}(t - t_2/2)$$
$$t \in [t_1 + t_2, T], \text{ Note } T = t_1 + 2t_2$$

4. Self Orientation:

People re-orient themselves easily on earth since gravity assures a force component and therefore an expectation of friction with the supporting surface. A space walker can move and make turns if equipped with SAFER (Simplified Aid for Extravehicular activity Rescue), which produces six degrees of freedom in movement. But how can an astronaut makes turns when floating inside the vehicle with no external force and torque acting on her body and no hand or foot restraints? In this section, we describe the dynamic theory -- based on the conservation of momentum and Featherstone's inverse recursive dynamics -- which allows the astronaut to do a self-orientation. Then we do the simulation and calculate the joint torques during the simulated motion.

A 0-gravity self-orientation motion can be completed in three steps: leg lifting, leg twisting, and leg closing.

Figure 7. Standing Posture

Figure 8. Leg Lifting

Figure 9. Leg Twisting

Figure 10. Leg Closing

For the purpose of this simulation, we use a simplified articulated body which includes two joints (the left and right hip joints) and three links (the torso link, the left and right leg links), which can be converted to a dynamic tree with six joints and seven links. (See Fig. 11.) The $O - XYZ$ frame is the reference frame. The torso link coordinate frame $O_0 - X_0Y_0Z_0$ is attached to the center of mass of the torso. The right hip joint private coordinate frame $O_1 - X_1Y_1Z_1$ and the left hip joint private coordinate frame $O_2 - X_2Y_2Z_2$ are attached to the right and left leg links, respectively. C_1 and C_2 are the center of mass of the right and left legs, respectively.

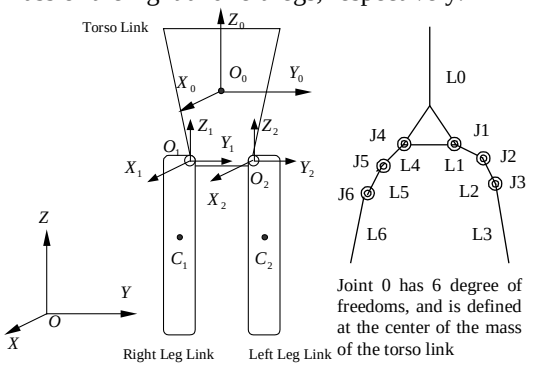


Figure 11. Coordinate frames for Articulated Body and its Dynamic Tree. Note: L1, L2, L4, and L5 are dummy links

Since the astronaut is floating inside the vehicle, there is no external force and torque exerted on the astronaut, and so spatial momentum (i.e. linear and angular momentum of the human body) is conserved during the motion. The dynamic notations we use in this paper are based on those in Featherstone's book. The conservation of momentum is expressed as:

$$\hat{\mathbf{I}}_0\hat{v}_0 + \hat{\mathbf{I}}_3\hat{v}_3 + \hat{\mathbf{I}}_6\hat{v}_6 = const = \hat{0}$$

where $\hat{\mathbf{I}}_i\,(i = 0 \cdots 6)$ is the spatial inertia for each link in the reference frame. $\hat{v}_i\,(i = 0 \cdots 6)$ is the spatial velocity for each link in the reference frame. The spatial velocities $\hat{v}_3$ and $\hat{v}_6$ of the left and right leg links can be written as:

$$\hat{v}_3 = \hat{v}_0 + \hat{S}_1\dot{q}_1 + \hat{S}_2\dot{q}_2 + \hat{S}_3\dot{q}_3$$
$$\hat{v}_6 = \hat{v}_0 + \hat{S}_4\dot{q}_4 + \hat{S}_5\dot{q}_5 + \hat{S}_6\dot{q}_6$$

where $q_i(t)$ $(i = 0\cdots 6)$ is the specified function for motion simulation during each step, computed from (1). $\hat{S}_i$ $(i = 0\cdots 6)$ is the spatial vector for the axis of joint *i*.

The spatial velocity $\hat{v}_0$ is obtained by solving the above equations:

$$\hat{v}_0 = -\left(\hat{\mathbf{I}}_0 + \hat{\mathbf{I}}_3 + \hat{\mathbf{I}}_6\right)^{-1} \left[\hat{\mathbf{I}}_3\left(\hat{S}_1\dot{q}_1 + \hat{S}_2\dot{q}_2 + \hat{S}_3\dot{q}_3\right) + \hat{\mathbf{I}}_6\left(\hat{S}_4\dot{q}_4 + \hat{S}_5\dot{q}_5 + \hat{S}_6\dot{q}_6\right)\right]$$

By integrating $\hat{v}_0$ or taking its derivatives, we can get the position and orientation of the torso link and the spatial acceleration of it as well.

Let $\hat{f}_0$ be the resultant spatial force exerted on the torso link through the right and left hip joints. $\hat{f}_0$ gives the overall rate change of momentum for the torso link.

$$\hat{f}_0 = \frac{d}{dt}\left(\hat{\mathbf{I}}_0\hat{v}_0\right) = \hat{\mathbf{I}}_0\hat{a}_0 + \hat{v}_0 \hat{\times} \hat{\mathbf{I}}_0\hat{v}_0$$

In order to get the left-hip joint torque, the spatial force $\hat{f}_3$ is computed as

$$\hat{f}_3 = \hat{\mathbf{I}}_3\hat{a}_3 + \hat{v}_3 \hat{\times} \hat{\mathbf{I}}_3\hat{v}_3$$

where

$$\hat{a}_3 = \hat{a}_0 + \hat{S}_1\ddot{q}_1 + \hat{S}_2\ddot{q}_2 + \hat{S}_3\ddot{q}_3 + \hat{v}_0 \hat{\times}\left(\hat{S}_1\dot{q}_1 + \hat{S}_2\dot{q}_2 + \hat{S}_3\dot{q}_3\right) + \hat{S}_1\dot{q}_1 \hat{\times}\left(\hat{S}_2\dot{q}_2 + \hat{S}_3\dot{q}_3\right) + \hat{S}_2\dot{q}_2 \hat{\times} \hat{S}_3\dot{q}_3$$

The joint torque about each joint axis is thus:

$$Q_X = \hat{S}_1^S \hat{f}_3,$$
$$Q_Y = \hat{S}_2^S \hat{f}_3,$$
$$Q_Z = \hat{S}_3^S \hat{f}_3.$$

5. Ladder Traversal:

Since an astronaut floats in space (0-gravity), it is hard to keep balance among the limbs when traversing a ladder. By letting the legs float free and only using the hands to grab the bars, the astronaut can control her body motion. In this Section we do the dynamic simulation and calculate the joint torques, so we can compute the maximum possible workload for an astronaut traversing a ladder. In the future, we plan to use computer vision techniques to capture an actual astronaut's movement for detailed comparison.

The articulated tree we used to simplify the astronaut body consists of eight joints and 7 links. (See Fig. 12.)

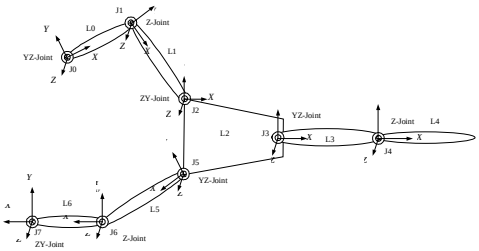


Figure 12. Simplified Articulated Tree with 13 degrees of freedom (All the frame are joint private coordinate frames.)

A dynamic tree with 13 joints and 13 links can be generated from the articulated tree easily. We add a dummy link to the left wrist z-joint if the right wrist y-joint is the root joint for the dynamic tree.

In order to simplify the simulation, we assume there are no dynamic closed loops during the motion. Namely, if the right hand grabs the bar then the left hand must be free, this is simplification is based on how astronauts traverse ladders. We can divide the motion into two kinds of motion periods: 1) Starting/ending motion period, 2) Middle motion period. The starting motion period begins with both hands on the bar, but only one grabbing it (say right hand), and ends with the free hand (left hand) on the front bar. The ending motion period is the reverse of the starting motion period. The middle motion period starts with one hand grabbing bar *i* (say right hand) and the other hand (left hand) on bar (*i-1*), and ends with the other hand (left hand) on bar (*i+1*). (See Fig. 13,14.)

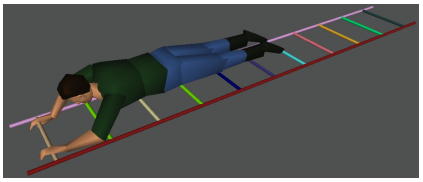
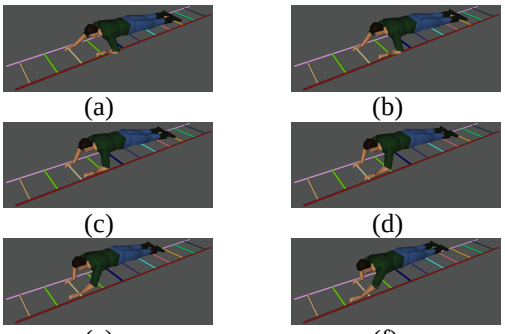
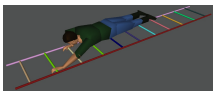


Figure 13. Ending Posture, which is the same as starting posture





(g)

Figure 14. Animation Sequence of Ladder Traversal in one period

The kinematic motion functions are computed in two levels. The global motion planning level gives the simulated motion for the torso link by the method we described in Section 2. The periodic motion planning level gives the simulated motions for each joint during that period. These functions are calculated by giving the initial and final positions for joint-variables and the maximum speed constraints. The initial and final joint-variables are computed based on inverse kinematics, with the position and orientation of the torso link specified, and the hands' positions are known at each period's initial and final stages. The velocity of the torso link (obtained from the periodic motion planning level) in the direction of motion should be consistent with the one computed from the global motion planning level. Once we have the functions of the simulated motion for each joint, we calculate the joint torques.

The above computes kinematics. In order to compute the joint torques we use an efficient calculation scheme, which is similar to Featherstone's and can be obtained by performing these calculations pertinent to link *i* in joint *i*-coordinate frame. The equations in joint *i*-coordinate frame, which is attached to link *i*, is as follows:

$$\hat{v}_i = {}_i\hat{\mathbf{X}}_{i.parent}\hat{v}_{i.parent} + \hat{S}_i\dot{q}_i, \qquad (\hat{v}_{0.parent} = \hat{0})$$

$$\hat{a}_i = {}_i\hat{X}_{i.parent}\hat{a}_{i.parent} + \hat{v}_i \hat{\times} \hat{S}_i\dot{q}_i + \hat{S}_i\ddot{q}_i, \ (\hat{a}_{0.parent} = \hat{0})$$

$$\hat{f}_i^* = \hat{\mathbf{I}}_i\hat{a}_i + \hat{v}_i \hat{\times} \hat{\mathbf{I}}_i\hat{v}_i,$$

$$\hat{f}_i = \hat{f}_i^* + \sum_{j\in\{i.child\}} {}_i\hat{\mathbf{X}}_j \hat{f}_j, \ (\text{if}\{i.child\} = \varnothing, \ \hat{f}_i = \hat{f}_i^*)$$

$$Q_i = \hat{S}_i^S \hat{f}_i.$$

6. Optimal Control:

The above simulations are done in real time. But we also interest in optimized motion of the human motion under certain specified constraints, which can not be done in real time. The optimal control problem is formulated by minimizing the joint torques, since it is suitable in space applications. The problem can be stated as [3]:

$$\textit{Optimize}: \ \min_{q_{ij}}\left\{\sum_{k=1}^{DOF}\int_{t_0}^{t_f}\left[\tau_k(t,q_{ij})\right]^2 dt\right\}$$

Subject to all the given constraints

where *DOF* is the total degrees-of-freedom of the articulated body. $\tau_k(t,q_{ij})$ is the joint torque at time *t* when the functions of the simulated motion are expressed in B-Spline functions [1] with the control points q_{ij}.

The nonlinear programming method requires the explicit calculation of the gradient. We define the optimization function based on a given set of control points q_{ij} and the weight coefficients λ_k for all the constraints:

$$F(q_{ij},\lambda_k) = \sum_{k=1}^{DOF}\int_{t_0}^{t_f}\left[\tau_k(t,q_{ij})\right]^2 dt + \sum_{k}^{n}\lambda_k C_k(t,q_{ij})$$

where *n* is the total number of constraints.

We optimize $F(q_{ij},\lambda_k)$ by using the gradient method:

$$\left[\nabla^2 F(q_{ij}^{(l)},\lambda_k^{(l)})\right]_{(m+n)\times(m+n)}\begin{pmatrix}\delta q_{ij}\\ \delta\lambda_k\end{pmatrix}_{(m+n)\times 1} = -\left(\nabla F(q_{ij}^{(l)},\lambda_k^{(l)})\right)_{(m+n)\times 1}$$

where $m = \sum_i^{DOF} N_i$, N_i is the number of B-spline control points for each joint's simulated motion function in B-spline form. Lo and Metaxas [6] give the details on how to compute the gradient of the optimization function $F(q_{ij},\lambda_k)$. By solving the above linear equation system, we obtain δq_{ij} and $\delta\lambda_k$. And we have the new set of the control points and the weight coefficients:

$$q_{ij}^{(l+1)} = q_{ij}^{(l)} + \delta q_{ij},$$
$$\lambda_k^{(l+1)} = \lambda_k^{(l)} + \delta\lambda_k.$$

If the iteration index *l* is larger than the specified maximum iteration times, or

$$\delta q_{ij}\delta q_{ij} << 1 \text{ and } \delta\lambda_k\,\delta\lambda_k << 1,$$

we choose $q_{ij}^{(l+1)}$ as the final control points for each joint's B-spline kinematic motion function.

7. Energy Expenditure:

It is commonly suggested that skilled human movements optimize certain criteria, which are related to the energy expenditure [13].

Among the suggested criteria, we elected to measure the virtual human's energy expenditure based on the following quantities:

a) the weighted sum of the absolute values of the joint torque,
b) the change of the total mechanical energy of the body,
c) the mechanical power generated or transferred in the joints, and
d) the rate of sudden change in movement (jerk).

The weighted sum of the absolute values of the joint torque

The time integral of the joint torque has been suggested as a useful measure of the energy expenditure when it is appropriately weighted for the negative and positive work conditions [13,16].

Among many suggestions, Williams[14] and Pierrynowski et al. [15] have reported that the negative work is approximately three times more efficient than the positive work. For a given joint, the work is positive when the torque and the angular velocity have the same direction, and negative otherwise. The positive joint work corresponds to the concentric muscle activity and the negative joint work corresponds to the eccentric muscle activity. So the sum of the weighted absolute joint moments is used as our energy expenditure measure where the absolute joint moment is defined as

$$\rho_i = |\tau_i| \quad \text{if } \tau_i * \omega_i \geq o$$
$$= \frac{1}{3}|\tau_i| \quad \text{otherwise}$$

where τ_i refers to the torque of the joint *i* and ω_i refers to the angular velocity of the joint *i*.
Fig. 15 shows the weighted sum of the absolute joint torque without any shoulder constraint where the high pitch denotes the time when the astronaut touches the ladder.

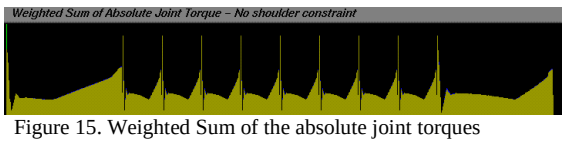


Figure 15. Weighted Sum of the absolute joint torques

The change of the total mechanical energy of the body

$$E = \sum_i (TKE_i + RKE_i)$$

where TKE_i refers to the Translational Kinetic Energy of the joint *i* and RKE_i the Rotational Kinetic Energy of the joint *i*.

The mechanical power generated or absorbed at the joints

$$P = \sum_i \tau_i(\omega_{i+1} - \omega_i)$$

where τ_i refers to the torque of the joint *i* and ω_i refers to the angular velocity of the joint *i*.

The mechanical power transferred at the joints

$$P = \sum_i p_i$$
$$p_i = \tau_i * \omega_i \quad \text{if } \tau_i * \omega_i \geq o$$
$$= \frac{1}{3}|\tau_i * \omega_i| \quad \text{otherwise}$$

where τ_i refers to the torque of the joint *i* and ω_i refers to the angular velocity of the joint i. This criterion incorporates both concentric and eccentric muscle activities.

The rate of sudden change in movement(Jerk)

$$A = \sum_i J_i$$

where J_i refers to the time derivative of acceleration of the joint *i*(jerk). The jerk is shown to be minimized for a certain class of activities [17].

8. Conclusion:

Dynamically correct human motion simulation requires a proper dynamic tree model efficiently created from a simplified articulated tree. It also requires the generation of an articulated body movement pattern from a starting posture to a final posture. The main focus of this paper is to simulate self-orientation and ladder traversal of an astronaut in 0-gravity. These and other dynamic simulations may be used for both visualization and analysis of 0- or micro-graity tasks. We implemented the methods developed here to generate the dynamic tree, simulate the

desired motions, and compute the energy expenditure functions. These procedures will be useful in the future analysis and safety evaluation of novel space activities.

Acknowledgments:

This research is partially supported by *NASA NRA NAG 5-3990* and *ONR YIP N00014-97-01803* to the second author.

Bibliography:

[1] G. Engeln-Mullges, F. Uhlig. *Numerical Algorithms with C.* Springer, 1996

[2] R. Featherstone. *Robot Dynamics Algorithm.* Kluwer Academic Publishers, Boston, 1987

[3] R. Fletcher. *Practical Methods of Optimization*. John Wiley & Sons, Ltd. 1981

[4] J.K. Hodgins, W.L. Wooten, D.C. Brogan, and J.F. O'Brien. *Animation of Human Athletics.* In Proc. *SIGGRAPH 95*, pages 71-78, August 1995

[5] K.W. Lilly. *Efficient Dynamic Simulation of Robotic Mechanisms.* Kluwer Academic Publishers, Boston, 1993

[6] J. Lo. Ph.D. Thesis: *Recursive Dynamics and Optimal Control Techniques for Human Motion Planning.* Mechanical Engineering and Applied Mechanics, University of Pennsylvania, 1998

[7] J. Lo and D. Metaxas. *Efficient Human Motion Planning Using Recursive Dynamics and Optimal Control Techniques.* Proc. Of the Second Symposium on Multibody Dynamics and Vibration at the 17th Biennial Conference on Mechanical Vibration and Noise, Las Vegas, NV, Sept. 12-15, 1999

[8] J. Lo and D. Metaxas. *Recursive Dynamics and Optimal Control Techniques for Human Motion Planning. Proc. Of Computer Animation Conference, CA'99. Geneva, Switzerland, May 26-29, 1999*

[9] A.J. Stewart and J.F. Cremer. *Beyond keyframing: An algorithmic approach to animation*. In *Proc. Of Graphics Interface*, pages 273-281, 1992

[10] H.W. Stone. *Kinematic Modeling, Identification, and Control of Robotic Manipulators.* Kluwer Academic Publishers, Boston, 1987

[11] J. Wilhelms and B. Barsky. *Using Dynamic Analysis to Animate Articulated Bodies such as Humans and Robots*. In Graphics Interface, 1985

[12] A. Witkin and M. Kass. *Spacetime Constraints*. *ACM Computer Graphics*, 22(4), 1988

[13] R.N. Marshall, G.A. Wood and L.S. Jennings. *Performance objectives in human movement: A review and application to the stance phase of normal walking*. Human Movement Science 8 (1989) 571-594

[14] K.R. Williams and P.R. Cavanagh. *A model for the calculation of mechanical power during distance running*. *Journal of Biomechanics* 16, 115-128, 1983

[15] M.R. Pierrynowski, D.A. Winter and R.W. Norman. *Transfers of mechanical energy within the total body and mechanical during treadmill walking*. Ergonomics 23, 1980, 147-156

[16] J.G. Andrews. *Biomechanical measures of muscular effort*. *Medicine and Science in Sports and exercise* 15, 199-207

[17] N. Hogan. *An organizing principle for a class of voluntary movements. Journal of Neuroscience* 4, 2745-2754