THE ROYAL SOCIETY

Published online 13 August 2003

Inverse and forward dynamics: models of multi-body systems

E. Otten

Institute of Movement Sciences, University of Groningen, A. Deusinglaan 1, 9713 AV, Groningen, The Netherlands
(*e.otten@med.rug.nl*)

Connected multi-body systems exhibit notoriously complex behaviour when driven by external and internal forces and torques. The problem of reconstructing the internal forces and/or torques from the movements and known external forces is called the 'inverse dynamics problem', whereas calculating motion from known internal forces and/or torques and resulting reaction forces is called the 'forward dynamics problem'. When stepping forward to cross the street, people use muscle forces that generate angular accelerations of their body segments and, by virtue of reaction forces from the street, a forward acceleration of the centre of mass of their body. Inverse dynamics calculations applied to a set of motion data from such an event can teach us how temporal patterns of joint torques were responsible for the observed motion. In forward dynamics calculations we may attempt to create motion from such temporal patterns, which is extremely difficult, because of the complex mechanical linkage along the chains forming the multi-body system.

To understand, predict and sometimes control multi-body systems, we may want to have mathematical expressions for them. The Newton–Euler, Lagrangian and Featherstone approaches have their advantages and disadvantages. The simulation of collisions and the inclusion of muscle forces or other internal forces are discussed. Also, the possibility to perform a mixed inverse and forward dynamics calculation are dealt with. The use and limitations of these approaches form the conclusion.

Keywords: modelling; locomotion; Lagrangian

1. INTRODUCTION

Systems consisting of rigid bodies, connected by well-defined connections, exhibit notoriously complex behaviour when driven by internal and/or external forces. This is directly related to the flow of energy along the chains of elements, which is a system property. The forces guided through the connections are partly formed by external forces (including gravity) and by inertial effects, both rotational and translational. There are several motivations for creating simulations of these systems. Sometimes the calculation of internal forces, or moments of force, from the movements and external forces is needed, to be able to understand, for example, how the nervous system drives such a complex system. This process is called inverse dynamics. Sometimes the calculation of movements and external reaction forces is required, based on known internal forces or moments of force. This process is called forward or direct dynamics. This is required to check whether a certain control strategy actually works, or in robotics, to be able to test the combination of a control strategy and inertial and spring properties of a robot.

Several useful approaches exist in the literature to solve this problem. Each of these approaches has advantages and disadvantages. Each of them allows an extension of algorithms, for instance to simulate extra constraints or collisions.

Given that the calculation of the dynamics of complex systems is very processor intensive, it is a great advantage that computer processors have doubled their computing power every year (Moravec 1998). In conjunction with improving motion capture capacity and accuracy, inverse dynamics calculations of 15-segment human body models can now be performed in real time on the more powerful desktop computers. This is based on 100 Hz motion capture data.

There are several computer packages, developed by the scientific community and commercially, that can perform forward dynamics computations, and they are based on a handful of principles, written in several languages. This paper outlines these principles, and in so doing partly opens the black boxes of these packages. The paper is also meant to indicate the limitations of the approaches used. It appears that in this complicated matter there is no single approach that combines all the favourable properties, like speed, stability and flexibility. Apart from being a review of these principles, the paper offers several additions. First it deals with the dynamic inclusion of contact points, either with the external world or between elements of the multi-body system. Part of this inclusion deals with the collision equations. Second, it shows that a mixture of forward and inverse dynamics is mathematically possible and useful for some studies. Third, it outlines a method to include muscle models in such a simulation.

One contribution of 20 to a Theme Issue 'Modelling in biomechanics'.

2. METHODS

Three methods will be discussed in this section for the solution of forward dynamics problems.

(a) *Newton–Euler method*

This method is usually associated with inverse dynamics approaches, but can also be used for forward dynamics by adding constraint equations.

A straightforward approach to the simulation of dynamics is to look at the elements of a connected system as free bodies. One can create three equations for the linear accelerations of an element (the Newton equations) and three equations for the rotational accelerations (the Euler equations). If one does this for all n elements of the system, one acquires $6n$ equations. However, the forces acting on each element are partly formed by the connections. Because these forces are equal but opposite in direction for each adjoining element, these $6n$ equations are not sufficient. These connection forces are unknowns and therefore we need to add as many equations as there are unknown contact force components. These equations can indeed be formulated: they are so-called constraint equations. They are formulated based on the demand that the accelerations at the contact points seen from both elements are equal. Thus, if there are p contact points in the system, there are $3p$ constraint equations and unknown force components. If there are more than three contact points between two elements, the system is overdetermined and can no longer be solved.

Because the equations are all linear, they can be solved using an appropriate algorithm for such systems of equations. The solution of a set of linear equations is an O(n3) process, meaning that doubling of the number of equations demands an eightfold increase in computer time (Press *et al.* 1986).

This approach makes the addition of external and internal forces very simple. In fact the administration of the equations is very easy, making this method the ideal choice for relatively small systems and planar systems up to about 10 elements. Two examples are worked out in Appendix A.

(b) *Lagrange's method*

Lagrange offered a mathematical description of the dynamics of mechanisms by having the insight that one can express the motion of a mechanism in terms of its degrees of freedom. For instance you can describe the position of the tip of the centre of mass of a door handle in terms of global coordinates, like in the Newton–Euler approach, resulting in six numbers. The first three numbers are the position coordinates of the centre of mass of the handle, and the tip can be calculated from the three orientation angles of the handle. However, Lagrange would describe it by giving the angle of the door relative to the wall, and the angle of the door handle relative to the door, which is only two numbers. Both angles are degrees of freedom of the mechanism. The only extra information needed concerns the dimensions of the mechanism, such as the distance of the handle axis to that of the door axis. The Lagrangian of a mechanism is the difference between the kinetic energy of a system and its potential energy. A mechanical system has the tendency to move in such a way that the integral of the Lagrangian over time is minimal. This principle is called Hamilton's principle. The integral is called the *action* of the system. For an excellent introduction to the principle of least action, see Feynmann *et al.* (1964, ch. 19). By taking the time derivative of the partial derivative of the Lagrangian of the system with respect to its generalized velocities, minus the partial derivatives of the Lagrangian with respect it its generalized coordinates, and stating that this should be equal to the generalized forces acting on the mechanism, we acquire the equations of motion.

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial q_i}\right) - \frac{\partial L}{\partial q_i} = F_i \quad i = 1,\ldots n.$$

Here, L is the Lagrangian (difference between kinetic and potential energy), q is a generalized coordinate or degree of freedom of the system (positions and angles) and F is a generalized force (forces and moments of force) acting on, or in, the system.

These equations are called Lagrange's equations. They can be used in combination with d'Alembert's principle, stating that in dynamic situations the sum of the forces on a system and its inertial forces is zero. The inertial forces are minus the accelerations multiplied by the masses. Stated in this way d'Alembert's principle is the same as Newton's second law, but can be extended to complex systems. Kane (Kane & Wang 1965) developed equations of motion for complex systems using a combination of Lagrange's equations and d'Alembert's principle. These equations of motion have since been called 'Kane's equations', 'Lagrange's form of d'Alembert's principle' and the 'principle of virtual power'. Huston (1990) gives a full mathematical treatise.

It is a much more complicated way of setting up equations of motion than the Newton–Euler approach described above, but the number of equations is lower. This number is exactly the same as the number of degrees of freedom of the mechanism. For instance, a human body model consisting of 15 elements and 14 joints of which four are hinge joints (1 d.f.) and 10 are ball and socket joints (3 d.f.) has a total of 40 d.f. and thus can be described with 40 coupled equations of motion using the lagrangian formalism. The Newton–Euler approach requires $6 \times 15 = 90$ equations to describe the free body dynamics of all elements and 42 constraint equations. These 42 equations follow from the fact that each of the 14 joints need to preserve their integrity in three directions. Because the solution of linear equations is an O(n3) process, the Newton–Euler approach takes 36 times more calculation time once the equations have been formulated. This number is found by raising the ratio of the number of equations $(90 + 42)/40 = 3.3$ to the power of three (figure 1).

(c) *Featherstone's method*

The Featherstone algorithm (Featherstone 1987) is a very economical way to solve multi-body dynamics, but it has its limitations. It is an O(n) process, meaning that doubling of the number of bodies results only in the doubling of calculation time. This implies that it is very useful for large systems of bodies. The algorithm consists of three passes. First, it evaluates the velocities (both linear and rotational) of all bodies, the static forces (forces needed to keep the bodies in position against external forces and moments of force) and the isolated inertia of all bodies. This first pass means that the chains of elements will have to be traversed from the central element (for instance the thorax in a human body model) to the most distal elements, because the velocities depend on each other in that direction. The second pass evaluates the articulated inertia and articulated static forces of each body and works its way from the most distal elements to the central element. The articulated inertia is the sum of the body's inertia and the inertia of all its connected more distal bodies. The articulated static force of a body is the force needed at its joint with the more central body to keep it

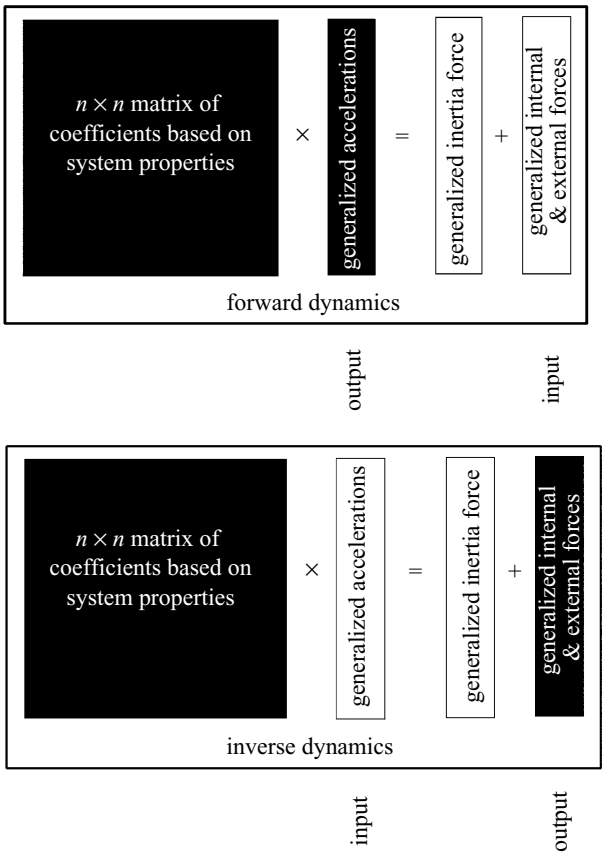

Figure 1. The set of equations of motion in the Lagrange approach in the inverse and forward dynamic simulations.

from accelerating. Again, the more distal elements are included. The third pass calculates the accelerations (both linear and rotational) of all bodies and works its way from the central element to the most distal ones.

Although the Featherstone algorithm is very economical, it has a very strong limitation: it can only be used on open chain systems, because it works its way along the branches of elements. No loop of elements is allowed, for instance loops that occur as soon as two feet of a human body model are on the ground. Because the contact forces are unknown, and as a circular dependence of joint forces occurs, the algorithm breaks down in this case. Some kind of optimization algorithm may be devised in such instances, in which one looks for the unknown external forces by trial and error, but that would lead to a loss of the major advantage of this method: its speed. It is possible to use the Featherstone algorithm when one element is in contact with the ground by renumbering the elements so that this ground element becomes the parent of all other elements. And so a hybrid algorithm may be used in which one uses the Featherstone algorithm in all time frames where one or no element is on the ground, while during the other time frames the Newton–Euler or Lagrangian approach is used. Major fluctuations in performance are expected because of the differences in processor load in each of the algorithms. But when no real-time performance is demanded, such a hybrid approach may save a lot of time.

3. FORWARD AND INVERSE DYNAMICS

Because equations of motion give the relation between motion and forces, they can be used in two directions: solving the motion from the forces (forward or direct dynamics) or solving the forces from the motion (inverse dynamics). Setting the equations up as mutually dependent linear equations, as in the Newton–Euler method with constraint equations and in the Lagrange method, ensures that all mechanical coupling in the system is taken into account, including the inverse dynamics simulations. However, in some instances it may be practical to solve for inverse dynamics by recursively going along the kinematic chains from distal to proximal (Winter 1990). Because, in inverse dynamics, the coupling in the system is already a given, one can find the joint moments of force by using this method, but only in the absence of closed loops. Mathematically, this means that the matrix that is multiplied with the linear and angular accelerations is sparse and can be arranged so that it shows a thin band of non-zero entries. However, if you have gone through the process of setting up Newton–Euler equations with constraint equations, it is expedient to multiply this matrix directly with the observed accelerations and obtain the joint moments of force as well as the contact forces in the joints. This avoids having to administrate the direction of calculations along the chains of elements and dynamically change it as soon as another element is in contact with the ground.

In conclusion, setting up the equations of motion is the most difficult part, but as soon as that has been done, both inverse and forward dynamics calculations can be done very easily by matrix multiplication and matrix inversion, respectively (figure 2).

4. FORWARD DYNAMICS AND PREDICTABILITY

It is very instructive to try to control a model of a human with 15 segments in three dimensions by adjusting the joint moments of force. It appears to be almost impossible: first of all it is hard to predict how a change in moment of force in the right hip joint for instance influences the ground reaction force and thus the acceleration of the centre of mass. Second, it is hard to predict the movements that result from such a change in moment of force in the rest of the body. Before long, you are adjusting moments of force everywhere, while the model starts to fall over. This problem is found in most simulations unless the model is heavily constrained, as in lying down on a floor. Therefore, it may be very helpful to use a recording of movements of a subject and the ground forces and calculate the moments of force at the joints by inverse dynamics. At some critical point in time one reverses the process into forward dynamics and takes the calculated moments of force from the latest time step as a guideline. Visualization of linear and angular accelerations while manually changing the moments of force in any one of the joints can form a rich source of information of the relation between local control and global movement. The only demand posed on the mathematics is that the process can indeed be reversed at will, and still produce the same instantaneous relation between generalized forces and movements. That is another reason for carrying out inverse dynamics in matrix form and not recursively along chains of elements.

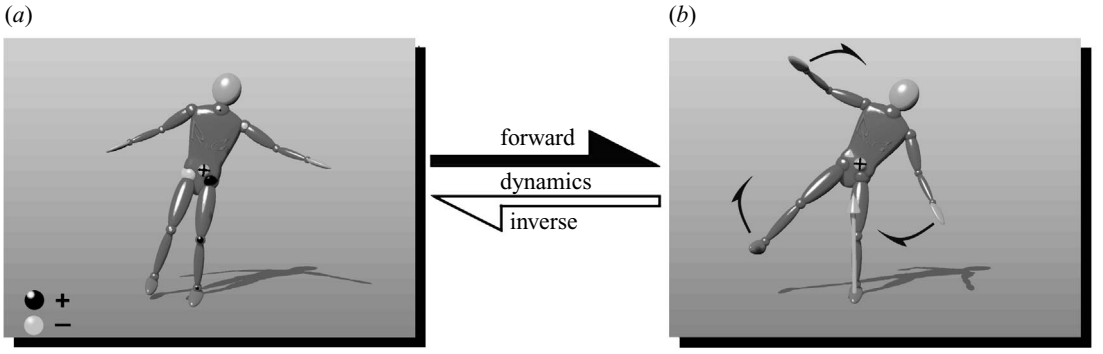

Figure 2. Forward and inverse dynamics of a 15-segment 3D human body model. (*a*) Joint moments of force; (*b*) movements and ground forces.

5. MIXED DYNAMICS

Because of the lack of predictability of forward dynamics, it may be worthwhile to eliminate some of the equations from the set, by presuming that the motion of some parts of the system (generalized coordinates) is known. For example, you may study the dynamics of walking, but you want to include some type of arm swing. The angles of hands, lower arms and upper arms relative to their parent elements can be tabulated as a time-varying list or as explicit functions of time. The accelerations of these angles can be calculated and multiplied with matrix A, which is a matrix containing the inertial properties of the system. In this way the rows and columns dealing with these known degrees of freedom disappear from the matrix by suitable rearrangement of the terms, and the system can still be solved for the unknown accelerations. Thus, what happens is the exclusion of the equations dealing with known motion from the forward simulation, but in the meantime including their inertial effects in the remaining equations. I have termed this method 'mixed dynamics' and it can be helpful in establishing the dynamic importance of movements of parts of the system for the dynamics of the remainder. One should, however, be cautious with this method, especially during impulses applied to the system or during collisions. Impossibly high moments of force may be required at the joints of the elements of which the motion is prescribed. This can easily be verified by inspecting the calculated moments of force and comparing them with known maximal joint moments of force. When these moments of force are supposed to originate from active muscle force, this verification should be done in the light of the physiological rise time of muscle force, which is between 50 and 100 ms in humans. When those requirements are met, mixed dynamics simulations can be helpful in understanding the motion and control of multi-body systems.

6. VALIDATION

The validation of a simulation is crucial before applying it in any piece of research. Fortunately, the laws of physics provide conservation laws that should be met by any simulation of the dynamics of multi-body systems. These are the laws of conservation of energy, momentum and angular momentum. When no work is generated (for instance by elastic connections such as muscles) or dissipated (for instance by damping elements such as muscles) and the system is suspended in a conservative force field such as gravity, these laws should hold. A part of the simulation should be devoted to testing whether the results comply with these laws. Naturally there will be small deviations due to rounding up of errors, but that can be tested by decreasing the tolerance for the integration of the differential equations or by going form 4 byte reals to 8 byte reals. When the error decreases, and tends to go to zero when the time-steps approach zero, the simulations comply with those conservation laws. Another type of validation is to measure the motion of a system, such as a chain of elements, and compare that to a forward dynamics simulation with the masses, inertias and geometry included. Finally, it can be helpful to compare simulation results between packages or approaches. For instance one may write a simulation using the Newton–Euler approach and the Lagrangian approach and compare the results. Although the mathematical route may be entirely different, the results should agree in detail. A further point of validation should be mentioned: that of the application of the model to live organisms. Clearly, the model is a simplification and so it is of importance to predict, for instance, ground forces from a simulation and inspect just how well they correspond with the measured values.

7. CARDAN ANGLES AND GIMBAL LOCK

Usually simulations of 3D systems use the cardan angle convention in which the orientation of a solid body is represented by three sequential rotations about global or local coordinate axes. There are several advantages and disadvantages to this convention. First, there is the advantage of ease of visualization for independent rotations about any of the coordinate axes. In fact, the anatomical terminology can match the cardan angle convention in its description of abduction and adduction, flexion and extension and pronation and supination. There are two disadvantages: the first is that combinations of rotations soon result in hard to visualize orientations. The second is gimbal lock. This phenomenon occurs when the rotation over the second axis is 90° plus or minus multiples of 180°. In this event, the third coordinate axis becomes aligned with the first axis and one rotational freedom is locked, hence its term 'gimbal lock'. A gimbal is used in the suspension of a compass to keep it horizontal

on a ship. Another familiar example is the tri-axial head on a tripod used by photographers. As soon as the camera is aimed at the ground by tilting forward over 90°, it can no longer be aimed sideways because of a gimbal lock. That is the reason some photographers prefer a ball head in which the rotations are not sequential but simultaneous. This can also be described by an angular rotation over an axis in any direction. That concept was formalized mathematically by Hamilton in the form of quaternions (Hamilton 1844). Quaternions have become very popular in computer animations and simulations. It is possible to write the Newton–Euler and Lagrange equations using quaternions. For the simulation of 3D dynamics it can be an advantage to use quaternions, but it is not strictly necessary. Exact gimbal lock happens rarely and in the neighbourhood of it precautions can be taken during the simulation to avoid large integration errors. This is necessary because some of the cardan angular velocities can become rather large close to the gimbal lock.

8. COLLISIONS

In forward dynamics simulations, collisions can be included, for instance those occurring between feet and floor. Such collisions can be modelled in three ways.

(i) By continuously checking for the intrusion of elements in a solid object (such as the floor) and creating a reaction force depending on the elastic properties of the floor–foot combination, their damping properties and the distance and velocity of intrusion.
(ii) By continuously checking for intrusion of elements in a solid object and stepping back one time-step. By decreasing the time-step to fit the moment of impact and by formulating collision equations. By solving the system for the collision and to continue the normal simulations but now using extra equations to simulate the newly formed contact point.
(iii) By using a so-called 'impulse based method' (Mirtich 1996) in which both collisions and contact constraint are modelled in the same way: by including a large number of small impulses at the contact interface between the bodies.

Method (i) is easiest to implement, because only an administration of collisions needs to be kept while known external forces are easy to implement, both in the Newton–Euler approach and the Lagrangian approach. However, when the integration time-step is too large and the stiffness of contact quite high, this method results in numerical instability.

Method (ii) is quite straightforward to implement, because the collision equations are the same as the normal equations of motion, but now forces are interpreted as impulses and accelerations as instantaneous changes in generalized velocities. In other words: the collision is modelled in a single time-step in which all generalized velocities are changed. This gives quite realistic simulations as long as the contact stiffness is not too low. In that case method (i) is to be preferred, extending the collision over many integration time-steps.

Method (iii) looks very promising and is relatively easy to implement. However, the integration time-step needs to be very small to avoid a slow drift in position and orientation of the elements in contact.

In the Newton–Euler approach with constraint equations and in the Lagrangian approach, the joints are modelled as very precise hinges or ball-and-socket joints. However, elastic tissue in joints may provide another propagation of impulses than that modelled by these approaches. For some studies this may be of importance, but it should be pointed out that impedance due to flexion of joints is modelled by these precise joint approaches and forms the main source of axial impedance in humans while standing. This is due to the non-alignment of the joints with impulse vectors from the ground.

Another omission in the above approaches is the elastic suspension of some organs and tissue in the human (or animal) body, such as the intestines and layers of fat. During impulsive forces these mass-spring-damper systems can have a major influence on the dynamics of the system. These systems can be added to multi-body system simulations.

9. MUSCLE MODELS

The inclusion of muscle models in simulations can be very revealing, especially in studies of movement control. Several properties of muscle models can be distinguished.

(i) Shape of the line of action between origin and insertion: straight or curved around other elements.
(ii) Inclusion of length-dependent force generation, both active and passive.
(iii) Inclusion of velocity-dependent force generation.
(iv) Inclusion of activation dynamics, resulting in more complex time-dependent force generation.
(v) Inclusion of a series elastic element.
(vi) Inclusion of force-dependent pressure generation.

These properties can be mixed depending on the purpose of the simulation. However, more realistic modelling of the shape of lines of action and the inclusion of series elastic elements requires more sophisticated mathematics and processor time.

Interestingly, muscle models are easier to implement in forward dynamics than in inverse dynamics. This is mainly due to the problem of splitting up of moments of force at the joints, as calculated in the inverse dynamics process, into muscle moments of force contributions. This is impossible without extra information on muscle activations, such as measured by using electromyography. Scaling and linear programming can be used to perform the distribution. In forward dynamic simulations this problem does not exist. Moreover, when series elastic elements are simulated, the differential equations involved can simply be part of the group of differential equations of motion of the system and solved accordingly. The inclusion of series elastic elements in muscle models shows that both the attainable rise time in muscle force (Otten & Hulliger 1994) and the dynamics of the system depend on it.

10. APPLICATIONS

There are numerous applications of inverse and forward simulations. Because the methods only translate one set

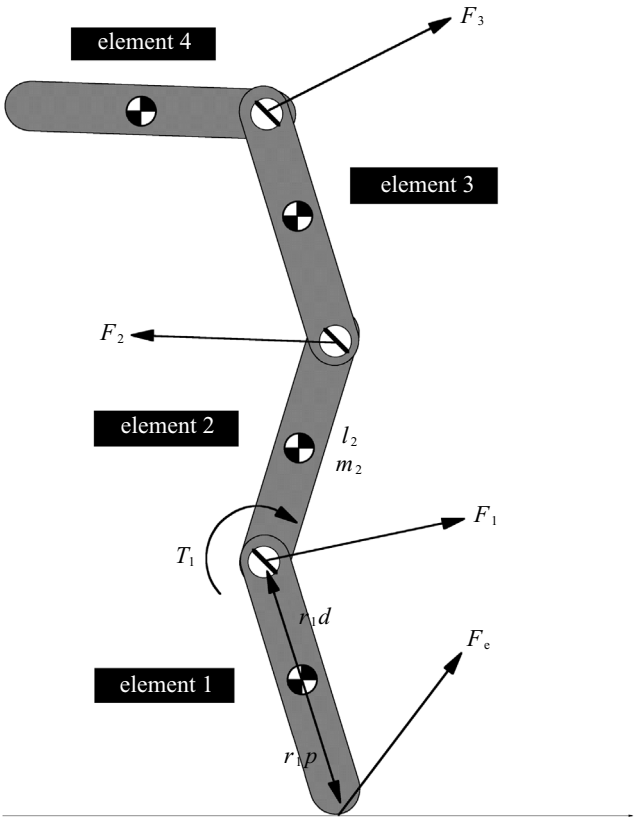

Figure 3. A planar system of four linked elements with joint torques and forces.

of time varying parameters into another set, one should be careful in creating yet another wealth of data that are left unexplained. A strong theoretical background should support these kinds of simulations so that the calculated curves can be interpreted. An example of such an application is found in the explanation of balancing on a narrow ridge, in which the centre of mass of a person sways outside the surface of support (Otten 1999). In this study, both inverse and forward dynamics simulations are used to explain and study the effect of changes in joint moments of force. Here, a 15-segment, 40 d.f. Lagrangian approach was used, whereas the control was left at the level of joint moments of force and not at the level of individual muscles.

APPENDIX A

A planar system of four elements linked in a serial chain with one contact point with the external world (figure 3).

Both the forward and inverse dynamics of this system will be solved using the Newton–Euler approach with constraint equations. All equations can be put in matrix form. In forward dynamics, the following variables are unknowns: F_1x and F_1y are the two components of the force acting from element 1 on element 2 by the joint. F_2 and F_3 are the forces on element 2 and element 3, respectively coming from elements 3 and 4, respectively. Variables $a1x$ and $a1y$ are the linear acceleration components of element 1. Vectors ***a2***, ***a3*** and ***a4*** are the accelerations of elements 2, 3 and 4. Variables *omd*1, *omd*2, *omd*3 and *omd*4 are the angular accelerations of elements 1, 2, 3 and 4. Variables *Fex* and *Fey* are the two force components acting on element 1 from the external world. This gives a total of 20 unknowns. Thus, to solve this system we need 20 equations. Indeed it is possible to formulate these equations. The system can be expressed as

$$\boldsymbol{A} \times \boldsymbol{X} = \boldsymbol{C},$$

in which $\boldsymbol{X}$ is the vector of unknowns, $\boldsymbol{A}$ the matrix of coefficients and $\boldsymbol{C}$ the vector of knowns. For this system, $\boldsymbol{A}$ is a 20×20 matrix, as shown in figure 4.

The constants $m1 \ldots m4$ are the masses of the elements. The constants $I1 \ldots I4$ are the second moments of inertia of the elements. The vectors ***r2p***, ***r3p*** and ***r4p*** are the vectors pointing from the centres of mass to the joints with the more proximal or parent elements. The vectors ***r1d***, ***r2d*** and ***r3d*** are the vectors pointing from the centres of mass of the elements to the joints with the more distal or child elements. Those vectors can be calculated from the angles of the elements in the global coordinate system and their length. For instance using the following expressions:

$$\boldsymbol{r1px} = -L1/2 \times \cos(\alpha 1),$$

$$\boldsymbol{r1py} = -L1/2 \times \sin(\alpha 1),$$

$$\boldsymbol{r1dx} = L1/2 \times \cos(\alpha 1),$$

$$\boldsymbol{r1dy} = L1/2 \times \sin(\alpha 1),$$

in which $\alpha 1$ is the angle of element 1 with the external world and $L1$ is the length of element 1. The vector ***re*** is pointing from the centre of mass of the first element to the contact point with the external world.

$\boldsymbol{C}$ is a vector 20 elements long:

1	flx
2	–9.8*ml+fly
3	f2x
4	–9.8*m2+f2y
5	f3x
6	–9.8*m3+f3y
7	f4x
8	–9.8*m4+f4y
9	–om2*om2*r2px+om1*om1*r1dx
10	–om2*om2*r2py+om1*om1*r1dy
11	–om3*om3*r3px+om2*om2*r2dx
12	–om3*om3*r3py+om2*om2*r2dy
13	–om4*om4*r4px+om3*om3*r3dx
14	–om4*om4*r4py+om3*om3*r3dy
15	+T1
16	–T1+T2
17	–T2+T3
18	–T3
19	om1*om1*rex+acx
20	om1*om1*rey+acy

Vectors ***f1***…***f4*** are forces acting on element 1…4 from the external world. $T1 \ldots T3$ are moments of force or torques acting at the three joints between the elements. The variables $om1 \ldots om4$ are the angular velocities of the four elements. The known components *acx* and *acy* are the accelerations at the contact point of the system with the external world. This provides us with 20 equations.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
	Flx	Fly	F2x	F2y	F3x	F3y	a1x	a1y	a2x	a2y	a3x	a3y	a4x	a4y	omd1	omd2	omd3	omd4	Fex	Fey
1	1	0	0	0	0	0	–m1	0	0	0	0	0	0	0	0	0	0	0	1	0
2	0	1	0	0	0	0	0	–m1	0	0	0	0	0	0	0	0	0	0	0	1
3	–1	0	1	0	0	0	0	0	–m2	0	0	0	0	0	0	0	0	0	0	0
4	0	–1	0	1	0	0	0	0	0	–m2	0	0	0	0	0	0	0	0	0	0
5	0	0	–1	0	1	0	0	0	0	0	–m3	0	0	0	0	0	0	0	0	0
6	0	0	0	–1	0	1	0	0	0	0	0	–m3	0	0	0	0	0	0	0	0
7	0	0	0	0	–1	0	0	0	0	0	0	0	–m4	0	0	0	0	0	0	0
8	0	0	0	0	0	–1	0	0	0	0	0	0	0	–m4	0	0	0	0	0	0
9	0	0	0	0	0	0	1	0	–1	0	0	0	0	0	–r12y	r21y	0	0	0	0
10	0	0	0	0	0	0	0	1	0	–1	0	0	0	0	r12x	–r21x	0	0	0	0
11	0	0	0	0	0	0	0	0	1	0	–1	0	0	0	0	–r22y	r31y	0	0	0
12	0	0	0	0	0	0	0	0	0	1	0	–1	0	0	0	r22x	–r31x	0	0	0
13	0	0	0	0	0	0	0	0	0	0	1	0	–1	0	0	0	–r32y	r41y	0	0
14	0	0	0	0	0	0	0	0	0	0	0	1	0	–1	0	0	r32x	–r41x	0	0
15	–r1dy	r1dx	0	0	0	0	0	0	0	0	0	0	0	0	–11	0	0	0	–rey	rex
16	r2py	–r2px	–r2dy	r2dx	0	0	0	0	0	0	0	0	0	0	0	–12	0	0	0	0
17	0	0	r3py	–r3px	–r3dy	r3dx	0	0	0	0	0	0	0	0	0	0	–13	0	0	0
18	0	0	0	0	r4py	–r4px	0	0	0	0	0	0	0	0	0	0	0	–14	0	0
19	0	0	0	0	0	0	1	0	0	0	0	0	0	0	–rey	0	0	0	0	0
20	0	0	0	0	0	0	0	1	0	0	0	0	0	0	rex	0	0	0	0	0

Figure 4. A matrix representing the dynamics of an open planar four-segment system.

The equations (1)–(8) are the dynamical equilibria of linear acceleration (Newton's second law). Equations (9)–(14) are the constraint equations demanding that the accelerations at the joints between the elements are equal. Equations (15)–(18) are the dynamical equilibria of angular acceleration (Euler's law). Finally, equations (19) and (20) are constraint equations of the contact point of the first element with the external world. These equations can be solved when matrix A can be inverted. Solving these equations needs to be carried out by the ordinary differential equations solver, because the equations produce linear and angular accelerations. These can be integrated over time to give linear and angular velocities. In turn, they can be integrated to give positions and angles of the system.

These equations can also be used in the inverse dynamics simulation. In that case A needs to be rearranged: columns 15–18 need to be changed into columns containing the coefficients of the unknown torques acting on the elements, whereas the components 15–18 of C will contain the known angular accelerations. Again the system can be solved and need not be integrated. As a function of time the torques and the external forces will be known. In addition, the linear accelerations of the elements are also produced, which are necessarily linked to the accelerations of the contact point and the angular accelerations and angular velocities, which are known in the inverse dynamical case.

Solving the inverse dynamics case in this way is by using 'brute force'. The advantage, however, is that the equations are the same as in the forward dynamics case. It also provides a clear overview of the system. Moreover, topological changes can easily be inserted and visualized in the matrix.

Let us consider a four-element chain system in a closed loop, namely a four bar mechanism with one contact point with the external world.

A joint is added between elements 1 and 4. Matrix A is now extended to the form shown in figure 5.

The vector of knowns ***C*** looks like this:

```
1     flx
2     –9.8*ml+fly
3     f2x
4     –9.8*m2+f2y
5     f3x
6     –9.8*m3+f3y
7     f4x
8     –9.8*m4+f4y

9     –om2*om2*r2px+om1*om1*r1dx
10    –om2*om2*r2py+om1*om1*r1dy
11    –om3*om3*r3px+om2*om2*r2dx
12    –om3*om3*r3py+om2*om2*r2dy
13    –om4*om4*r4px+om3*om3*r3dx
14    –om4*om4*r4py+om3*om3*r3dy
15    –om1*om1*r1px+om4*om4*r4dx
16    –om1*om1*r1py+om4*om4*r4dy

17    +T1
18    –T1+T2
19    –T2+T3
20    –T3

21    om1*om1*rex+acx
22    om1*om1*rey+acy
```

The same definitions have been used as in the former open chain model. However, two constraint equations have been added and the loop has been closed. This also affects many other equations, because forces and torques need to be administrated acting from element 1 on 4 and vice versa (see figure 5). These equations can readily be typed into any development environment and some graphics can be added so that the movements produced by the simulations can be seen.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
	Flx	Fly	F2x	F2y	F3x	F3y	F4x	F4y	a1x	a1y	a2x	a2y	a3x	a3y	a4x	a4y	omd1	omd2	omd3	omd4	Fex	Fey
1	1	0	0	0	0	0	–1	0	–m1	0	0	0	0	0	0	0	0	0	0	0	1	0
2	0	1	0	0	0	0	0	–1	0	–m1	0	0	0	0	0	0	0	0	0	0	0	1
3	–1	0	1	0	0	0	0	0	0	0	–m2	0	0	0	0	0	0	0	0	0	0	0
4	0	–1	0	1	0	0	0	0	0	0	0	–m2	0	0	0	0	0	0	0	0	0	0
5	0	0	–1	0	1	0	0	0	0	0	0	0	–m3	0	0	0	0	0	0	0	0	0
6	0	0	0	–1	0	1	0	0	0	0	0	0	0	–m3	0	0	0	0	0	0	0	0
7	0	0	0	0	–1	0	1	0	0	0	0	0	0	0	–m4	0	0	0	0	0	0	0
8	0	0	0	0	0	–1	0	1	0	0	0	0	0	0	0	–m4	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	1	0	–1	0	0	0	0	0	–r12y	r21y	0	0	0	0
10	0	0	0	0	0	0	0	0	0	1	0	–1	0	0	0	0	r12x	–r21x	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	1	0	–1	0	0	0	0	–r22y	r31y	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	1	0	–1	0	0	0	r22x	–r31x	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	1	0	–1	0	0	0	–r32y	r41y	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	–1	0	0	r32x	–r41x	0	0
15	0	0	0	0	0	0	0	0	–1	0	0	0	0	0	1	0	r11y	0	0	–r42y	0	0
16	0	0	0	0	0	0	0	0	0	–1	0	0	0	0	0	1	–r11x	0	0	r42x	0	0
17	–r12y	r12x	0	0	0	0	r11y	–r11x	0	0	0	0	0	0	0	0	–11	0	0	0	–rey	rex
18	r21y	–r21x	–r22y	r22x	0	0	0	0	0	0	0	0	0	0	0	0	0	–12	0	0	0	0
19	0	0	r31y	–r31x	–r32y	r32x	0	0	0	0	0	0	0	0	0	0	0	0	–13	0	0	0
20	0	0	0	0	r41y	–r41x	–r42y	r42x	0	0	0	0	0	0	0	0	0	0	0	–14	0	0
21	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	–rey	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	rex	0	0	0	0	0

Figure 5. A matrix representing the dynamics of a closed planar four-segment system.

REFERENCES

Featherstone, R. 1987 *Robot dynamics algorithms*. Dordrecht, The Netherlands: Kluwer.

Feynmann, R. F., Leighton, R. B. & Sands, M. L. 1964 *The Feynmann lectures on physics*. Reading, MA: Addison-Wesley.

Hamilton, W. R. 1844 On a new species of imaginary quantities connected with a theory of quaternions. *Proc. R. Irish Acad.* **2**, 424–434.

Huston, R. L. 1990 *Multibody dynamics*. Boston, MA: Butterworth–Heinemann.

Kane, T. R. & Wang, C. F. 1965 On the derivation of equations of motion. *J. Soc. Indust. Appl. Math.* **13**, 487–492.

Mirtich, B. 1996 Impulse-based dynamic simulation of robotic systems. PhD thesis, University of California, Berkeley.

Moravec, H. 1998 *Robot: mere machine to transcendent mind*. Oxford University Press.

Otten, E. 1999 Balancing on a narrow ridge: biomechanics and control. *Phil. Trans. R. Soc. Lond.* B **354**, 869–875. (DOI 10.1098/rstb.1999.0439.)

Otten, E. & Hulliger, M. 1994 A finite elements approach to the study of functional architecture in skeletal muscle. *Zool. Anal. Complex Syst.* **98**, 233–242.

Press, W. H., Flannery, B. P., Teukolsky, S. A. & Vetterling, W. T. 1986 *Numerical recipes*. Cambridge University Press.

Winter, D. A. 1990 *Biomechanics and motor control of human movement*. New York: Wiley.