Journal on Perceptual Control Theory
Vol 1 No 1, 1999

An experiment with a simple method for controlling an inverted pendulum

by

William T. Powers

Directory

Last Revision Date: March 16, 1999 ISSN 1437-9325

An experiment with a simple method for controlling an inverted pendulum

by

William T. Powers

A simulated inverted pendulum consisting of a bob on a shaft hinged to a movable cart is controlled by a hierarchy of 5 simple control systems. The simulation of the physical system treats the cart and bob as free masses connected by a spring acting in the direction of the shaft. No trigonometric functions are required. The user of the program can set a reference position with the mouse, after which the control systems move the cart carrying its balanced pendulum to place the bob at any selected position in the x direction. This is an experimental design, with no serious attempt to optimize performance. Nevertheless, performance is better than a human being could produce.

An experiment with a simple method for controlling an inverted pendulum

by

William T. Powers

1. Introduction

This paper addresses two subjects relating to simulation and control, the first being a simplified method for simulating a mechanical system and the second being a method for achieving control of an inverted pendulum. The methods may be suggestive of broader applications, and are presented here despite their incomplete form so that others may help extend the principles.

The test bed is a computer-simulated pendulum mounted upside down on a simulated movable cart. The shaft and bob are hinged to a cart that can roll in the x direction on rails. The plane of motion of the pendulum is vertical and includes the direction of motion of the cart. The first phase of this project involves simulating the pendulum itself in a way that appeals only to fundamental physical relationships rather than solutions of analytical equations. The second phase involves developing a control system consisting of subsystems that control successively higher time-integrals of physical variables until a level is reached that can control the position of the bob in the x direction. Each control subsystem is very simple.

2. Simulating the pendulum

The mechanical assembly being simulated consists of a rolling cart and a bob, each treated as a point-mass, and a shaft connecting the bob to a hinge on the cart. The bob weighs 1 Kg, the cart weighs 0.1 Kg, and the shaft is considered weightless. See the upper part of Fig. 1.

The cart is confined to motion along the x axis; the bob can move in two dimensions, x and y. The two masses are connected by the shaft, which is not considered rigid as in the usual physical approximations, but is treated as a spring with a resting length L0. The spring can extend or shorten, but does not bend. When the bob and the cart are in specific positions, the distance between their centers is the length L of the shaft, and in general this length implies a force of a magnitude

$$F = ke * (L - L0) \tag{1}$$

where

F = force in newtons,
ke = spring constant, newtons / meter
L = actual length of shaft, stretched or compressed,
L0 = resting length of shaft

The force generated by the spring acts along the direction of the shaft between the bob and the cart, pulling them together or pushing them apart. The result is to accelerate both objects: the cart to the left or right along its rail, and the bob in some direction in x-y space. Computing the acceleration is simplified by computing x and y acceleration separately.

For the bob, the x force Bob.fx is simply the ratio of the x displacement of the bob relative to the cart divided by the shaft length L, times the force in the direction of the shaft. Since the force along the hypotenuse of the triangle is known, we can compute the x and y forces using similar triangles instead of trigonometric functions. The acceleration is the force divided by the mass, Bob.Mass:

Bob.Ax = F*(Bob.x - Cart.x) / (Bob.Mass*L)

The y acceleration is

Bob.Ay = F*(Bob.y - Cart.y) / (Bob.Mass*L)

For the cart, the expressions are the same except that ”cart” is substituted for ”bob” in the variable names (Note 1).
We have the x and y accelerations of the bob and the x acceleration of the cart, so we can proceed to integrate once to get velocity and again to get position in both the x and y directions for both masses. The integration is done over a very short time-duration (here 0.0001 second) to get a new set of x and y positions for the bob and cart. Then, with the bob and cart in slightly different new positions, we can compute the new length of the shaft, new forces and accelerations, and new bob and cart positions to use during the next ten-thousandth of a second. This is the basic process of simulation in which we compute the new state of a system after a very short time interval, and then compute the new forces acting on the system during the next time interval. This process is repeated millions of times during a run of a simulation.

This permits us to simulate the behavior of the system without going through an abstract mathematical analysis. This method is very close to working with the physical system itself, and has the great advantage that nonlinear relationships are just as easy to work with as linear ones.

The computer program actually used calculates an added force dependent on the velocity of extension or contraction of the shaft; the amount of this ”viscous damping” force is selected to make any high-frequency oscillations of the masses at the ends of the spring damp out rapidly. The spring constant used for the shaft is about 10 million newtons per meter, so the shaft is very stiff: a weight of 1 kilogram hanging from the shaft would stretch it by about one micron or 25 millionths of an inch. The shaft is hardly distinguishable from the classic ”rigid rod” used in mathematical analysis of similar mechanical systems. But its non-rigidity makes all the difference in the analysis.

There are several practical advantages of this method of simulating a mechanical system. It is not necessary to find mathematical forms for all the relationships. No differential equations have to be

solved analytically. No trigonometric functions, which are slow to compute, are used. One point that does need investigation is how close this way of treating the physical system comes to an exact analytical representation of its behavior.

3. The Control Systems

Two sets of control systems are used. The first set positions the cart, and the second set positions the pendulum bob by using the cart-positioning systems.

A force applied to the cart in the x direction will cause it to accelerate, its velocity increasing at a constant rate for a constant applied force. In Fig. 1 , the smallest closed loop in the lower right corner is the cart velocity control system. For all control systems it is assumed that a suitable sensor for the controlled variable, here linear velocity, exists.

The sensed velocity in the x direction, Cart.vx, is compared with a reference velocity signal coming from above, and the error signal, the difference, is amplified and converted to a force applied to the cart. Everything in this loop responds proportionally except for the time-integration in the box, which represent the conversion of applied force to acceleration and the conversion of acceleration to velocity, which are basic physical relationships. This loop, with one integration in it, is inherently stable. If the output parameter (here, a factor

of 200) is large enough, the sensed velocity will closely track the reference velocity, shown entering the comparator (C) from above. The feedback involved in this control process will see to it that changes in the sensed velocity are nearly simultaneous with variations in the reference signal, so the control system as a whole behaves very nearly like a proportional link. It is this property of negative feedback control that makes the hierarchical control process so easy to stabilize.

The controlled velocity is integrated again to calculate the position of the cart – in effect, velocity is multiplied by elapsed time to get distance traveled (over a period of 0.0001 second). In the next hiigher control system, the sensed distance is compared with a reference distance signal by the second level comparator (C in a box), the error signal being amplified to become the reference velocity for the first-level system. Because the first-level integration has been made almost into a proportional response, the second loop can be made very sensitive without causing instability. In this case the error signal is multiplied by 200. The result is that the cart position follows the reference position signal very closely and quickly.

The cart, with the bob standing approximately vertically above it, must move in the direction of lean of the bob to create a lean in the other direction and slow the bob to a stop. To achieve this, the present strategy was first to get control of bob acceleration, use that to get control of bob velocity, and finally to use that to get control of bob position.

The acceleration of the bob is affected by the distance of the cart to the left or right underneath the bob. The first bob control system senses the cart position relative to the bob, and keeps it at whatever relative position is set by the reference signal. As the bob accelerates left or right, the cart also accelerates, keeping the angle of lean and the acceleration constant.

The bob acceleration is integrated (by the physics of nature) to generate the bob velocity. In the second bob control system, bob velocity is sensed and compared with a reference velocity, and the difference or error is amplified to produce the reference signal for the acceleration control system.

To establish a velocity to the right, the cart must move at first to the left, creating a lean of the bob to the right. Then, as the velocity

increases toward the reference velocity, the lean decreases (the cart moves right and catches up to the bob), and the bob then continues moving at the specified velocity while remaining upright above the cart. All this happens completely automatically; the cart is made to move left when the acceleration is too low, and right when it is too high, and that is all that is necessary to do.

Finally, bob velocity is integrated as in nature to produce bob position, and bob position is compared with a reference position to produce a position error signal. This error signal is amplified to generate the velocity reference signal, closing the final loop.

4. Limits of performance

The forces that balance the bob are actually produced by gravity. The control systems, by moving the supporting cart left and right underneath the bob, can direct these gravitational forces to create the necessary balancing forces. However, this can work only if the bob remains within some fairly small angle of the vertical over the cart (about plus and minus 30 degrees). Within this range, a leftward movement of the cart produces a rightward acceleration of the bob. As the bob moves out of this range, the direct effect of cart movements on the bob (through the shaft) begins to predominate. This can be seen by imagining the bob to have toppled over by 90 degrees so the shaft is parallel to the x axis. Now when the cart moves leftward, the bob can only move left (instead of right). Somewhere between the vertical and this 90 degree orientation, there is a transition from one sign to the opposite sign of the effect of moving the cart. Since the direct effect is opposite to the gravitational effect, there comes a point where the negative feedback in our control systems turns into positive – and very much larger – feedback effects. At that point

the program goes into runaway and halts when the variables head towward infinite values.

In a truly complete model, one that works as much like a human system as possible, runaway would not happen. Instead, a higher-level system would turn off the balancing control systems before they get into a runaway state, and substitute another system, perhaps one that swings the bob in large loops back and forth until it comes to a stop for a moment somewhere within the effective control range. Then balancing could be resumed. Real human beings behave just like this. If a disturbance moves the bob out of the critical range, the human being will start into a runaway process, but before it can go very far, a completely different mode of behavior will take the place of balancing.

Since this higher-level control is not part of this model, runaway is avoided here simply by limiting the output of the position control system, which limits the speed reference signal.

5. Performance of the simulation

When the program starts, the bob is initialized to a position 0.1 radian off the vertical, with the position reference signal set to zero and the cart at the zero of the x-axis. The computer mouse controls the position reference signal.

The cart immediately moves under the bob and the bob quickly comes into balance. It remains in balance indefinitely with no visible movement. See screenshot in Fig. 2.

Now the user can use the mouse to move the reference position to either side. On the screen, the cart immediately moves opposite to the mouse movement, making the shaft and bob lean the way the mouse went. The cart and bob accelerate to a constant velocity and coast for a while with the bob vertical again. Then, as the bob approaches the new reference position, the cart speeds up and gets ahead of the bob. The resulting backward lean decelerates the bob. As the bob approaches the reference position, the backward lean decreases, becoming zero just as the bob becomes stationary at (or very close to) the new reference position.

This behavior looks complex and programmed, but it is in fact neither. The complex motions are a natural result of the behavior of the organization shown in the lower half of Fig. 1. There are no tests for different logical conditions, and there are no logical rules in effect (as in ”fuzzy logic” controllers). This is a set of 5 very simple continuous analog controllers.

6. Conclusions

There were two primary objectives in constructing this stimulation. One was to test the idea of simulating mechanical systems in terms of fundamental physical laws, not employing ”rigid rods.” The other objective was to test the idea that complex control systems could be analyzed into hierarchical levels, with the lowest levels controlling the highest derivatives of the variables to be controlled. The results are encouraging in both cases.

In this preliminary effort, no attempt was made to find optimum parameter settings to get the greatest stability and speed possible. In fact this could be claimed as another positive result, for the idea was to see if there could be an approach to simulating control processes that bypasses all the complexities so often found in textbooks on this subject. The author, however, would be embarrassed to make that claim, since the truth is that complex methods of control system analysis are mostly beyond his abilities.

It may be, however, all such disclaimers aside, that the methods outlined here could be developed into a much more systematic and

useful approach to both simulation and control. Simulating physical systems in the usual way gets extremely complex and requires advanced mathematical abilities; the same applies to simulating and analyzing control systems. Any method that promises to simplify and streamline such analyses, while of little interest to mathematical geniuses, might well be worth developing for the sake of the rest of us. Anyone interested is warmly encouraged to help carry this exploration further.

William T. Powers
Durango, CO
May 24, 1998

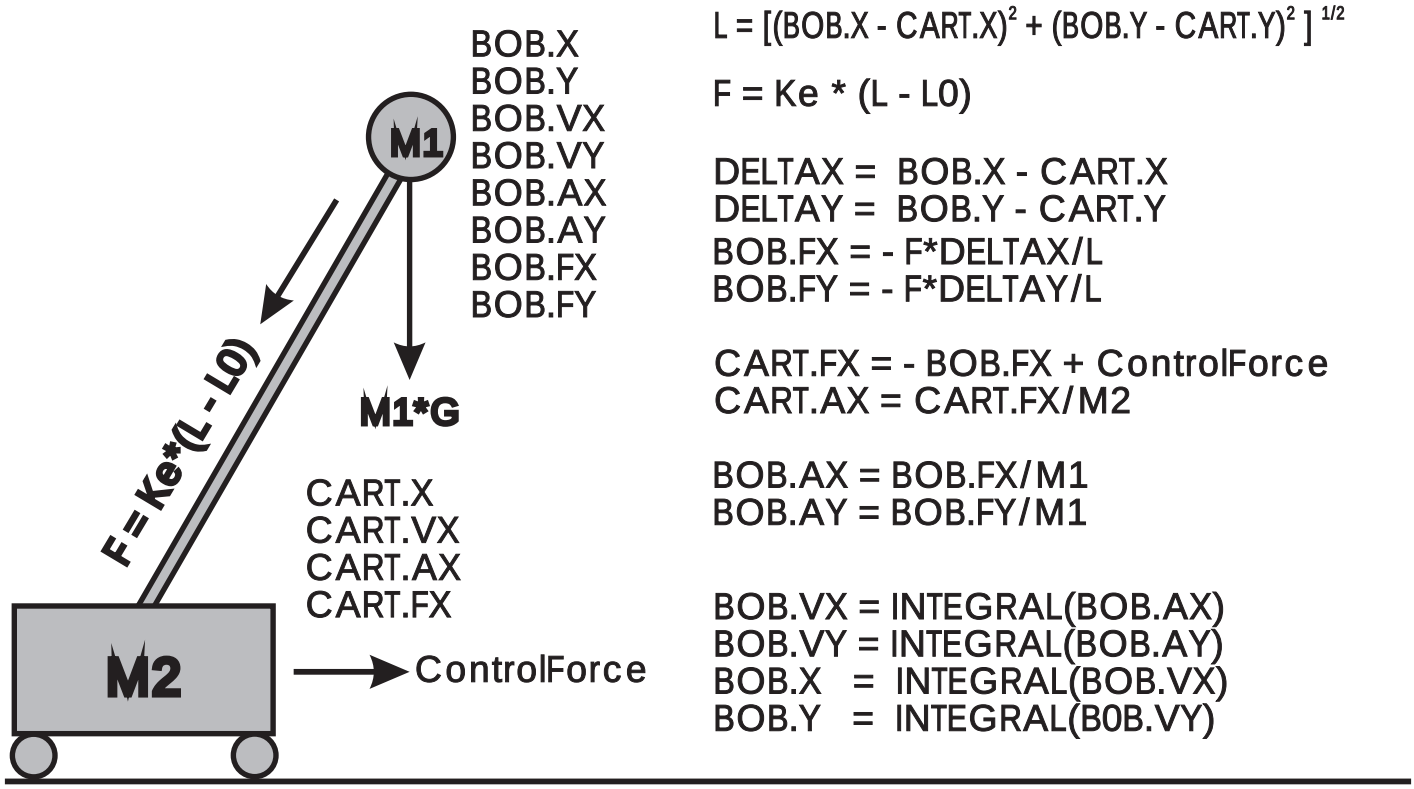
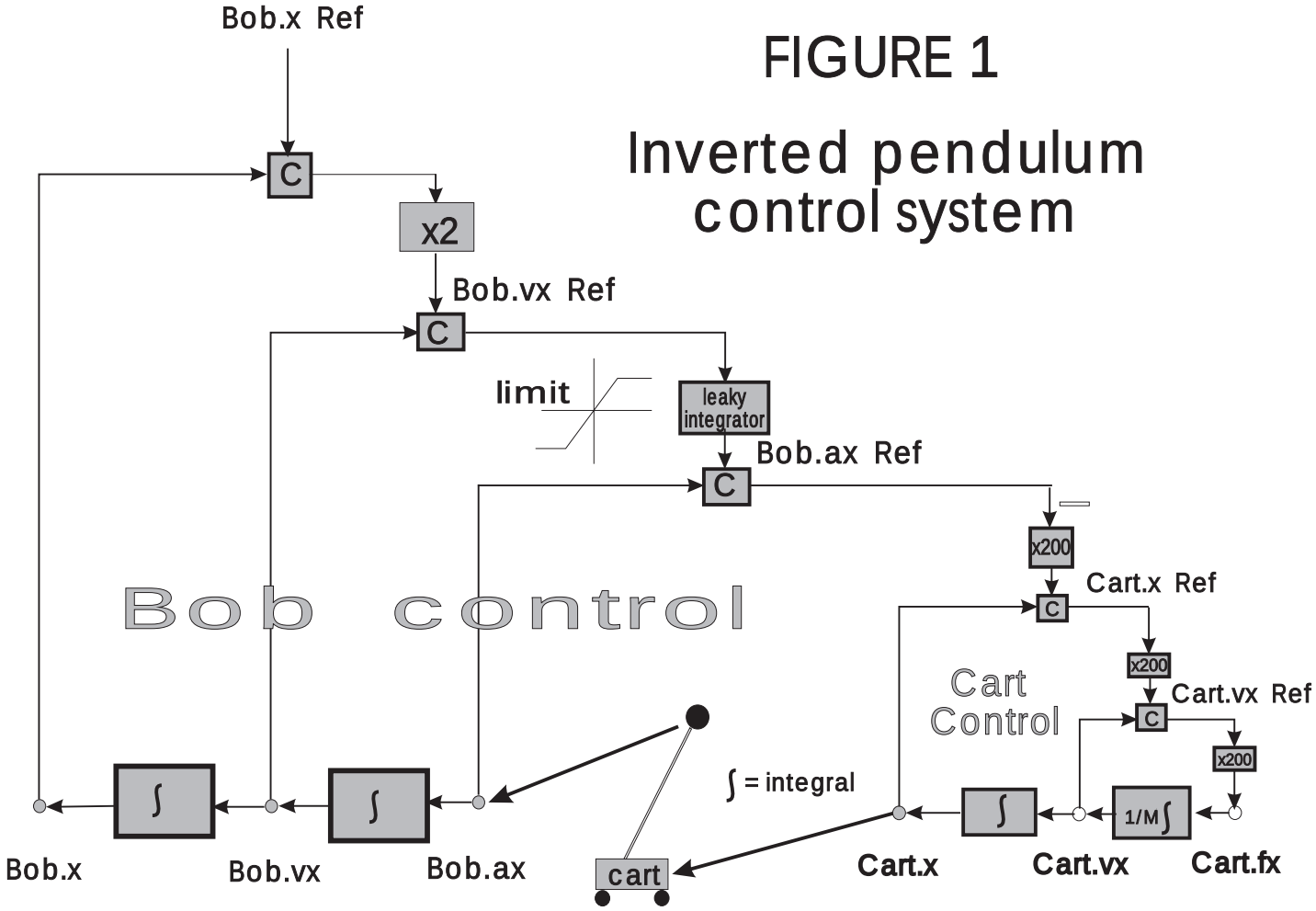


FIGURE 1

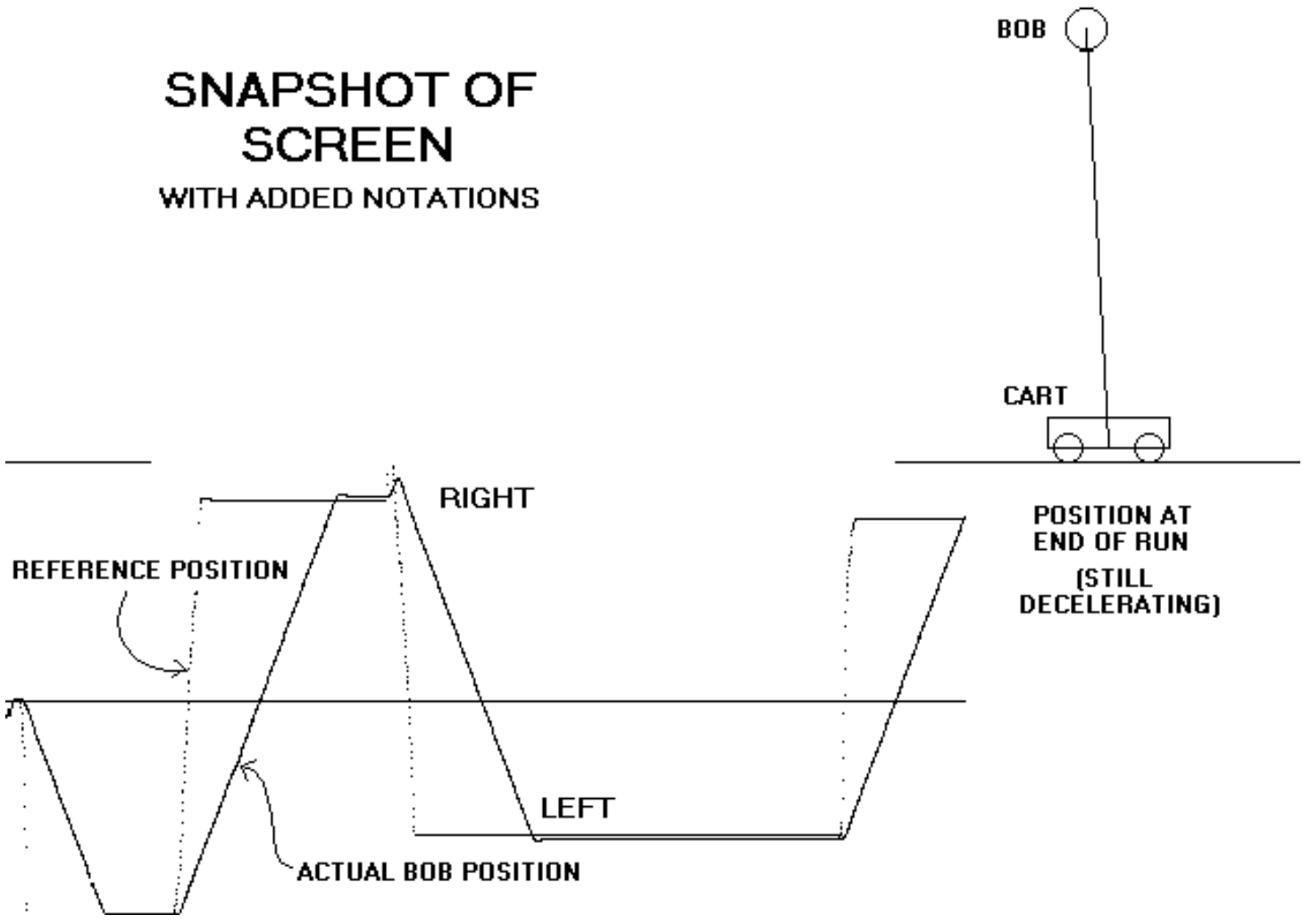
Inverted pendulum control system



PRESS ANY KEY TO QUIT

SNAPSHOT OF SCREEN

WITH ADDED NOTATIONS



The notation here is that of computer programming, not normal mathematics. The explicit multiplication sign (*) allows variables to be given multiple-letter names rather than being represented by single letters. Variables are grouped into ”records” which can contain lists of symbols. For example, the record named ”Bob” has subsymbols x,vx,ax,fx,y,vy,ay, and fy. The letter x indicates position, v is velocity, a is acceleration, and f is force. Thus Bob.ay means the y direction of acceleration of the Bob, Cart.vx means the x direction of velocity of the Cart, and Bob.y means the y position of the Bob. With this key, the equations in the upper part of Fig. 1 become self-explanatory.