

# Video Modeling: Combining Dynamic Model Simulations with Traditional Video Analysis

Douglas Brown

Cabrillo College, Aptos, CA [dobrown@cabrillo.edu](mailto:dobrown@cabrillo.edu)

## Introduction

The free Tracker video analysis program<sup>1</sup> allows users to overlay simple dynamic particle models on a video clip. In a typical video modeling experiment students capture and/or open a digital video file, calibrate the scale, and define appropriate coordinate axes just as for traditional video analysis. But instead of tracking objects with a mouse, students define theoretical force expressions and initial conditions for a dynamic model simulation that synchronizes with and draws itself on the video. The behavior of the model is thus compared directly with that of the real-world motion. Tracker uses the Open Source Physics<sup>2</sup> code library so relatively sophisticated models are possible.

This poster showcases video modeling experiments conducted by students in my Fall 2007 Introductory Mechanics course. The assignment was very open-ended: *Identify a mechanical system for which the behavior of the system can be (a) captured on video and (b) modeled as a set of one or more particles. Construct models, capture videos and compare. Students worked in small groups on this multi-week project in both scheduled labs and a required drop-in learning center with peer tutor and instructor support.*

## Building models

Students define dynamic models by entering force functions and initial conditions in a "Model Builder" as shown in Figure 1. Function expressions are interpreted and evaluated using a parser which recognizes parameter and variable names, mathematical and logical operators, common math functions and *if* statements. The motion of the models is computed with an ODE solver using a Runge-Kutta algorithm with an adaptive step size.

mass  $m$  is a built-in parameter

students may define additional parameters (names and values) as desired

parameter and initial value expressions may be numbers or functions of parameters

force expressions are functions of motion variables ( $t, x, y, vx, vy$ ) and parameters

students can undo/redo all edits

Figure 1. Model Builder for a monkey falling with a parachute

## Monkey with a parachute

One student group updated the traditional monkey/hunter demo by substituting a coffee filter for the falling monkey. Before trying to shoot this "monkey with a parachute," they modeled it as a particle experiencing gravitational and drag forces as shown in Figure 1. Using a similar model "dart," they then predicted the initial velocity angle needed to hit the monkey. After making their prediction, they hit the real coffee filter on the first try (Figure 2)!

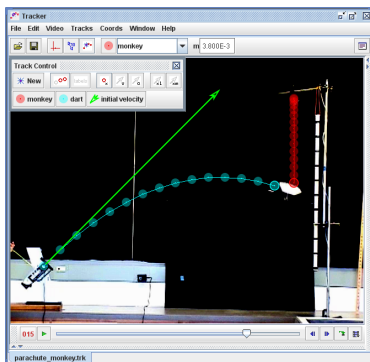


Figure 2. Shooting a monkey falling with a parachute

## Spring wars

Here students modeled a cart being pulled by opposing springs and slowed by friction. The Model Builder shows the use of a separately defined frictional force " $f$ " and an *if* statement to specify its direction (Figure 3). The model displays a linear amplitude decay that agrees well with the motion of a real cart captured on video and marked with a mouse (Figure 4).

Name	Expression
$f_x$	$-k_1 * x_1 + k_2 * x_2 + f * \text{sign}(v_x)$
$f_y$	$\mu * m * g$

Figure 3. Use of an *if* statement

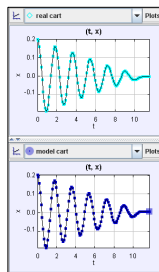


Figure 4. Frictional damping

## Bouncing cart

This group modeled a cart bouncing from a spring at the lower end of a tilted air track. The " $f_x$ " force expression uses an *if* statement to apply the spring force only when the cart has reached the spring position " $x_s$ " (Figure 5). The model cart was placed just above the real cart in the video for easy comparison as seen in Figure 6. The decreasing bounce heights were assumed to result from friction between the cart and track rather than from an imperfect spring.

Parameters		
Add	Copy	Paste
Name	Expression	
$m$	$0.5$	
$k$	$0.5$	
$x_s$	$0.5 * \pi * 0.08$	
$c$	$0.01$	
$\mu$	$0.258$	
$\mu_0$	$0.0075$	
$f_0$	$0.0001 * \text{Data}$	

Initial Values		
Add	Copy	Paste
Name	Expression	
$s$	$0.0$	
$v$	$0.0$	
$a$	$0.0$	
$\omega$	$0.0$	
$\omega_0$	$0.0$	

Force Functions		
Add	Copy	Paste
Name	Expression	
$f_x$	$-m * g * \sin(\theta) + k * (x - x_s) + c * v_x$	
$f_y$	$-m * g * \cos(\theta) + \mu * m * g$	

Figure 5. Model Builder

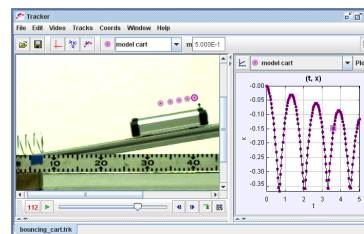


Figure 6. Comparing the model (purple) with a real cart

## Accelerating cart pendulum

What is the motion of a ball suspended from an accelerating cart? My students first discovered "experimentally" that the model ball's acceleration was sinusoidal—then were able to relate the motion theoretically to a pendulum in a skewed gravitational field!

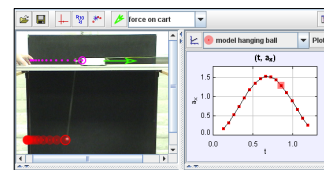


Figure 7. Ball suspended from an accelerating cart.

## Discussion

Computer modeling that engages students directly in the development of physical models is becoming an increasingly important part of physics education<sup>3</sup>. Dynamic particle models based on Newton's laws provide a natural first exposure to the modeling process for students in introductory physics.

Other modeling tools such as Easy Java Simulations<sup>4</sup> also give students a relatively simple way to build particle (and many other) models and to control virtually every aspect of the modeling process. While Tracker's particle models are more limited, its ability to compare the models with real-world videos combines features of both traditional video analysis<sup>5,6</sup> and animation-only modeling and offers these additional advantages:

1. Students study real objects but focus on identifying forces rather than tracking and analyzing motion.
2. Visual comparisons are rapid, intuitive and capable of discerning fine differences.
3. Visual agreement may validate models more convincingly than numerical or graphical comparisons for beginning students.
4. Visual disagreement leads naturally to explorations of limitations and refinements of models.
5. Quantitative analysis of model and/or video data can support and extend the visual comparison to any degree desired.

## Acknowledgements

Many thanks to Wolfgang Christian and the Open Source Physics team Mario Belloni, Francisco Esquembre, Anne Cox and Bill Junkin for their invaluable expertise and guidance. The Open Source Physics project is supported in part by the National Science Foundation grant DUE-0442481.

## References

1. Tracker: <http://www.cabrillo.edu/~dbrown/tracker/>
2. Open Source Physics: <http://www.opensourcephysics.org/>
3. W. Christian and F. Esquembre, "Modeling Physics with Easy Java Simulations," *Phys. Teach.* **45**, 475-480 (Nov 2007).
4. Easy Java Simulations: <http://www.um.es/fem/Ejs/>
5. D. Brown and A. J. Cox, "Innovative Uses of Video Analysis," *Phys. Teach.* (early 2009)
6. P. Laws and H. Pfister, "Using Digital Video of Video in Introductory Mechanics Projects," *Phys. Teach.* **36**, 282-287 (May 1998).