HYBRID STATE CHART MODELLING FOR NONLINEAR BOUNCING BALL DYNAMICS

Rouzbeh Karim[1], Pavol Bauer[1], Bernhard Heinzl[1], Matthias Rößler[1], Günter Schneckenreither[1], Andreas Körner[1], Katharina Breitenecker[1], Günther Zauner[2], Nikolas Popper[2]

[1]Vienna University of Technology, Institute for Analysis and Scientific Computing
Wiedner Hauptstraße 8-10, 1040 Vienna, Austria
[2]dwh Simulation Services,
Neustiftgasse 57-59, 1070 Vienna, Austria

e9427229@student.tuwien.ac.at (Rouzbeh Karim)

Abstract

This contribution highlights two aspects of the classical bouncing ball modeling and simulation. On modelling level, the ball characterises more a big bubble than a ball, so that drag forces cannot be neglected and impacts must be modelled by distortion, and additionally the implication of the big-sized ball for different atmospheres (comparing Earth and Mars). What looks like fun – may be used for education in modelling and simulation, and may become serious science once. (http://saturn.astrobio.net/pressrelease/63/having-a-ball-on-mars)

On implementation level, the presented MATLAB/Stateflow version is a purely discrete approach: fly and distortion are modelled by state charts, updated by triggered stepsize events which drive an ODE solver. Switching between impact, distortion, and fly is triggered by state events following a predictive event finding strategy: as the event function is the state ball height, also first and second derivative of event function are known. Thus allows approximating the event function by a polynomial of first order or second order near the impact, the known zero of which may be used for adjusting a smaller step size, and in last consequence, to give the impact time. The idea may be generalized for contact problems in mechanical systems.

Keywords: Bouncing Ball, State Flow, State Charts, Hybrid Modelling.

Presenting Author's biography

Pavol Bauer is working on a master degree in Medical Informatics and Modeling and Simulation, currently writing his thesis at TU Vienna, ARGESIM Group. His research focus emphasis is on hybrid models, cellular automata and agent based modeling. He is involved in a project on discrete modeling approaches, supported by state chart descriptions.

1 Introduction

A Google-search for ‘bouncing+ball model’ results in about 550.000 hits – raising two questions: what else can be published on this topic, and why this small model makes so much rumour?

The answer to the second question may be the fact, that the bouncing ball model (BBM) is not only a model for a bouncing ball, it is the simplest but most realistic illustration for the concept of entropy change as a result of the redistribution of energy in a system to available microstates in case of a state-dependent rebound. In simple versions, the ball is a point particle, more complex ball models design the ball as agglomeration of particles, which are hooked (with spring and damper), the bounce itself can be seen as simple event, changing velocity discontinuously, or as begin of distortion of the ball (with spring and damper); whereby one- or more-particle ball models are used. And from application viewpoint, the BBM is a simplification from rather complex processes, like airbag dynamics, etc. From classification point of view, the BBM is the simplest model for contact problems, which all have the same problem: the condition for the contact is state-dependent and the state is not known analytically, so that contact time cannot be determined in advance.

The answer to the first question may be the assumption, the presented approach is partly a novelty.

The following considerations highlight two aspects of BBM simulation. On modelling level, the ball characterises more a big bubble than a ball, so that drag forces cannot be neglected and impacts must be modelled by distortion, and additionally the implication of the big-sized ball for different atmospheres (comparing Earth and Mars). What looks like fun – may be used for education in modelling and simulation, and may become serious science once.
(http://saturn.astrobio.net/pressrelease/63/having-a-ball-on-mars)

The second aspect deals with the implementation of the BBM. Classical implementations follow ODE modelling with state events modelling. Accurate state event handlers are forced to intervene into the ODE solver, to step back in time and/or to calculate with physical wrong equations - in order to determine the impact time by iteration or by interpolation. The presented implementation is a purely discrete approach: fly and distortion are modelled by state charts, updated by triggered stepsize events, which update the state by an ODE algorithm of the user’s choice. Impact/begin of distortion and end of distortion/restart of fly are triggered by events, which switch between the fly model with drag force (BBFM) and the distortion model (BBDM), and which are determined approximately in advance.

The idea behind is the fact, that in case of BBM, not only the event function can be evaluated, but also its derivative(s). The event function, the zero of which must be found, is the ball height – a state variable; but also the velocity – another state – is available, and, if necessary also the acceleration – the derivative of a state. This allows approximating the event function by a polynomial of first order or second order near the impact, the known zero of which may be used for adjusting a smaller step size, and in last consequence, to give the impact time. This contribution illustrates a basic implementation of the suggested state chart approach and event algorithm in MATLAB/Stateflow®. The idea may be generalized for contact problems in mechanical systems, where these derivates are available.

A bouncing ball is a hybrid model, and it is an example whose model consists on the one hand of a motion equation and on the other hand of a force equation. This paper introduces the ball and its bouncing behaviour with its drag force.

In aerospace science there are some models with airbags, whereas they have a model which is similar to a bouncing ball hybrid model, which - with some modification - can be used as a tiny part of a complex aerospace mission. The idea of landing exploration robots on other planets (Mars, for example) opens up new horizons in knowledge and science when such operations can be achieved in reality, for example MER atmospheric entry [1, 2]. In the final phase of landing of some aerospace exploration robots, the robot (capsule), which is centred in airbags with definite stiffness and damping characteristics, should be launched (near to the ground) from a parachute (or dropped from a beam). In this case, the destruction of the robot or lander, which is in the airbag, must be avoided.

The content of this paper below is not related to any aerospace applications; it merely introduces a very simple preface example to describe the model of a bouncing ball on Earth and on Mars.

2 Model Description

The first part of the simulation algorithm describes the kinematic relationships of the ball (falling of the ball). The two important relations are the velocity and acceleration differential equation. If the ball falls from a height h, then the velocity is

$$\dot{h} = v \tag{1}$$

where v is velocity, and acceleration a is given as

$$\dot{v} = a \tag{2}$$

The second part of the simulation algorithm concerns the dynamic equation of the ball.

In this part, drag force is considered. To determine drag force, the density of the atmosphere needs to be found, to compute the density of the atmosphere it is necessary to evaluate the temperature and pressure of the atmosphere [3, 4]. So the temperature θ (in degrees Celsius) on Earth can be defined by SI units as follows

$$\theta_e = 15.04 - 0.00649 \cdot h \tag{3}$$

where h is given in meters. The temperature of the atmosphere is used to calculate the atmospheric pressure. The pressure of the atmosphere P is given in kilo-Pascal units. The equation of the pressure of the atmosphere on Earth is given by

$$P_e = 101.29 \cdot \left[\frac{\theta_e + 273.1}{288.08}\right]^{5.256} \tag{4}$$

The equation of the density of the atmosphere ρ on Earth (5) can be seen below.

$$\rho_e = \frac{P_e}{\left(0.2869 \cdot \left(\theta_e + 273.1\right)\right)} \tag{5}$$

The unit of ρ is kg/m^3. These equations for Earth are valid for the troposphere layer or altitudes less than $h_e \prec 11000$ meters. The equations of temperature pressure and density for Mars are defined with the following functions:

$$\theta_m = -31 - 0.000998 \cdot h \tag{6}$$

$$P_m = 0.699 \cdot e^{-0.00009 \cdot h} \tag{7}$$

$$\rho_m = \frac{P_m}{\left(0.1921 \cdot \left(\theta_m + 273.1\right)\right)} \tag{8}$$

The equations for Mars are valid for altitudes less than $h \prec 7000$ meters. These equations can be redefined for higher altitudes. Now the density of the atmosphere can be substituted in the drag force equation. The drag force equations are given with the following relations for Earth and for Mars [5]:

$$D_e = \frac{1}{2} \cdot c_D \cdot \rho_e \cdot A \cdot v_e^2 \tag{9}$$

$$D_m = \frac{1}{2} \cdot c_D \cdot \rho_m \cdot A \cdot v_m^2 \tag{10}$$

The atmosphere friction force establishes the force against motion direction. The velocity of the ball on Mars and on Earth is given by v_m and v_e. The parameter A is the cross section area, and C_D is the drag coefficient. Drag coefficient C_D depends on the shape of the ball (or airbag).

The gravitation force (gravity or weight equation) without drag for falling objects is given by

$$m \cdot a = -m \cdot g \tag{11}$$

This equation with drag force is

$$m \cdot a - D = -m \cdot g \tag{12}$$

If a ball falls and encounters the ground, then a mass-spring-damper model is used to describe the dynamic behaviour of the ball.

The following figure (Fig. 1) shows the ball before contacting with the ground,

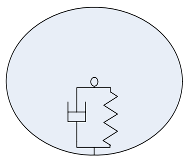

Fig. 1: The ball before contacting, no deformation

The next figure (Fig. 2) shows the ball when contacting with the ground.

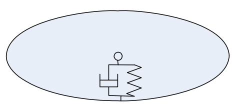

Fig. 2: The ball when contacting with the ground

The elastic force of the ball (spring character) can be calculated as

$$f_e = -\kappa \cdot \left(h - r\right) \qquad h \le r \tag{13}$$

where κ is the elasticity constant of the ball, h is altitude, and r is the radius of the ball. The viscous damping character of the ball is described by following the damping force model below

$$f_d = -\beta \cdot v \qquad h \le r \tag{14}$$

in which β is the damping constant. Three final equations can be combined with gravity equation, and the result is elaborated as follows:

$$\begin{cases} m \cdot a - D = -m \cdot g & h \succ r \\ m \cdot a - D = -\kappa \cdot (h - r) - \beta \cdot v - m \cdot g & h \le r \end{cases} \tag{15}$$

3 Implementation and Simulation

The presented implementation is a purely discrete approach: fly and distortion are modelled by state chart(s), updated by triggered step size events, which update the state by an ODE algorithm of user's choice. Impact/begin of distortion and end of distortion/restart of fly are triggered by events, which switch between the fly state with drag force and the distortion state, which are determined approximately in advance. For simplicity, fly and distortion model can be combined, when switching between different forces [7, 8].

For a-priori approximation of the impact/switching, the fact is used, that in case of BBM, not only the event function can be evaluated, but also its derivative(s). The event function $h_{event}(x,t)$, the zero of which must be found, is the ball height $h(t)$ – a state variable (minus ball radius r); but also the velocity and if necessary also the acceleration are state variables or derivative variables and available in the model.

$$h_{event}(t,\vec{x}) = h(t) - r, \ \frac{d}{dt} h_{event}(t,\vec{x}) = v(t), \\ \frac{d^2}{dt^2} h_{event}(t,\vec{x}) = a(t) \tag{16}$$

This (Eq. 16) allows approximating the event function by a polynomial of first order or second order near the impact:

$$p_1(t) = v_k \cdot t + h_k - v_k t_k - r \\ h_k = h(t_k), v_k = v(t_k) \\ t_{ae1} = t_k + \frac{1}{v_k}(r - h_k) \tag{17}$$

The zero t_{ae1} of the polynomial of first order may be used for adjusting a smaller step size, and in last consequence, as approximation for the event time h_{event}.

While the first order approximation requires in general some case-by-case analysis because of concave/convex type of *h(t)*, a second order approximation takes into account convexity of *h(t)*, but is more complex. The following equations show the definition of a second order system:

$$p_2(t) = \frac{1}{2} a_k \cdot t^2 + (v_k - a_k t_k) \cdot t + a_0 \\ h_k = h(t_k), v_k = v(t_k), a_k = a(t_k) \\ a_0 = \frac{1}{2} a_k t_k^2 - v_k t_k + h_k - r \\ p_2(t_{ae2}) = 0 \tag{18}$$

The BBM in this paper is implemented and simulated with the simulation tool MATLAB/State-flow®.

The simulation algorithm proceeds with two states. State transition depends on the altitude of the centre of the ball. If altitude of the centre of the ball is smaller than the radius *r* of the ball, then the force equation is a mass-spring-damper model, and if the altitude of the ball is greater than the radius of the ball, then the gravity drag equation is considered in a hybrid simulation algorithm. Drag and damping force depend on velocity. When velocity falls to zero, then these two forces are reduced, and fall to zero too.

The model of the bouncing ball is extended with equations of temperature, atmospheric pressure, density of the atmosphere and drag force, so the model structure is a hybrid model. In Fig. 3, an abstract implementation model for BBM on Earth is shown.

The simulation parameters are chosen in metric units as follows [6]:

Fall initial altitude $h = 5m$

Initial velocity $v = 2.5\frac{m}{s}$

Mass of ball $m = 200kg$

Radius of ball $r = 2.5m$

Cross section area $A = 19.625m^2$

Drag coefficient $c_D = 0.47$ dimension less

Spring constant $\beta = 250\frac{kg}{s}$

Damping constant $\kappa = 1.0e + 4\frac{kg}{s^2}$

Earth's gravity $g_e = 9.80665 \frac{m}{s^2}$

Mars' gravity $g_m = 3.693 \frac{m}{s^2}$

Euler number $e = 2.71828182\,8458$

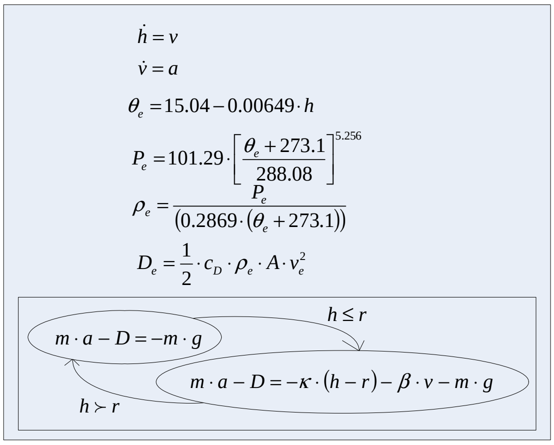

Fig. 3: simulation algorithm structure as an abstract model on Earth. The state chart defines the hybrid model switch

Mars' gravity $g_m = 3.693 \frac{m}{s^2}$

Euler number $e = 2.71828182\,8458$

3.1 Simulating with Stateflow

The state update is explicitly implemented by an Euler algorithm, as shown in Fig. 4. Within the state chart BouncingBall, an event graph switches between fly and distortion.

The update is driven by external events (Edge Sense), synchronised with step size *Ts*. In the synchronising state chart BouncingBallUpdate, step size *Ts* may be adjusted with respect to the approximated event time t_{ae1} or t_{ae2}, so that the $t_{k+1} = t_k + T_s < t_{ae1}$ (t_{ae2}) or may be set to $T_s = t_{ae1} - t_k$ or $T_s = t_{ae2} - t_k$ (matching of approximated event time).

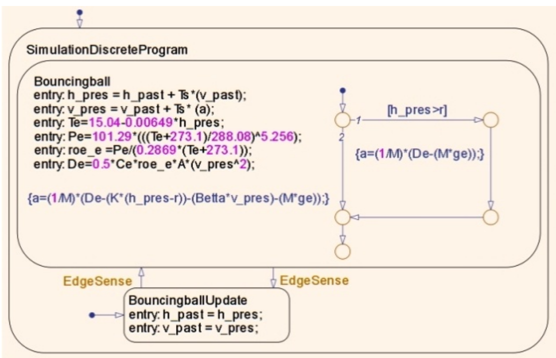

Fig. 4: State chart bouncing ball model for Earth using MATLAB environment

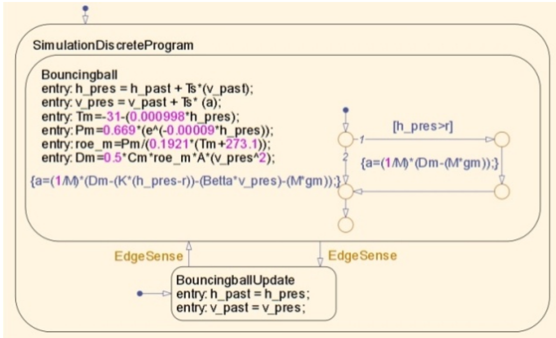

Fig. 5: State chart bouncing ball model for Mars

The hybrid algorithm for the BBM on Mars is implemented in the same way (Fig. 5).

For comparison of results for Earth and Mars, both models have been implemented in parallel – Fig. 6, using the superstate construct of Stateflow. Both superstates are triggered by a pulse generator with frequency/step size *Ts*, which can be adjusted from the inner state charts as given above.

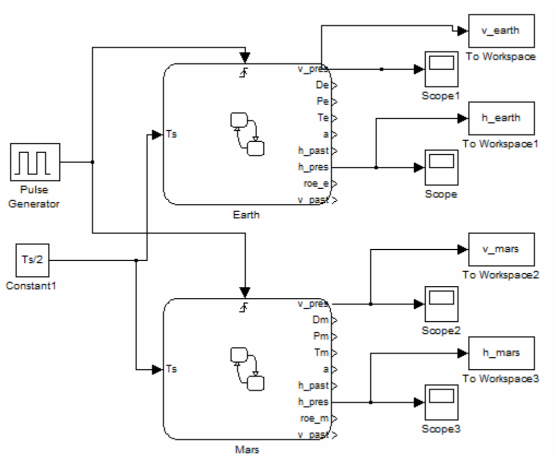

Fig. 6: Block diagram of the overall simulation model of bouncing ball using Stateflow.

The variables of altitude, acceleration, force balance, drag force, pressure of atmosphere and density of the atmosphere and temperature in this example are calculated in SI units, so all the results of the simulations are shown in SI units.

The following figures show altitude (Fig. 7), velocity (Fig. 8), and acceleration (Fig. 9) for BBM simulation comparing situation on Earth and on Mars. The drag forces for BBM simulation on Earth and on Mars are presented separately in Fig. 10 and Fig. 11, respectively.

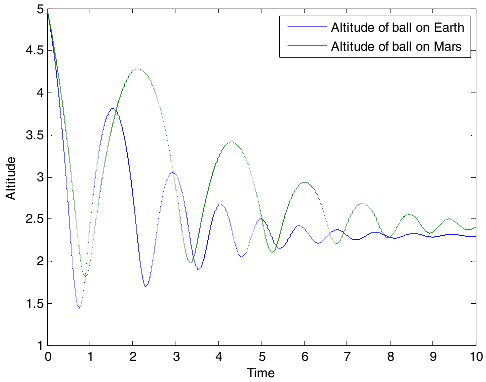

Fig. 7: Altitude of the bouncing ball on Earth (blue) and on Mars (green)

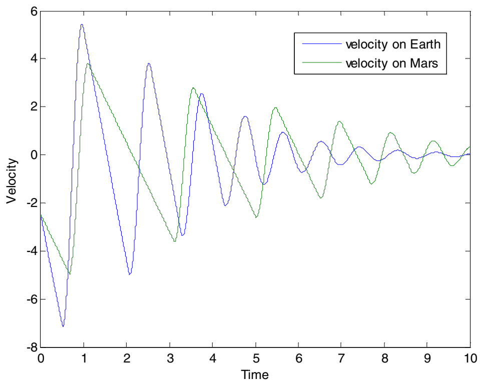

Fig. 8: Velocity of the bouncing ball on Earth and on Mars

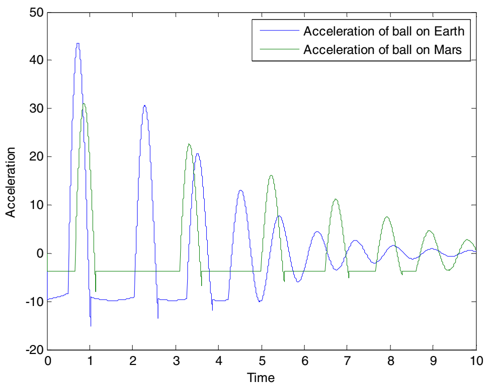

Fig. 9: Acceleration of the bouncing ball on Earth and on Mars

4 Summary and Conclusion

This model is presented to demonstrate the flexibility and extensibility of state charts and differential equations as a hybrid combination in the field of physical modelling. The Bouncing Ball can be used for various purpose of teaching or basic research, the advantage of the model is the simplicity of the state flow structure and the comprehensible physical system, which can be connected and redefined with any type of linear or nonlinear equations.

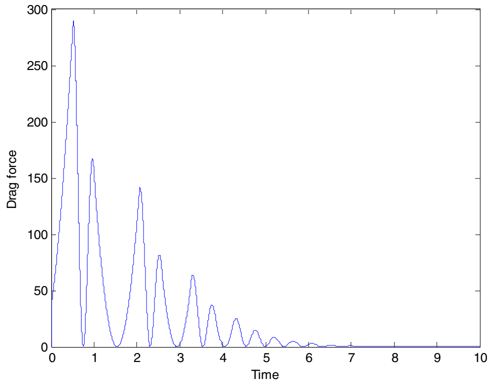

Fig. 10: The drag force of a bouncing ball on Earth

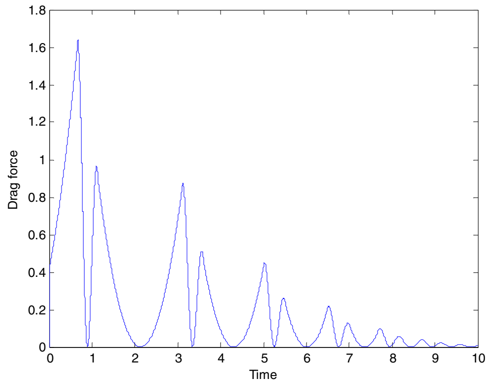

Fig. 11: The drag force of a bouncing ball on Mars

5 References

[1] G. G. Wawrzyniak and M. E. Lisaon. Using Inertial Measurements for the Reconstruction of 6-Dof Entry, Descent, and Landing Trajectory and Attitude Profiles, 2001.

[2] R. D. Braun, R. M. Manning. Mars Exploration Entry, Descent and Landing Challenges, 2005.

[3] Earth Atmosphere Model (Metric units) NASA Glenn Research Centre www.grc.nasa.gov/WWW-12/airplane/atmosmet.html 23.03.2010.

[4] Mars Atmosphere Model (Metric units) NASA Glenn Research Centre www.grc.nasa.gov/WWW-12/airplane/atmosmrm.html 23.03.2010.

[5] E. L. Houghton, P. W. Carpenter. Aerodynamics for Engineering Students 2003.

[6] Earth Atmosphere Model (Imperial units) NASA Glenn Research Centre www.grc.nasa.gov/WWW-12/airplane/atmos.html 31.07.2008

[7] M. Gyimesi, P.Einzinger, F. Breitenecker. Developing Simulation Models using Statechart Methodology, Eurosim Proceedings 2010

[8] PH.D. Michael M. Tiller, Introduction to Physical Modeling with Modelica 2004.