

Bouncing Ball

Bernd Ulmann
ulmann@vaxman.de

Vintage Computer Festival Europe 2007
29th April – 1st May 2007
Munich

- 1 Introduction
- 2 The modell
- 3 Calculating y
- 4 Calculating x
- 5 Displaying a ball
- 6 The program
- 7 Request for support

Introduction

The following slides describe the simulation of a bouncing ball in a box using a small electronic analog computer (in this case a Dornier DO-80).

- First of all, a precise description of the system to be simulated is required.
- Using this description as a starting point a set of (usually differential) equations can be derived which will serve as the foundation of the simulation.

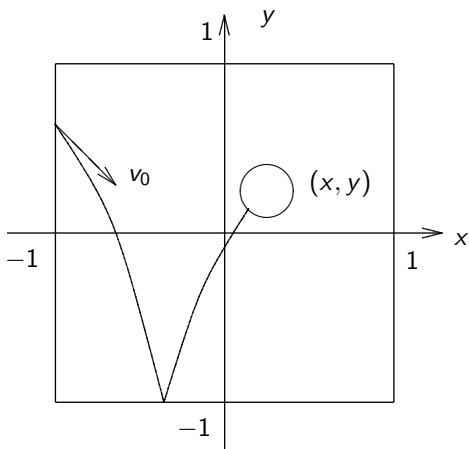
In this case the setup is fairly simple – it consists of an elastically rebounding ball moving in a rigid box. All coordinates are expected to be cartesian with ranges $-1 \leq x \leq 1$ and $-1 \leq y \leq 1$ (delimited by the box). This fits well into the picture of an electronic analog computer with its natural range of values between 1 and -1 .

The model

The picture on the next slide shows the model to be implemented:

- Key element of the model is the elastic ball with coordinates (x, y) .
- This ball is elastically bouncing inside a quadratic box having sides of length 2.
- At the start ($t = 0$) of the simulation the ball is thrown into the box from the left side of the box with an initial velocity v_0 (and an initial acceleration of \dot{v}_0).
- The ball will loose energy by air friction while it bounces completely elastic.
- In the following both coordinates of the ball movement will be handled separately – we will first have a look at the y -component and then at the x -component.

The model



Calculating y

Let us have a look at \ddot{y} – it depends on three factors:

- 1 The constant gravitation g ,
- 2 a frictional loss being proportional to \dot{y} and, finally,
- 3 a sudden change when the ball hits one of the boundaries of the surrounding box.

All in all we get

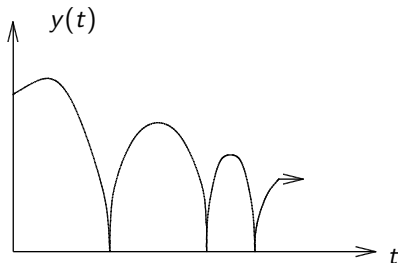
$$\ddot{y} = -g + d\dot{y} \begin{cases} +\frac{c}{m} (|\dot{y}| + 1) & \text{if } \dot{y} < -1 \\ -\frac{c}{m} (\dot{y} - 1) & \text{if } \dot{y} > 1. \end{cases}$$

Integrating yields

$$\dot{y} = \int_0^T \ddot{y} dt + \dot{y}_0 \text{ and finally } y = \int_0^T \dot{y} dt + y_0.$$

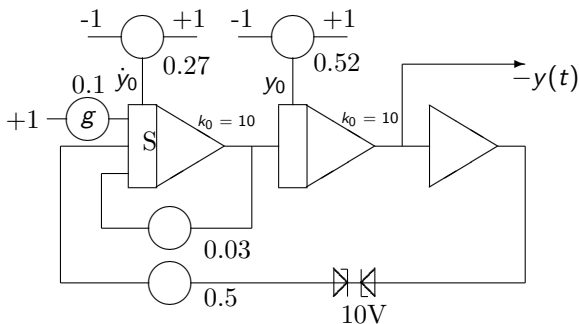
Calculating y

Plotting $y(t)$ against t results in a pictures like this:



Calculating y

Combining these three equations yields the computer circuit shown below for calculating the y -coordinate of the bouncing ball:

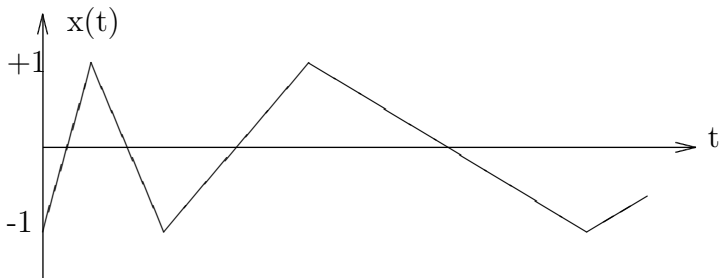


Using the two Z-diodes is a trick and saves an adjustable limiter at the expense of a box having always height ± 1 .

Calculating x

Due to the ideal elasticity of the ball the calculation of the x -movement is also rather simple:

The ball starts with an initial velocity v_0 which decreases linearly due to air friction, so the movement in the x -direction looks a bit like this:

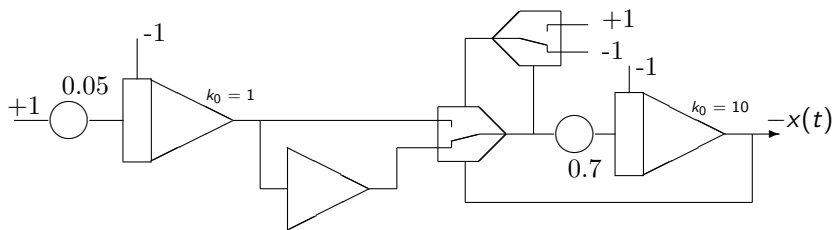


Calculating x

Starting with

$$\dot{x}(t) = - \left(\int_0^T 0.05 dt - 1 \right)$$

only an additional integrator and two comparators are needed to generate an x -signal bouncing between -1 and 1 with decreasing velocity:



Displaying a ball

These two partial computer circuits are sufficient to calculate (x, y) for the bouncing ball. This coordinate pair can now be used to deflect a circle on an oscilloscope screen yielding a quite realistic impression of a bouncing ball inside its box.

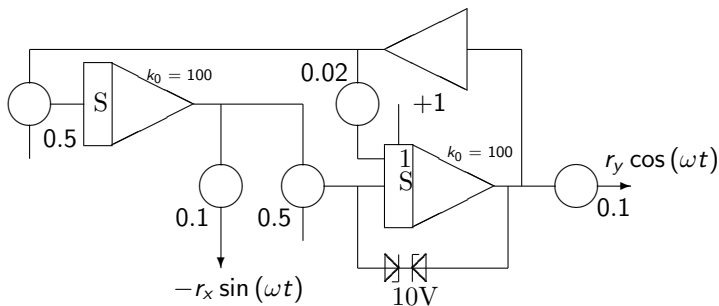
To generate a circle, a sine/cosine-signal-pair is necessary which can be generated easily by an electronic analog computer solving the differential equation

$$\ddot{y} + \omega^2 y = 0.$$

This is a standard problem and easily solved using two integrators, a summer and maybe two coefficient potentiometers to determine ω .

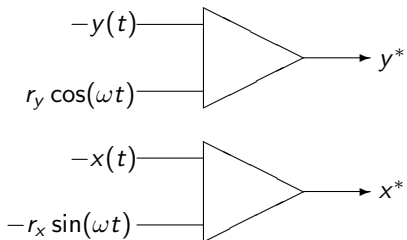
Displaying a ball

Due to the non-ideal computer components, a direct solution of the equation above will not yield a stable harmonic oscillation – instead its amplitude will increase or decrease over time which is undesirable for a simulation. So we will use a trick circuit like this to generate stable harmonic oscillations:



Displaying a ball

Using $r_y \cos(\omega t)$ and $-r_x \sin(\omega t)$ to display a circle on an oscilloscope, all what is left to do is to mix (*add*) these two signals with the coordinates (x, y) of the bouncing ball:



The program

The picture on the following slide will show the program patched on a DO-80, a small electronic analog computer made by Dornier in the 1970s.

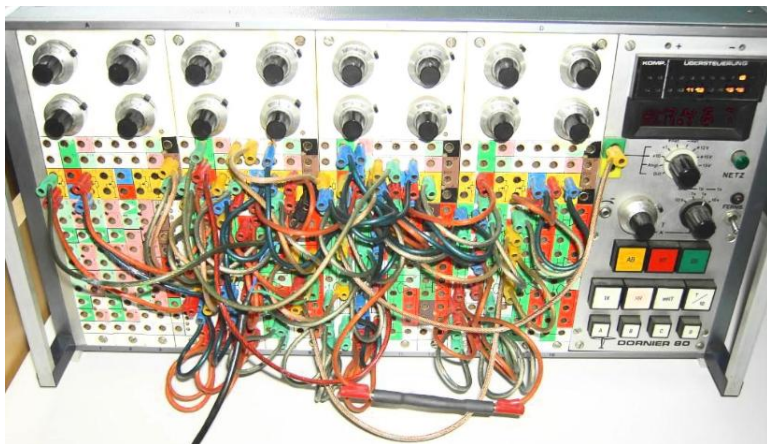
The leftmost quadrant of this machine is occupied by digital elements like flip flops and simple gates, so only three of four quadrants are populated with analog computing elements.

The bouncing ball simulation needs more than 80 percent of this computer's computing elements – all in all

- 6 integrators,
- 5 summers,
- 2 comparators and
- 12 coefficient potentiometers

are used.

The program



Request for support

I would like to ask you for your support:

- I save analog computers from scrap where ever I can. If you happen to know about a system looking for a good home, please let me know. (I am not afraid of large systems and I would really love to get an RA770 or an RA800(H), for example! I will pay for all expenses to save the machine from scrap, I will arrange shipping, etc.)
- I am interested in trivia, documentation, computer setups, sales brochures, etc.

Please spread the word and help to save these machines from getting scrapped, lost and forgotten.

You can reach me always at ulmann@vaxman.de or by mobile phone at 0177/5633531 (in Germany) in case of an emergency.

Thank you very much!



Demonstrationsbeispiel Nr. 5, Ball im Kasten, AEG Telefunken



Bernd Ulmann, *Simulating a bouncing ball in a box*,
[http://www.vaxman.de/analog_computing/
bouncing_ball_bouncing_ball.html](http://www.vaxman.de/analog_computing/bouncing_ball_bouncing_ball.html)