

A Zipliner's Delight

Subject Area(s) Physical Science, Science and Technology

Associated Unit

Yellow highlight = required component

Associated Lesson

Activity Title A Zipliner's Delight

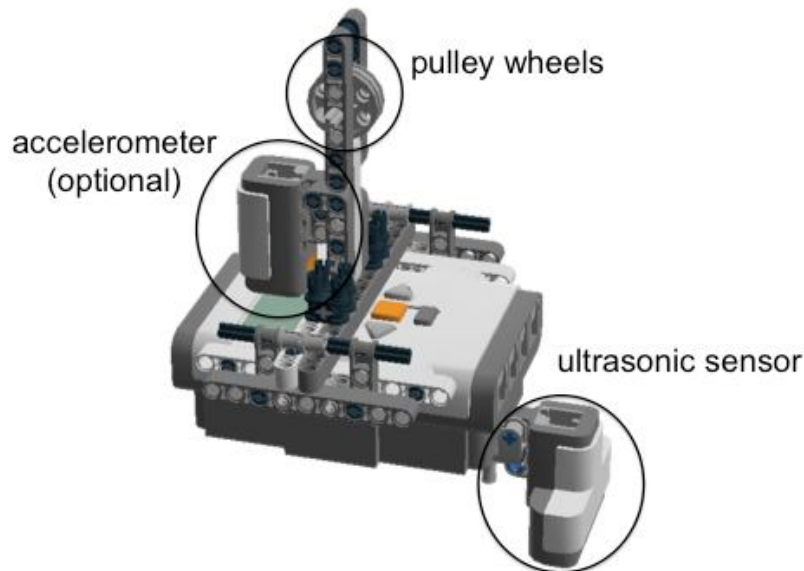


Figure 1

ADA Description:

Caption: Figure 1: Zipline robot basic assembly and components

Image file: figure_1.tiff

Source/Rights: Wikimedia Commons (left) Copyright © 2011 Polytechnic Institute of NYU.

Grade Level 06 (05-07)

Activity Dependency

Time Required 2x(40-60 minutes)

Group Size

Expendable Cost per Group US\$ ___

Summary

Students learn about potential energy, as expressed as the height of an object along a linear one-dimensional zipline track. A robot, designed to traverse the track, converting stored potential energy into kinetic energy, also is capable of monitoring the instantaneous speed of the robot using various sensors. Thus, students are able to quantify and compare the starting potential energy (height) of a robot and the conversion thereof into kinetic energy (linear displacement). The system that is presented is that of a

single robot, which is built using the LEGO Mindstorms robotics platform and installed with Lejos 0.9 firmware.

Engineering Connection

Engineering Category = #2

Keywords

speed, potential energy, kinetic energy, robot, data collection

Educational Standards

Choose from <http://www.jesandco.org/asn/viewer/default.aspx>.

ITEEA (provide standard number, grade band, benchmark letter and text):

State/national science/math/technology (provide source, year, number[s] and text):

NY Science Standard 1.2 Scientific Inquiry

NY Science Standard 6.2 Models

Pre-Requisite Knowledge

Teacher should be familiar with LEGO NXT Mindstorms as a building and programming platform.

Knowledge of pairing NXT Intelligent Bricks via Bluetooth is necessary. cursory knowledge of Java is beneficial.

Learning Objectives

After this activity, students should be able to:

- Clearly and confidently explain the conversion of potential energy to kinetic energy using a zipline experiment.
- Compute the average speed of a vehicle using distance and start and stop times.
- Understand that speed can change as an object travels.
- Understand the difference between instantaneous speed and average speed.
- Predict the shape of a curve that defines the speed of a *free* object as it traverses a sloped path.

Materials List

1 LEGO NXT Intelligent Brick (1x\$144.99) is needed for each setup.

Each setup needs:

- A constructed zipline robot and a laptop with Bluetooth connectivity to the NXT brick. Each laptop should be installed with the Java SDK (> 1.6) and Lejos SDK (> 0.9).

Introduction / Motivation

5th grade science classes investigate how single variables affect experimental setups. In keeping with a progression of mechanical components of earlier grades, this activity aims to introduce students to a more kinetic robotic model. Instead of being limited by reaction times in controlling stopwatches to record average and instantaneous travel speeds, the accompanying robotic programming circumvents the need for student control of data logging; the speed of a robot as it traverses a linear zipline path is automatically logged to practically infinitesimal time resolution. This allows educators to focus class discussions on the nature of the conversion of potential energy to kinetic energy. Specifically, “instantaneous speed” can be compared to “average speed.” This lends itself as also a nice introduction to more basic physics.

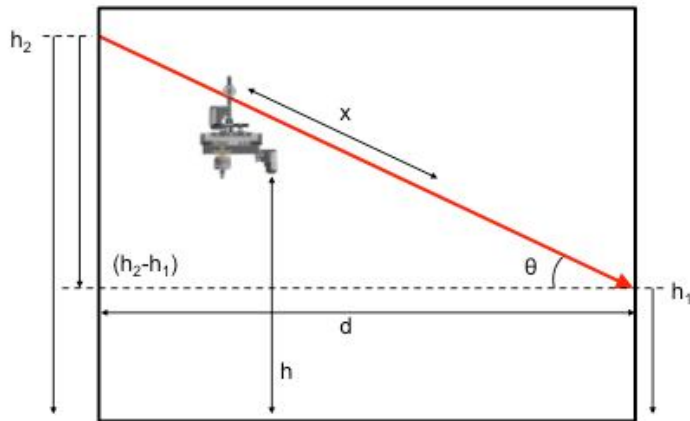
Procedure

Background

In this lesson, students can be presented with several setups (using a single zipline robot for each setup) that are assigned to distinct zipline paths, defined by different starting and ending heights. This effectively situates each robot on separate tracks with different slopes and corresponding potential

energies. The laptops for each robot may be connected to the robots wirelessly via Bluetooth. After a connection is established, the activity's open-source program may be executed to monitor and graph the displacement and zipline speed with respect to time. Students can be asked to collect curves of numerous trials on the zipline paths as well as try different robots to compare the effects of potential energy change. The robot calculates the speed of along its path using the calculations shown in Figure 2.

Image Insert Figure 2 here, [note position: left justified]



Variables

- x = zipline displacement
- t = time variable
- θ = angle made by zipline with ground

Measurable Variables

- h = height of robot from ground (ultrasonic sensor)
- h_2 = height of highest potential
- h_1 = height of lowest potential
- d = horizontal span of zipline

$$\theta = \tan^{-1} \left(\frac{(h_2 - h_1)}{d} \right)$$

$$x = (h - h_1) \csc \theta$$

$$\frac{dx}{dt} = \csc \theta \frac{dh}{dt} = \csc \left[\tan^{-1} \left(\frac{(h_2 - h_1)}{d} \right) \right] \frac{dh}{dt}$$

Figure 2

ADA Description:

Caption: Figure 2: Internal trigonometry calculations performed by the robot to compute its speed along its zipline one-dimensional linear traversal.

Image file: figure_2.tiff

Source/Rights: Copyright © 2011 Polytechnic Institute of NYU. Used with permission

Image Insert Figure 3 here, [note position: left justified]

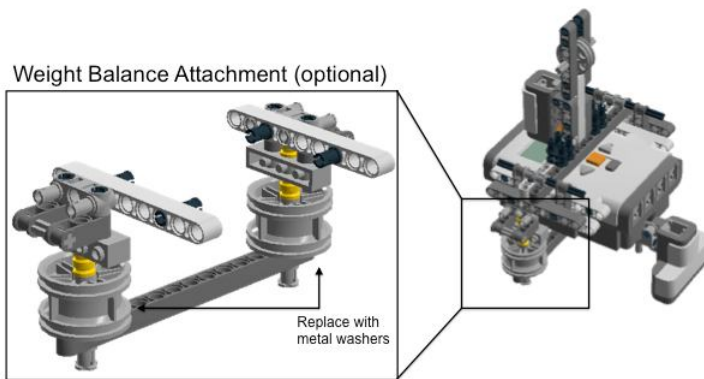


Figure 3

ADA Description:

Caption: Figure 3: Optional weight addition to robot

Image file: figure_3.tiff

Source/Rights: Copyright © 2011 Polytechnic Institute of NYU. Used with permission

Students can also assemble the robot with added weight to study how weight affects the conversion of energies (Figure 3).

Before the Activity

The activity requires significant setup by the instructor. These may be divided into the following categories: building, programming, and demo setup. After each robot is built according to the accompanying instructions, Lejos firmware (available from <http://lejos.sourceforge.net/>) may be uploaded to the robot and connected to PC or Mac platforms using Bluetooth. After a connection is established, the accompanying Java program may be run from the computer to poll data from the robot(s). Successful program execution should prompt a window similar to the one shown in Figure 4.

Image Insert Figure 4 here, [note position: left justified]

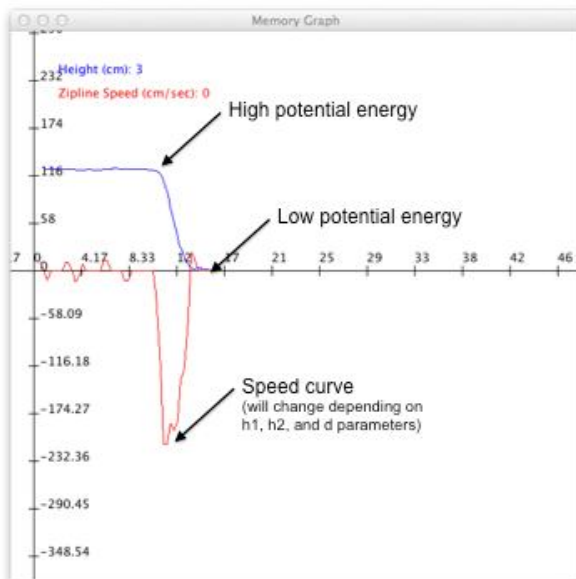


Figure 4

ADA Description:

Caption: Figure 4: Sample chart from logger.java program

Image file: figure_4.tiff

Source/Rights: Copyright © 2011 Polytechnic Institute of NYU. Used with permission

Following proper and successful robot operation, zipline setups may then be setup. Thin plastic twine may be used for this purpose. The high tension required may be achieved using opposite ends of an outdoor handball court (40 ft. across) or similar space with fenced boundaries.

Once assembled and programmed, the activity requires that the robots be calibrated per setup. The variables h1, h2, and d, the meanings for which are indicated in Figure 2, must be measured and entered into the program prior to compilation and running (Figure 5). d must be measured using a tape measure or yard stick. h2 and h1, however, may be measured with the robot itself at the start and end positions, respectively.

Image Insert Figure 5 here, [note position: left justified]

```
        break;
    case PathIterator.SEG_CLOSE:
        System.out.println("close");
        break;
    default:
        break;
    }
}

public static void main(String[] args) throws Exception {
    final int AVG_WINDOW = 6;
    final int VEL_WINDOW = 2;
    final int VELA_WINDOW = 2;
    final double h2 = 130;
    final double h1 = 10;
    final double d = 400;

    UltrasonicSensor sonic = new UltrasonicSensor(SensorPort.S4);
    /*AccelMindSensor tilt = new AccelMindSensor(SensorPort.S1);
    */
    Logger gc = new Logger();
    gc.show();
}
```

Figure 5

ADA Description:

Caption: Figure 5: Required parameters to be set in code

Image file: figure_5.tiff

Source/Rights: Copyright © 2011 Polytechnic Institute of NYU. Used with permission

With the Students

1. Discuss/review the concept of potential energy and its conversion into kinetic energy. Linear one-dimensional travel on sloped gradients is preferable as an example, i.e. downhill skiers, cars on hilly roads, roller coasters.
2. Describe the experimental setup using figures similar to Figure 2 above. For novice groups, simply study the effect of different potential energy setups on the kinetic energy of the robot, i.e. speed. For more advanced and independent groups, discuss with the students and allow them, in their groups, to

design their own experiment. These may include the original experiment of studying different potential energy setups, adding additional weight to the robot using the optional attachment, or monitoring the acceleration of the robot (measured by the optional accelerometer).

3. Instruct the students to create data tables for measuring the average speed for each trial of each experiment that they intend to carry out. 5 trials should be sufficient.
4. Instruct the students on how to operate the software from the command line/terminal window using Java. Future versions of the software will package the execution in an icon on the desktop.
5. The current version of the software does not save graphed data (soon to be implemented), so instruct the students on how to take screenshots of the logging window using either *Ctrl-Print Screen* on PC or the *Preview* program on Mac.
6. Instruct the students to compare their tabulated results to the real-time charted data. Ask them if the robot ever appears to go faster than the computed average speed. Ask them if the shape of the speed curve appears to conform to a pattern or general shape. Ask them what they think this pattern may be due to and why it does not look differently.

Attachments

1. [Lejos program: logger.java](#)
2. [Building instructions \(Lego Digital Designer required\): zip_liner.lxf](#)

Safety Issues

- Ziplines are typically setup with moderately high tension to withstand deflection from the robot's weight. These lines can injure or trip participants, particularly when they run or are unaware.

Troubleshooting Tips

Investigating Questions

Assessment

Activity Extensions

None

Activity Scaling

None

Additional Multimedia Support

None

References

Other

None

Redirect URL

None

Contributors

Primary Developer: Carlo Yuvienco

Copyright

None

Supporting Program

School:

Polytechnic Institute of NYU

Grant:

This TeachEngineering Lesson was developed with the support of the AMPS Project via
National Science Foundation GK-12 Grant # 0741714

Version: September 2010