Pi from an Intelino Smart Train and Scratch

by Richard Born
Associate Professor Emeritus
Northern Illinois University
rborn2@niu.edu

Introduction

There is probably no number that has received more interest since ancient times than the number pi, symbolized by the Greek letter π. Originally defined as the ratio of the circumference of a circle to its diameter, it has been given approximate values including 3.14 and 22/7. Proven to be an irrational number, supercomputers have computed the value of pi to more than one trillion digits.

With this background in mind, what better number could there be to investigate with an intelino Smart Train and Scratch than the number pi? This combination provides a perfect math or statistics activity for middle school and high school students.

In this lesson, students will use provided Scratch programs and an intelino Smart Train to determine an approximate value for pi. Students will also calculate the percent error in their value and use critical thinking to explain possible reasons for the error. Clearly, this activity supports STEM as well as NGSS practices.

The Equation for Pi

An approximate value for pi will be determined from time data collected by Scratch while the intelino Smart Train is moving. No meter sticks or other distance measuring devices are used. Students will time two events:

- T_c, the time for intelino to travel one circumference around a circle of intelino curve tracks
- T_d, the time for intelino to travel a length of one diameter of the circle on a series of straight tracks.

By definition, π is the ratio of the circumference C of a circle to its diameter D, $\pi = C/D$. If the speed of the intelino engine is kept the same for both the circle and the straight track, then the distance traveled would be proportional to the time, *i.e.*, $C/D = T_c/T_d$. Therefore, ***we can compute the value of π by finding the ratio T_c/T_d of the two times!***

Now with an experimental value for π, the percent error can be calculated by the usual method for computing percent error:

$$\% \, Error = \frac{|Experiment\ value - Actual\ value|}{Actual\ Value} \text{ X } 100\%$$

Finally, a discussion regarding possible reasons for the error would be in order.

Lab Setup

Figure 1 shows the setup for this lab. The train is loaded with the Scratch file *Get Time for Diameter.sb3*. The figure shows intelino at its starting position on a length of straight track, facing the direction shown by the arrow. Color action snaps ① and ② are placed so that they are separated by a distance D equal to the diameter of the circular track. The intelino action snap ① (white, magenta, yellow) is associated with a Scratch block that resets the Scratch timer to zero—like starting a stopwatch. Action snap ② (white, magenta, blue) is associated with a Scratch block that captures the current time of the timer—like stopping a stopwatch. This is the time T_d, discussed in the previous section. When intelino reaches action snap ③ (white, magenta, red), intelino is told to move backward. Intelino does just that, ignoring action snaps ① and ②. Upon reaching action snap ④ (white, magenta, green), intelino is told to move forward again. You may be wondering what the purpose of action snaps ⑤ and ⑥ is. They act as co-pilots. In the rare event that intelino doesn't act on snaps ③ and ④, then action snaps ⑥ and ⑤ act as 'backups', respectively.

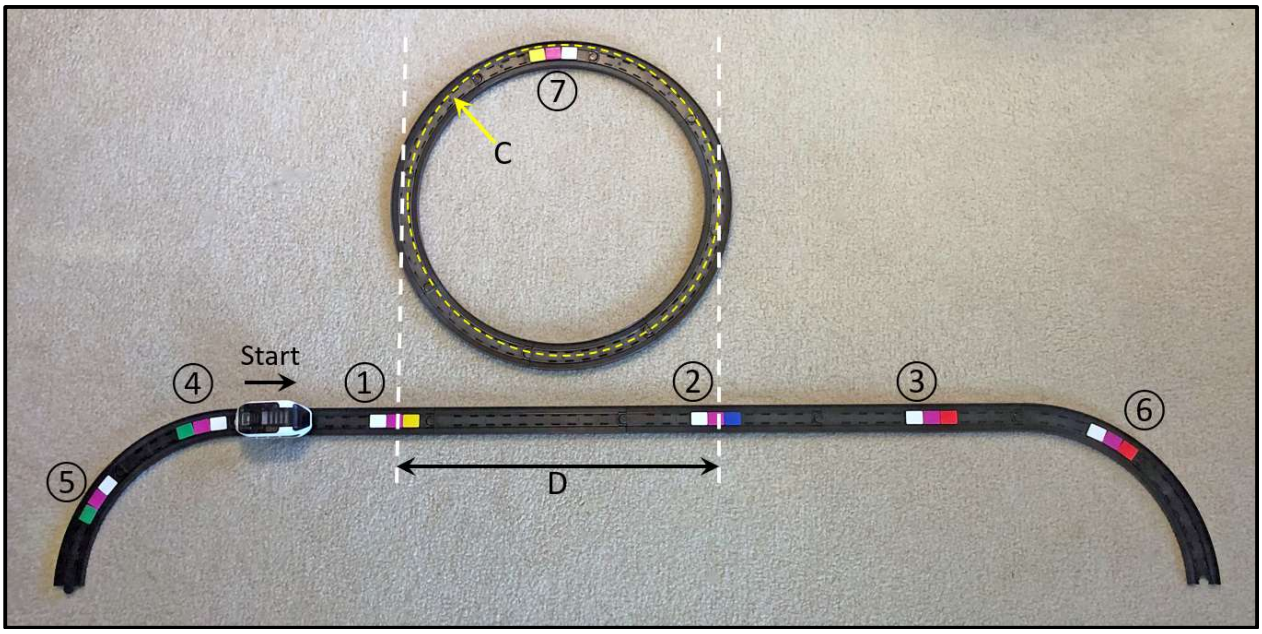


Figure 1

After collecting data on T_d, intelino is placed on the circular track facing counterclockwise. The train is loaded with the Scratch file *Get Time for Circumference.sb3*. Action snap ⑦ is used to mark a position where the Scratch timer captures the lap time T_c.

An accompanying video, *Capturing Timing Data to Approximate Pi.m4v*, shows motion of the intelino smart engine while capturing T_d three times and T_c three times. The next section explains how the accompanying Scratch programs are used to obtain these times. Remember that our approximation for pi is simply T_c/T_d!

The Scratch Programs

Figure 2 shows the two Scratch programs, *Get Time for Diameter.sb3* on the left and *Get Time for Circumference.sb3 on the right*. Computer science students should be able to do a code review of each program, explaining the purpose of all blocks. Math students primarily interested in finding an approximation of pi need only consider the *User Defined Parameters*. There are two such parameters—one for intelino's *speed* and the other for the number of *trials* for which you want times averaged. Both programs show the speed set to 35 cm/s and the number of trials set to 6. Best results will be obtained using speeds between 30 and 40 cm/s. The programs are started by clicking the green "Go" flag icon in the upper right corner of the Scratch screen.

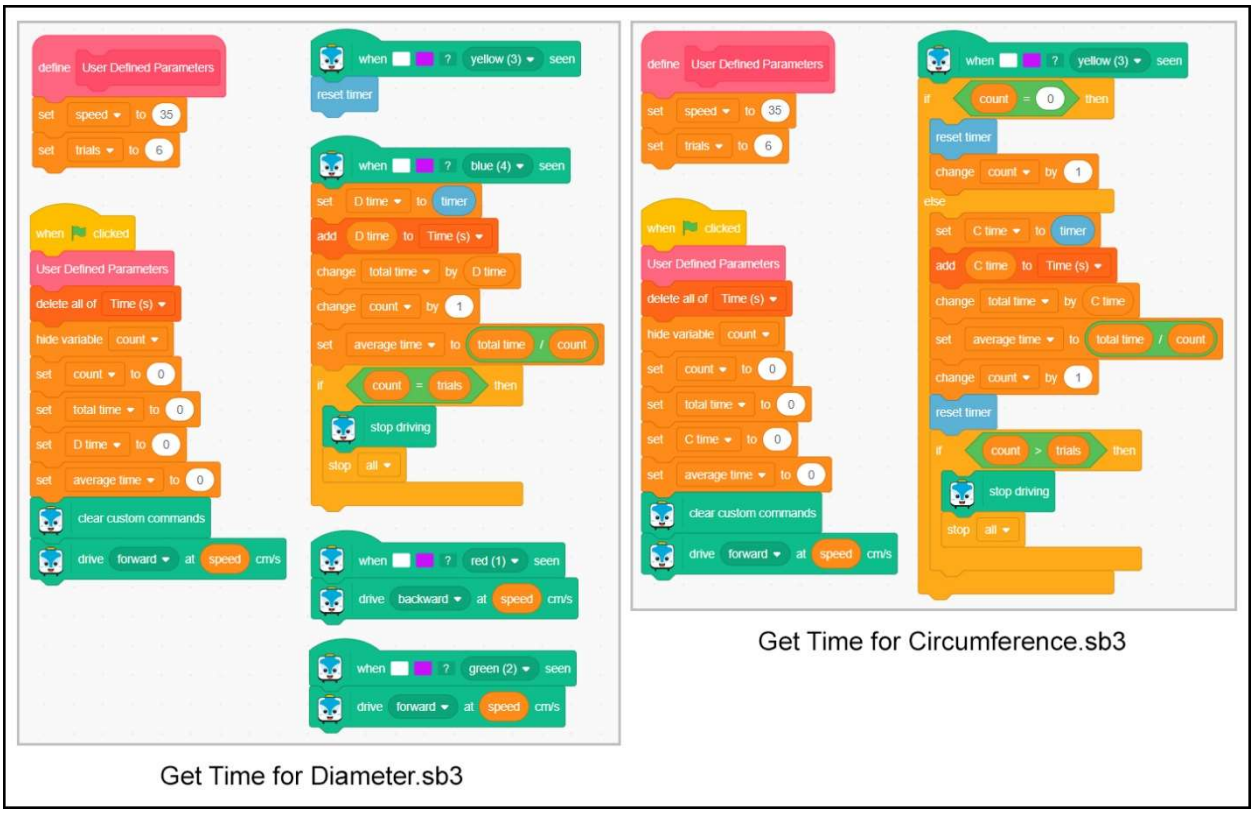


Figure 2

Scratch Program Output Statistics

Figure 3 shows typical output from each of the two programs. The speed was set to 37.5 cm/s and the trials was set to 6. The output shown is the output at the conclusion of the requested 6 trials. The user requested speed and number of trials are shown in the upper left. D time (T_d) and C time (T_c) for trial number 6 are shown in the upper right. These times for each of the six trials are shown in their respective lists. The total time for all six trials is shown, along with the

most important piece of information—the average time. All times are in seconds. Using the average time values, this series of trials results in an approximation of pi equal to T_c/T_d = 3.3498/1.0667 = 3.14!

Students may not always be this lucky, so they should be prepared to hypothesize on possible sources of error. Are the nominal speeds of the intelino Smart Train always equal to the actual speeds? Is the diameter equal to the outer diameter of the track, the inner diameter, or something between these values?



Figure 3