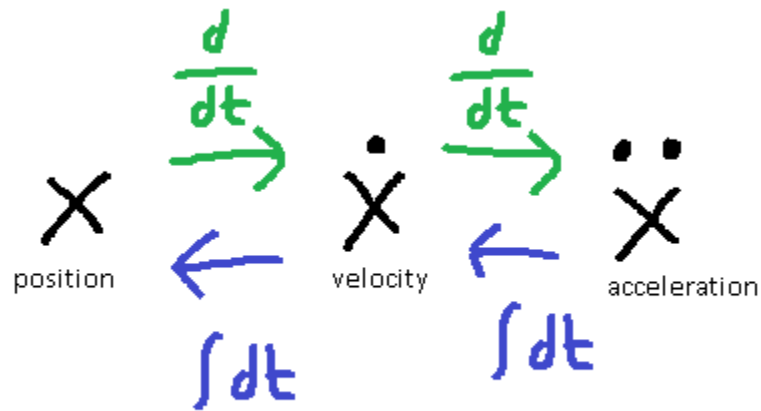


Implementing differential math using printer encoder



Written by :

Muhammad Husni

Goal of this experiment :

- Understanding what is position, speed, and acceleration
- Understanding how to implement that equation and make coding using Arduino IDE.

Theory behind that

- Position, speed, and acceleration

Position is also known location. In cartesian graph (x,y) position is how many value x and y.

The velocity of an object is the rate of change of its position with respect to a frame of reference, and is a function of time. Remember this equation :

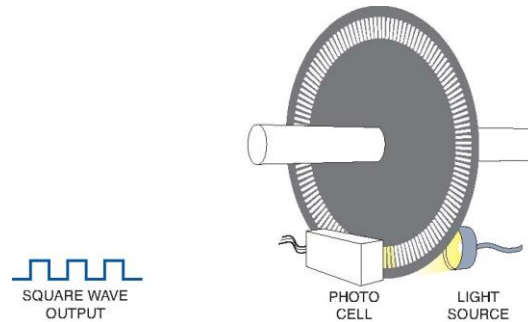
$$v = \lim_{\Delta t \rightarrow 0} \frac{\Delta x}{\Delta t} = \frac{dx}{dt}.$$

<https://en.wikipedia.org/wiki/Velocity>

Acceleration is the rate of change of the velocity of an object with respect to time. Accelerations are vector quantities (in that they have magnitude and direction).

$$\mathbf{a} = \frac{d\mathbf{v}}{dt} = \frac{d^2\mathbf{x}}{dt^2}$$

- How Encoder work



Encoder is a device that convert rotary motion to a square wave electronics signal. More speed more square wave. To read the position of encoder we need a microcontroller. We use Arduino uno microcontroller to make easier.

Little bit math

- Numerical derivation

W Trapezoidal rule - Wikipedia x W Numerical differentiation - Wiki x position velocity acceleration de x Kinematics and Calculus - The P x +

https://en.wikipedia.org/wiki/Numerical_differentiation#:~:text=In%20numerical%20analysis%2C%20numerical%20differentiation...

WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Current events
Random article
About Wikipedia
Contact us
Donate

Contribute
Help
Learn to edit
Community portal
Recent changes
Upload file

Tools
What links here
Related changes
Special pages
Permanent link
Page information
Cite this page
Wikidata item

Print/export
Download as PDF
Printable version

Languages

Numerical differentiation

From Wikipedia, the free encyclopedia

In **numerical analysis**, **numerical differentiation** describes *algorithms* for estimating the **derivative** of a **mathematical function** or function *subroutine* using values of the function and perhaps other knowledge about the function.

Contents [hide]

- 1 Finite differences
- 2 Step size
- 3 Other methods
 - 3.1 Higher-order methods
 - 3.2 Higher derivatives
- 4 Complex-variable methods
- 5 Differential quadrature
- 6 See also
- 7 References
- 8 External links

Finite differences [edit]

Further information: *Finite differences*

The simplest method is to use finite difference approximations.

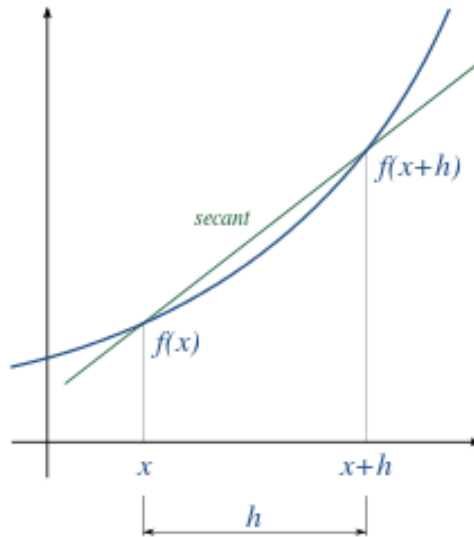
A simple two-point estimation is to compute the slope of a nearby *secant line* through the points $(x, f(x))$ and $(x + h, f(x + h))$.^[1] Choosing a small number h , h represents a small change in x , and it can be either positive or negative. The slope of this line is

$$\frac{f(x + h) - f(x)}{h}.$$

This expression is *Newton's difference quotient* (also known as a *first-order divided difference*).

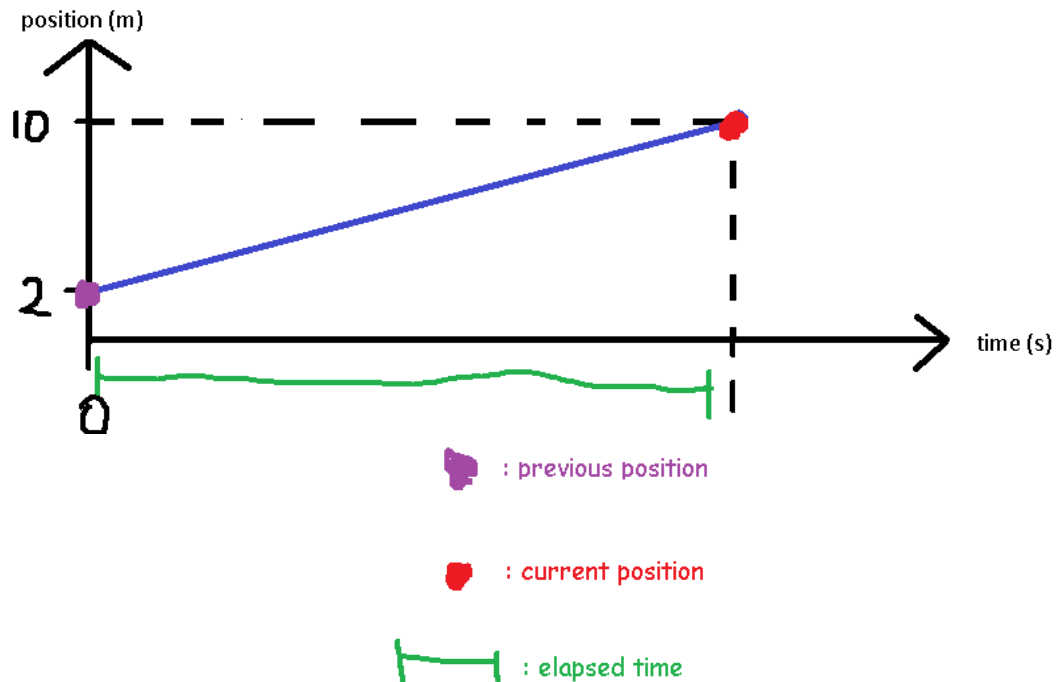
https://en.wikipedia.org/wiki/Numerical_differentiation#:~:text=In%20numerical%20analysis%2C%20numerical%20differentiation,other%20knowledge%20about%20the%20function.

Are you still confused ? me too 😊 hahaha nonono just kidding. On Wikipedia there are lot of equation that you must confused which one you will use. I will pick and modify the equation so you can use easily 😊



$$\frac{f(x+h) - f(x)}{h}.$$

I guess you have no idea what this is. Okay remember speed equation that you got from position differentiation. I will make simpler I promise haha



This graph have so many similarity. Let's replace every variable on that equation. So the equation will looks like this

$$\frac{f(x+h) - f(x)}{h} = \frac{\text{current position} - \text{previous position}}{\text{elapsed time}}$$

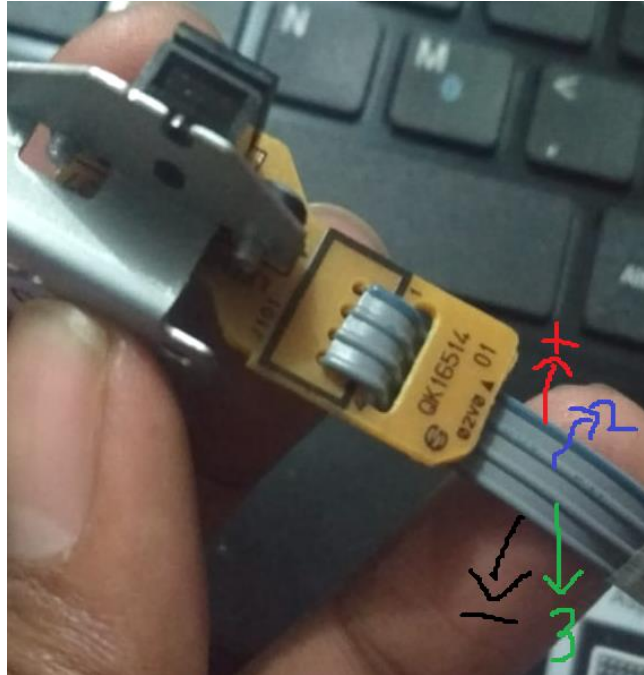
Let's insert that value. The result will surprise you

$$\begin{aligned} \frac{f(x+h) - f(x)}{h} &= \frac{\text{current position} - \text{previous position}}{\text{elapsed time}} \\ &= \frac{10 \text{ m} - 2 \text{ m}}{1 \text{ s}} = 8 \text{ m/s} \end{aligned}$$

..... VOILAAA you got velocity from position derivation 😊

So you can calculate the acceleration by replacing position with velocity on that equation. So let's try by yourself

Electronics circuit



Red = +5v
Blue = D2
Green = D3
Black = GND

Connect to Arduino uno



Code explained line by line

```
/////////////////////////////////////////  
////Implementing differential math using printer encoder////  
////written by : Muhammad Husni          ////  
/////////////////////////////////////////
```

```
#define encoderOPinA 2
```

```
#define encoderOPinB 3
```

```
#define encoderOBtn 4
```

```
int encoderOPos = 0;
```

```
float Time;
```

```
float elapsedTime;
```

```
float timePrev;
```

```
void setup() {
```

```
    Serial.begin(9600);
```

```
    pinMode(encoderOPinA, INPUT_PULLUP);
```

```
    pinMode(encoderOPinB, INPUT_PULLUP);
```

```
    attachInterrupt(0, doEncoder, CHANGE);
```

```
}
```

```
int x,last_x;
```

```
int x_dot, last_x_dot;
```

```
int x_dot_dot;
```

```
void loop() {
```

```
//direction
```

```
if(x>last_x)
```

```
{
```

```
Serial.print(" CW");
```

```
}
```

```
if(x<last_x) {
```

```
Serial.print(" CCW");
```

```
}
```

```
//got position value from interrupt
```

```
//pulse
```

```
Serial.print(" position : ");
```

```
Serial.print(x);
```

```
//got velocity/speed value from position using numerical derivation
```

```
//pulse per second
```

```
x_dot = (x-last_x)/elapsedTime;
```

```
Serial.print(" speed : ");
```

```
Serial.print(x_dot);
```

```
last_x = x;
```

```
//got acceleration value from velocity using numerical derivation
```

```
//pulse per second square
```

```
x_dot_dot = (x_dot-last_x_dot)/elapsedTime;
```



```
Serial.print(" acceleration : ");
```

```
Serial.print(x_dot_dot);
```

```
last_x_dot = x_dot;
```

```
Serial.println(" ");
```

```
delay(250);
```

```
//TIMER
```

```
timePrev = Time;           // the previous time is stored before the actual time  
read
```

```
Time = millis();           // actual time read
```

```
elapsedTime = (Time - timePrev) / 1000;
```

```
}
```

```
//position value interrupt
```

```
void doEncoder()
```

```
{
```

```
if (digitalRead(encoderOPinA) == digitalRead(encoderOPinB))
```

```
{
```

```
encoderOPos++;
```

```
}
```

```
else
```

```
{
```

```
encoderOPos--;
```

```
}
```

```
x = encoderOPos/2.5;
```

```
}
```

Serial monitor result

serial monitor result			manual calculation		
pulse	pulse/s	pulse/s^2	pulse	pulse/s	pulse/s^2
0	0	0	0	0	0
4	16	64	4	16	64
28	95	314	28	96	316
47	75	-79	47	76	-80
92	180	420	92	180	420
168	302	486	168	304	488
210	167	-537	210	168	-540
226	63	-414	226	64	-416
245	76	52	245	76	52
242	-12	-352	242	-12	-352
243	3	59	243	4	60
382	556	2212	382	556	2212
460	310	-980	460	312	-984
598	549	952	598	552	956
702	414	-537	702	416	-540
733	123	-1159	733	124	-1164

Here is the comparison between Arduino calculation and excel calculation. Not to far because of sampling rate and Arduino performance. But all and all you can use Arduino for differential math to calculate speed and acceleration.