

# **Continuous Time Causal Block Diagrams (CBDs)**

Ken Bauwens 20143225  
and Baturay Ofluoglu 20174797

November 8, 2017

# 1 Tasks

## 1.1 Integral Blocks

Integrator block for CBD is implemented by using Forward Euler Series. Forward Euler Series provide approximate solution for integrals. Refer to equation (1) for the formula of Forward Euler Series. Basically, it converts continuous time to discrete by using small time steps, called  $\Delta t$ . It calculates the area of rectangle at each step  $F(x[s], s\Delta t)\Delta t$  and add them cumulatively. Therefore, lower  $\Delta t$  requires more steps but lower error.

$$X^{n+1} = X^n + F(x[s], s\Delta t)\Delta t \quad (1)$$

For the CBD implementation of integral block refer to Appendix A. The block requires three inputs which are initial condition, input function and  $\Delta t$ . Delay and sum block are used to add the area of rectangles cumulatively and multiplication block is used to calculate  $F(x[s], s\Delta t)\Delta t$ .

## 1.2 Derivative Blocks

Derivative block for CBD is implemented by using Newtons Difference Quotient. This method is also an approximation to find the derivative of the function. It finds the slope of a function at a given  $\Delta t$ . As in the integral blocks, if  $\Delta t$  is smaller, then derivative prediction will be more accurate. Refer to equation (2) for the formula of Newtons Difference Quotient.

$$\frac{x^{[s+1]} - x^{[s]}}{\Delta t} = F(x[s+1]) \quad (2)$$

For the CBD implementation of integral block refer to Appendix B.

To implement equation (2) in CBD form, we converted it as equation (3) and (4). (Gomes, Denil, and Vangheluwe, Gomes et al.)

$$\frac{x^{[s]} - x^{[s-1]}}{\Delta t} = y^{[s]} \quad (3)$$

$$x^{[s-1]} = x^{[s]} - y^{[s]}\Delta t \quad (4)$$

$$x^{[-1]} = x^{[0]} - y^{[0]}\Delta t \quad (5)$$

Since, we know that at the initial state we cannot take the derivative, we calculate the initial state  $x^{[-1]}$  by using delay block. At the delay block as an IC input, we calculate equation (5). Then, after each state delay block keeps the previous states value to calculate differences between  $x^{[s]} - x^{[s-1]}$ .

## 1.3 Test Approximations

### 1.3.1 Harmonic Oscillator CBD with Integral Blocks

For the CBD implementation of Harmonic Oscillator with integral block refer to Appendix C.

For the plot of Harmonic Oscillator with integral block refer to Appendix F. Note that in this plot  $\Delta t = 0.01$  and it iterates 2000 times.

Integrator1 takes input from negator block which is equal to  $\frac{d^2x}{dt^2}$ . Then the output of Integrator1 block will be  $\frac{dx}{dt}$  and this output will be used by Integral2 as an input to generate  $x$ .

### 1.3.2 Harmonic Oscillator CBD with Derivative Blocks

For the CBD implementation of Harmonic Oscillator with integral block refer to Appendix D.

For the plot of Harmonic Oscillator with derivative block refer to Appendix E. Note that in this plot  $\Delta t = 0.01$  and it iterates 2000 times.

Derivator1 takes input from negator block which is equal to  $-x$ . Then the output of Derivative1 block will be  $-\frac{dx}{dt}$  and this output will be used by Derivative2 as an input to generate  $-\frac{d^2x}{dt^2}$ . The output is returned after negator because we are looking for a positive  $\frac{d^2x}{dt^2}$ .

## 1.4 Measure Error

The error between the approximated differential equation and real differential equation is calculated as equation (6).

$$e(t) = \int_0^t |\sin(t) - x(t)| dt \quad (6)$$

For the CBD implementation of Error Block refer to Appendix G.

In this Error block,  $\Delta t$  is a constant block that is directly used by Derivative or Integral block. However, for SinusBlock it needs to be multiplied with TimeBlock to obtain the same time with the Derivative or Integral blocks. Because if SinusBlock only uses TimeBlock as an input, then the input time would be the current iteration. By default current iteration increases by 1 and suppose  $\Delta t = 0.1$ . In this case, the time for SinusBlock would be 1 while the time for Derivative or Integral Block is 0.1. To avoid inconsistencies, we multiplied TimeBlock and  $\Delta t$ , to obtain the same time. Note that, you should use default current iteration as 1. It should not be changed to use this Error Block. However,  $\Delta t$  may change base on preference.

#### 1.4.1 Harmonic Oscillator Error Comparison between Derivative and Integral Block implementations

Harmonic Oscillator approximation by using derivative blocks is plotted at E. As it can be seen, when time increases the amplitude decreases and  $\lim_{t \rightarrow \infty} x(t)$  goes to 0.

In addition, Harmonic Oscillator approximation by using integral blocks is plotted at F. For this case when time goes to infinity the amplitude also goes to infinity and for the  $\lim_{t \rightarrow \infty} x(t) = \infty$ .

The value for  $\sin(t)$  is always between 1 and -1. Thus, without plotting the error results, it is strait forward that in the long run derivative block will generate less error for the same  $\Delta t$  values based on equation (6).

The two different error plot for Harmonic Oscillator approximation by using Integral Blocks is given at H for 10000 iterations and 0.001  $\Delta t$  and at I for 100 iterations and 0.1  $\Delta t$ .

For Harmonic Oscillator approximation by using Derivative Blocks is given at J for 10000 iterations and 0.001  $\Delta t$  and at K for 100 iterations and 0.1  $\Delta t$ .

The iteration sizes are different but all plots have the same time intervals. The ending simulation time will be calculated as  $\Delta t(\#Iterations) = SimulationEndingTime$  for both cases. Thus, we evaluate the simulation between time 0 and 10.

For small  $\Delta t$  value as 0.001, the error difference between the integral and derivative implementations is almost negligible. However, when  $\Delta t$  increases to 0.1, then derivative implementation performs much better than integral. For  $t=10$ , derivative has almost 1.2 error score. However, integral has 1.7 error point. Thus, for this case using derivative blocks performs 42% better than using integral blocks.

Increasing  $\Delta t$  also increases the error. For instance, when  $\Delta t = 0.1$  using integral blocks at time=10 has 1.7 error. However, for smaller  $\Delta t$  as 0.001 the error is around 0.015.

Note that there is a tradeoff between  $\Delta t$  and execution time. When  $\Delta t$  decreases, accuracy of the model increases but execution time increases as well. For very small  $\Delta t$  values, computing may be impossible in a feasible time.

## 2 Driverless Train

### 2.1 Computerblock

Our implementation of the computerblock is simple. The table of values is hardcoded into the block. It takes the current time as input and outputs the correct target velocity depending on the current time.

### 2.2 Driverless Train CBD

The CBDmodels for the driverless train can be found in the appendix (L, M, N).

To simplify the implemented ODE's we precalculated some of the constants. If we have the system of ODE's:

$$\begin{cases} \frac{dv_{passger}}{dt} = \frac{k(-x_{passger}) + c(-v_{passger}) - m_{passger} * \frac{F_{traction}}{(m_{train} + m_{passger})}}{m_{passger}} \\ \frac{dv_{passger}}{dt} = \frac{F_{traction} - \frac{1}{2} \cdot p \cdot v_{train}^2 \cdot C_D \cdot A}{(m_{train} + m_{passger})} \\ \frac{dx_{passger}}{dt} = v_{passger} \\ \frac{dx_{passger}}{dt} = v_{train} \end{cases}$$

We can precalculate  $\frac{1}{2} \cdot p \cdot v_{train}^2 \cdot C_D \cdot A$ ,  $m_{train} + m_{passger}$  and  $m_{passger} * \frac{F_{traction}}{(m_{train} + m_{passger})}$  using the constants given in the assignment to simplify the resulting CBD's.

#### 2.2.1 Delta.t

We tried using a delta.t of 1 at first, executing a step every second. This resulted in large errors caused by the integralblocks in the PlantCBD. A better value for delta.t is 0,1. This value causes the cbd to run for a couple of seconds, but the resulting errors are within acceptable ranges.

#### 2.2.2 Results

The plots for the train velocity and person displacement using (200,0,0) for (Kp, Ki, Kd) can be found in the appendix (O, P).

## 2.3 Tuning the PIDController

We took inspiration from the *hillclimbing* algorithm and the *Simulated annealing* algorithm and created a simple algorithm to try and find a better combination for  $(K_p, K_i, K_d)$ . The algorithm can be described as follows:

1. Choose a random configuration with values between 0 and 400
2. Pick 3 random values in the interval  $[-5,5]$
3. Add each of the items to the correct value  $\rightarrow (K_p + r_1, K_i + r_2, K_d + r_3)$
4. Run the cbd with the new values for  $(K_p, K_i, K_d)$
5. Evaluate the score
6. If score < bestscore, save the configuration and the score somewhere
7. Increment a counter with 1
8. If counter%10 == 0, go to 1 else go to 2

This loop can be ran as long as the user wants. Longer runs will be more likely to find a better solution. We ran this for 500 loops.

Step 8 exists to prevent getting stuck in local maxima.

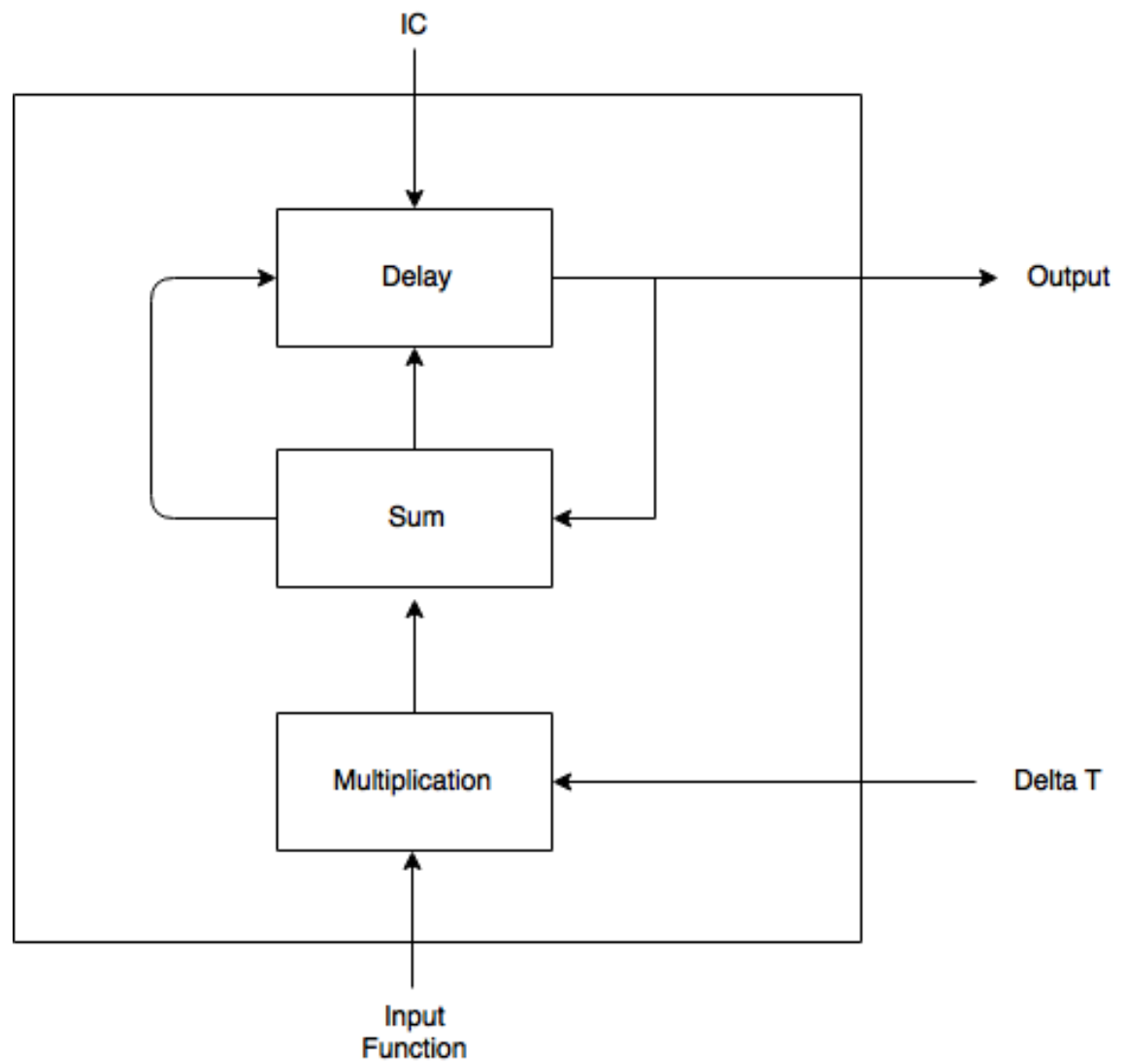
The best solution by using this algorithm was these parameters (429,252,224) with a score of 1055. For reference, the solution (200,0,0) has a score of 157010,5. Keep in mind that this is not necessarily the absolute best configurations, but it gives a low score while still staying within the requirements.

The plot for the velocity of the train can be found in appendix Q. We see that the velocity oscillates alot, but still converges to the desired speed.

The plot for the displacement of the passengers can be found in appendix S. We see that the displacement is never greater than 0,4. In appendix R we plotted the displacement together with the acceleration of the train.

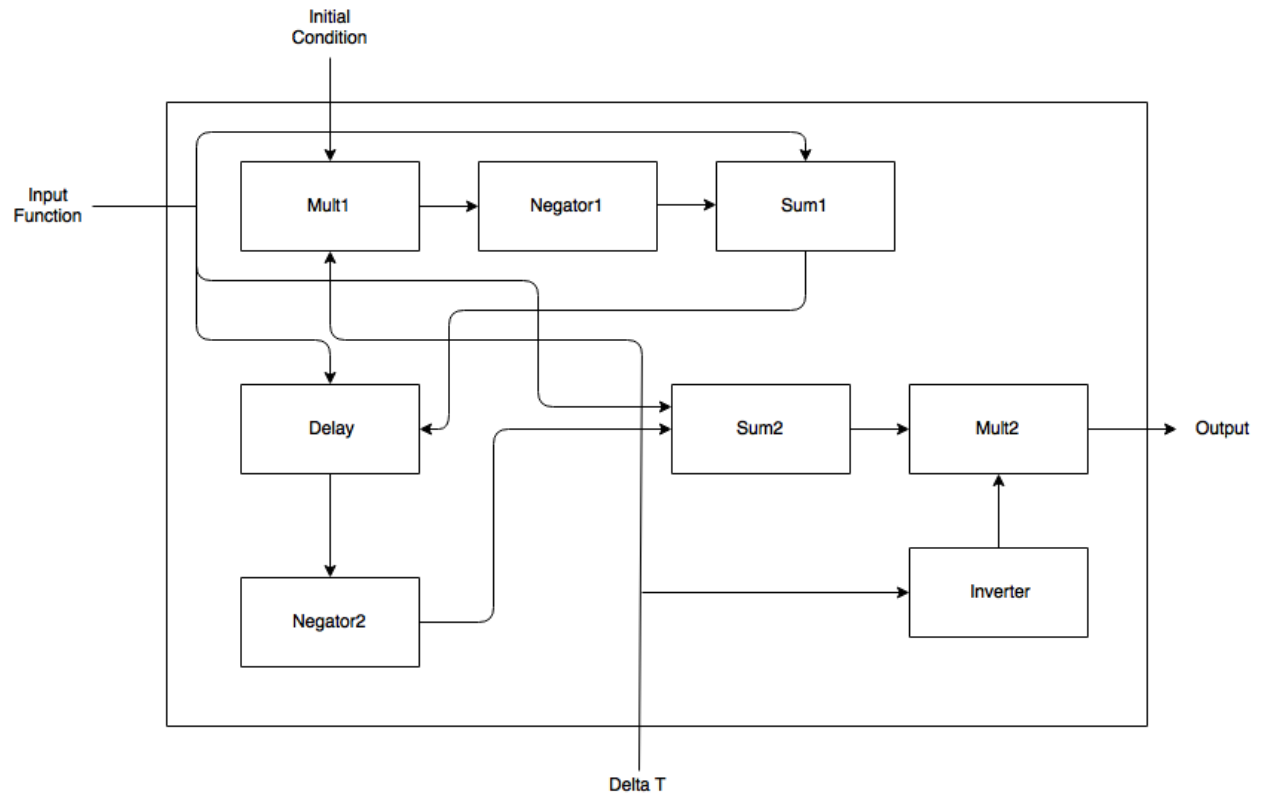
# Appendices

## A Integral Block CBD



**INTEGRAL BLOCK**

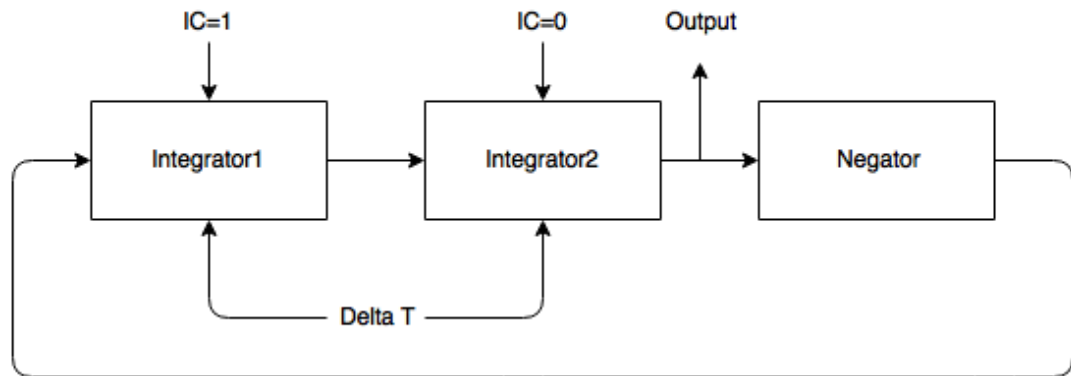
## B Derivative Block CBD



**Derivative Block**

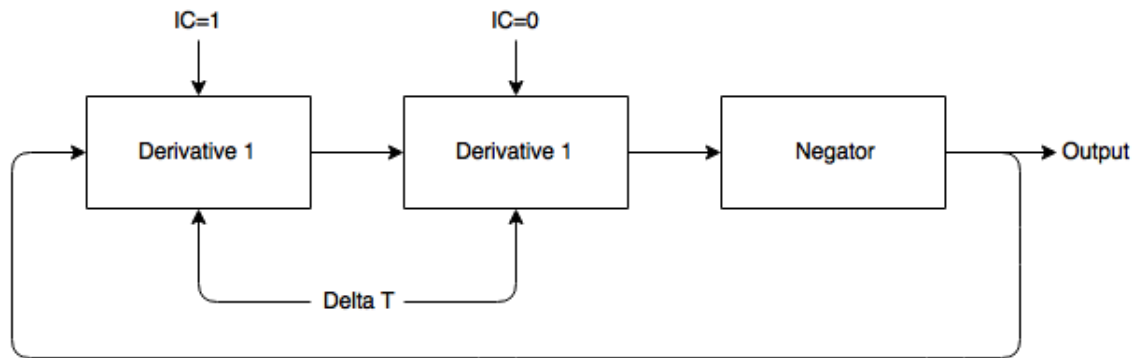


### C Harmonic Oscillator Integral Block CBD



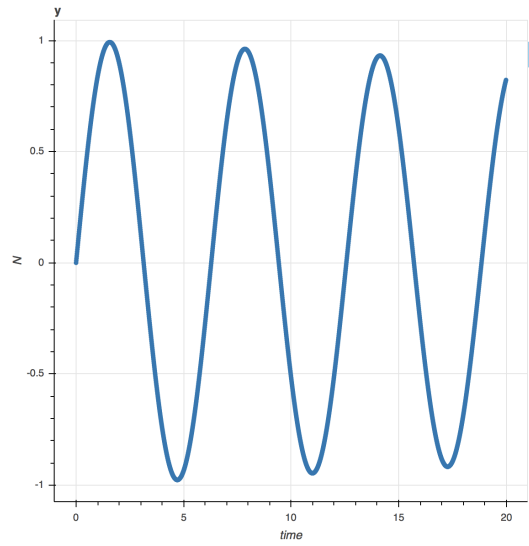
### Harmonic Oscillator with Integral Blocks

### D Harmonic Oscillator Derivative Block CBD

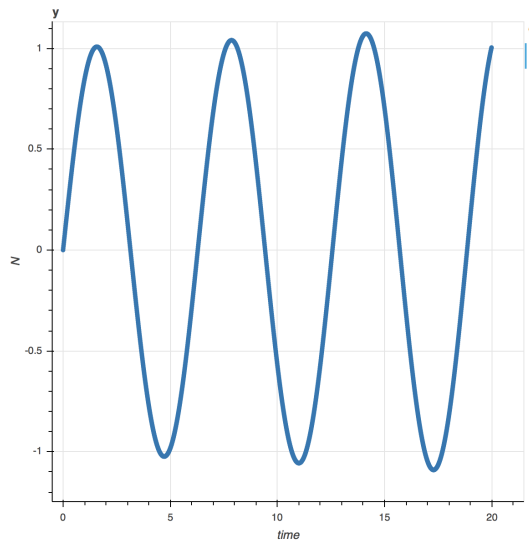


### Harmonic Oscillator with Derivative Blocks

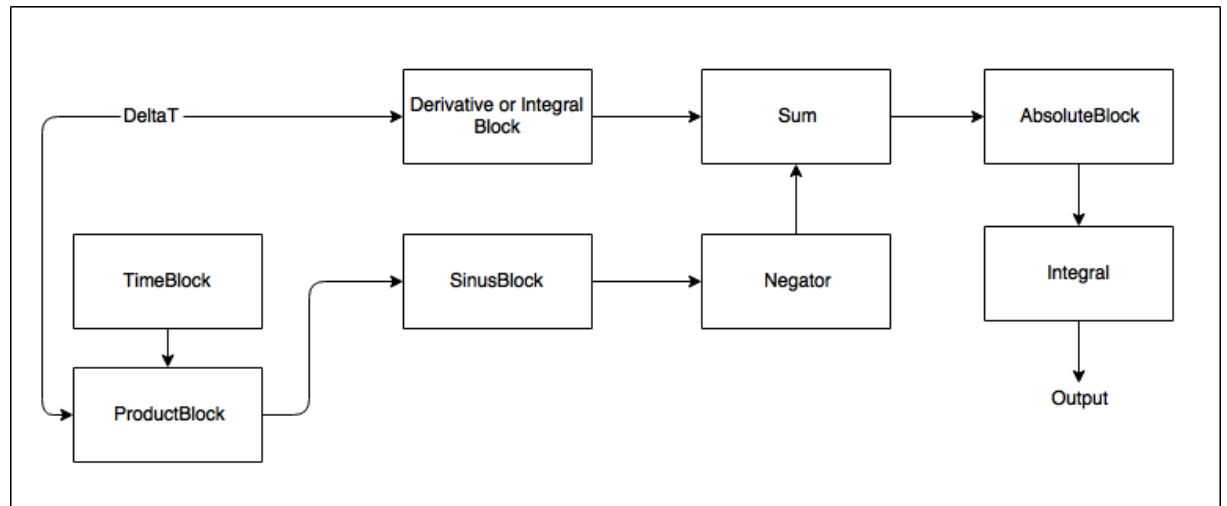
## E Harmonic Oscillator Derivative Block Plot



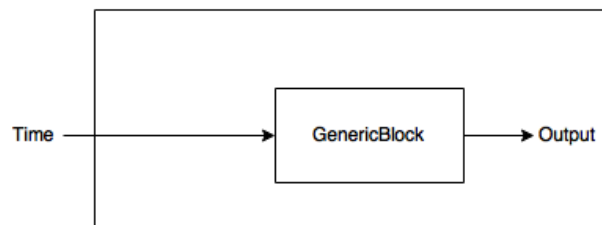
## F Harmonic Oscillator Integral Block Plot



## G Harmonic Oscillator Integral/Derivative Error Calculator CBD

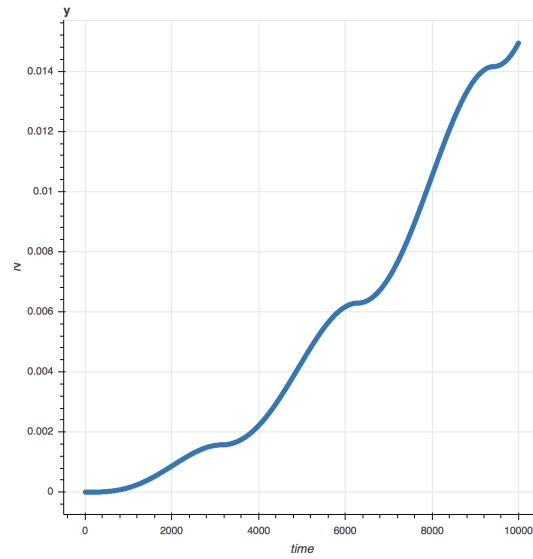


### ErrorBlock

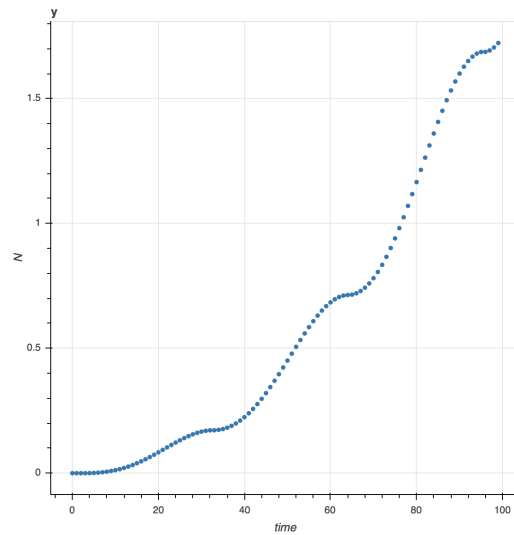


### SinusBlock

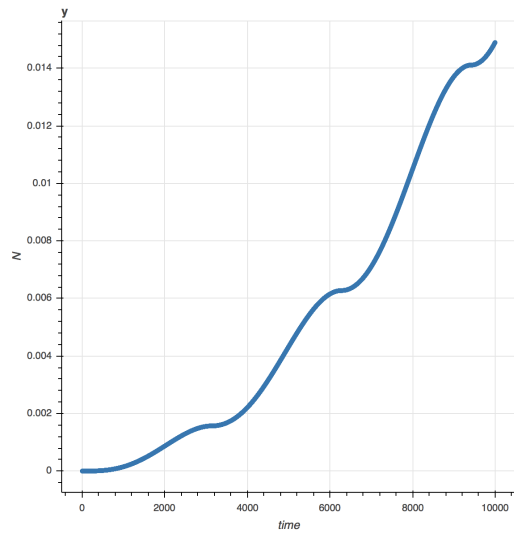
## H Harmonic Oscillator Integral Block Error Plot (Iteration=10000 $\Delta t = 0.001$ )



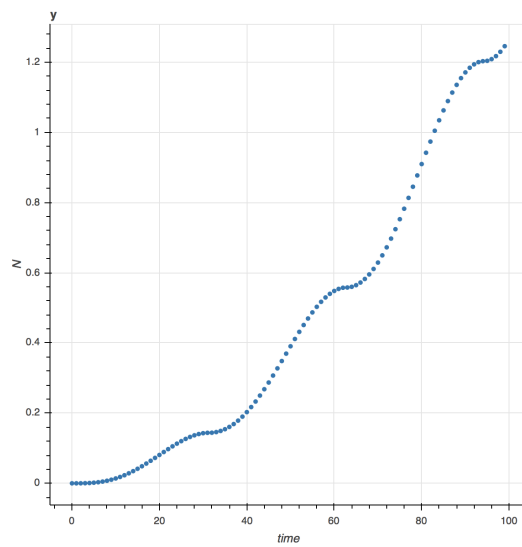
## I Harmonic Oscillator Integral Block Error Plot (Iteration=100 $\Delta t = 0.1$ )



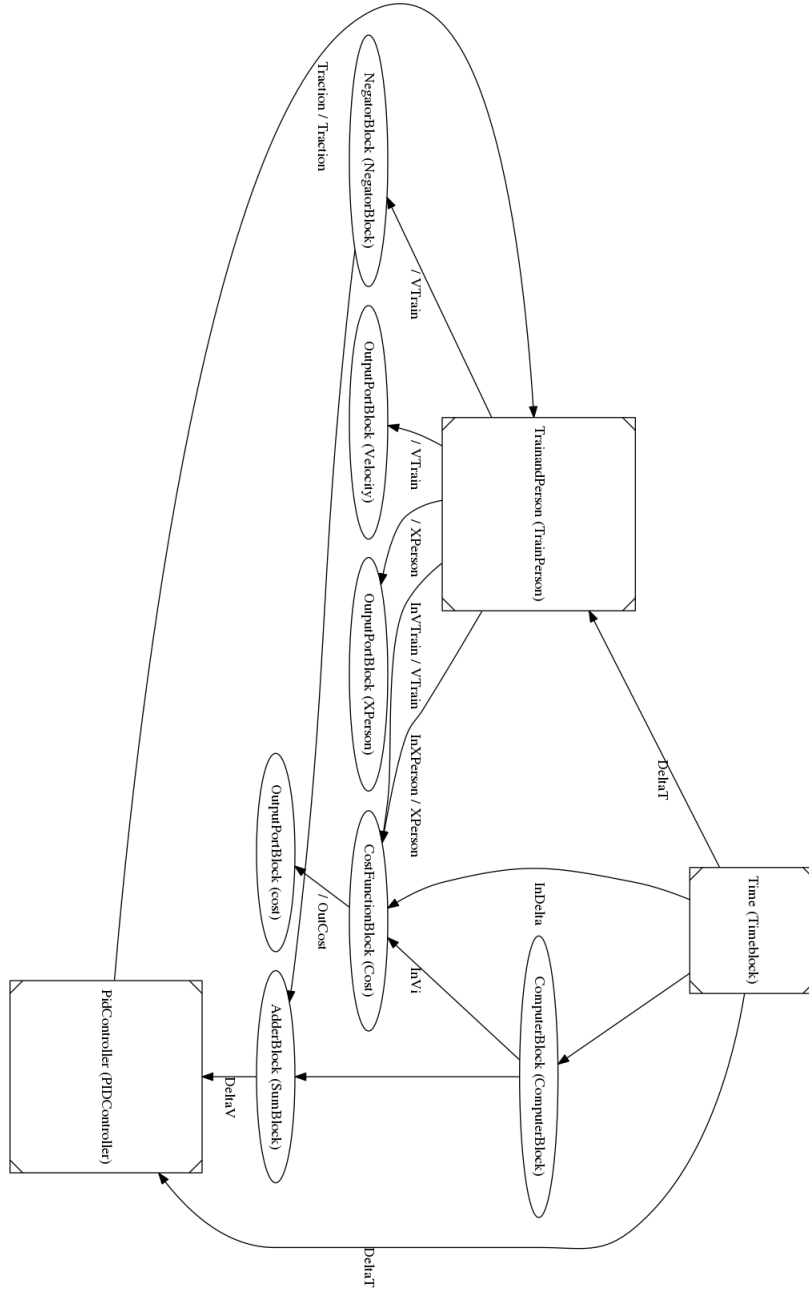
**J Harmonic Oscillator Derivative Block Error  
Plot (Iteration=10000  $\Delta t = 0.001$ )**



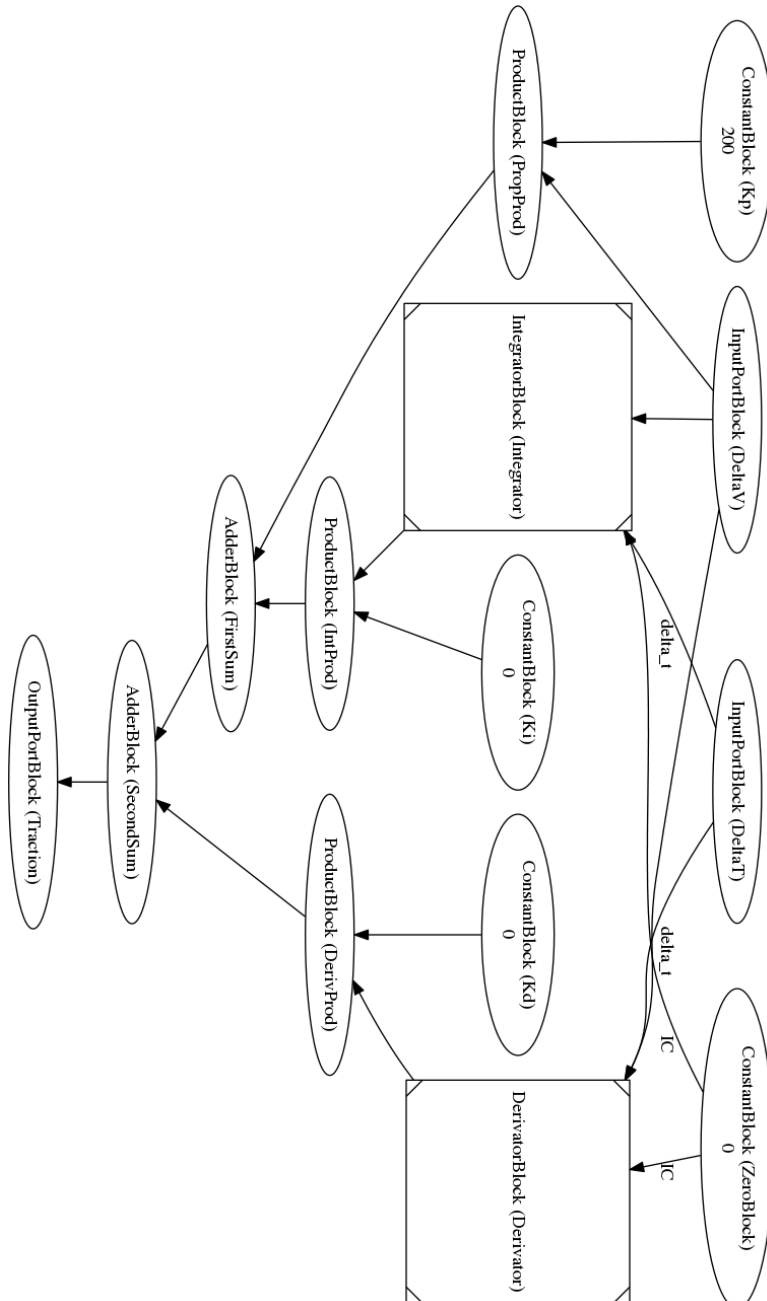
**K Harmonic Oscillator Derivative Block Error  
Plot (Iteration=100  $\Delta t = 0.1$ )**



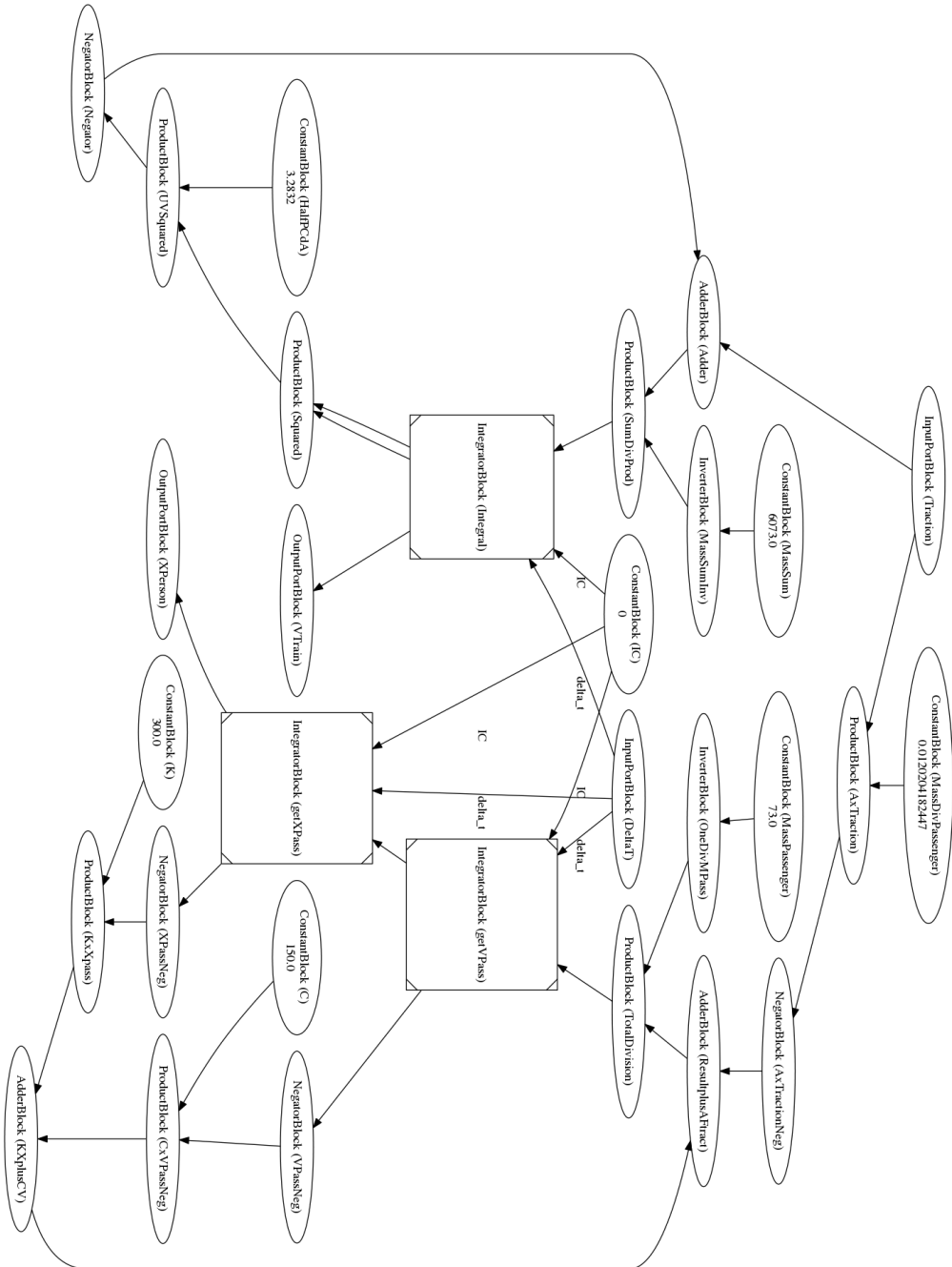
## L CBD Diagram DriverlessTrain



## M CBD Diagram PIDController

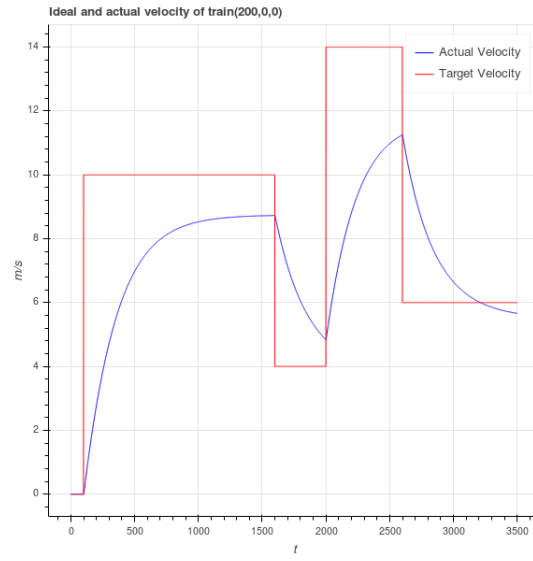


## N CBD Diagram Plant

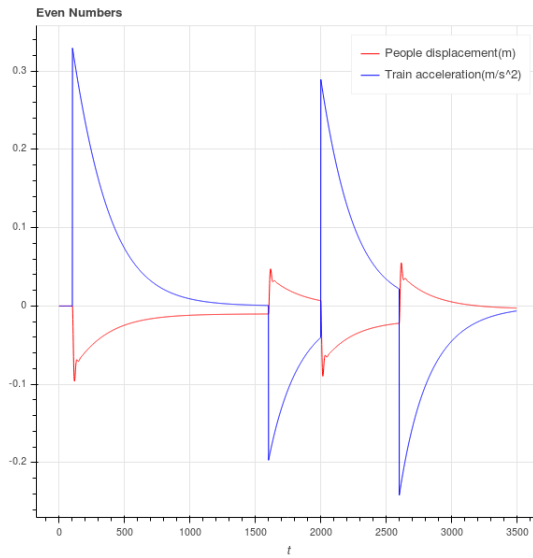




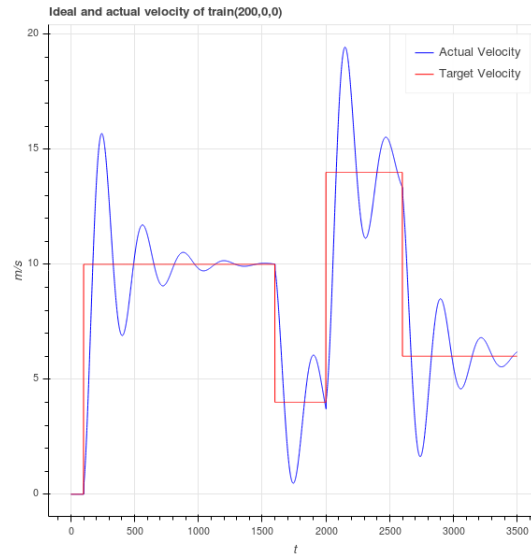
## O Velocity (200,0,0)



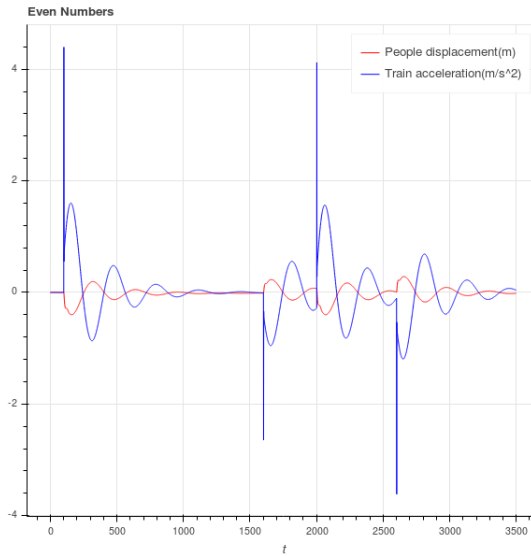
## P Displacement (200,0,0)



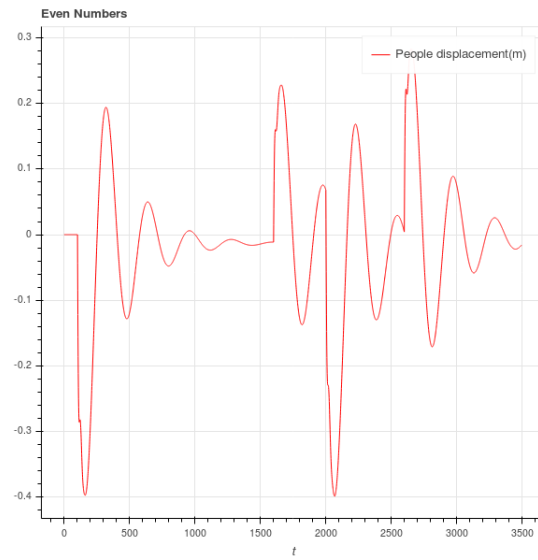
## Q Velocity with Tuned Parameters



## R Displacement with Tuned Parameters



## S Only displacement with Tuned Parameters



## References

Gomes, C., J. Denil, and H. Vangheluwe. Causal-block diagrams. Technical report.