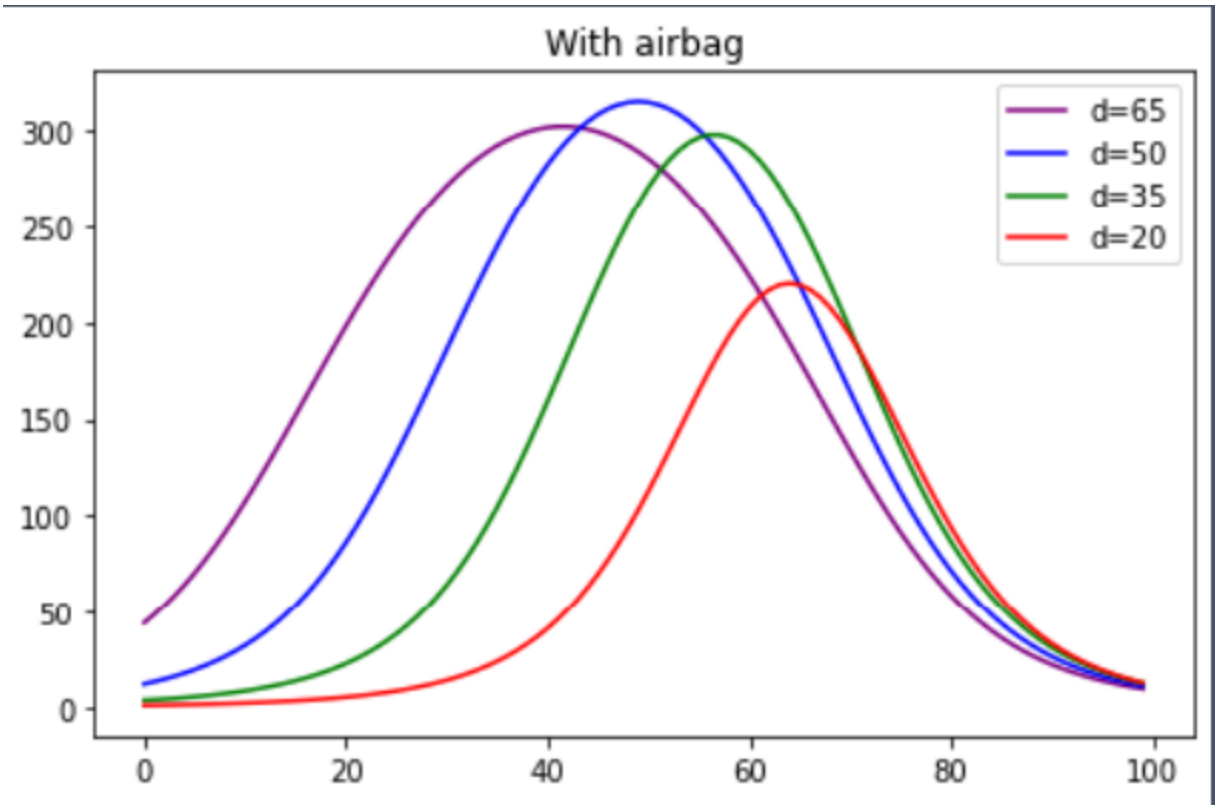
Lab 5 Documentation

HID Crash test modeling:

Below are the waveforms of the HIC system that is being modeled through Python



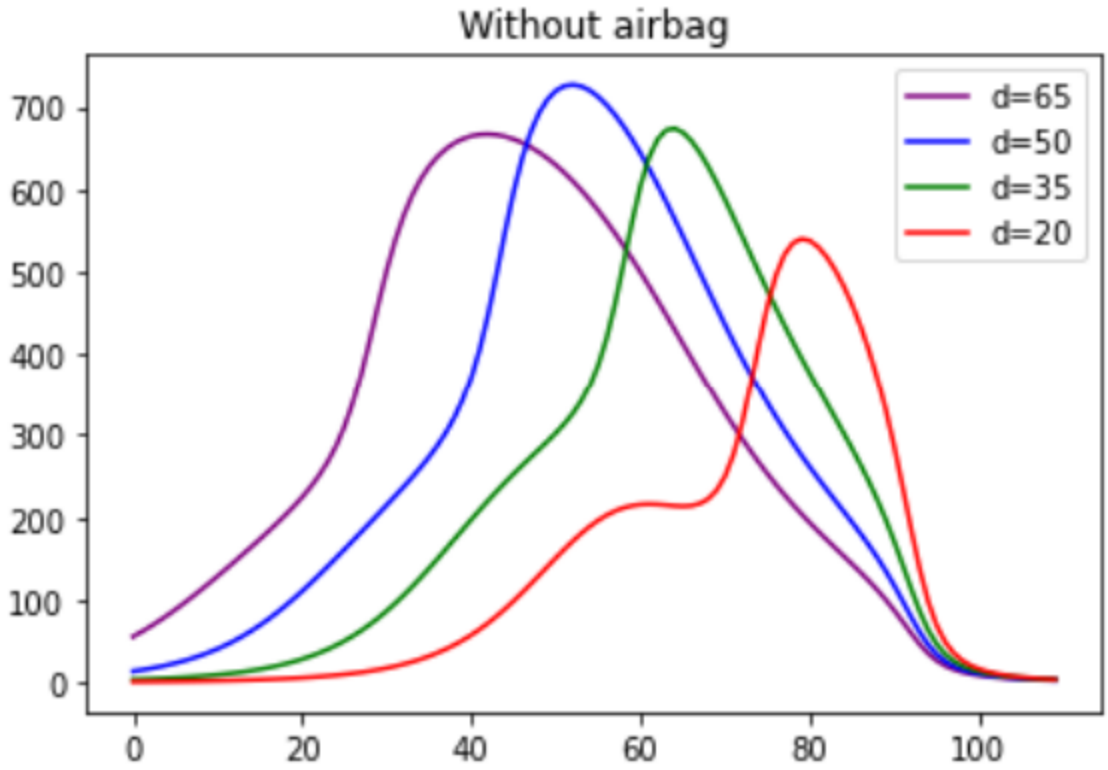
Function used to generate code:

```
def f1(t):
    if t == 74:
        return(22000/500)
    else:
        return (22000/(((t-74)*(t-74))+500))

def f2(d, t):
    return integ.quad(f1,t,(t+d))

def f3(d, t):
```

```
    result=f2(d, t)
    return (1/d*(result[0]))

def f4(d, t):
    return (d*(f3(d,t)**2.5))
```

The model is simpler for the airbag case, as the deceleration is smoother and is almost bell-shaped. Some modelling achieves the following expression for the acceleration:

$$a(t) = \frac{22000}{(t-74)^2 + 500}$$



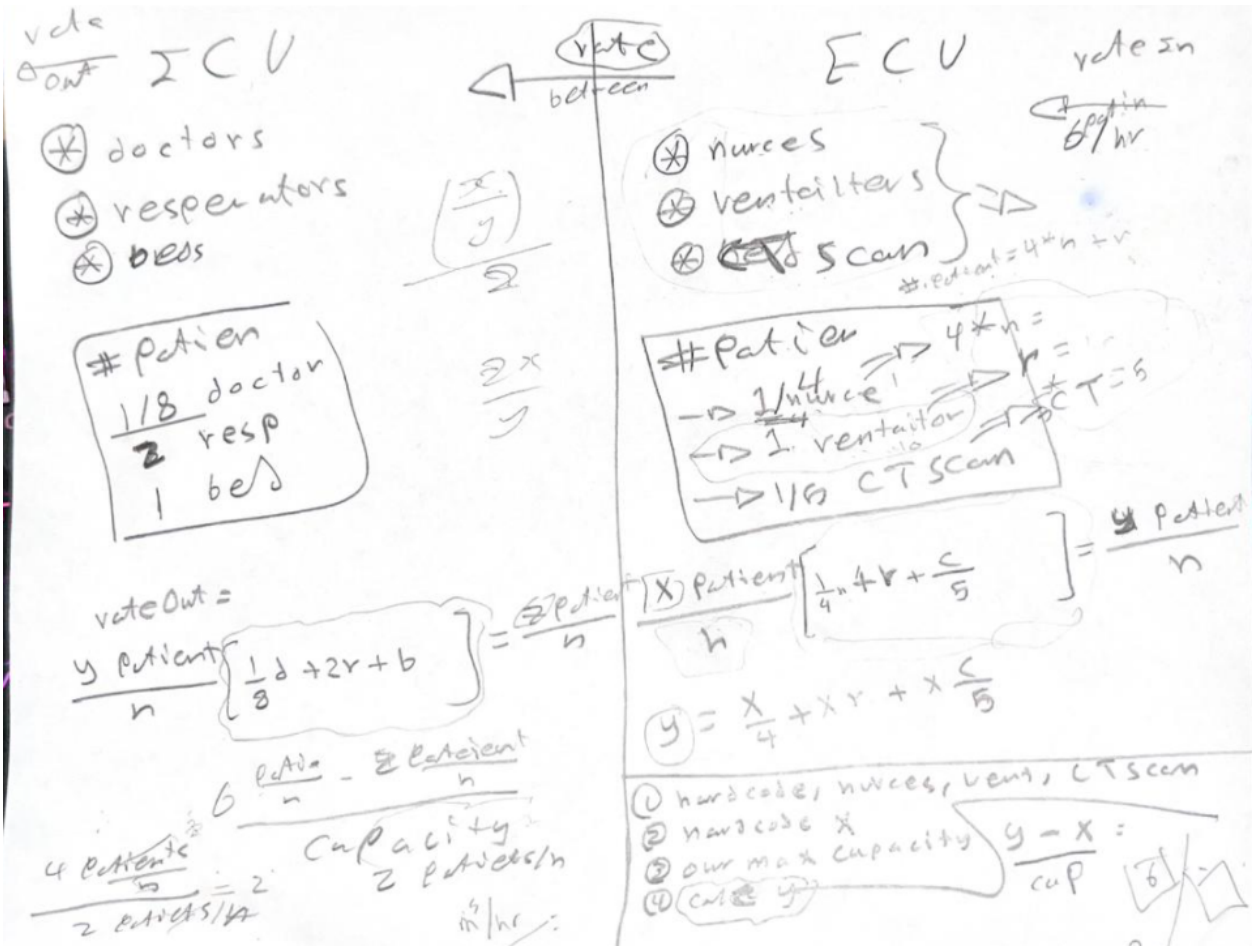
We can simplify that part of the HIC formula in braces, { }, for different values of *d*= *t*2− *t*1. We define a family of curves:

$$H_{t,d} = d\left(\frac{1}{d}\int_t^{t+d} a(T)dT\right)^{2.5}$$

We then vary the value of *d.* The value of the highest peak of the family of curves obtained gives us the HIC.

```
def f1(t):
    t = (16400/((t-68)**2 + 400) + (1480/((t-93)**2 + 18)))
    return t

def f2(d, t):
    return integ.quad(f1,t,(t+d))

def f3(d, t):
    result=f2(d, t)
    return (1/d*(result[0]))

def f4(d, t):
    return (d*(f3(d,t)**2.5))
```

ECU and ICU Hospital Modeling

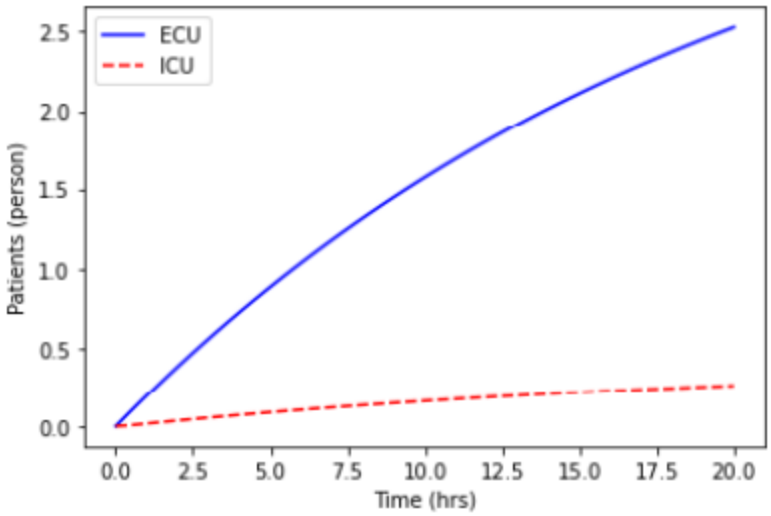


Scrathe paper on equation.

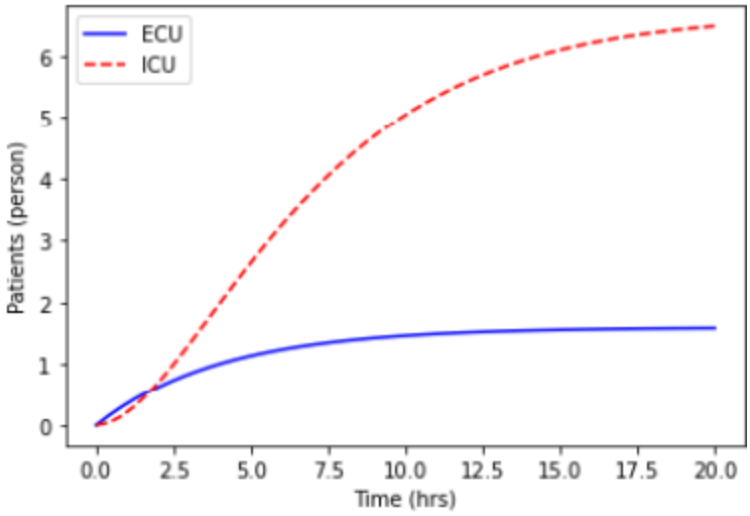
We used the code provided on Solve Differential Equations in Python (apmonitor.com)

Where we did the differential equation using the ODIENT function

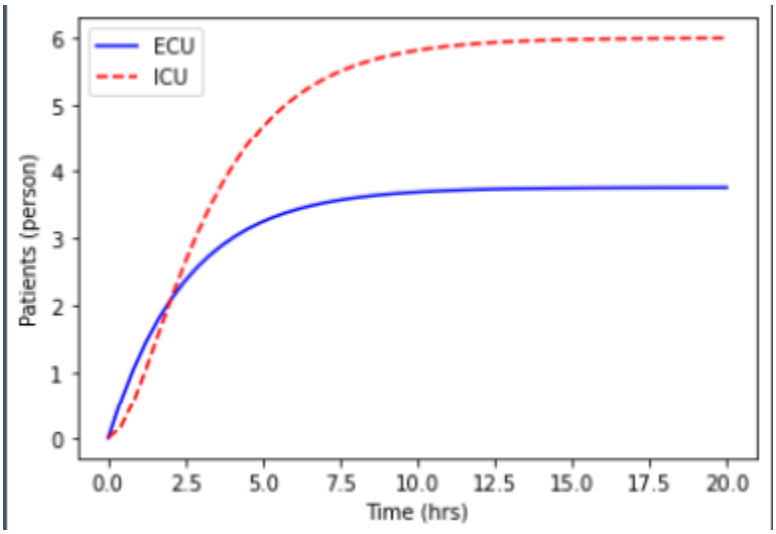
ICU saturated:



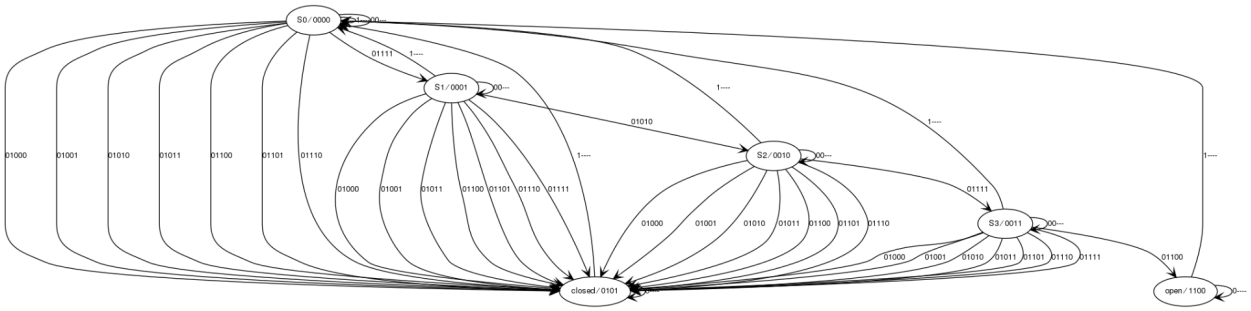
ECU saturated:



ECU and ICU both saturated



Passcpde: 7274

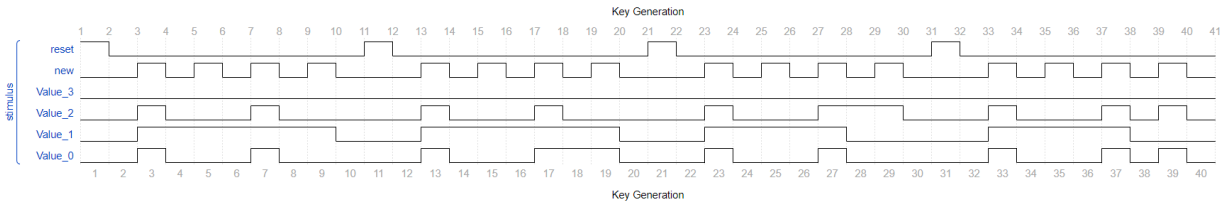


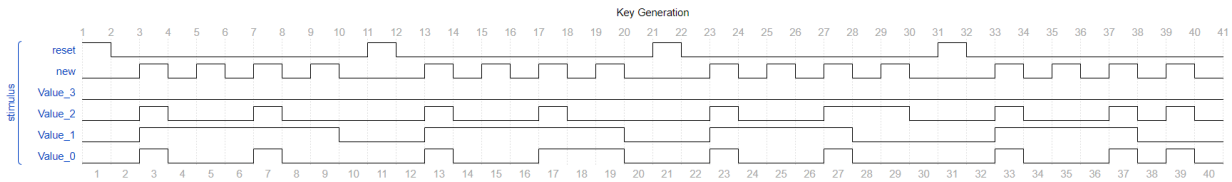

```
#HS (Harkiret Singh) = 7274
fsm_spec = {
    'inputs': [('reset', 'D0'), ('new_code_digit', 'D1'),
               ('digit_bit2', 'D2'),
               ('digit_bit1', 'D3'),
               ('digit_bit0', 'D4')],
    'outputs': [('lock_status', 'D5')],
    'states': ['S0', 'S1', 'S2', 'S3', 'open', 'closed'],
    'transitions': [['1----', 'S0', 'S0', '0'],
                    ['1----', 'S1', 'S0', '0'],
                    ['1----', 'S2', 'S0', '0'],
                    ['1----', 'S3', 'S0', '0'],
```

```
                    ['1----', 'closed', 'S0', '0'],

                    ['00---', 'S0', 'S0', '0'],
                    ['01000', 'S0', 'closed', '0'],
                    ['01001', 'S0', 'closed', '0'],
                    ['01010', 'S0', 'closed', '0'],
                    ['01011', 'S0', 'closed', '0'],
                    ['01100', 'S0', 'closed', '0'],
                    ['01101', 'S0', 'closed', '0'],
                    ['01110', 'S0', 'closed', '0'],
                    ['01111', 'S0', 'S1', '0'], #input '7' makes jump to S1

                    ['00---', 'S1', 'S1', '0'],
                    ['01000', 'S1', 'closed', '0'],
                    ['01001', 'S1', 'closed', '0'],
                    ['01010', 'S1', 'S2', '0'],#input '2' makes jump to S2
                    ['01011', 'S1', 'closed', '0'],
                    ['01100', 'S1', 'closed', '0'],
                    ['01101', 'S1', 'closed', '0'],
                    ['01110', 'S1', 'closed', '0'],
                    ['01111', 'S1', 'closed', '0'],

                    ['00---', 'S2', 'S2', '0'],
                    ['01000', 'S2', 'closed', '0'],
                    ['01001', 'S2', 'closed', '0'],
                    ['01010', 'S2', 'closed', '0'],
                    ['01011', 'S2', 'closed', '0'],
                    ['01100', 'S2', 'closed', '0'],
                    ['01101', 'S2', 'closed', '0'],
                    ['01110', 'S2', 'closed', '0'],
                    ['01111', 'S2', 'S3', '0'],#input '7' makes jump to S3

                    ['00---', 'S3', 'S3', '0'],
                    ['01000', 'S3', 'closed', '0'],
                    ['01001', 'S3', 'closed', '0'],
                    ['01010', 'S3', 'closed', '0'],
                    ['01011', 'S3', 'closed', '0'],
                    ['01100', 'S3', 'open', '0'],#input '4' opens the safe
                    ['01101', 'S3', 'closed', '0'],
                    ['01110', 'S3', 'closed', '0'],
                    ['01111', 'S3', 'closed', '0'],

                    ['0----', 'closed', 'closed', '0'],

                    ['1----', 'open', 'S0', '1'],
                    ['0----', 'open', 'open', '1']]}
```

```
reset =          'hl........' + 'hl........' + 'hl........' + 'hl........'
new_code_digit = 'l.hlhlhlhl' + '..hlhlhlhl' + '..hlhlhlhl' + '..hlhlhlhl'
```

```
# digits:  _______7_2_7_2_______7_2_7_3_______7_2_7_4_______7_2_7_5

digit_bit_3 = 'l.........' + '..........' + '.........' + '.........'
digit_bit_2 = 'l.hl..hl....' + 'hl..hl....' + 'hl..h..l..' + 'hl..hlhl'
digit_bit_1 = 'l.h......l' + '..h......l..' + 'h....l....' + 'h....l..'
digit_bit_0 = 'l.hl..hl....' + 'hl..h..l..' + 'hl..hl....' + 'hl..hlhl'

key_gen = {'signal': [
    ['stimulus',
     {'name': 'reset', 'pin': 'D19',
      'wave': reset},
     {'name': 'new', 'pin': 'D18',
      'wave': new_code_digit},

    {'name': 'Value_3', 'pin': 'D17',
      'wave': digit_bit_3},
     {'name': 'Value_2', 'pin': 'D16',
      'wave': digit_bit_2},
     {'name': 'Value_1', 'pin': 'D15',
      'wave': digit_bit_1},
     {'name': 'Value_0', 'pin': 'D14',
      'wave': digit_bit_0}]],
    'foot': {'tock': 1, 'text': 'Key Generation'},
    'head': {'tick': 1, 'text': 'Key Generation'}}

waveform = Waveform(key_gen)
waveform.display()
```





Lock opens as it touches 7274