MODELLING OF A VEHICLE SUSPENSION SYSTEM

ÉPREUVE SYNTHÈSE DE PROGRAMME (ESP)

Differential Equations 201-HTL-VA
Probability and Statistics 201-HTH-05

BY

TADEUS TALMACI
KAMILA YAHIAOUI

Vanier College
May 28, 2020

Abstract

Dynamic systems involving springs, masses and dampers are commonly used in the automotive industry. The suspension system is an example of dynamic system that is essential to the proper functioning of any road vehicle. This article explains the physics and the numerical solvers necessary to model a realistic vehicle suspension system. A system of second order differential equations is derived from Newton's Second Law, which is then converted into a system of four coupled ordinary differential equations in order to be solved using the Runge-Kutta 45 numerical approach. Poisson process randomises the intervals of time between jolts affecting the suspension and helps to realistically model the system rather than simply illustrating its ideal behavior. Finally, graphs of the Gamma distributed jolts and the suspension system animations are computed using Python coding in Jupyter Notebook.

1 Introduction

A suspension system is a very complex mechanism with multiple forces acting on it, such as: vibrations coming from the vehicle itself, vibrations coming from the roughness of the road, the down-force of the wind, the braking and the acceleration process of the vehicle. However, in this article, only the essentials of such a system will be discussed, focusing mainly on modelling the behavior of the suspension system in randomly potholed road conditions.

The suspension system is the only mechanism that separates the chassis of the vehicle from the wheels directly in contact with the road. This dynamic system plays an important role in the handling, the stability and the comfort of the ride. Without an appropriate suspension system, driving would become a tiring and unpleasant experience.

The shock absorbers, the linkages and the springs work together to isolate the body of the vehicle along with the passengers from shocks arising from the road, and to enables braking, driving and steering systems to operate correctly. Briefly, linkages link the suspension components to the wheels of the car and to the chassis, springs soften the vehicle by oscillating and dampening the shock absorbed from a bump or a pothole, and shock absorbers, in addition to absorbing the shock using the motion of a hydraulic piston moving in a cylinder filled with viscous oil, they also dampen the oscillations of the springs after the impact.

There are three ways the suspension system operates: passively (the cheapest way; the most commonly used in the automotive world), semi-actively (more expensive; contains an integrated regulator to control the stiffness of the shock absorber by using a magnetorheological fluid rather than regular oil), and actively (the most expensive; contains an actuator, a sensors and a central processing unit (CPU) that allows the suspension, by the use of additional force, to adapt in advance to the bumpiness of the road in front of the vehicle).

To realistically model this system, we begin by deriving and explaining the differential equations representative of a quarter car suspension system. Then, the numerical solver used to solve the system of equations is discussed. Runge-Kutta 45 is the numerical solver used to approximate the solution to this system because it is a more accurate solver than other similar methods that will be described in this article. Moreover, a random number generator along with a Poisson process will randomise the time between each jolt on the suspension system. The distance between Poisson events is exponentially dis-

tributed with parameter lambda for Poisson rate. The shocks themselves will have Gamma shapes and will be plotted using Python language in Jupyter Notebook. Graphs of the displacement of the shock absorbers will also be presented in addition to an animation of the suspension system of a quarter car (wheel with tire and suspension components holding a quarter of the vehicle's weight). Three different runs of the code, which is released at **GitHub**, will be included and analysed in the article with variation of constants.

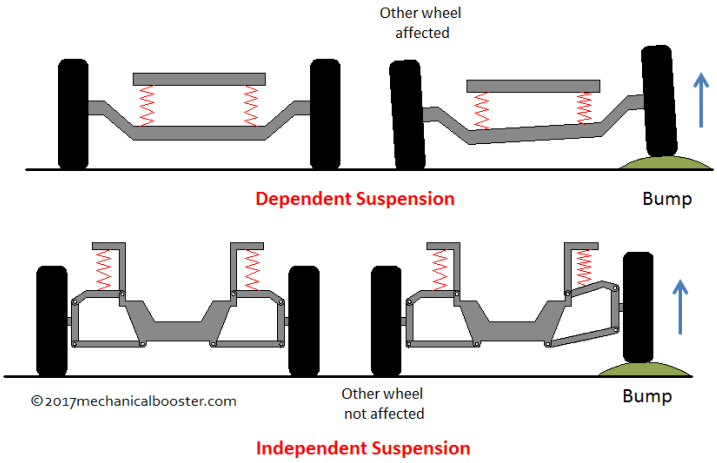
2 Physics of the suspension system

2.1 Types of Suspension Systems

To begin with, there are two main types of suspension systems that can be installed on a vehicle: a dependent system or an independent system. The main difference between these two types is that in the dependent system, a solid rod connects the wheels from opposite sides together, which makes the forces acting on the wheels of one side of the vehicle depend on the forces acting on the wheels on the opposite side. If the right wheel hits a bump, the left wheel will also be affected by the shock taken by the right wheel. However, in an independent system, the shock on the right wheel will only affect that same wheel and not the one on the opposite side as the two wheels are not rigidly linked together. They are said to be independent as they both can move vertically with no effect on the parallel wheel. In such a system, one might wonder how is the power transmitted through the drive shaft to wheels on opposite sides if no linkage is made between them. This is why engineers have designed drive shafts that are synchronised with the suspension components and that are able to move vertically along with the suspension while transmitting torque from the transmission to the wheels simultaneously. *Figure 1* illustrates well the difference between these two types of suspension.

For the purpose of this paper, we will use an independent system model as all computations and explanations will be based on a quarter car suspension model and will focus on the forces acting on a single wheel at a time. Obviously, we will assume that the weight of the car is equally distributed and that one wheel supports a quarter of the total car's weight.

Figure 1: Dependent and Independent Suspension Systems



Vehicle manufacturers use multiple suspension designs in their products as one specific design fits better a certain model of vehicle for optimal driving experience. Within those suspension designs, one finds:

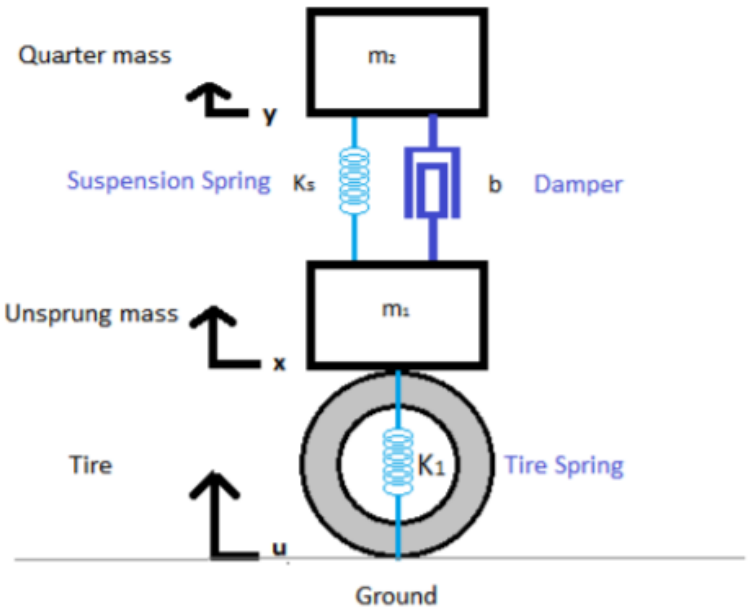
- The Macpherson Strut Suspension, the simplest one and the most commonly used in urban vehicles in which the wheel and the chassis are connected through a spring;
- The Leafspring Suspension, commonly used in heavier vehicles such as trucks and pick-ups, consists of long metal flexible bars that absorb the shocks;
- The Double Wishbone Suspension, made of two A-shaped arms that look like a bird's wishbone, is mostly used on high performance sports cars;
- The Pushrod Suspension has a single spring in between the two wheels and it is called a monoshock suspension (often used on motorcycles).

2.2 Quarter Car Diagram

From *Figure 2*, m_2 represents the quarter of the total car's mass (sprung mass) which lays on the suspension spring, with spring constant K_s, and on the shock absorber which acts as a damper of the oscillations with damping

constant b. The unsprung mass m_1 consists of the weight of the wheel and of the suspension components themselves. Finally, the tire also acts as a spring in this context, with spring constant K_1, but the damping factor of the tire is not considered in this illustration. The tire is the object that separates the unsprung mass from the ground falsely illustrated as being straight and linear, but it will not be considered as so in our computations. The vertical positions of m_1, m_2 and the tire on the ground are x, y and u respectively. As the road is not perfectly linear, any fluctuations will result in changes in the masses' positions and velocities which will be denoted as $x(t)$ and $x'(t)$ for the unsprung mass, and as $y(t)$ and $y'(t)$ for the sprung mass.

Figure 2: Quarter Car Suspension Diagram



Whenever the wheel hits a hole, the spring elongates by $(y - x)$ and the tire does the same, by a length of $(x - u)$, because the positive vertical displacement is assumed upwards in the illustration. As the spring elongates, the size of its deformation is directly proportional to the force applied on it. This concept is stated in Hooke's Law with the following formula:

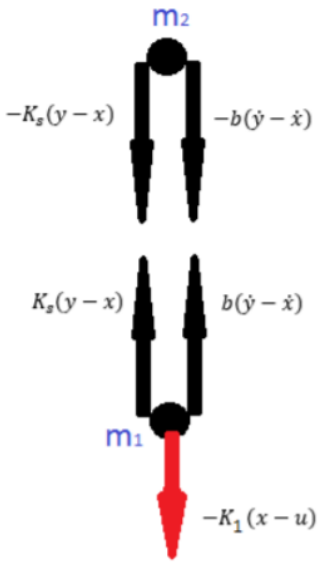
$$F = -kx$$

In this formula, F represents the force, x stands for displacement, and k is the spring constant. In this situation, the spring constant is negative because the force acts in the opposite direction of the displacement. The spring tries to resist the changes produced by the applied force as it always goes back to its equilibrium position. Moreover, just like k is related to x, the viscous damping coefficient b is related to velocity x'. Why viscous? Because the damper in this system is the shock absorber and this component is made of a piston that moves up and down in a cylinder filled with a viscous fluid (oil) to create resistance. Therefore, the dampening effect is created by bringing to zero the velocity of the piston's motion as fast as possible.

Hitting a hole on the road generates the spring's deformation. As it does so, it also brings down the sprung mass (the body of the vehicle) and due to its spring constant (K_s) and dampening coefficient (b), it tries to bring back up the unsprung mass m_1 simultaneously. A very similar scenario occurs between the tire and the unsprung mass m_1. The free body diagram helps to analyse the forces acting on each mass and component. Briefly, as we hit the hole, K_s and b act as negative forces on m_2 and as positive forces on m_1. In addition, as K_1 pulls m_1 downwards, K_1 acts as a negative force on m_1. Consequently, m_1 opposes to the force and acts as a positive force on the tire, trying to bring it back up (in the opposite direction). The forces acting in this system are purely applications of Newton's Second Law from which we derive the system of two second-order differential equations.

2.3 Free Body Diagram

Figure 3: Free Body Diagram



From the free body diagram (*Figure 3*), the forces acting on the two masses are resumed as follows:

- a superposition of two downward forces act on the chassis of the vehicle (m_2), which are $-K_s(y-x)$ and $-b(y'-x')$ (as explained previously, b is related to the velocity of the shock absorber motion);
- a superposition of two upward forces act on the unsprung mass (m_1) (to oppose the other two downward forces on m_2), along with a downward force $-K_1(x-u)$ caused by the tire forcing m_1 down.

Obviously, an upward force of $K_1(x-u)$ is also applied on the tire and opposes the downward force on m_1, but it is not illustrated on the diagram.

2.4 Derivation of DE's

As stated by Newton's Second Law,

$$F_{net} = ma \tag{1}$$

where m is the mass of an object and a is its acceleration. F_{net} stands for the sum of the forces acting on the system.
Following Newton's law, we get to this first equation of forces acting on m_2:

$$m_2 y'' = -K_s(y - x) - b(y' - x') \tag{2}$$

which simplifies to:

$$m_2 y'' + K_s y + b y' = K_s x + b x' \tag{3}$$

Similarly, when combining the forces acting on m_1:

$$m_1 x'' = K_s(y - x) + b(y' - x') - K_1(x - u) \tag{4}$$

which further simplifies to:

$$m_1 x'' + (K_s + K_1)x + b x' = K_s y + b y' + K_1 u \tag{5}$$

we get that equations (3) and (5) are the two second-order differential equations representative of the quarter car suspension system studied in this article. Now lets turn this system of second-order ODEs into a system of four coupled first-order equations.
Assume that:

$$y' = v \tag{6}$$

and that:

$$x' = w \tag{7}$$

Then, by replacing the new variables in the equations (3) and (5), we have:

$$y'' = v' = \frac{K_s x}{m_2} + \frac{b x'}{m_2} - \frac{K_s y}{m_2} - \frac{b y'}{m_2} \tag{8}$$

$$v' = \frac{K_s}{m_2} x + \frac{b}{m_2} w - \frac{K_s}{m_2} y - \frac{b}{m_2} v \tag{9}$$

And,

$$x'' = w' = \frac{K_s y}{m_1} + \frac{b y'}{m_1} + \frac{K_1 u}{m_1} - \frac{(K_s + K_1)x}{m_1} - \frac{b x'}{m_1} \tag{10}$$

$$w' = \frac{K_s}{m_1} y + \frac{b}{m_1} v + \frac{K_1}{m_1} u - \frac{(K_s + K_1)}{m_1} x - \frac{b}{m_1} w \tag{11}$$

In sum, our new system of ODEs is composed of the following four first-order differential equations: eq.(6), eq.(7), eq.(9), eq.(11). This system of equations is representative of the quarter car model suspension system explained above.

3 Numerical Solver

In the context of mathematically modelling a complex mechanical system such as an automotive suspension system, the numerical approach is preferred over the analytical one, as it turns out to be very demanding and complex. A numerical approach does not necessarily allow to find the exact solution of a system of differential equations. It will rather give us an accurate and efficient approximation of the solution and this, way faster than analytically. Multiple numerical methods are available to solve more or less complex differential equations. Within those methods we find Euler's Method, Improved Euler's Method, Midpoint Method, Runge-Kutta Method and a few others. Some are more accurate than others in estimating the real solution, but briefly, the logic behind all these methods is to approximate the area under the curve of a given function.

A known fact is that computers have a hard time with continuous processes such as techniques of differentiation. The goal is to replace derivatives by simpler operations like differences which are discrete processes easily computed by computers. Consequently, in order to approximate a solution, Taylor's series technique has a disadvantage over the Runge-Kutta methods. Taylor's series require a derivative computation at every step whereas the Runge-Kutta methods require no actual differentiation of equations of higher orders. When dealing with mechanical systems, to compute the solution using Taylor's formula (12) becomes messy and complex after a few calculations. Here is the formula showing that the more accurate the approximation is, the more work needs to be done to compute the higher order derivatives:

$$\text{Taylor's series formula: } y_{t+h} = y_t + hy'_t + \frac{h^2}{2}y''_t + \frac{h^3}{3!}y'''_t + ... + \frac{h^n}{n!}y_t^{(n)} \quad (12)$$

3.1 Runge-Kutta Methods

The Runge-Kutta is a family of methods of different orders of accuracy that are designed to approximate Taylor's series by using linear combinations of values of $f(t, y_t)$ to approximate $y(t)$. With the needed information on a point (t_n, y_n), one calculates $f(t, y_t)$ at different positions without having to derive the function itself. Also, to know the value of the following point, y_{n+1}, we simply need the value of the point y_n. Therefore, Runge-Kutta are said to be one-step methods and self-starting methods. Within these methods, one

finds the well-known Euler's Method, the Improved Euler's Method and the Runge-Kutta 4 Method. Euler's Method and Improved Euler's Method are very simple and efficient to be used both numerically or analytically, mostly for less complex first order systems. Yet, these methods are first and second order methods respectively, meaning that their local truncation error (e_n) is similar to h^2 and h^3 respectively from the Taylor's series. In other words, Euler's Method agrees with Taylor's solution only on first degree term and fails to correctly approximate the area under the curve at the second degree term whereas its Improved Method agrees with Taylor's computations up to second degree terms and fails afterwards. Similarly, the Runge-Kutta 4's first error occurs in Taylor's fifth degree term, which makes this technique fourth order with a local truncation error similar to h^5. Consequently, RK4 is two orders of magnitude more accurate than the Improved Euler's Method. In summary:

$$\text{Euler's Method: } e_n \sim h^2$$

$$\text{Improved Euler's Method: } e_n \sim h^3$$

$$\text{Runge-Kutta 4 Method: } e_n \sim h^5$$

The derivation of the Runge-Kutta fourth order method is a very complex and laborious process, hardly found in any literature. Hence, putting its derivation aside (which starts with a Taylor's series expansion), the following is a brief explanation on how the Runge-Kutta methods work. In Euler's Method (the simplest one), the approximations of the actual function's slope are made through explicit time steps denoted Δt.

$$\Delta t = t_{n+1} - t_n$$

The y value is incremented with respect to time, and to get the value of y_{n+1} knowing y_n:

$$y_{n+1} = y_n + \Delta t \frac{dy}{dt}$$

After computing the value of y_{n+1}, the same process is re-used at the new time step Δt until the final time is reached. This easily programmable method is simple but also less efficient as it requires very small time steps to follow the slope. Nevertheless, it assumes that the slope remains constant as we go from step n to step $n+1$, which is clearly not the case in complex differential equations. The following two sources of error end up occurring in Euler's algorithm:

1. The approximations at each n steps are straight lines rather than curves following the actual trajectory;

2. After very few steps, the y position is no longer on the actual curve but completely inaccurate;

To correct those errors, we need to evaluate the slope between t_n and t_{n+1} with more time points than previously. Here come in play the Runge-Kutta methods of higher orders. The value of y_{n+1} is now calculated as follows:

$$y_{n+1} = y_n + \Delta y_{final}$$

and Δy_{final} is the weighted average of four different increments as explained subsequently.

3.1.1 Runge-Kutta 45

For a better understanding on the functioning of the numerical differential equations solver, here are the basic steps of the algorithm that performs the Runge-Kutta 4 Method:

Step 1: Define $f(t, y)$ (provide the function)

Step 2: Input the initial values for t and y (t_o and y_o)

Step 3: Input the step size (h) and the number of steps (n)

Step 4: Output the t_o and y_o values

Step 5: Create a *for* loop to execute *Step 6* for i from 1 to n

Step 6: $k_1 = f(t, y)$

$$k_2 = f(t + \frac{h}{2}, y + \frac{h}{2}k_1)$$

$$k_3 = f(t + \frac{h}{2}, y + \frac{h}{2}k_2)$$

$$k_4 = f(t + h, y + hk_3)$$

$$y = y + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

$$t = t + h$$

Step 7: Output t and y values

Whenever $f(t, y_t)$ does not depend on y, we have:

$$k_{n1} = f(t_n)$$
$$k_{n2} = k_{n3} = f(t_n + \frac{h}{2})$$
$$k_{n4} = f(t_n + h)$$

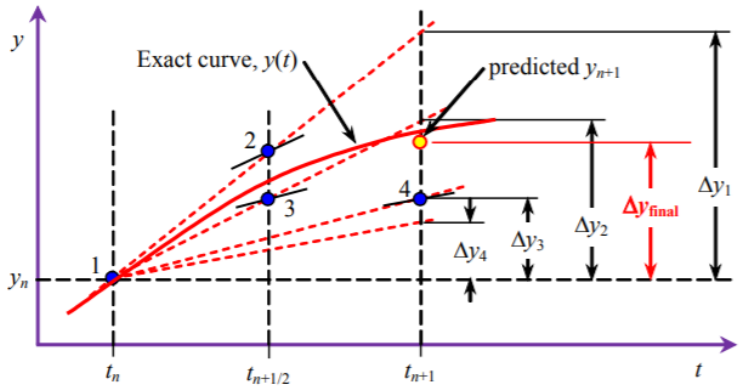
and the Runge-Kutta formula simplifies to:

$$y_{n+1} = y_n + \frac{h}{6}(f(t_n) + 4f(t_n + \frac{h}{2}) + f(t_n + h)) \tag{13}$$

From equation (13), one can see that the RK4 formula implies a weighted average of values of $f(t, y)$ at four different points, and these points are found in the interval of:

$$t_n \leq t < t_{n+1}$$

Figure 4: Representation of the Runge-Kutta 4 final slope approximation using four points of increment



Moreover, the middle increments (k_{n2} and k_{n3}), which represent the inner slopes (Δy_2 and Δy_3 in *Figure 4*), are twice as important as the outer increments (k_{n1} and k_{n4}), which represent the outer slopes of the estimate (Δy_1 and Δy_4 in *Figure 4*). The step-size (h) equals to:

$$h = t_{n+1} - t_n$$

Finally, adaptive Runge-Kutta methods are used numerically in order to get an approximation as close as possible to the actual result. One of these methods is called the Runge-Kutta 45: the method we use to solve the four suspension system modelling equations through Python's numerical solver in Jupyter Notebook. This method modifies the step-size h automatically as it computes the values at each step. It is also known as the "Automatic Error Control" and it works as follows: the step-size is modified in order to maintain the local truncation error (e_n) as close as possible (usually less than or equal) to the specified tolerance level in a certain context. Otherwise, by always keeping the step-size very small, it often happens that it is small enough for good accuracy in some parts of the function but it might be way smaller than necessary in other sections of the interval.

With this improved method, the computer needs to estimate the e_n at every step of the computation. In order to estimate a new e_n, the solver runs computations using fifth order Runge-Kutta ($e_n \sim h^6$) and fourth order Runge-Kutta ($e_n \sim h^5$) simultaneously. The difference between the results from the fourth and the fifth orders becomes the new e_n. The combination of the fourth and fifth order Runge-Kutta results in the RK45 method used by our numerical solver. Its accuracy and precision to approximate the actual solution outdo the simple RK4 or RK5 methods, without mentioning Euler or Improved Euler's methods. The disadvantage of RK45 is that at each step, the described method requires extra evaluations of $f(t, y)$ to reach the fifth order of accuracy. Even though the computations made for the fourth order can be reused in the fifth order process, extra calculations are added to the multiple computations already made by the solver. However, let's keep in mind that a more accurate algorithm is more efficient and consists in better results at the end of the process.

It is primordial to mention that Runge-Kutta family of methods can be applied to multiple differential equations simultaneously and also to any order of differential equations. Yet, the higher order equations have to be first converted into a system of coupled first-order ordinary differential equations to be solved using Runge-Kutta. This is mainly due to the fact that, as explained previously, RK methods use linear combinations of values of $f(t, y_t)$ in order to approximate Taylor's series $y(t)$. Higher order equations would require second and higher complex derivatives computations which cannot result in linear combinations. Thus, the differential equations modelling a suspension system, being second-order equations, will first be converted into a system of four coupled first-order differential equations and then be solved

using the numerical RK45 solver in Python.

4 Modelling of Shocks on The Road

In order to realistically model the behaviour of the suspension system when it is submitted to disturbances on the road, the Poisson and Gamma distributions are used. In this context, the Poisson distribution models the number of shocks like potholes or cracks a suspension system could encounter on the road. Indeed, we assume that shocks occur independently and not simultaneously, thus meeting the conditions to qualify as Poisson events. In fact, a specific length of road has a constant expected number of shocks per distance or unit of time which will be designated as the rate parameter λ, but the shocks will occur at random. The Python code is set to generate 20 different intervals of time, therefore 20 random Poisson events, which represent jolts on the road. Each event is dispersed from one another with an average time of 10 seconds. The average time between events is represented by β, the reciprocal of λ. The intervals of time between the Poisson events, are, in themselves, exponentially distributed. However, since we are interested in the n^{th} shock on the suspension system rather than the first one when sampling, the generalization of the Exponential distribution, the Gamma distribution, will be used to model the time between the disturbances. The shape-determining parameter λ of the Gamma distribution will be set to $\frac{1}{10}$.

4.1 Poisson Process

A Poisson Process is used to model a sequence of discrete events where the expected number of events per unit of time is known, but the eaxct times at which the events occur are random. Furthermore, the occurrence of an event does not depend of the event before as the waiting time between events is exponentially distributed and,so, is memoryless (Koehrsen, 2019).

The probability mass function of a Poisson distributed discrete random variable, X, is given by

$$P(X = x) = \frac{e^{-\lambda t}(\lambda t)^x}{x!}, \text{for x=0,1,2,...} \tag{14}$$

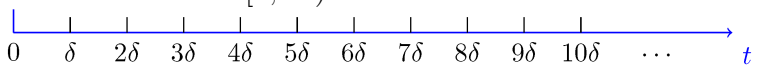
where λ is the rate parameter which represents the average number of events per unit of time.

4.2 Poisson Process modeled as a Bernoulli Process

A Bernoulli process is used to model a sequence of random experiments, each of which has solely two outcomes: success or failure. The success is designated as 1 and is associated with a probability p, whereas the failure is designated as 0 and will have a probability of 1-p (Alto, 2019). Now, let's apply the Bernoulli process in order to demonstrate the Poisson process.

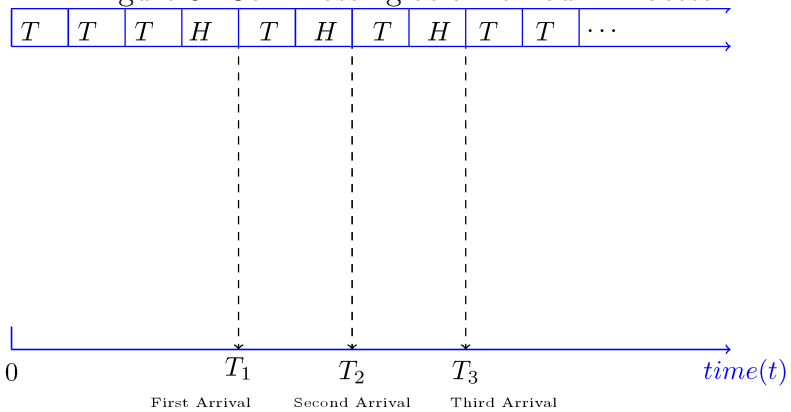
Suppose that we want to model random events occurring at a rate λ per unit of time. First, we fractionate the half-line $[0, \infty)$ into multiple intervals of length δ as exhibited in Figure 5.

Figure 5: The half-line $[0, \infty)$ fractionated into intervals of length δ



From the figure, we get that, essentially, the k^{th} interval is given by [(k-1)δ, kδ]. We suppose that, in each interval of time, we toss a coin for which the probability of getting a head is P(H)=p= $\lambda\delta$. If the coin shows head when it lands, we say that an arrival (success) occurred, and vice-versa. This process is depicted in Figure 6.

Figure 6: Coin Tossing as a Bernoulli Process



We define the number of successes, or arrivals, in a given interval [0, t] as N(t). At t=0, no event occurred yet, thus N(0) = 0. The number of intervals of time in [0,t], n, is given by $\frac{t}{\delta}$. This means that, in this case, N(t) is the

number of heads that we get in n coin flips, or trials. Thus, we conclude that N(t) $\sim$ Binomial(n,p). Since P(H)= p= $\lambda\delta$, we get

$$np = n\lambda\delta$$
$$np = \frac{t}{\delta} \cdot \lambda\delta$$
$$np = t\lambda$$

As $\delta \to 0$, n$\to \infty$ and p$\to 0$. As know, a binomial distributed random variable with $n > 50$ and $p < 0.1$ can be approximated with a Poisson distribution as follows:

$$\text{X} \sim \text{Binomial (n,p)} \Rightarrow \text{X} \sim \text{Poisson (n}\cdot\text{p)}$$

We conclude that as $\delta \to 0$, the PMF of N(t) converges to a Poisson distribution with expected number of events λt.

From the above demonstration, we can deduce that the distribution of the number of occurrences in a Poisson process depends solely on the length of the interval, thus, it is composed of stationary increments (Pishro-Nik, 2016).

4.3 Definition of the Poisson Process

Let $\lambda > 0$ be a constant. The counting process $\{N(t), t \in [0, \infty]\}$ is said to be a Poisson process with rate λ if the following conditions are met:

1. N(0) = 0;
2. The number of occurrences N(t) in an interval of time $t > 0$ is Poisson distributed with rate λ;
3. All the events in N(t) are independent and stationary increments.

4.4 Expected Value and Variance of a Poisson Distributed Discrete Random Variable

For a discrete random variable, the expected value is the sum of the product between the value of the discrete random variable and its analogous probability.

To find the expected value of a Poisson (λ) distributed random variable X, we calculate:

$$E(x) = \sum_{x=0}^{\infty} x \frac{e^{-\lambda t}(\lambda t)^x}{x!} \tag{15}$$

Since the $x = 0$ term of the sum is equal to 0, we find:

$$\sum_{x=1}^{\infty} x \frac{e^{-\lambda t}(\lambda t)^x}{x!} \tag{16}$$

We divide both numerator and denominator by x:

$$\sum_{x=1}^{\infty} \frac{e^{-\lambda t}(\lambda t)^x}{(x-1)!} \tag{17}$$

We factor out $e^{-\lambda t}$ and λt:

$$\lambda t \cdot e^{-\lambda t} \sum_{x=1}^{\infty} \frac{(\lambda t)^{x-1}}{(x-1)!} \tag{18}$$

Which is equivalent to:

$$\lambda t \cdot e^{-\lambda t} (\frac{(\lambda t)^0}{0!} + \frac{(\lambda t)^1}{1!} + \frac{(\lambda t)^2}{2!} + ...) \tag{19}$$

In its turn, equivalent to:

$$\lambda t \cdot e^{-\lambda t} \sum_{x=0}^{\infty} \frac{(\lambda t)^x}{x!} \tag{20}$$

Which simplifies to:

$$\lambda t \cdot e^{-\lambda t} e^{\lambda t} \tag{21}$$

Which equals to:

$$\lambda t \tag{22}$$

Thus, we find that E(X)= λt. As for the variance of a discrete random variable, it is designated as follows:

$$\mathrm{V(X)} = E(X^2) - [E(X)]^2$$
$$\mathrm{V(X)} = E[(X)(X-1) + X] - [E(X)]^2$$

$$\mathrm{V(X)}= E[(X)(X-1)] + E(X) - [E(X)]^2$$

So, we need to calculate:

$$E[(X)(X-1)] + E(X) - [E(X)]^2 = \sum_{x=0}^{\infty} x(x-1)\frac{e^{-\lambda t}(\lambda t)^x}{x!} + \lambda t - (\lambda t)^2 \quad (23)$$

Since the $x = 0$ and $x = 1$ terms of the sum are equal to 0:

$$\sum_{x=2}^{\infty} x(x-1)\frac{e^{-\lambda t}(\lambda t)^x}{x!} + \lambda t - (\lambda t)^2 \quad (24)$$

We divide both numerator and denominator by x and $(x-1)$:

$$\sum_{x=2}^{\infty} \frac{e^{-\lambda t}(\lambda t)^x}{(x-2)!} + \lambda t - (\lambda t)^2 \quad (25)$$

We factor out $(\lambda t)^2$ and $e^{-\lambda t}$:

$$(\lambda t)^2 e^{-\lambda t} \sum_{x=2}^{\infty} \frac{(\lambda t)^{x-2}}{(x-2)!} + \lambda t - (\lambda t)^2 \quad (26)$$

Which is equivalent to:

$$(\lambda t)^2 e^{-\lambda t}(\frac{(\lambda t)^0}{0!} + \frac{(\lambda t)^1}{1!} + \frac{(\lambda t)^2}{2!} + ...) + \lambda t - (\lambda t)^2 \quad (27)$$

In its turn, equivalent to:

$$(\lambda t)^2 e^{-\lambda t} \sum_{x=0}^{\infty} \frac{(\lambda t)^x}{x!} + \lambda t - (\lambda t)^2 \quad (28)$$

Which simplifies to:

$$(\lambda t)^2 e^{-\lambda t} e^{\lambda t} + \lambda t - (\lambda t)^2 \quad (29)$$

$$(\lambda t)^2 + \lambda t - (\lambda t)^2 \quad (30)$$

$$\lambda t \quad (31)$$

In essence, we found that $\mathrm{V(X)} = \lambda t$.

Therefore, a Poisson distributed discrete random variable, X, has the following properties:

1. $\mathrm{E(X)} = \lambda t$
2. $\mathrm{V(X)} = \lambda t$

4.5 Gamma Function

The Gamma function, $\Gamma(x)$, broadens the factorial function to complex and real numbers, rather than only natural numbers. It states that, for any r > 0,

$$\Gamma(r) = \int_0^\infty x^{r-1} e^{-x} dx$$

The Gamma function has the following properties:

$$1.\ \Gamma(r) = (r-1)!$$

$$2.\ \Gamma(r) = (r-1)\Gamma(r-1)$$

$$3.\ \Gamma(\frac{1}{2}) = \sqrt{\pi}$$

4.6 Gamma Distribution

The Gamma distribution is a distribution in which the waiting time between events is pertinent. It models the waiting time for the r^{th} Poisson event rather than the waiting time for the first event. For this reason, it can be designated as the generalization of the Exponential distribution.

The probability density function of a Gamma distributed continuous random variable, X, with parameters $\alpha > 0$ and $\lambda > 0$, is given by

$$f(x) = \begin{cases} \frac{1}{\Gamma(\alpha)} \cdot \lambda^\alpha \cdot x^{\alpha-1} \cdot e^{-\lambda x}, & \text{if } x > 0 \\ 0, & \text{otherwise} \end{cases} \tag{32}$$

where λ is the rate parameter which represents the average number of events per unit of time.

4.7 Expected Value and Variance of a Gamma Distributed Continuous Random Variable

From the definition of the expected value of a continuous random variable, we get that the expected value of X is:

$$E(X) = \int_{-\infty}^{\infty} x \cdot f(x) dx$$

So, we solve:

$$\frac{\lambda^{\alpha}}{\Gamma(\alpha)} \int_0^{\infty} x^{\alpha} e^{-\lambda x} dx \tag{33}$$

We substitute λx by t:

$$\frac{\lambda^{\alpha}}{\Gamma(\alpha)} \int_0^{\infty} (\frac{t}{\lambda})^{\alpha} e^{-t} \frac{dt}{\lambda} \tag{34}$$

We factor out $\lambda^{-\alpha}$ and λ^{-1}:

$$\frac{\lambda^{\alpha}}{\lambda^{\alpha+1}\Gamma(\alpha)} \int_0^{\infty} t^{\alpha} e^{-t} dt \tag{35}$$

Using the definition of a Gamma function which stipulates that, for any r>0, then:

$$\Gamma(r) = \int_0^{\infty} x^{r-1} e^{-x} dx$$

We solve the integral in eq. 35:

$$\frac{\lambda^{\alpha}\Gamma(\alpha+1)}{\lambda^{\alpha+1}\Gamma(\alpha)} \tag{36}$$

Which simplifies to:

$$\frac{\alpha\Gamma(\alpha)}{\lambda\Gamma(\alpha)} \tag{37}$$

And equals to:

$$\frac{\alpha}{\lambda} \tag{38}$$

From the definition of the expected value of a continuous random variable, we get that the variance of X is:

$$\int_{-\infty}^{\infty} x^2 f(x) dx - [E(X)]^2 \tag{39}$$

$$\frac{\lambda^{\alpha}}{\Gamma(\alpha)} \int_0^{\infty} x^{\alpha+1} e^{-\lambda x} dx - (\frac{\alpha}{\lambda})^2 \tag{40}$$

We substitute λx by t :

$$\frac{\lambda^{\alpha}}{\Gamma(\alpha)} \int_0^{\infty} (\frac{t}{\lambda})^{\alpha+1} e^{-t} \frac{dt}{\lambda} - (\frac{\alpha}{\lambda})^2 \tag{41}$$

We factor out $\lambda^{-(\alpha+1)}$ and λ^{-1}:

$$\frac{\lambda^{\alpha}}{\lambda^{\alpha+2}\Gamma(\alpha)} \int_0^{\infty} t^{\alpha+1} e^{-t} dt - \frac{\alpha^2}{\lambda^2} \tag{42}$$

Using the definition of a Gamma function which stipulates that, for any r>0, then:

$$\Gamma(r) = \int_0^{\infty} x^{r-1} e^{-x} dx$$

We solve the integral:

$$\frac{\lambda^{\alpha}\Gamma(\alpha+2)}{\lambda^{\alpha+2}\Gamma(\alpha)} - \frac{\alpha^2}{\lambda^2} \tag{43}$$

Using the second property of the Gamma function which stipulates that, for any r>0, then:

$$\Gamma(r) = (r-1)\Gamma(r-1)$$

We get:

$$\frac{(\alpha+1)\alpha\Gamma(\alpha)}{\lambda^2\Gamma(\alpha)} - \frac{\alpha^2}{\lambda^2} \tag{44}$$

After putting everything on the common denominator: $\lambda^2\Gamma(\alpha)$:

$$\frac{\alpha(\alpha+1)\Gamma(\alpha) - \alpha^2\Gamma(\alpha)}{\lambda^2\Gamma(\alpha)} \tag{45}$$

We factor out $\Gamma(\alpha)$ from the numerator:

$$\frac{\Gamma(\alpha) \cdot (\alpha(\alpha+1) - \alpha^2)}{\lambda^2\Gamma(\alpha)} \tag{46}$$

Which simplifies to:

$$\frac{\alpha}{\lambda^2} \tag{47}$$

Therefore, a Gamma distributed discrete random variable, X, has the following properties:

1. $\mathrm{E(X)} = \dfrac{\alpha}{\lambda}$
2. $\mathrm{V(X)} = \dfrac{\alpha}{\lambda^2}$

5 Graphs and Analysis of Results

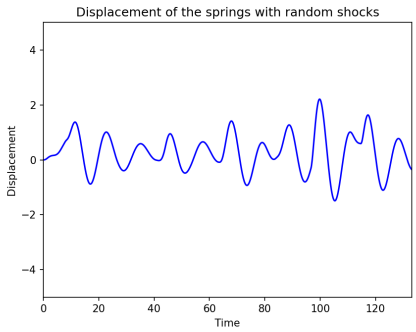


Figure 7: Displacement 1

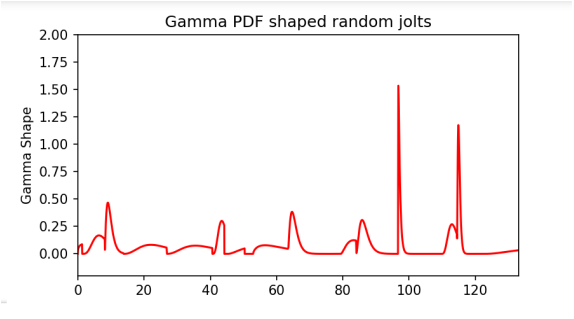


Figure 8: Gamma 1

```
L1 = 3 #Length of the suspension spring at equilibrium (m)
Dt = 3 #Diameter of the tire acting as a second spring (at equilibrium) (m)
m2 = 2 #Mass of quarter car = unsprung mass (kg)
m1 = 1 #Mass of wheel + suspension components = sprung mass (kg)
b = 0.75 #Damping coefficient (Ns/m)
Ks = 1 #Spring constant (N/m)
K1 = 2 #Tire 'spring' constant (N/m)
n = 100 #Number of evaluation points between jolts
```

Figure 9: Initial Conditions 1

In this first run, initial conditions for the parameters are shown in *Figure 9*. The spring displacement graph (*Figure 7*) looks like an average behavior with a greater shock at around 100 seconds and another one at around 120 seconds, just after the system started to stabilize back to its normal oscillations. The intensity of the shock is also illustrated in *Figure 8* by the sharpness of the Gamma PDF shape which elevates to over 1.5. Considering these initial parameters, it represents the behavior of a more or less proportional vehicle as we know that the length of the suspension spring is similar to the diameter of the tire which acts as a spring; the mass of the quarter body car is greater than the mass of the wheel, the brakes and the suspension components by

around 10 times; the 'spring constant' of the tire is also much greater than the spring constant of the shock absorber itself.

All these values are far from being actual numbers used in the automotive industry, but for the purpose of animating the behavior of such a system, it is preferable to use convenient values for the numerical solver's computations. In this particular scenario, we can associate the system with a critically damped case because the spring returns back to its initial equilibrium position in a small amount of time before a new shock is absorbed. The frequency of jolts is high enough to make the suspension oscillate almost continuously. However, around 70 seconds, we can see that the system had a shock and then oscillated once before considerably decreasing its oscillations (almost back to normal), then took a new shock.

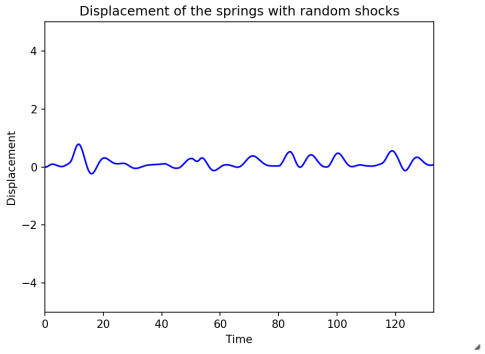


Figure 10: Displacement 2

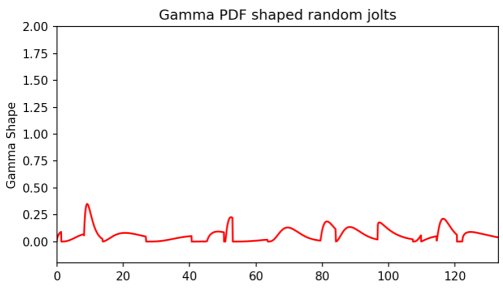


Figure 11: Gamma 2

```
L1 = 3 #length of the suspension spring at equilibrium (m)
Dt = 3 #Diameter of the tire acting as a second spring (at equilibrium) (m)
m2 = 4 #Mass of quarter car = unsprung mass (kg)
m1 = 1 #Mass of wheel + suspension components = sprung mass (kg)
b = 5 #Damping coefficient (Ns/m)
Ks = 3 #Spring constant (N/m)
K1 = 5 #Tire 'spring' constant (N/m)
n = 100 #Number of evaluation points between jolts
```

Figure 12: Initial Conditions 2

For our second run, we significantly increased both spring constants, the

damping coefficient and the mass of the quarter car, which again, represents the sprung mass (m_2), so a quarter of the car's weight resting on the springs. This mass does not take into consideration the mass of the wheels and all the other components that are part of the unsprung mass in our example. Now, m_2 is four times m_1 as it should approximately be in a realistic example. In this scenario, we can compare the behavior to an over-damped case where the mass returns to equilibrium without oscillating almost at all as it can be seen in *Figure 10*. Between 20 and 40 seconds, the displacement graph is almost a flat line, meaning that the mass barely moves. The Gamma shapes are also very small (*Figure 11*), going slightly above 0.25 in the first jolt only.

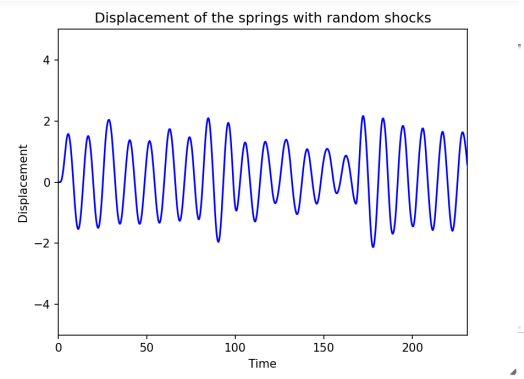


Figure 13: Displacement 3

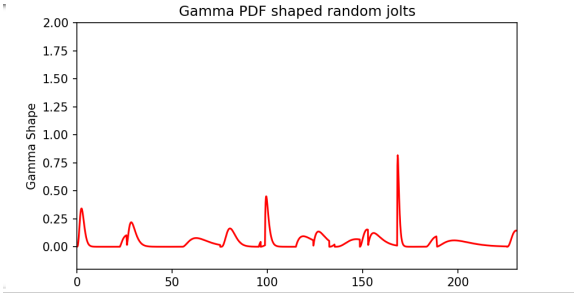


Figure 14: Gamma 3

```
L1 = 3 #Length of the suspension spring at equilibrium (m)
Dt = 3 #Diameter of the tire acting as a second spring (at equilibrium) (m)
m2 = 1 #Mass of quarter car = unsprung mass (kg)
m1 = 0.5 #Mass of wheel + suspension components = sprung mass (kg)
b = 0.01 #Damping coefficient (Ns/m)
Ks = 0.5 #Spring constant (N/m)
K1 = 1 #Tire 'spring' constant (N/m)
n = 100 #Number of evaluation points between jolts
```

Figure 15: Initial Conditions 3

In our final run, the system is similar to an under-damped system in which the oscillations continue until enough energy is dissipated to completely

stop the movement of the mass and bring it back to zero. As it is depicted by *Figure 13*, the mass oscillates much more than in the previous two runs and this, due to a decrease in the weight parameters, the damping coefficient, and both spring constants. Lowering the values of all these parameters make the system much more likely to oscillate for a longer period of time. A good illustration of this occurs following the big jolt at around 170 seconds (*Figure 14*) when the mass has a large displacement and then continues to oscillate for a while with slowly decreasing displacement.

```
print(jt) #Shows the random intervals of time
```

```
[ 1.27616016  6.85036093  5.59397971 13.0723894  13.80262025  3.61796984
  1.04241302  5.10033779  0.43042213  2.13569939 10.49545573 16.01452951
  4.55411036 12.6602422  10.67079831  2.43010586  4.69572147  6.12512564
  1.64516823 10.8521846 ]
```

Figure 16: Random Times 1

```
print(jt) #Shows the random intervals of time
```

```
[19.55351697  3.04791427  3.42948814 29.4124945  19.7758518   0.27474924
 20.06671801  0.91547638  2.26056343 16.2458572   9.07653397  8.38688136
  2.78143645 13.26863549  4.25707591 15.33868815 15.07758041  5.53279255
 36.98805262  5.40717553]
```

Figure 17: Random Times 2

```
print(jt) #Shows the random intervals of time
```

```
[ 5.91584369  6.03967179 28.48969235  6.69128354  1.2442366  17.61274604
 14.79201092  6.64077513  5.04713093 13.74458955  1.54878086  3.42280055
 19.05521918  1.74668921  7.10134987 11.40351028  8.31750388  3.23961906
  6.43919909  2.25878928]
```

Figure 18: Random Times 3

Figures 16, 17, 18 are simply snapshots from Jupyter Notebook to show how the intervals of time are programmed to be randomised at every single new code run. These random intervals are the times between each Poisson event (between each jolt) and the parameter is set to 20 different intervals, and therefore around 20 jolts, but it can obviously be changed, just like the value of λ, which is the reciprocal of the average time between events in the code.

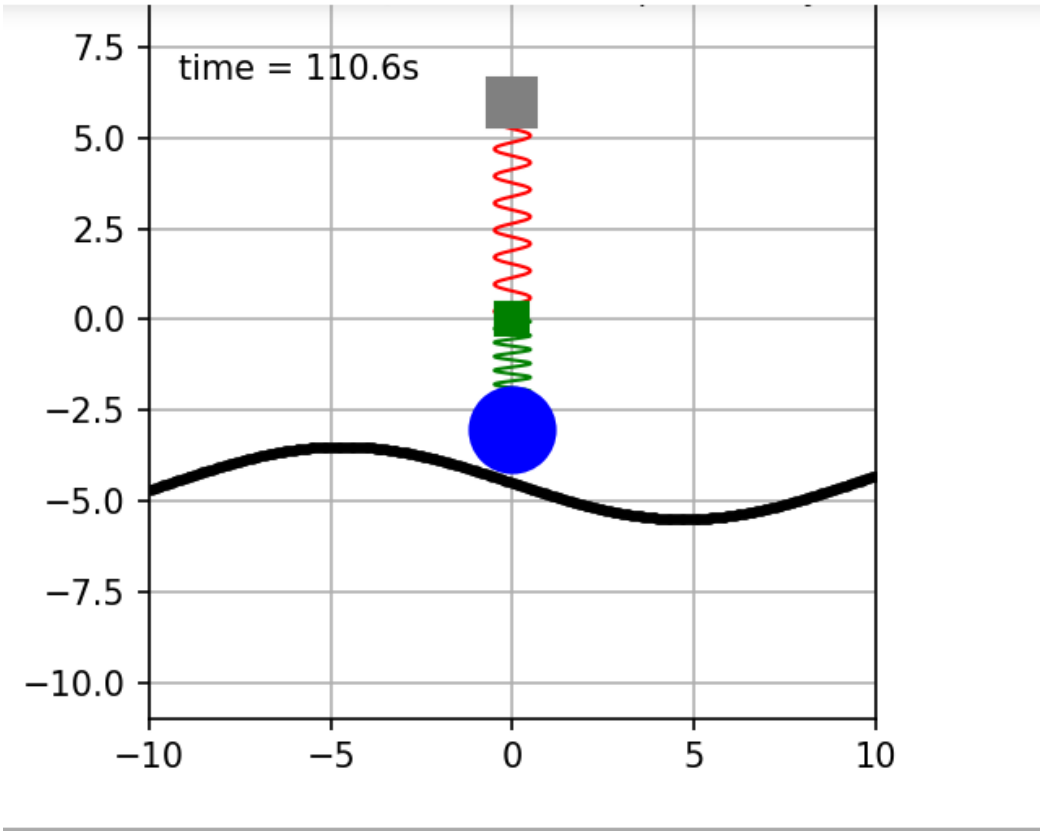


Figure 19: Quarter Car Suspension System Animation

Lastly, *Figure 19* is a snapshot of the numerical animation made using matplotlib tools in Python. Due to a lack of coding skills, the animation is more or less representative of a real suspension system. Yet, it is closely related to the quarter car suspension diagram analysed previously.

- The grey square represents the sprung mass (m_2), or the chassis of a quarter car;
- The red spring is the suspension spring (K_s and b) linking the chassis to the components of the wheel;
- The green square illustrates the unsprung mass (m_1), so the weight of all the components related to the system except for the chassis' weight;
- The green spring located right on top of the blue circle (wheel/tire) simply represents the spring coefficient of the tire (K_1) and illustrates how the tire deforms during the shocks;
- The blue circle is an illustration of the wheel/tire of the car;
- The black curvy line shows the road on which the wheel rolls. It is not assumed to be a flat road as explained previously in this article. Moreover, the wheel goes under the road line on purpose, only when there are big enough jolt, to illustrates the 'potholes' on the road.

Click ***here*** to view the animated version of the suspension system

6 Conclusion

To model a vehicle suspension system realistically, two second-order differential equations from Newton's Second Law were successfully derived. These equations were based on a quarter car sample as we assumed that the suspension system was independent, which allowed us to model the wheel and its suspension components separately from the rest of the vehicle. Then, in order to solve the system of equations numerically, we had to convert it into a system of four coupled first-order ordinary differential equations. Runge-Kutta 45 numerical solver was used for greater accuracy and approximation of the actual solution. The differential equations were needed to plot the displacement vs. time graphs, and knowledge related to Poisson and Gamma distributions was needed to plot the Gamma Probability Density Function graphs. Solving the derived equations also allowed us to animate the quarter car model in Python.

After multiple runs and modifications, we can conclude that depending on the spring constants and on the damping coefficient, the system will

behave as an under-damped, over-damped or critically damped system. In the case where the damping coefficient is high enough, the mass might go to its equilibrium position as fast as possible, without oscillating a single time (over-damped). If the coefficient is less exaggerated, the mass might take longer to get back to normal but still will, without further oscillations. And, if damping is not present at all or is very low, the mass could oscillate infinitely without stopping or stopping once all the energy of the system is dissipated. Depending on the type of vehicle (weight, size, shape), its use, the road condition, and multiple other factors, automotive manufacturers will choose the convenient type and design of suspension system. However, most vehicles use a system composed of a shock absorber (piston moving through viscous oil) and a coil (spring-shaped flexible metal) around it to absorb shocks and damp oscillations.

Taking into consideration that this model is not the most representative of a realistic system, it can always be improved and shaped closer to a more realistic one. Such a system involves many more mechanical components and absorbs shocks from more than just the road. Let's not forget that all operating systems in a vehicle are somehow related and have an impact on each other, but this is subject to more advanced research. In further researches, plots of velocity vs. time in addition to displacement vs. time could be interesting in order to analyze in more depth the behavior of the system. Also, in this article, the tire was considered as a second spring with its own spring coefficient, but its damping effect was neglected. Hence, taking a second damping coefficient into consideration would complicate the differential equations even more. Finally, another interesting approach to examine such a dynamic system is to test the impact on the suspension from a bump at a certain speed and compare it to the impact from a pothole at that same speed.

References

[1] V. Alto, "Understanding bernoulli and binomial distributions", *Towards Data Science*, 2019.

[2] W. E. Boyce, R. C. DiPrima, and D. B. Meade, "Elementary differential equations", in. John Wiley & Sons, 2017, ch. 8.3.

[3] J. M. Cimbala, "Runge-kutta method for solving ordinary differential equations", 2016.

[4] D. Collins. (2018). Complete guide to car suspension, [Online]. Available: `https://www.carbibles.com/guide-to-car-suspension/`. (accessed: 22.05.2020).

[5] J.-W. Hu and H.-M. Tang, "Numerical methods for differential equations", *City University, Hong Kong*, 1999.

[6] T. Iyenger, K. B. Gandhi, S. Ranganatham, and M. Prasad, "Engineering mathematics volume - ii (numerical methods and complex variables)", in. S. Chand Publishing, 2017, pp. 211–264.

[7] B. Kanber. (2012). Physics in javascript: Car suspension - part 1 (spring, mass, damper), [Online]. Available: `https://burakkanber.com/blog/physics-in-javascript-car-suspension-part-1-spring-mass-damper/`. (accessed: 22.05.2020).

[8] W. Koehrsen, "The poisson distribution and poisson process explained", *Towards Data Science*, 2019.

[9] A. Materdey, "Higher-order numerical solutions of the quarter car suspension model", 2018. DOI: `10.11159/icmie18.135`.

[10] A. Nicholson and Y.-D. Wong, "Are accidents poisson distributed? a statistical test", *Accident Analysis & Prevention*, vol. 25, no. 1, pp. 91–97, 1993.

[11] J. Orloff and J. Bloom, "Expectation, variance and standard deviation for continuous random variables class 6, 18.05, jeremy orloff and jonathan bloom", 2014.

[12] H. Pishro-Nik, "Introduction to probability, statistics, and random processes", in. 2016, ch. 4.2.4 and 11.1.2.

[13] T. O. Terefe and H. G. Lemu, "Solution approaches to differential equations of mechanical system dynamics: A case study of car suspension system", *Advances in Science and Technology Research Journal*, vol. 12, 2018.

[14] W. Wu. (2018). Modelling car suspension with ode's: Damped free oscillations part 1, [Online]. Available: `https://steemit.com/steemstem/@masterwu/modelling-car-suspension-with-second-order-odes-damped-free-oscillations`. (accessed: 22.05.2020).