New Application of The Kermack-McKendrick model

Abdalftah Elbori

Department of Mathematics Azzaytuna University, Tarhuna-Libya
e-mail: abdalftah81@yahoo.com

Abstract. This study analyses the mathematical structure of the Kermack-McKendrick model and its stability. This paper also investigates how to optimize the Kermack-McKendrick systems in order to generate rhythmic patterns for one leg with two degrees of freedom by using a hybrid function. It also discusses how to optimize the different types of the Kermack-McKendrick to generate rhythmic patterns similar to the rhythmic patterns derived from real data without any sensory feedback.
Keywords: *The Kermack-McKendrick Model, Modelling of one leg, Numerical Solutions, Hybrid function.*

1 Introduction

The Kermack-McKendrick model is a mathematical model used to describe the spread of infectious diseases in a population. It was developed in the 1920s by William Ogilvy Kermack and Anderson Gray McKendrick [1] and [2], and is based on the idea that the transmission of a disease is dependent on the number of susceptible individuals in a population, as well as the number of infected individuals. The model can be used to predict the trajectory of an outbreak, and has been applied to the spread of diseases such as influenza, HIV, and Ebola. It is a useful tool for public health officials and policymakers in understanding the potential impact of an infectious disease and implementing effective control measures.
Recently, many study this model is based on a set of differential equations that describe the time evolution of the number of individuals in a population who are susceptible to an infectious disease, the number of individuals who are infected with the disease, and the number of individuals who have recovered from the disease or have died from it [3] and [4]. The model is commonly referred to as the SIR model, where "S" represents the number of susceptible individuals, "I" represents the number of infected individuals, and "R" represents the number of recovered or deceased individuals. The model is used to understand the dynamics of infectious disease outbreaks and to inform public health policy. This study focuses on optimizing this model as central patterns generators to generate rhythmic patterns for one leg's model for more details see[5], [6] and [7]. The optimization of the Kermack-McKendrick model can be a novelty in this paper. Based on the cost function, this paper also uses a new algorithm to find the optimum parametric values for this model. The paper is organized as follows: The kinematic model has been discussed in the next section. A numerical solution and strategy to the Kermack-McKendrick systems are given in Section 3. In Section 4, the real data is discussed. Section 5 is devoted to the optimization results. In Section 6, some conclusions are drawn and suggestions for future research are given.

2 The Model of the One leg

The kinematic model of one leg with two degrees of freedom (DOFs) is designed to conducted a base analysis. To model one leg as in the Figure 1, which this figure presents the leg structure in two cases of motion: a swing mode and a stance mode. Where, L_1 and L_2 are the lengths from the hip joint to the knee joint, and from the knee joint to the end effector, respectively; and θ_1 and θ_2 are the angular position of the hip and is the angular position of the knee respectively, y_g is the distance between the lower body and the ground. The coordinates of the lowest part of the hip and knee are denoted by

(x_A, y_A) and (x_f, y_f), respectively.

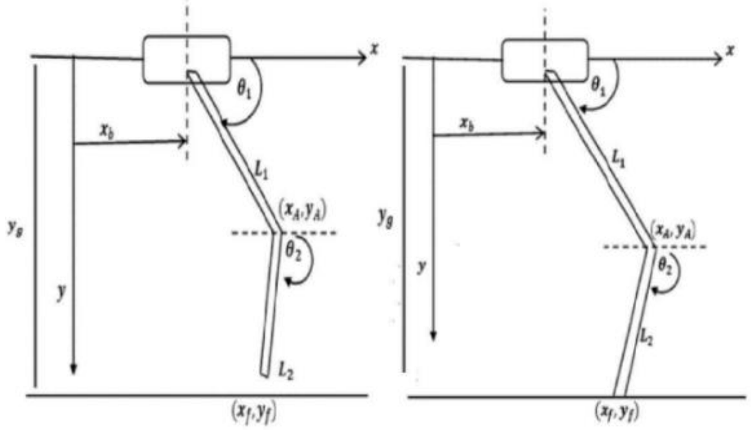

Figure 1: Swing and stance modes of the leg

The simple kinematic equations are

$$x_A = x_b + L_1 \cos\theta_1\,, \qquad \& \quad y_A = L_1 \sin\theta_1,$$

and

$$x_f = x_A + L_2 \cos\theta_2\,, \qquad y_f = y_A + L_2 \sin\theta_2$$

There are two cases to consider during motion. The first case is when $y_f = y_g$, that is, the leg touches the ground. This case is known as the stance mode in which case the leg behaves as a revolute joint. In stance mode, the hip joint angle θ_1 is computed in terms of the knee angle θ_2, which is established by the Kermack-McKendrick model. Thus, in this mode, the kinematic model has one degree of freedom (DOF). Moreover, only in stance mode will the body move. The second case is when $y_f < y_g$, which is the time when the leg does not touch the ground. This mode is known as the swing mode.

3 The Kermack-McKendrick model

The Kermack-McKendrick model for the course an epidemic in a population is given by the system of ODEs

$$\left.\begin{aligned} y_1' &= -cy_1y_2 \\ y_2' &= cy_1y_2 - dy_2 \\ y_3' &= dy_2 \end{aligned}\right\} \qquad (3.1)$$

Where y_1 represents susceptible, y_2 represents infective in circulation, and y_3 represents infective removed by isolation, death, or recovery and immunity. The parameters c and d represent the infection rate and removal rate, respectively, these parameters will determine by using hybrid function later to generate motion to angular positions θ_1 and θ_2 in the model. To solve this system, by two ways.

The first way the explicit Euler: With $h = 0.01$ to solve this system numerically, with the parameter values $c = 1$ and $d = 5$, and initial values $y_1(0) = 95$, $y_2(0) = 5$, $y_3(0) = 0$. Integrate from $t = 0$ to $t = 1$. Plot each solution component on the same graph as a function of t it is shown in the figure 2. As expect with an epidemic, it should see the number of infective grow at first, then diminish to zero. Experiment with other values for the parameters and initial conditions. It easily finds values for which the epidemic does not grow, or for which the entire population is wiped out, it is going to discuss in following steps.

The explicit Euler's method with $h = 0.01$ to solve this system numerically, with the parameter values $c = 1$ and $d = 5$, and initial values $y_1(0) = 95$, $y_2(0) = 5$, $y_3(0) = 0$. Integrate from $t = 0$ to $t = 1$.
Firstly, the systems (3.1) can be written as

$$\left.\begin{aligned} y_1' &= f_1(y_1, y_2, y_3) = -cy_1y_2 \\ y_2' &= f_2(y_1, y_2, y_3) = cy_1y_2 - dy_2 \\ y_3' &= f_3(y_1, y_2, y_3) = dy_2 \end{aligned}\right\} \tag{3.2}$$

Secondly, we apply the explicit Euler's to this system

$$\left.\begin{aligned} y_{1(n+1)} &= y_{1(n)} + hf_1\big(t_n, y_{1(n)}, y_{2(n)}, y_{3(n)}\big) \\ y_{2(n+1)} &= y_{2(n)} + hf_2\big(t_n, y_{1(n)}, y_{2(n)}, y_{3(n)}\big) \\ y_{3(n+1)} &= y_{3(n)} + hf_3\big(t_n, y_{1(n)}, y_{2(n)}, y_{3(n)}\big) \end{aligned}\right\}, \qquad n = 0,1,2,\ldots \tag{3.3}$$

We have,

$$\left.\begin{aligned} y_{1(n+1)} &= y_{1(n)} - h * c * y_{1(n)} * y_{2(n)} \\ y_{2(n+1)} &= y_{2(n)} + h(c * y_{1(n)} * y_{2(n)} - d * y_{2(n)}) \\ y_{3(n+1)} &= y_{3(n)} + h * d * y_{2(n)} \end{aligned}\right\}, \qquad n = 0,1,2,\ldots \tag{3.4}$$

Start with $y_{1(0)} = 95$, $y_{2(0)} = 5$, $y_{3(0)} = 0$ and $N = 100$ the parameter values $c = 1$ and $d = 5$.
Thirdly, we need to write Computer Programming by MATLAB as in **appendix A**, the numerical solution is shown as in the figure 2

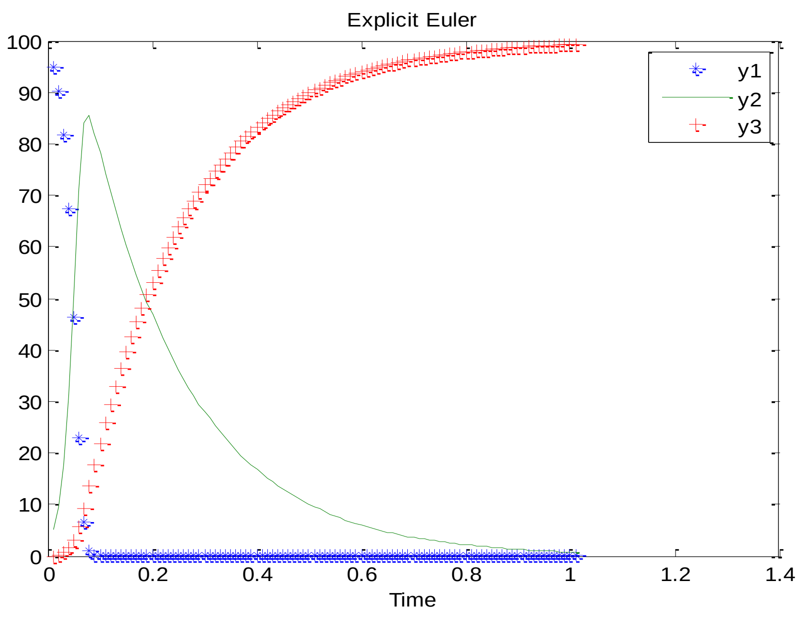

Figure 2: The numerical solution by using Explicit Euler method when $y_{1(0)} = 95$, $y_{2(0)} = 5$, $y_{3(0)} = 0$ and $N = 100$

It notices that the number of infective grow at first, then diminish to zero because the infection rate was 1 and removal rate was 5. When the parameters are changed rates then, then, the number of infective grow at first, then diminish to zero because but if value of the parameters of infection rate greater than the value of removal rate then the population is wiped out, whereas if this value of

infection rate c less or equal the value of removal rate d then, epidemic will not grow forever. It is a clear to infer that as bigger as the value of removal rate the epidemic will never grow.
The second way, the Implicit midpoint rule with $h = 0.01$ to solve this system numerically, with the parameter values $c = 1$ and $d = 5$, and initial values $y_1(0) = 95$, $y_2(0) = 5$, $y_3(0) = 0$. Integrate from $t = 0$ to $t = 1$. Firstly; it is needed to apply the explicit Euler's to this system

$$\left.\begin{aligned} y_{1(n+1)} &= y_{1(n)} + hf_1\left(\frac{1}{2}(t_n + t_{n+1}), \frac{1}{2}\left(y_{1(n)} + y_{1(n+1)}\right), \frac{1}{2}(y_{2(n)} + y_{2(n+1)}), \frac{1}{2}(y_{3(n)} + y_{3(n+1)})\right) \\ y_{2(n+1)} &= y_{2(n)} + hf_2\left(\frac{1}{2}(t_n + t_{n+1}), \frac{1}{2}\left(y_{1(n)} + y_{1(n+1)}\right), \frac{1}{2}(y_{2(n)} + y_{2(n+1)}), \frac{1}{2}(y_{3(n)} + y_{3(n+1)})\right) \\ y_{3(n+1)} &= y_{3(n)} + hf_3\left(\frac{1}{2}(t_n + t_{n+1}), \frac{1}{2}\left(y_{1(n)} + y_{1(n+1)}\right), \frac{1}{2}(y_{2(n)} + y_{2(n+1)}), \frac{1}{2}(y_{3(n)} + y_{3(n+1)})\right) \end{aligned}\right\}$$

$$n = 0,1,2, \ldots$$

Then,

$$\left.\begin{aligned} y_{1(n+1)} &= y_{1(n)} - h * c * \frac{1}{2}\left(y_{1(n)} + y_{1(n+1)}\right) * \frac{1}{2}(y_{2(n)} + y_{2(n+1)}) \\ y_{2(n+1)} &= y_{2(n)} + h(c * \frac{1}{2}\left(y_{1(n)} + y_{1(n+1)}\right) * \frac{1}{2}(y_{2(n)} + y_{2(n+1)}) - d * \frac{1}{2}(y_{2(n)} + y_{2(n+1)})) \\ y_{3(n+1)} &= y_{3(n)} + h * d * \frac{1}{2}(y_{2(n)} + y_{2(n+1)}) \end{aligned}\right\}$$

$$n = 0,1,2, \ldots$$

Since, the above system is nonlinear system, it is difficult to find unknown term $y_{1(n+1)}$, $y_{2(n+1)}$ and $y_{3(n+1)}$ directly by Matlab, here, it is necessarily to use Newton's Method as Jacobian Matrix.
Let $x = y_{1(n+1)}$, $y = y_{2(n+1)}$ and $z = y_{3(n+1)}$.

$$f_1(x, y, z) = x - y_{1(n)} + h * c * \frac{1}{2}\left(y_{1(n)} + x\right) * \frac{1}{2}(y_{2(n)} + y)$$

$$f_2(x, y, z) = y - y_{2(n)} - h(c * \frac{1}{2}\left(y_{1(n)} + x\right) * \frac{1}{2}(y_{2(n)} + y) + d * \frac{1}{2}(y_{2(n)} + y))$$

$$f_3(x, y, z) = z - y_{3(n)} - h * d * \frac{1}{2}(y_{2(n)} + y)$$

$$\begin{bmatrix} x_{m+1} \\ y_{m+1} \\ z_{m+1} \end{bmatrix} = \begin{bmatrix} x_m \\ y_m \\ z_m \end{bmatrix} - J^{-1}(x_m, y_m, z_m) \begin{bmatrix} f_1(x_m, y_m, z_m) \\ f_2(x_m, y_m, z_m) \\ f_3(x_m, y_m, z_m) \end{bmatrix}, m = 0,1,2, \ldots$$

$$\text{where } J = \begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} & \frac{\partial f_1}{\partial z} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} & \frac{\partial f_2}{\partial z} \\ \frac{\partial f_3}{\partial x} & \frac{\partial f_3}{\partial y} & \frac{\partial f_3}{\partial z} \end{bmatrix} \text{ is Jacobian matrix}$$

$$J(x, y, z) = \begin{bmatrix} 1 + 0.25 * h * c * (y_{2(n)} + y) & 0.25 * h * c * \left(y_{1(n)} + x\right) & 0 \\ -0.25 * h * c\left(y_{2(n)} + y\right) & 1 - 0.25 * h * c * \left(y_{1(n)} + x\right) + \frac{d}{2} & 0 \\ 0 & -\frac{hd}{2} & 1 \end{bmatrix} \Rightarrow$$

We have, $y_{1(0)} = 95$, $y_{2(0)} = 5$, $y_{3(0)} = 0$ and $N = 100$ the parameter values $c = 1$ and $d = 5$. $x_{(0)} = y_{1(n)}$, $y_{(0)} = y_{2(n)}$, $z_{(0)} = y_{3(n)}$

Secondly, the Computer Programming is given by **Appendix B**. the numerical solution is shown as in the figure 3

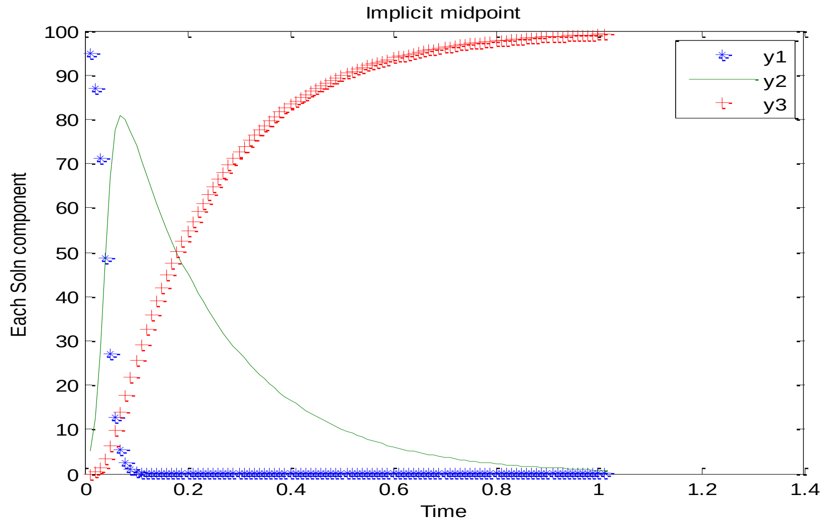

Figure 3: **The numerical solution by using Implicit midpoint rule when $y_{1(0)} = 95$, $y_{2(0)} = 5$, $y_{3(0)} = 0$ and $N = 100$**

In introduction of this paper, the Kermack-McKendrick model is defined explicitly. The mathematical differential equations are presented the the Kermack-McKendrick systems in general formula (3.2), we will focus on this paper only Uncoupled two differential equations with connected three equation together as as below.

$$\left.\begin{aligned} y_1' &= f_1(y_1, y_2, y_3) = -c_1 y_1 y_2 + k_1 y_3 \\ y_2' &= f_2(y_1, y_2, y_3) = c_1 y_1 y_2 - d_1 y_2 + k_1 y_3 \\ y_3' &= f_3(y_1, y_2, y_3) = d_1 y_2 + k_1 y_3 \end{aligned}\right\} \tag{3.5}$$

$$\left.\begin{aligned} y_4' &= f_4(y_4, y_5, y_6) = -c_2 y_4 y_5 + k_2 y_6 \\ y_5' &= f_5(y_4, y_5, y_6) = c_2 y_4 y_5 - d_2 y_5 + k_2 y_6 \\ y_6' &= f_6(y_4, y_5, y_6) = d_2 y_5 + k_2 y_6 \end{aligned}\right\} \tag{3.6}$$

Figure 4 shows One Equation of the Kermack-McKendrick structure in Simulink.

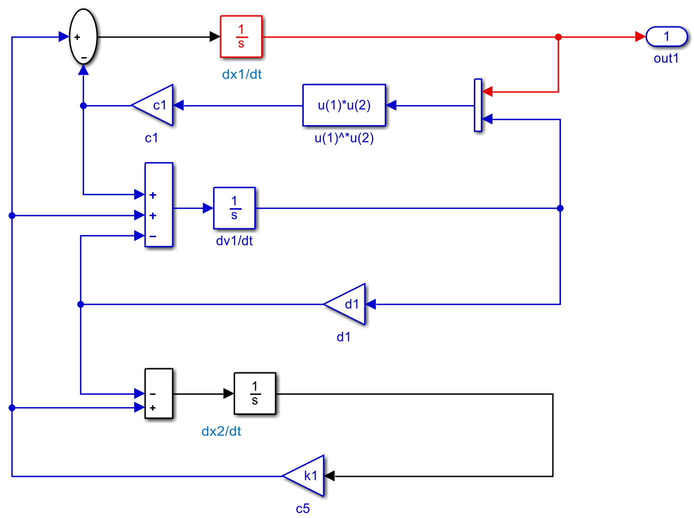

Figure 4: Internal Dynamics of single Equation (Modified Kermack-McKendrick)

The outputs of the systems are $x_1 = \theta_1$ and $x_2 = \theta_2$, where θ_1 and θ_2 are previously defined in our model., we just focus on the stability bidirectional two Equations, consider the equation (3.5), this system has four equations, it is a clear that, the equilibrium point is the origin point $(0,0,0,0)$. The Jacobian matrix at the origin point is

$$J(0,0,0,0) = \begin{bmatrix} 0 & 0 & k_1 \\ 0 & -d_1 & k_1 \\ d_1 & 0 & k_1 \end{bmatrix}$$

The characteristic polynomial is

$$\begin{vmatrix} \lambda & 0 & -k_1 \\ 0 & \lambda + d_1 & -k_1 \\ -d_1 & 0 & \lambda - k_1 \end{vmatrix} = 0 \Rightarrow$$

$$\lambda(\lambda + d_1)(\lambda - k_1) - k_1 d_1^2 = \lambda^3 + (d_1 - k_1)\lambda^2 - k_1 d_1 \lambda - k_1 d_1^2 = 0 \qquad (3.7)$$

It is clear $k_1 = 0$, then we will have multiplicity eigenvalues 0, this is enough to see that the system (3.5) is unstable at origin, we will just focus on modified system to avoid usability. The equation (3.7) is not easily to analysis it manually, we used both simulation and optimization to compare with these eigenvalues be in stable or not.

4 Optimizing Movement of one leg

There are two system (3.5) and (3.6), these two systems have to generate outputs angular patterns for each joint. To evaluate one leg with 2 DOFs generation, we have to know that, the optimization of parameters in an equation refers to finding the values of the parameters that result in the best performance of the equation. This is often done by minimizing an objective function that measures the error or deviation of the equation's output from the desired result. There are many different algorithms and techniques that can be used to optimize the parameters of an equation, including gradient descent, simulated annealing, and genetic algorithms. The choice of algorithm will depend on the specific characteristics of the equation and the optimization problem.
Actually, it is needed to find the optimal parameter sets by using the modified Kermack-McKendrick system, the parameter sets for each joint's Equation is given below. $P_1 = \{c_1, k_1, d_1, c_2, k_2, d_2\}$., there is one cost function is utilized; to obtain one leg, it should be depended on this cost function below.

$$J = -C_1 \sum_{k=1}^{n} x_b(k) + C_2 [\sum_{k=1}^{n} (\theta_1^2(k) + \theta_2^2(k))]/N, \qquad 10$$

where $C_1, C_2 \in [0,1]$ with $C_1 = \frac{1}{4}, C_2 = 3/4,$ n is the number of elements of position vector in simulation, and N is the length of the time. The aim here to maximize both the displacement or the velocity, as a results of that , we have to minimize J. If $C_2 = 0$, then the aim is to maximize the displacement. However, if $C_1 C_2 \neq 0$, then there will be another cost function involving energy related terms in addition to the position. The goal is to minimize the energy while changing the position
Still, there are two constraints $0 \leq \theta_1, \theta_2 \leq \pi$. Evolutionary optimization algorithms reveal the gait below in case constraints applied for joint angles. In this study, it is used the hybrid function during the optimization. A hybrid function is a mathematical function that combines different types of functions, such as polynomial and trigonometric functions. It can also refer to a function that combines different optimization techniques, such as gradient descent and simulated annealing. In optimization, hybrid functions are used to improve the performance

of an optimization algorithm by combining the strengths of different techniques. This can be especially useful when the optimization problem has multiple local minima or when the objective function is highly nonlinear. Hybrid optimization techniques can also be more computationally efficient and converge to a solution faster than using a single optimization technique. A hybrid function is an optimization function that runs after the genetic algorithm terminates in order to improve the value of the fitness function. The hybrid function uses the final point from the genetic algorithm as its initial point, we may conclude that locomotion is achievable by using the cost function J for the case of the systems (3.5) and (3.6) respectively as uncoupled case as shown in such as in the Figures 6, 7 and 8. While we could not achieve any locomotion for system (3.2) because there are multiplicity eigenvalue zero by using optimization.

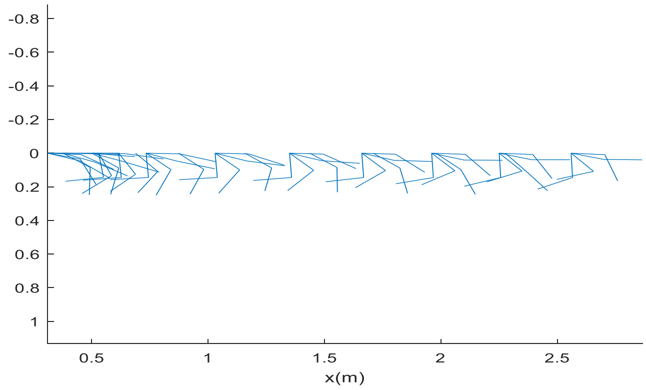

Figure.6: Simulation of Walking Gait with Constraints

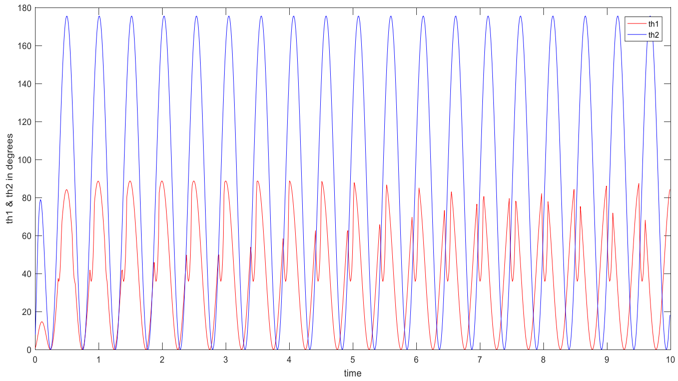

Figure 7: Joint angles against Time

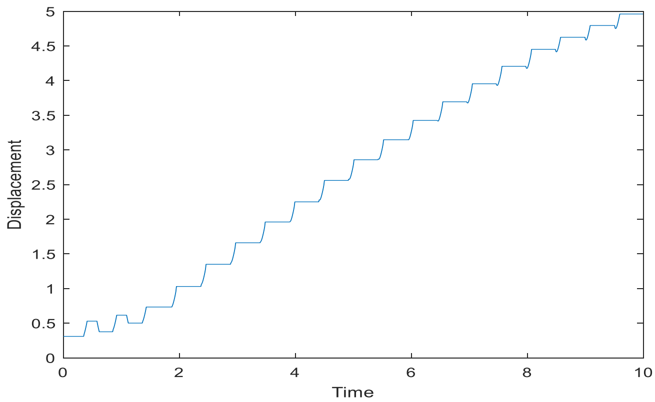

Figure 8. Displacement against Time

6 Conclusion and Future Directions

In this paper, the modified Kermack-McKendrick model are used to generate motion similar to the rhythmic patterns for one leg with two degrees of freedom, the one leg starts moving quickly and normally as seen in simulation using optimizing this model, which lead us to think about the future work to use this model in different seniors to obtain different type of movements for arm or leg by using different algorithm.

7 Appendixes

Appendix A

```
clear all
N=100;
c=1;
d=5;
h=1/N;
t=h*(1:(N+1))';
y1(1)=95;
y2(1)=5;
y3(1)=0;
for n=1:N;
    y1(n+1)=y1(n)-h*c*y1(n)* y2(n); % y1 by explicit Euler method
    y2(n+1)=y2(n)+h*(c*y1(n)* y2(n)-d* y2(n));% y2 by explicit
Euler method
    y3(n+1)=y3(n)+h*d*y2(n);% y3 by explicit Euler method
end
[t y1' y2' y3']
plot(t,y1,'*',t,y2,'-',t,y3,'+','linewidth',1);
title('Explicit Euler');
xlabel('Time')
ylabel('Each Soln component')
legend('y1','y2','y3')
```

Appendix B

```
clear all
N=100;
g=1;
c=1;
d=5;
Tol = 1.e-10;
h=1/N;
%n=1;
t=h*(1:(N+1))';
y1(1)=95;
y2(1)=5;
y3(1)=0;
%for n=1:N;
 %y1(n+1)=y1(n)-0.25*h*c*(y1(n)+y1(n+1))*(y2(n)+y2(n+1)); % y1 by
Implicit midpoint rule
```

```
 % y2(n+1)=y2(n)+h*(0.25*c*(y1(n)+y1(n+1))*(y2(n)+y2(n+1))-
0.5*d*(y2(n)+y2(n+1)));% y2 by Implicit midpoint rule
   % y3(n+1)=y3(n)+0.5*h*d*(y2(n)+y2(n+1));% y3 by Implicit
midpoint rule
%end
x(1)=95;
y(1)=5;
z(1)=0;
for n=1:100;
for m=1:30;   % number of iteration by Jacobian matrix
%J(m)=[1+0.25*h*c*(y2(n)+y(m)) 0.25*h*c*(y1(n)+x(m)) 0;-
0.25*h*c(y2(n)+y(m))...
%-0.25*h*c*(y1(n)+x(m))+d/2 0;0 -h*d/2 1];
%jacobian matrix
f1(m)=x(m)-y1(n)+0.25*h*c*(y1(n)+x(m))*(y2(n)+y(m));
f2(m)=y(m)-y2(n)-h*(0.25*c*(y1(n)+x(m))*(y2(n)+y(m))-
0.5*d*(y2(n)+y(m)));
f3(m)=z(m)-y3(n)-0.5*h*d*(y2(n)+y(m));
%[x(m+1);y(m+1) ;z(m+1)]=[x(m); y(m); z(m)]-
inv([1+0.25*h*c*(y2(n)+y(m)) 0.25*h*c*(y1(n)+x(m)) 0;-
0.25*h*c*(y2(n)+y(m)) -0.25*h*c*(y1(n)+x(m))+d/2 0;0 -h*d/2
1])*[f1(m); f2(m) ;f3(m)];
k=[x(m); y(m); z(m)];
b=[1+0.25*h*c*(y2(n)+y(m)) 0.25*h*c*(y1(n)+x(m)) 0;-
0.25*h*c*(y2(n)+y(m)) 1-0.25*h*c*(y1(n)+x(m))+d/2 0;0 -h*d/2
1];%Jacobian
s=[f1(m); f2(m) ;f3(m)];
r=inv(b)*s;
L=k-r;
k=L;
x(m+1)=k(1);y(m+1)=k(2);z(m+1)=k(3);
if abs(k(1))<Tol;
    abs(k(2))<Tol;
    abs(k(3))<Tol;
end
%x(m+1)=k(1);y(m+1)=k(2);z(m+1)=k(3);

end
g=g+1;

y1(g)=x(m);y2(g)=y(m);y3(g)=z(m);
end
%x(m)=y1(n+1);y(m)=y2(n+1);z(m)=y3(n+1);
[t y1' y2' y3']
plot(t,y1,'*',t,y2,'-',t,y3,'+','linewidth',1);
title('Implicit midpoint');
xlabel('Time')
ylabel('Each Soln component')
legend('y1','y2','y3')
```

7 References

1. Kermack, W.O., A.G. McKendrick, and E. Ponder, *XV.—The Stability of Suspensions. III. The Velocities of Sedimentation and of Cataphoresis of Suspensions in a Viscous Fluid.* Proceedings of the Royal Society of Edinburgh, 1930. **49**: p. 170-197.
2. Kermack, W.O., A.G. McKendrick, and P.L. McKinlay, *Death-rates in Great Britain and Sweden: expression of specific mortality rates as products of two factors, and some consequences thereof.* Epidemiology & Infection, 1934. **34**(4): p. 433-457.
3. Kaxiras, E., G. Neofotistos, and E. Angelaki, *The first 100 days: Modeling the evolution of the COVID-19 pandemic.* Chaos, Solitons & Fractals, 2020. **138**: p. 110114.
4. Yakubu, A.-A., *A discrete-time infectious disease model for global pandemics.* Proceedings of the National Academy of Sciences, 2021. **118**(42): p. e2116845118.
5. Elbori, A.E.G., M. Turan, and K.B. Arikan, *Evaluation and Optimization of Nonlinear Central Pattern Generators for Robotic Locomotion.* Journal of Control Engineering and Applied Informatics, 2018. **20**(3): p. 89-98.
6. Elbori, A.E.G., M. Turan, and K.B. Arıkan, *Optimization of Central Patterns Generators.* 2017.
7. Elbori, A., M. Turan, and K.B. Arikan, *Evaluation and Optimization of Nonlinear Central Pattern Generators for Robotic Locomotion.* 2017.
8. Abdalsmd, L.A., *Optimization of Human's Arm by Using Central Pattern Generators (CPGs)*, in *Material Science and Engineering Department*. 2018, Kastamonu University. p. 89.