Project Title

Name1 Surname1 123456789, Name2 Sureame2 987654321

1 Introduction

(General information of your project, WHAT, WHY)

Example: System dynamics modeling is a methodology to analyze, explain and design a desired system. Most of approaches used in system dynamics modeling are based on computer software. System dynamics modeling is not only applied on engineering fields, but many other fields, i.e., economic, managerial, complex social, or ecological systems. In modeling and simulation, the system is formed to differential equations or integral equations. The system is easily simulated by partitioning time into discrete intervals. Each time interval (Δ_T), all parameters in the system are calculated and updated. Normally, a desired system is analyzed and modeled for applying an appropriate control algorithm to it to make it works as expected. Nowadays, there are many tools that can be used for system modeling and controller design, e.g., Matlab, Octave, Mathematica, Maple, SciLab, Spyder, Maxima, and many other tools. Even though those tools are very useful for System dynamics modeling and controller design but they require a computer to run. They can also be used for mathematical proofs perfectly. In real-world applications, especially in industrial and real-time control applications, all ideas, concepts and algorithms are needed to be converted for the real-world. Practically, any system and controller can be modeled in computer world using those tools before converted into an embedded system (microcontroller). Unfortunately, because of limitation of microcontrollers, the algorithms obtained from the computer software cannot be deployed directly to a microcontroller to make it runs properly. Even if the algorithms are converted to discrete algorithms which can be implemented on a microcontroller, but some unexpected computational errors are often occurred. In microcontroller, especially in fixed-point microcontrollers, computational errors can be easily produced, because of round-off error or rounding error. In addition, in digital systems the Δ_T must be carefully controlled. Varying of the Δ_T will make unexpected behaviors of the system.

2 System Overview (Block diagram, Circuit/Schematic)

Example: The proposed system works as system emulators. It can emulate every system that can be converted into differential equations. We design and develop this system to run on a microcontroller platform. It implies that a microcontroller can emulate itself as system dynamics. In this paper, two systems are considered, one is a Mechanical system and another is an Electrical system. Both of the systems are analyzed and converted to discrete models. Fig.1 shows a block diagram of the proposed system. All modules shown in the block diagram are implemented in a microcontroller. The Mechanical Model and Electrical Model are differential equations that represent to those systems. The Model Parameters are used to store all constant parameters of the

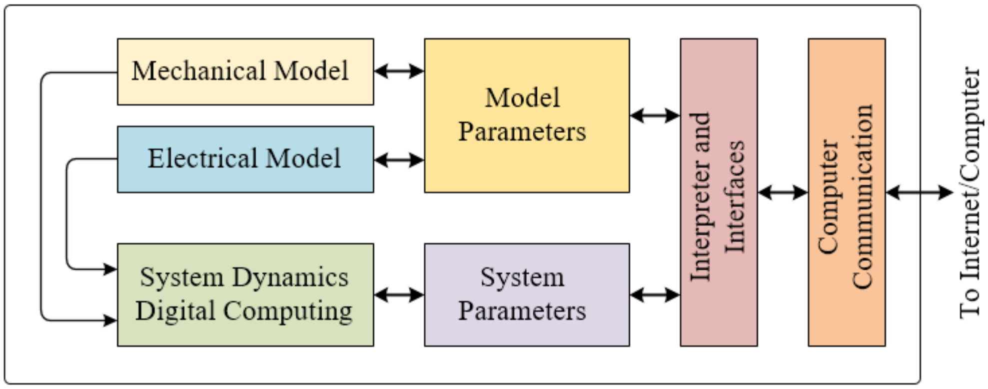

Fig.1. Proposed system block diagram

models, e.g., R, L, C, m, k, b and others. The System Dynamics Digital Computing composes of digital processing algorithms. It reads a model (differential equations) and calculates all parameters of the systems. The System Parameters are used to store and control system's parameters, e.g., sampling time, algorithm types and other internal operations. The Interpreter and Interfaces interpret parameters or data that flow back and forth between the system and computer over internet network. For example, when the host computer sends a data frame to change the sampling time, the Interpreter and Interfaces will convert that data frame and write to the System Parameters module to update the system operations. Finally, the Computer Communication is used to communicate between the system and computer. In this research the Universal Asynchronous Receiver/Transmitter (UART) protocol with 115200 bps, 1 start bit, 1 stop bit and no parity, no hardware flow control, is used. It connects to a standard 802.11b/g Wi-Fi module (WiFly R-717). Therefore, the system can communicate with computers and other devices through internet network. The system is controlled and monitored by web-applications running on a web-browser. It means that we can adjust system's parameters and examine all parameters through a web-page. The web-applications are implemented by JavaScript and HTML5. It supports WebSockets real-time communication. Therefore, we can control emulation parameters and visualize the emulation results on a Web-browser, e.g., Chrome, Firefox, Safari, Opera and others in real-time operation.

3 Example: System Dynamics Modeling (Details of each unit your system)

Example: Generally, all of system dynamics can be modelled by differential equation methods. In this paper, we demonstrate two types of the systems, a Mechanical system (a second-order Damped Harmonic Oscillator) and an Electrical system (a second-order RLC series circuit).

3.1 Example: Mechanical System

Example: The Mechanical system shown in Fig.6 is a Damped Harmonic Oscillator system. In this case, an output of the system is a displacement represented by $y(t)$. Let m denotes the mass, Newton's law becomes

$$F = m\frac{d^2y(t)}{dt^2}. \tag{4}$$

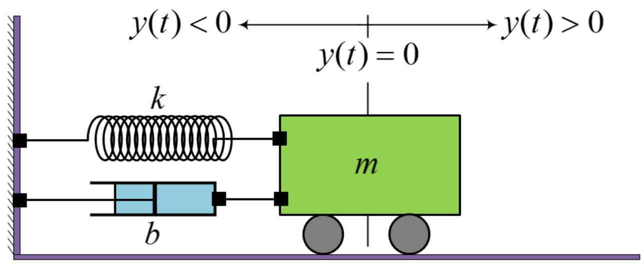

Fig.2. Mechanical diagram for modeling Mechanical system

Example: By applying the Hooke's law of spring, the restoring force exerted by a spring is lineally proportional to the spring's displacement from its rest position and the restoring force is directed toward the rest position. Thus, it can be described as following equation

$$F_s = -k_s y, \tag{5}$$

Example: where k_s is the spring constant, that greater than zero. The restoring force F_s equals to the Newton's law, ($F_s = F$), can be explained as

$$-k_s y(t) = m\frac{d^2y(t)}{dt^2}. \tag{6}$$

Example: Now, a second-order differential equation of *un-damped harmonic oscillator* can be obtained as follow

$$\frac{d^2y(t)}{dt^2} + \frac{k_s}{m}y(t) = 0. \tag{7}$$

3.2 Example: Electrical System

(Separate them into sub-topics and explain them clearly)

Example: An electrical circuit used in this work is the RLC series circuit which is shown in Fig.7. In a series circuits, the summation of voltages across all devices connected in the circuit equal to the voltage of the power supply. At $t=0$, the switch is at position *on*, by applying Kirchhoff's voltage law (KVL), we get

$$V_R(t)+V_L(t)+V_C(t)=V_S(t), \tag{8}$$

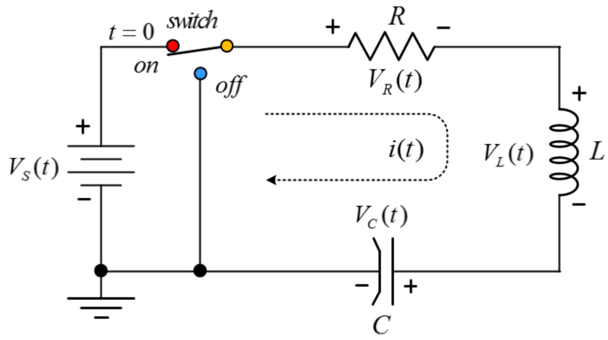

Fig.3. Electrical diagram for modeling Mechanical system

and well-known that the voltage dropping on each device can be computed by appropriate equation that related to its quantity and the current flowing in the circuit. So that the current $i(t)$ and voltage across R and L can be descripted as the following equations

$$i(t) = C\frac{dV_C(t)}{dt}, \tag{9}$$

$$V_R(t) = Ri(t), \tag{10}$$

$$V_L(t) = L\frac{di(t)}{dt}. \tag{11}$$

Example: Now we can replace the equations (12), (13) and (14) into (11). In this case, the voltage source is a DC voltage, so that the $V_S(t)$ can be replaced by a constant value V_S. A new equation is obtained, shown in the following equation

$$RC\frac{dV_c(t)}{dt} + LC\frac{d^2V_c(t)}{dt^2} + V_C(t) = V_S. \tag{12}$$

Example: To make it to be a standard form of second-order differential equation, divide all terms in (15) by LC and rearrange it. Now the standard form equation of the RLC series circuit is obtained as shown in the following equation

$$\frac{d^2V_c(t)}{dt^2} + \frac{R}{L}\frac{dV_c(t)}{dt} + \frac{V_C(t)}{LC} = \frac{V_S}{LC}. \tag{13}$$

Example: In this equation, we get a non-homogenous second-order differential equation. So the solution can be computed by a combination of the particular and homogenous solutions. In this case, by considering at $t \to \infty$, the right hand side becomes V_S and the left hand side becomes $V_C(t)$ (LC is cancelled out) which means that the voltage across the capacitor equals to voltage of the power supply V_S at the steady state.

In the same way as the mechanical system, we can rewrite the differential equation of electrical system in a standard form, let $p = R/L$ and $q = 1/LC$, equation (16) can be rewritten as

$$\frac{d^2V_c(t)}{dt^2} + p\frac{dV_c(t)}{dt} + qV_C(t) = qV_S. \qquad (17)$$

4 Example: Reduction of Order Technique

Example: Because the system equations of the mechanical and electrical systems, explained in (10) and (17), cannot be implemented directly on a microcontroller, the numerical approximation methods, i.e., integral and differential approximation approaches, described in Section III, are required to transform those continuous time-domain equations to discrete time-domain equations. It is known that any discrete time-domain equation, and any discrete s-domain equation (z-domain) can be implemented on a microcontroller. The system equations of the Mechanical and Electrical are second-order differential equations. So, the Euler's methods, explained in the Section III, cannot be applied on those system equations. Therefore, an additional technique is required to convert the second-order system equations to first-order system equations. The Reduction of order technique is selected for solving the second-order system equations [4]. Let say we have a second-order differential equation as shown in the following equation

$$\frac{d^2y(t)}{dt^2} = -p\frac{dy(t)}{dt} - qy(t), \qquad (18)$$

which can be written in a short form like this

$$y'' = -py' - qy. \qquad (19)$$

Example: To convert the second-order ode into two first-order differential equations, we define two differential equations as shown below

$$y' = v, \qquad (20)$$

$$y'' = v'. \qquad (21)$$

Example: We are now ready to approximate the two first-order differential equations by the Euler's methods. Let rewrite our system equations of the Mechanical system and Electrical system, explained in (9) and (16), in the same format of (18), we get

$$y'' = -\frac{b}{m}y' - \frac{1}{m}k_s y + 0, \qquad (22)$$

$$y'' = -\frac{R}{L}y' - \frac{1}{LC}y + \frac{1}{LC}V_S. \tag{23}$$

Example: Let z_k is an approximated solution of $y(t)$ at time t_k and w_k is an approximated solution of $v(t)$ at time t_k. Then we get the following recursion formulas for z_k and w_k

$$z_{k+1} = z_k + \Delta_T w_k, \tag{24}$$

$$w_{k+1} = w_k + \Delta_T y'', \tag{25}$$

Example: where the y'' is a differential equation in (22) or (23). Now replace y'' of (22) and (23) into (25), we get

$$w_{k+1} = w_k + \Delta_T \left\{ -\frac{b}{m}w_k - \frac{1}{m}k_s z_k + 0 \right\}, \tag{26}$$

$$w_{k+1} = w_k + \Delta_T \left\{ -\frac{R}{L}w_k - \frac{1}{LC}z_k + \frac{1}{LC}V_S \right\}. \tag{27}$$

Example: Finally, we achieve final version of the system equations that can be implemented on any microcontroller. A combination of (24) and (26) is used for the Mechanical system. While the Electrical system is represented by a combination of (24) and (27).

5 Web-based User Interface (Application) (Explain your design, its fractures, how to use it just like user manual)

Example: To evaluate the proposed system precisely, some numerical and graphical comparison techniques are required. The emulation results are carefully compared with reliable tool running on a computer, i.e., Matlab and Octave. For the Electrical system, it is also compared with circuit simulators, PSPICE and PROTEUS.

The Mechanical system is simulated without external forces. The system's parameters are: m=1.5kg, spring constant k=5.0N/m and damping factor b=1.5. The displacement is set to 10 meters with zero of velocity and acceleration. The simulation starts at t=0, then the mass moves back and forth with exponential decay oscillation. And finally, it goes to steady state around 10 seconds. To examine the side effect of the h or Δ_T, the number of steps are set to 50, 100 and 400

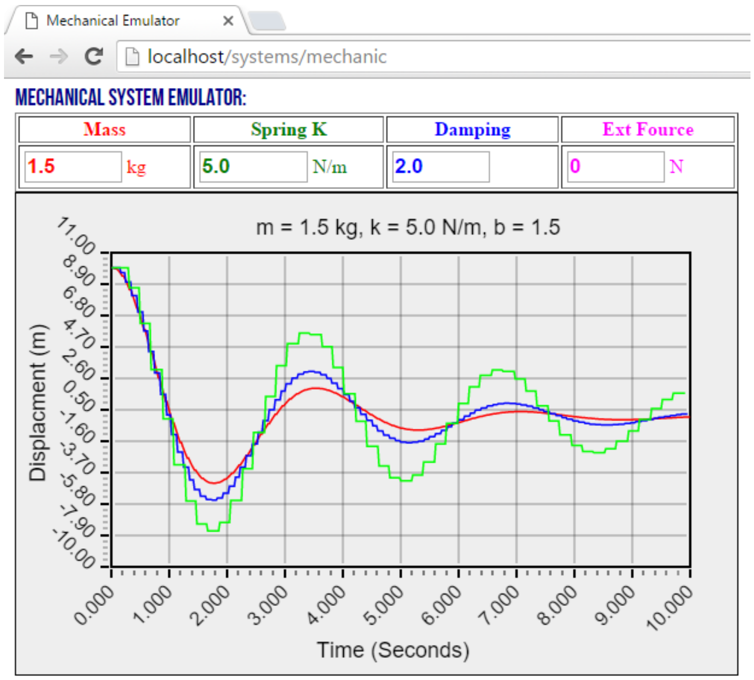

Fig.4. Result of the Mechanical system (m=1.5kg, ks=5N/m, b=1.5)

steps and the system runs for 10 seconds. Note that, the lower step is set, the higher Δ_T is produced. The experimental results are graphically visualized on a web-page as shown in Fig.8.

For the Electrical system, the parameters are set as following: R=10Ω, L=680uH, C=4.7uF and Vs=5V. The switch is turned *off* for a long time, then the system starts when the switch is turned

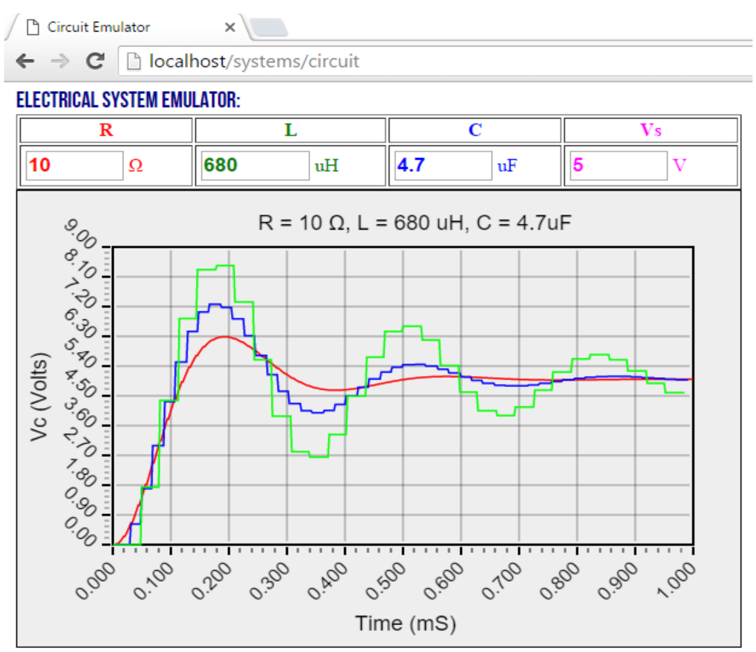

Fig.4. Results of the Electrical system (R=10Ω, L=680uH, C=4.7uF)

on. It means that no voltage and current in the circuit at t=0. With these parameters, the voltage across the capacitor exponentially increases as expected. One again, to examine the side effect of the Δ_T, the number of simulation steps are set to different values, 30, 50 and 400 steps respectively. Likewise, the experimental results are clearly displayed on a web-page as shown in Fig.9.

As expected, both of the systems dynamic emulators run correctly. The step size Δ_T or h is a parameter that must be considered to make the systems work precisely. The smaller Δ_T, the higher precision.

6 Conclusion

Example: In this project, the design and implementation technique of system emulator for Mechanical system and Electrical system is proposed. Both systems are designed and implemented on a small fixed-point microcontroller (no floating-point unit). The numerical approximation techniques based on Euler's methods are studied and used. The Reduction of order is then applied to the system equations, the second-order differential equations to convert to first-order differential equations. The final version of the system equations is then implemented on a fixed-point microcontroller. The experimental results of the Mechanical system and Electrical system are archived as expected. We closely analyze and compare the experimental results with the results produced by reliable software tools, Matlab and Octave, at the same Δ_T, and we realize that an insignificant error is emerged in the system because the microcontroller operates the mathematic operations on fixed-point processing unit. So, the insignificant error is produced by the effect of rounding error. However, it acquires many desired requirements of system dynamics modeling and it ensures that many fixed-point microcontrollers can be properly used for system dynamics emulator.