Journal
International I J of Modern
M E
Engineering
WWW.IJME.US
Print ISSN: 2157-8052
Online ISSN: 1930-6628

INTERNATIONAL JOURNAL OF MODERN ENGINEERING

Spring/Summer 2015
Volume 15, No. 2

The Leading Journal of Engineering, Applied Science and Technology

Industrial
Electronics
Biomedical
Civil
Aerospace
Computer

Electrical
Chemical
Mechanical

ENGINEERING

THE LEADING JOURNAL OF ENGINEERING, APPLIED SCIENCE, AND TECHNOLOGY

Print ISSN: 2157-8052
Online ISSN: 1930-6628

www.ijme.us

www.iajc.org

INTERNATIONAL JOURNAL OF MODERN ENGINEERING

ABOUT IJME:

- IJME was established in 2000 and is the first and official flagship journal of the International Association of Journal and Conferences (IAJC).
- IJME is a high-quality, independent journal steered by a distinguished board of directors and supported by an international review board representing many well-known universities, colleges and corporations in the U.S. and abroad.
- IJME has an impact factor of **3.00**, placing it among the top 100 engineering journals worldwide, and is the #1 visited engineering journal website (according to the National Science Digital Library).

OTHER IAJC JOURNALS:

- The International Journal of Engineering Research and Innovation (IJERI)
  For more information visit www.ijeri.org
- The Technology Interface International Journal (TIIJ).
  For more information visit www.tiij.org

IJME SUBMISSIONS:

- Manuscripts should be sent electronically to the manuscript editor, Dr. Philip Weinsier, at philipw@bgsu.edu.

  For submission guidelines visit
  www.ijme.us/submissions

TO JOIN THE REVIEW BOARD:

- Contact the chair of the International Review Board, Dr. Philip Weinsier, at philipw@bgsu.edu.

  For more information visit
  www.ijme.us/ijme_editorial.htm

INDEXING ORGANIZATIONS:

- IJME is currently indexed by 22 agencies. For a complete listing, please visit us at www.ijme.us.

Contact us:

Mark Rajai, Ph.D.

Editor-in-Chief
California State University-Northridge
College of Engineering and Computer Science
Room: JD 4510
Northridge, CA 91330
Office: (818) 677-5003
Email: mrajai@csun.edu

www.tiij.org

www.ijeri.org

INTERNATIONAL JOURNAL OF MODERN ENGINEERING

The INTERNATIONAL JOURNAL OF MODERN ENGINEERING (IJME) is an independent, not-for-profit publication, which aims to provide the engineering community with a resource and forum for scholarly expression and reflection.

IJME is published twice annually (fall and spring issues) and includes peer-reviewed articles, book and software reviews, editorials, and commentary that contribute to our understanding of the issues, problems, and research associated with engineering and related fields. The journal encourages the submission of manuscripts from private, public, and academic sectors. The views expressed are those of the authors and do not necessarily reflect the opinions of IJME or its editors.

EDITORIAL OFFICE:

Mark Rajai, Ph.D.
Editor-in-Chief
Office: (818) 677-2167
Email: ijmeeditor@iajc.org
Dept. of Manufacturing Systems Engineering & Management
California State University-Northridge
18111Nordhoff Street
Northridge, CA 91330-8332

THE INTERNATIONAL JOURNAL OF MODERN ENGINEERING EDITORS

TABLE OF CONTENTS

IN THIS ISSUE (P.13)
NUMERICAL SIMULATION OF A TRANSESTERIFICATION AND SENSITIVITY STUDY

Philip Weinsier, IJME Manuscript Editor

Greenhouse gases and global warming! For years, many have claimed that humans, and the animals we raise for food, are contributing to the warming of our planet by the gases we create, thereby causing potentially catastrophic events in our not-so-distant future. To be fair, others disagree and feel that Earth's climate has cycled continuously, hot to cold and back to hot again, throughout its history and is being only mildly affected by our current production of greenhouse gases (GHG). That is, we are currently in a global warming "stage", which will eventually reverse itself, irrespective of human activities. That does not necessarily mean that we should stop caring about our environment and throw caution—and greenhouse gases—to the wind.

Under relatively optimal conditions, the energy striking Earth through sunlight is equal to the energy radiated or reflected back into space, thereby allowing our temperature to remain roughly constant. Greenhouse gases in the atmosphere trap the energy reflected from the planet's surface causing, over time, the overall temperature of the Earth to increase. The logic is, then, that reducing greenhouse gases will reduce our perceived warming of the planet. Examples of naturally occurring gasses include water vapor, carbon dioxide, methane, nitrous oxide, and ozone. Man-made gases, typically coming from industrial processes, include hydrofluorocarbons (HFCs), carbon dioxide (CO_2), and chlorofluorocarbons (CFCs). Over the last few decades, CFCs have been banned by many countries. CO_2, however, continues to be a problem for the Earth's protective ozone layer. The U.S. produces about 25 percent of the CO_2 emissions worldwide (according to the National Energy Information Center's Energy Information Administration in Washington, DC), as we meet 85 percent of our energy needs by burning fossil fuels. And, worldwide, CO_2 emissions are expected to increase by about 1.9 percent by the year 2025.

A major contributor to the world's pollution and source of greenhouse gases is the burning of fossil fuels. But scientists and researchers are looking for ways to reduce the production of greenhouse gases. One way is to reduce our use of fossil fuels by adding small amounts (around 10 percent) of biofuels (ethanol and biodiesel) to the standard crude-oil-based fuels used in combustion engines. These biofuels offer 40-80 percent fewer GHG emissions than gasoline, do not require modifications to regular gasoline engines, and deliver similar horsepower, torque, and MPH to crude-oil-based fuels. Biodiesel is, in fact, so successful that the U.S., Brazil, and France are the top three producers in the world and have associated Federal mandates.

Biofuels are biodegradable, non-toxic, and often made from vegetable oils, waste cooking oil (WCO), tall oil (a by-product of the pulp and paper industry), and animal fats. Using a catalyst, sodium hydroxide for example, these oils are subjected to a reaction with an alcohol; this process is called transesterification. But biofuels are not new. Consider that the transesterification of vegetable oils dates back to the mid-1800s, when two scientists, E. Duffy and J. Patrick, received credit for first using the process in the production of soap. Their product was later named biodiesel. Today, many people feel that there is a growing trend towards biofuels; and perhaps there is. But that was also true from the end of the 19th century through the early 20th century, before fossil fuels became a cheaper source of fuel.

In 2013, roughly 2 percent of the energy used in the U.S. came from biofuels. Biofuels are considered carbon-neutral because the corn, sugarcane, soybeans, palm oil trees, etc. used to make them absorb CO_2 during growth and may offset the CO_2 produced when the biofuels are burned. But, not unexpectedly, there exists much controversy over the use of crop lands for biofuels instead of food crops, due to the decrease in available land for food crops and the additional use of fertilizers and energy to grow the crops.

In the featured article in this issue, the authors developed a predictive transesterification model to aid the process of reactor development. A Navier-Stokes reactive, multiple species flow model was developed and applied to a three-dimensional annular reactor, which ensured process modularity and product continuity. The chemistry of the flow was governed by four reversible reactions with kinetics data, and results demonstrated that higher conversion and product yield could be achieved at higher methanol-to-triglyceride molar ratios, lower velocities, and with a longer reactor. A high fidelity reactive flow model, based on CFD for transesterification was developed. The model was validated against experimental data for alcohol, TG, FAME, and glycerol yields obtained by Analytical GC/MS measurements.

Editorial Review Board Members

HDL-BASED IMPLEMENTATION OF A NOVEL DYNAMIC RECONFIGURABLE DATA ACQUISITION SCHEDULER-ON-CHIP (SCHOC)

M. Abdallah, State University of New York Polytechnic Institute

Abstract

Data acquisition is the process of sampling a physical quantity from the environment. In the case of more than one input, a reliable scheduler is needed. In this study, a number of signals with different frequencies were acquired via the proposed Scheduler-on-Chip (SchoC). The speed of the acquisition/storing process, the amount of data being acquired, and the signal reconstruction quality were the three main design characteristics that were taken into account. Moreover, the proposed system was dynamic, which means that it could detect any change in any of the input channels/ signals. It could also adapt to the change and try to find the optimal sampling rate for each signal on the fly. Performance evaluations showed that the speed of the SchoC was 24% faster than a comparable software-based scheduler. The SchoC reduced the amount of data acquired by up to 59%, which in turn decreased memory requirements. It also achieved a high signal reconstruction quality and achieved a better root mean square of errors than a comparable software-based DAQ.

Introduction

In most real-time systems, task scheduling is the most important problem, because it is the scheduling plan that ensures that tasks meet their deadlines. Each task has its own timing restrictions or deadlines. Existing scheduling algorithms can be classified as shown in Figure 1. Dynamic scheduling is more flexible, allowing characteristic-varying tasks to be scheduled. In a round-robin process, processor time is equitably divided among all processes. Each process gets an equal time slot of a shared processor. If heterogeneous multi-rate processes are scheduled by a single core processor, the round-robin scheduling technique assigns the shared processor to all processes with a fixed processing rate [1]. In a Rate Monotonic (RM) process, tasks are assigned different priorities. Tasks with higher priority will interrupt the current task and replace it. This also means that the system is preemptive. The priorities are assigned to tasks based on their frequency. Priorities are assumed to be static, so the task periods also need to be static. Hence, RM cannot be used if tasks have varying frequencies [2]. Earliest Deadline First (EDF) places processes in a priority queue. The process which is closest to its deadline will be scheduled for execution. However, EDF has some drawbacks such as situations in which deadlines are not known in advance; thus, they are provided for but subject to change and/or situations that require uniform process spacing [2]. Scheduler-on-Chip (SchoC) is a new technique to dynamically schedule a large number of tasks using a single processor without increasing the requirements of the processor. It can fit under the quality-driven item in Figure 1.

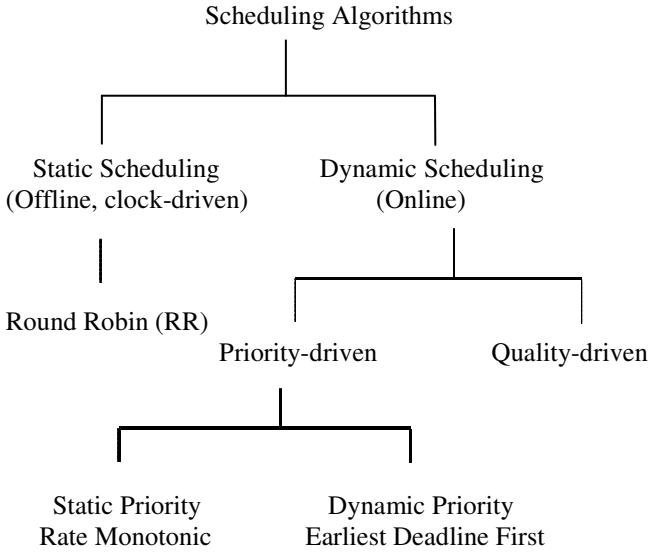

Figure 1. Classification of Existing Scheduling

The focus of this current study was to develop a low-cost, small-circuit-size data acquisition system for a multiple-channel system, where several applications must schedule their tasks on a single processor. There are several channels and each requires its own optimal sampling rate. Therefore, the problem of scheduling the tasks can be formulated as an optimization problem. Given a maximum sampling rate, F_s, of an analog-to-digital converter (ADC) for a total number of N channels, the problem will be how to assign an optimal sampling rate, F_{si}, for each channel. A mathematical formulation of the problem was developed. This algorithm allowed for maximizing of the total number of channels but also minimized the amount of data being acquired and stored and which would eventually be transmitted. The sys-

tem was prototyped on an Altera Cyclone II Field Programmable Gate Array (FPGA). The scheduling algorithm that finds the optimal sampling rate was developed using only Hardware Description Language (HDL) modules (i.e., hardware) and compared against known scheduling techniques (round-robin, RM, and EDF). The speed of the proposed scheduler was compared against a software-based scheduling implementation on the NIOS II processor using C++. The power consumption calculations and comparison are currently under investigation and will be reported in future studies.

Applications

Using a single ADC is the best solution in terms of circuit size and power consumption, compared to one dedicated ADC per channel. Using a single ADC to acquire multi-channel signals is the optimal solution in nanotechnology applications, since circuit size is a critical issue. The proposed SchoC can be utilized in many applications, since different applications need to schedule multi-rate tasks on a single-core processor. Different analog sensors and quasi-digital sensors and transducers such as frequency can use the proposed SchoC [3]. The environmental measurements [4] and spectroscopic imaging [5], [6] are examples of applications that can utilize the proposed idea.

The proposed SchoC can be used in order to acquire human body signals such as heartbeat, pressure, and lung sounds. Using a varying sampling rate per channel is the optimal solution in terms of scalability, power consumption, and memory requirements [7], as well as for applications having only one channel of varying data sources. Each time period, the sampling rate should be configured according to the input channel characteristics. In the case of a single sensor of varying data sources, such as irradiation or vibration, the proposed dynamic SchoC can be used as well [8].

Design Methodology

For dynamic heterogeneous multi-channel signals, it is proposed that each channel be sampled at the appropriate sampling frequency in order to maximize the total number of channels and minimize the amount of data being acquired, which eventually will be stored or transmitted. The SchoC was designed and implemented in an FPGA to manage the variable switching time of the ADC and the multiplexer such that an arbitrarily large number of channels can be acquired with good reconstruction quality. One of the advantages of the proposed system that is not considered here is power consumption. According to Equation (1), the dynamic power increases as the activity factor, α, increases which, in turn, will increase the total power consumption. If the sampling rate increases, the activity factor will be increased.

$$P_{total} = \alpha\, C\, V^2\, F + I\, V + P_{short} \quad (1)$$

where, *C* is capacitance, *V* is voltage, and *F* is the processor clock frequency.

Generally speaking, dynamic power can be reduced by using different, lower rates of activity factor for less frequent sampling for channels that require a much smaller minimum service rate, as compared to channels with higher service rates.

Optimization Problem

The problem can be formulated as an optimization problem. Given a maximum sampling rate, F_s, of the ADC and a total number, N, of channels, an optimized sampling rate, F_{si}, for each channel needs to be assigned, as determined by Equation (2):

$$\begin{aligned} & Max \sum_{i=1}^{N} \log_2 (OSR_i) \\ & subject\ to: \sum_{i=1}^{N} F_{si} \le F_s \\ & \qquad T_{si}\ \%\ T_s = 0 \\ & \qquad T_{si} \times M_1 \ne T_{sj} \times M_2 + T_s \times M_3 \end{aligned} \quad (2)$$

where, the oversampling ratio for channel (i) = $OSR_i = F_{si} / F_i$; i, j =1,...,N ; $i \ne j$; $T_{si} = 1/ F_{si}$; $T_s = 1/ F_s$; M1, M2, and M3 are integers ($\le 2\ Max(T_{si})$); and, M3 is the number of time periods, T_s, between channel j and channel i.

The objective of this optimization problem was to maximize the assigned sampling rate (or minimize the sampling period) for each channel. However, there were various restrictions that limited the OSR_i. First, the sum of the assigned sampling rates for all channels had to be less than or equal to the total sampling rate of the available ADC. Second, the assigned sampling rate for any channel could not be an arbitrary number. Its inverse, the time period, had to be a multiple of the inverse of the total sampling rate of the available ADC, T_s. Third, at any given time, no more than one channel could be sampled. For example, assume that T_s = 0.625 nsec; Ch_1 has T_{s1} = 2.5 nsec; and, Ch_2 has T_{s2} = 1.875 nsec. For this scenario, there was a problem at time T = 2.5 nsec. Both Ch_1 and Ch_2 needed to be sampled at the same time. This problem is called Same Time Sampling (STS) and has to be avoided in order to maintain high signal reconstruction quality.

SchoC Implementation

In this section, the implementation of the proposed SchoC using an Altera FPGA Cyclone II is presented. Four design issues are considered: 1) the proposed scheduler logic; 2) the software characteristics of the software implementation of the proposed—the software implementation is only introduced for comparison purposes with the hardware implementation of the scheduler; 3) the Digital Signal Pre-Processing Unit (DSPU)—the DSPU detects each channel's frequency; 4) the Frequency Detection and Monitoring (FREDM) Module, which detects any change in the frequency and reports it. In Figure 2, a SchoC flowchart is presented. The inputs to the SchoC were frequency bandwidths of input channels. These frequency bandwidths were calculated on-the-fly by a proposed special DSP module, implemented in the FPGA using only HDL modules. Therefore, it was different than the EDF scheduler, where the user/operator had to input these frequency bandwidths in advance.

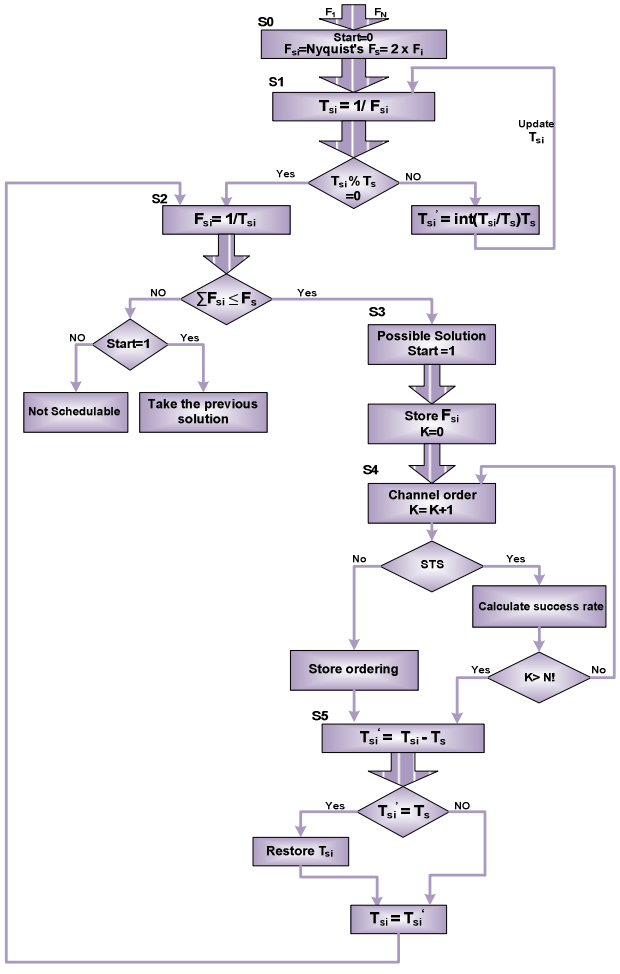

Figure 2. The Proposed SchoC Flowchart

The SchoC technique starts with state 0 (S0), where each channel has its own sampling rate, F_{si} and equals the Nyquist sampling rate. Therefore, each channel's sampling period, T_{si}, had to be calculated by taking $1/F_{si}$ in state 1 (S1). T_{si} should be multiples of T_s; if it is, the flow goes to state 2 (S2), otherwise, T_{si} should be updated to be a multiple of T_s. In state 2 (S2), a brute-force search method was used. It was necessary to check the total sum of the sampling frequencies of the channels. If it exceeded the maximum sampling rate of the available ADC, the task would not be schedulable. Otherwise, the control goes to state 3 (S3). Current sampling frequencies could be a solution; however, the STS problem should be checked before judging the current solution. In state 4 (S4), channels order iterations to take place. In each order, STS problems are checked. If there is an STS problem, a success rate is calculated and then the control goes to state 4 (S4) again for the next iteration. If there is no STS problem, a valid solution is approached, which is needed to be stored as a possible optimal solution. After storing the solution or after all iterations are completed without finding a solution, state 5 (S5) should be reached. In state 5 (S5) for each channel, T_{si} is decremented once by the value of T_s, taking into account that T_{si} cannot be smaller than T_s. Then, the flow goes again to state 2 (S2).

Implementation: Software versus Hardware

The proposed system was implemented twice, in hardware and in software, in order to compare the results. A NIOS II processor was used to implement the software solution. The NIOS II processor consists of three types of cores, namely NIOS-II/e (economy), NIOS-II/f (fast), and NIOS-II/s (standard). The NIOS-II/f was the core used in this study. It is a 32-bit RISC processor, is pipelined, has an instruction cache, data cache, and 51 Dhrystone million instructions per second. The NIOS II processor is also provided with a JTAG debug module, which has software debug capabilities that enable a designer to debug the application when launched in hardware. The NIOS II processor is based on Harvard architecture, where program and data are stored separately. It has a 4 kB instruction cache and a 2 kB data cache. The processor runs on a 50 MHz clock. The NIOS-II processor core is connected to its components through the Avalon switch interface.

Digital Signal Pre-processing Unit

The digital signal pre-processing unit (DSPU) performs automatic signal conditioning such as bias addition and removal, adaptive signal scaling, and filtering [9]. An exten-

sion of the DSPU is presented here to have the built-in capability to perform digital signal processing such as FFT and DCT for one-dimensional digital signals. Its added function in this study was to perform real-time Fast Fourier Transformation (FFT) for each channel in order to monitor each channel's varying frequency. As shown in Figure 3, DSPU has two main tasks. It performs FFT for each channel. In addition, a power density calculation is performed in order to detect each channel's frequency bandwidth (F_1, F_2, …, F_N). Moreover, it passes these values to the Frequency Detection and Monitoring (FREDM) module which, in turn, automatically decides if a frequency change occurred.

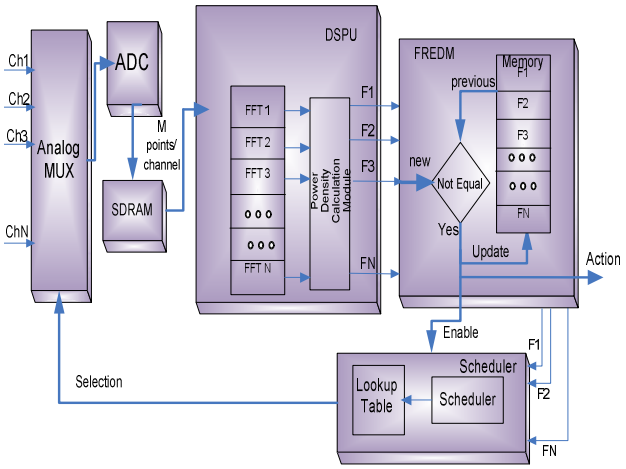

Figure 3. DSPU Module, FREDM, and SchoC

In the beginning, a number of samples (M) was acquired from each channel and the DSPU performed the FFT on these samples using the Altera FFT MegaCore function. The FFT MegaCore function's outputs are M points, where M is the number of points used in FFT processing. Each point has real and imaginary parts. For each channel, M output points were applied to a power density calculation module. The power density was calculated using Equation (3):

$$P_i = \frac{(realpart_i)^2 + (imagpart_i)^2}{M}, i = 1, \ldots, M \quad (3)$$

It is well known that only the first half of the M points are used to determine the frequency bandwidth. The second half is a mirror of the first half. An input signal is not necessary for a single-frequency signal; it may have multiple frequency components. In order to detect the frequency bandwidth in this case, the greatest frequency component should be detected. The greatest frequency component is used to decide the optimal sampling rate for a channel.

The Frequency Detection and Monitoring Module

After detecting frequency bandwidths F_1, F_2, …, F_N via the DSPU, the Frequency Change Detector in the FREDM compared each channel's frequency with its previous values. If both values are within a small preset range, no action is taken. Otherwise, new frequency values are stored in memory and an enable signal is activated. The enable output signal is the input to the SchoC. Moreover, the Action signal can simply report the change in frequency to a user and indicates which channel has the frequency change. The action signal can drive a certain device to perform a predefined action such as an alarm or frequency restoration (in power systems). Figure 3 illustrates the intercommunication between the DSPU, FREDM, and SchoC.

SchoC Evaluation

Six parameters are presented here to verify the performance of the proposed SchoC. Signal reconstruction and speed were used to compare the software and hardware implementations of the proposed scheduler. Accuracy was used to compare the hardware implementation of the scheduler and a well-known existing technology (NI). Memory requirements and a schedulability analysis were performed with the proposed SchoC and the well-known round-robin scheduling technique.

Signal Reconstruction

Both EDF and RM scheduling, and the proposed SchoC techniques were implemented in software using NIOS II processor. Moreover, the proposed SchoC was implemented again in only hardware. When the scheduler implementation is done via software processor, low-quality reconstruction signals are generated especially with high-frequency signals. The NIOS II processor was used to implement the proposed scheduler, EDF, and the RM scheduling techniques. Its instructions have a non-uniform execution time. In other words, the time between each acquired sample is not equal. A Tektronix logic analyzer was used to prove this observation. The logic analyzer was connected to the acquisition clock of the NIOS II processor via one of the I/O pins of the FPGA. Figure 4 shows the non-uniform behavior of the generated acquisition clock of the NIOS II processor. This affects the frequency of the stored signal. So, one can see that the stored reconstructed signal starts to slow down and deviate from the original signal. This comes from the fact that the NIOS II processor is instruction-based, which makes data acquisition non-uniform. In other words, the time between each acquired sample is not equal. This af-

fects the frequency of the stored signal. Therefore, the proposed hardware SchoC was better than the software-based implementation, in terms of the signal reconstruction quality.

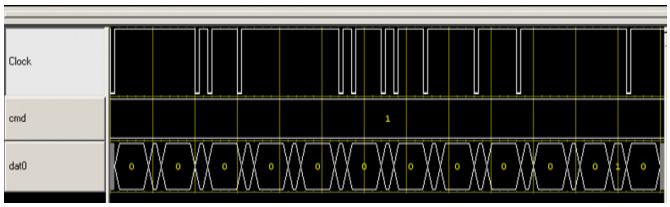

Figure 4. Logic Analyzer Showing the Non-uniform Behavior of the NIOS II Processor Acquisition Rate

Speed

In the implemented software-based DAQ systems, the speed of data acquisition, processing and storing is slow. It reaches 46 seconds to process and store only 8000 blocks of data, where each block contains 512 samples. Hence, another design approach was implemented in this study. The SchoC hardware acquisition was implemented in order to get faster data acquisition and processing, and a better storing system as well as to maintain an accurate signal reconstruction in terms of the frequency. A comparison was done between the proposed SchoC and the software-based NIOS II acquisition and storing systems. Different data sizes were applied to both systems. The performance evaluation showed that the speed of the proposed hardware SchoC was 24% faster than a comparable software-based scheduler. Figure 5 shows the time saved using the SchoC over the software-based schedulers. The horizontal axis represents the data size (number of blocks) and the vertical axis represents the time saved in seconds. As shown, as the data size increased, the time saved also increased.

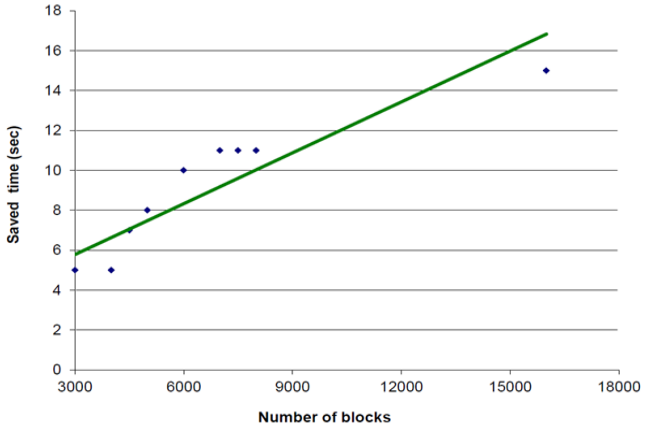

Figure 5. Time Saved Using the Proposed SchoC

Accuracy

The National Instruments (NI) data acquisition card was chosen because it has the closest similarity to the proposed DAQ structure (although it is a computer-based device). The NI test-bench is a PCI-6115 4-channel, 12-bit DAQ card [10]. This National Instruments DAQ was used for comparison. To be fair, the sampling rate was fixed for both systems at 40 MSPS. The proposed SchoC utilized a Texas Instruments ADC12D040 12-bit ADC [11]. Different signals were applied to both systems. Acquired signals from both systems were tested in terms of root mean square of errors. A signal generator was used to generate the sine waves under investigation. The proposed SchoC works as a standalone device without any interfere from the computer. All of the processing and control was done by the FPGA. On the other hand, the NI-based DAQ needs a LabVIEW program that runs on a computer in order to store the input signal into a file. Both acquired/stored signals by both systems were tested using Matlab. The root mean square of errors was used as an evaluation parameter.

Four analog sinewaves of 1, 1.5, 2, and 3 MHz were applied to the inputs of both DAQ systems. Different sample sizes of N = 100 and 1000 were considered. A comparison between the acquired/stored signals via both DAQ systems with respect to the source signal was made. The root mean square of errors RMS (e) was calculated for both systems. Table 1 shows a comparison of the proposed SchoC and a comparable NI-based system. From Table 1, although the NI card has two signal conditioning stages, the performance of the proposed SchoC was better than the NI-DAQ. The best performance for the proposed SchoC had 15% better (smaller) RMS (e) values than the NI-based DAQ, in the case of 1000 samples of 2 MHz input signals.

Table 1. Root Mean Square of the Error: Hardware FPGA-based Acquisition SchoC and Ni-based DAQs

	RMSe (proposed SchoC) for (N) samples	RMSe (NI-DAQ) for (N) samples	% Improvement N=100		
Signal	100	1000	100	1000	
1 MHz	0.07	0.075	0.07	0.076	1
1.5 MHz	0.05	0.065	0.07	0.077	15
2 MHz	0.09	0.091	0.093	0.093	2
3 MHz	0.092	0.096	0.099	0.11	12

Memory Requirements

Both the round-robin and the proposed SchoC schedulers were implemented in hardware. The goal was to test the ability of the proposed SchoC to minimize the memory requirements. Various case studies were considered for verification. One is presented in this section. Eight sinusoidal analog signals were considered, with channel frequencies of 500 kHz, 200 kHz, 190 kHz, 185 kHz, 160 kHz, 100 kHz, 100 kHz, and 100 kHz, and a maximum sampling rate, F_s, of the available ADC of 10 MSPS (T_s = 100 nsec). The SchoC is shown in Figure 6. The x-axis represents time slots separated by T_s. The y-axis represents the acquired channels (1 to 8). On the other hand, the round-robin scheduler can schedule the given signals with a fixed sampling rate for all channels (1/800nsec = 1.25MSPS). This leads to oversampling of low-frequency signals (such as the 100 kHz signal) which, in turn, causes extra memory storage requirements and power consumption. The SchoC scheduled the eight input channels to be sampled at a varying sampling rate. This reduced the amount of data being acquired which, in turn, decreased the required memory. The total amount of data being acquired using the proposed SchoC was less than that acquired by the round-robin scheduler by 59%.

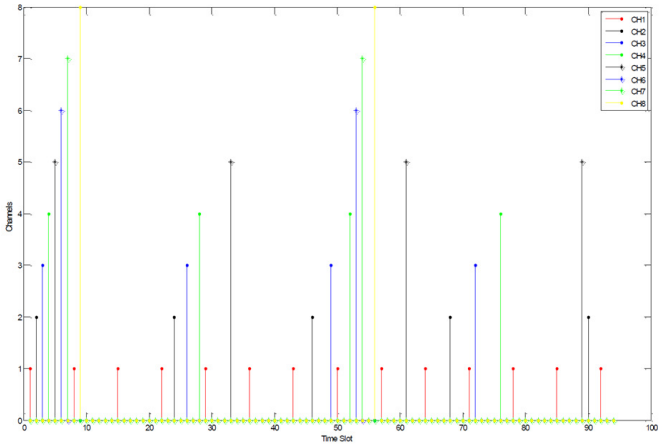

Figure 6. Eight Channels with a Varying Sampling Rate

Schedulability Analysis

Various case studies were considered for verification. One is presented in this section. Consider a case study where the proposed SchoC can schedule a given task and a round-robin technique cannot get a solution for the same task. Let the maximum sampling rate, F_s, of the given ADC be 100 MS/s. The time period is 10 nsec. Three analog signals were applied to the proposed SchoC: 25 MHz, 10 MHz, and 2 MHz. In addition, assume that no channel has priority. Applying Nyquist's law, the following must be satisfied: $F_{s1} \geq 50$ MS/s, $F_{s2} \geq 20$ MS/s, and $F_{s3} \geq 4$ MS/s, where F_{si} is the sampling frequency for channel (i). In other words, $T_{s1} \leq 20$ nsec, $T_{s2} \leq 50$ nsec, and $T_{s3} \leq 250$ nsec, where T_{si} is sampling time for channel (i). If a round-robin sampling technique is applied, these three signals cannot be sampled using the available ADC. Applying the proposed scheduling technique, these analog signals can be optimally scheduled using the available ADC. As shown in Figure 7, the optimal scheduling is: T_{s1} = 20 nsec, T_{s2} = 40 nsec, and T_{s3} = 240 nsec (F_{s1} = 50 MS/s, F_{s2} = 25 MS/s, and F_{s3} = 4.166 MS/s). The three constraints, mentioned in the optimization problem, are satisfied.

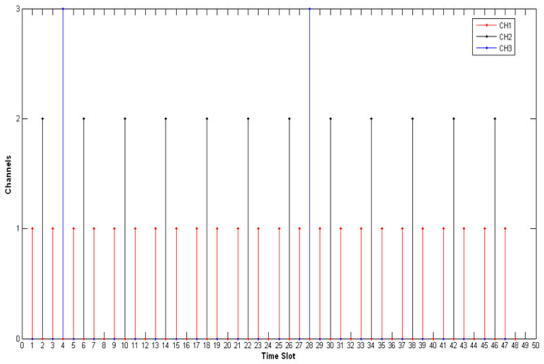

Figure 7. The Round-robin Technique Cannot Schedule the Given Signals, but the Proposed SchoC Can

In Figure 7, the horizontal axis represents the time slots, where the period between each consecutive time slot is T_s, which, in this example, is equal to 10 nsec. The vertical axis represents the channel being sampled at a certain time slot. As shown in the figure, three channels are mentioned on the vertical axis. At time slots 1, 3, 5,..., channel 1 is sampled. At time slots 2, 6, 10,..., channel 2 is sampled. At time slots 4, 28,..., channel 3 is sampled.

Synthesis Analysis of the Proposed SchoC

From the Altera developing tools and the synthesis report, it was found that the prototype of the proposed SchoC only used 11,779 logic elements (35% of the available logic elements on the FPGA), 2770 registers, 244 pins (51% of the available pins), 75,904 memory bits (16% of the available memory bits), and four embedded multiplier elements (6% of those available). It can sample up to 32 channels using a single ADC.

Dynamic SchoC

The proposed dynamic SchoC was tested to verify that it could dynamically detect any change in the input channel

frequencies. Moreover, that it could adapt to the frequency change and adjust the sampling rate for each channel/signal. Three signals were considered here (25 MHz, 15 MHz, and 5 MHz). A single ADC (100 MSPS) was used to acquire the three signals. In the FFT calculations, 512 points were used. As shown in Figure 8, source_real and source_imag were the real and imaginary parts of each output point, respectively. The signal py was the Power Spectral Density (PSD) of the first 255 points. The signal s_pyy_max was the highest PSD and s_pyy_max_index was its index. In other words, s_pyy_max_index was the location, fci, of the greatest frequency component. In Figure 8, F_s1 means the sampling frequency for channel 1, F_s2 the sampling frequency for channel 2, and F_s3 the sampling frequency for channel 3. T_s1 means the sampling period for channel 1, T_s2 the sampling period for channel 2, and T_s3 the sampling period for channel 3. As shown, the proposed dynamic hardware SchoC detected each channel's frequency and determined the optimal sampling rate for each channel (F_{s1}=50 MSPS, F_{s2}=33 MSPS, and F_{s3}=10 MSPS). The sum of all sampling rates was 100 MSPS. It has to be less than or equal to the given ADC's sampling rate of 100 MSPS. Therefore, the single ADC was fully utilized in this case.

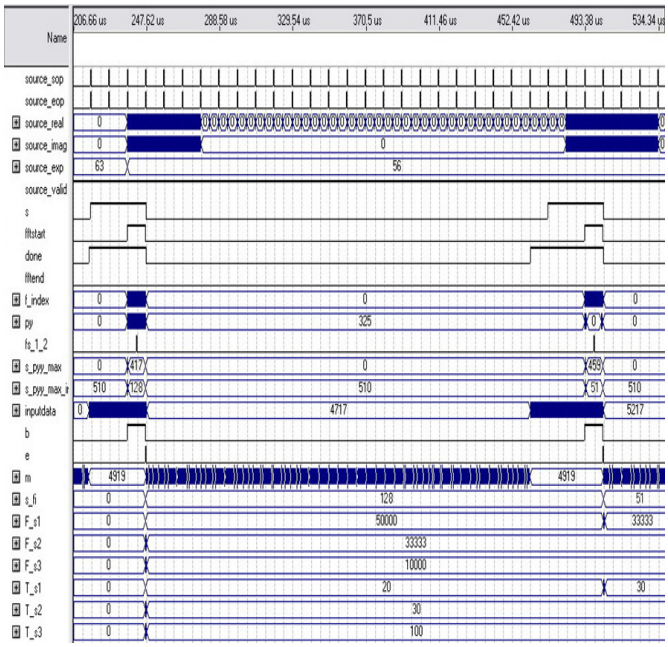

Figure 8. Dynamic SchoC (case study I)

At a certain point, channel one's input signal was changed from 25 MHz to 16 MHz. As shown in the Figure 8, Fs_1 (the sampling rate for channel one) was changed accordingly from 50 MSPS to 33 MSPS, and T_s1 (the sampling period of channel one) was changed from 20 nsec to 30 nsec. As a second case study, three signals were considered as well. As shown in Figure 9, signals two and three were changed. Therefore, sampling frequencies for both channels were changed accordingly.

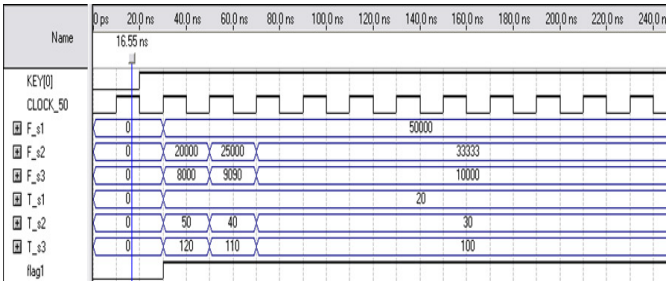

Figure 9. Dynamic SchoC (case study II)

Cost Comparison

Nallatech [12] does utilize the FPGA in their dual 3 Gsps ADC board (i.e., all main design modules are performed using an FPGA), but they use a dedicated ADC per channel which, in turn, increases the power consumption, cost, and circuit size. The cost of the Nallatech standalone BenADC-3G is currently $22,000 [12]. It also has only dual channels. Therefore, it is not scalable. It also lacks the adaptive/ reconfigurable optimized sampling technique. Moreover, according to Venture Development Corporation [13], a study regarding the global and product shifts in the DAQ market was made. They expected that the total worldwide shipments of data acquisition products would greatly increase. In particular, a 22% increase to $321.5 million was projected for 2014 in just the U.S.

Conclusions

In various DAQ applications, a fixed sampling rate is not the optimal solution. The dynamic proposed scheduler-on-chip (SchoC) achieves the optimal solution for a large number of channels. It also reduces power consumption and memory requirements. Frequency detection and monitoring capabilities are integrated and implemented in the proposed SchoC. The proposed research provided design, implementation, and verification of a scheduler that takes full advantage of the capabilities of the FPGA as well as using a single multiplexed ADC for multi-channel analog signal acquisition and processing. It also detects and monitors their frequencies. Any change in input channel frequency is reported via Frequency Detection and/or Monitoring (FREDM) system. If the input signals have different frequency bandwidths, then the proposed SchoC is required to perform adaptive sampling, instead of using the highest frequency as a fixed sampling rate for all channels. Oversampling low-frequency spectrum channels leads to unnecessary data, which in turn requires extra storage capabilities and more power consumption.

Different case studies were presented here. As a result, the proposed hardware dynamic SchoC was found to be able to schedule tasks that were not schedulable via the round-robin technique. It also reduced the amount of data being acquired by up to 59%, which in turn decreased memory requirements. Moreover, the implementation of the SchoC as a hardware component on a chip achieved fast data acquisition, processing, and storing system, as well as maintained an accurate signal reconstruction in terms of the frequency. The performance evaluation shows that the speed of the proposed SchoC was 24% faster than a comparable software-based scheduler. It reached 15% better (smaller) RMS (e) than the NI software-based DAQ. Different case studies showed that the proposed dynamic SchoC could detect any frequency change in the input signals. It also could adapt to the change and determined new optimal sampling rates for each channel.

References

[1] Leung, J., & Anderson, J. H. (2004). *Handbook of Scheduling: Algorithms, Models, and Performance Analysis*. Boca Raton, FL: CRC Press.

[2] Brucker, P. (2007). *Scheduling Algorithms*. (5th ed.). New York: Springer.

[3] Yurish, S. (2009). Intelligent Multichannel Data Acquisition Systems Based on Intelligent Universal Sensors and Transducers Interfacing Circuits. *IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems*, (pp. 38-43).

[4] Thomass, J., & Tang, F. (1996). A Proposed Bridge to Soil Moisture Estimation Using Current and Future Operational Satellite Radars. *Geoscience and Remote Sensing Symposium*.

[5] Hartov, A., Mazzarese, R. A., Reiss, F. R., Kerner, T. E., Osterman, K. S., Williams, D. B., et al. (2012). A Multichannel Continuously Selectable Multifrequency Electrical Impedance Spectroscopy Measurement System. *IEEE Transactions on Biomedical Engineering*, *47*(1), 49-57.

[6] Avramova, Z., Wittevrongel, S., Bruneel, H., & De Vleeschauwer, D. (2009). Dimensioning of a Multi-Rate Network Transporting Variable Bit Rate TV Channels. *IEEE International Conference on Communications*, (pp. 1-6).

[7] Rieger, R., & Taylor, J. (2009). An Adaptive Sampling System for Sensor Nodes in Body Area Networks. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *17*(2), 183-189.

[8] Abdallah, M. (2013). FPGA-Based Adaptive Data Acquisition Scheduler-on-Chip (SchoC) for Heterogeneous Signals. *Reconfigurable Computing: Architectures, Tools and Applications, Lecture Notes in Computer Science*, 7806, 223.

[9] Abdallah, M., & Elkeelany, O. (2011, August). A Low-Cost Stand-Alone Multichannel Data Acquisition Monitoring and Archival System with On-Chip Signal Preprocessing. *IEEE Transaction on Instrumentation and Measurement*, *60*(8), 2813–2827.

[10] National Instruments. (2013). PCI-6115. Retrieved from http://sine.ni.com/nips/cds/ view/p/lang/en/ nid/11886

[11] Texas Instruments. (n.d.) ADC12D040. Retrieved from http://www.ti.com/product/ adc12d040

[12] Nallatech. (2010). Virtex-4, Dual 3 GSPS ADC. Retrieved from http://www.nallatech.com

[13] Taylor, J. (2008). *Global and Product Shifts in the Data Acquisition Market*. Boston, MA: Venture Development Corporation Press.

Biography

M. ABDALLAH received his Ph.D. in Electrical Engineering and works as an Assistant Professor at SUNYIT. His research area is reconfigurable computing and embedded systems. He works as a reviewer for many journals. He has published more than 40 conference papers and journal articles. Dr. Abdallah may be reached at abdallm@sunyit.edu

NUMERICAL SIMULATION OF A TRANSESTERIFICATION AND SENSITIVITY STUDY

Isam Janajreh, Masdar Institute (UAE); Mohamed Almusharrekh, Masdar Institute (UAE); Mohammed Husain, Masdar Institute (UAE)

Abstract

Transesterification is mainly achieved through a batch or semi-continuous reactor of the two immiscible reactants (methanol or ethanol and triglyceride) in the presence of a catalyst. A continuous reactor has the advantage of high throughput and is characterized as requiring minimal labor, while providing higher process efficiency, profitability, and less variability in product quality. As the process is influenced by a long list of geometrical, physical, and environmental parameters, their interactions have been problematic for the process engineer. High fidelity modeling and simulation of a reactive flow can isolate the influence of these parameters. In this study, a predictive transesterification numerical model was developed to aid the process of reactor development. A Navier-Stokes, reactive, multiple species flow model was developed and applied to a three-dimensional annular reactor, which ensured process modularity and product continuity. The chemistry of the flow was governed by four reversible reactions with kinetics data that were inferred from the work of Noureddini and Zhu [1] and closely reproduced in this current study [2]. In particular, this work entailed a sensitivity study of the methanol/lipid ratio, inlet velocity, temperature, and tube geometry in an attempt to find the optimal reactor design and process conditions. Results demonstrated that higher conversion and product yield could be achieved at higher methanol-to-triglyceride molar ratios, lower velocities, and with a longer reactor. This work also provided detailed information on the spatial distribution and progression of each of the reaction kinetics and molar fractions of the reactant and product species.

Nomenclature

Symbol	Meaning
K_1	Forward reaction constant for the first reaction
K_2	Backward reaction constant for the first reaction
K_3	Forward reaction constant for the second reaction
K_4	Backward reaction constant for the second reaction
K_5	Forward reaction constant for the third reaction
K_6	Backward reaction constant for the third reaction
K_7	Forward reaction constant for the overall reaction
K_8	Backward reaction constant for the overall reaction
E_1	Activation energy for the first reaction in the forward direction
E_2	Activation energy for the first reaction in the backward direction
E_3	Activation energy for the second reaction in the forward direction
E_4	Activation energy for the second reaction in the backward direction
E_5	Activation energy for the third reaction in the forward direction
E_6	Activation energy for the third reaction in the backward direction
E_7	Activation energy for the overall reaction in the forward direction
E_8	Activation energy for the overall reaction in the backward direction
u	Velocity component
$S\phi$	Dependent variable/turbulent scalars
T	Temperature
ρ	Density
μ_t	Eddy viscosity
$D_{i,m}$	Diffusion coefficient
f,C	User-defined constants
σ_k , σ_ε	Empirical constants
m_i	Species
Sc_t	Turbulent Schmidt number
k_f , k_b	Forward and backward reaction constants
C_j	Molar concentration of the j^{th} species
η	Reaction order
v	Stoichiometric coefficient
M_i	Molar weight
R_i	Reaction source term
m_f	Mass fraction

Introduction

Biodiesel is considered a sustainable alternative to diesel fuel, as it can be utilized in current internal combustion diesel engines without any modification. It is transesterified from the combination of methanol or ethanol and triglycerides/lipid to form mono-alkyl ester in the presence and aid of a catalyst. The process is driven by a long list of parameters (e.g., molar ratio of reactants, their inlet conditions and properties, reactor geometry, etc.) making experimentation extremely tedious and sometimes inconclusive. Trapped grease, waste cooking oil (WCO), animal fat tallow, and algae oil are considered substitute feedstock and, hence, their psycho-chemical properties extend the list of parameters. High fidelity computational fluid dynamics (CFD) that model and simulate reactive flow is sought to gain more insight into the reacting flow parameters. Modeling and simulation of reactive flow finite volume discretization in a coupled system of equations (continuity, momentum, energy, transport of species, and turbulence scalars) have appeared in previous studies by the authors [3-5]. Recent advances in solution algorithms, computer memory, and processing speed facilitated this research field. Reactive flow simulations became a complementary laboratory tool and resulted in sever reduction in experimental setup and sometimes total replacement.

Transesterification still lacks robust modeling and simulation, despite well-defined process reactions [6]. The intent of this current study was to demonstrate the feasibility of pursuing numerical simulation for the transesterification process under different sets of flow parameters, thereby hoping to attract more interest for further research that could lead to improved and optimized reactor development. Transesterification can be conducted in a batch or continuous reactor. A continuous reactor has the advantage of high throughput, decreased labor, higher efficiency, and less product variability. The reactants are typically premixed by a head pump and are introduced into the reactor as a homogenous mixture. These reactants, triglyceride (TG) and alcohol (AL), are immiscible and under poor mixing mass-transfer limitations deter the reactions. As the reactions proceed, products are generated, the immiscibility is reduced, and kinetics drives the reaction [3], [6], [7].

Boer and Bahri [8] investigated the mixing of oil and alcohol and observed the two-phase mass transfer limitation in the form of a sludge flow of a dispersed biphasic mixture. Mixing or induced turbulence, a higher temperature reaction, and an increase in the alcohol amount enhance the preferable forward reaction. Similarly, the presence of a catalyst reduces the reaction activation energy and allows the process to proceed faster. In this current study, the stoichiometric molar ratio was three moles of alcohols to one mole of triglyceride; however, practically speaking, this ratio rarely achieves the desired conversion. Results from this and other studies [1-3] showed a sigmoidal reaction progress for the transesterification. Hence, significant agitation was required to increase the reaction rate, as indicated by Boocock et al. [9]. The main reaction starts slowly, followed by a sharp increase and finally entering another slow asymptotic stage. This could be attributed to the reduction in droplet sizes, surfactant action of the intermediates, and the effect of solvability properties of the biodiesel product, as observed by Boocock et al. [9], Ma et al. [10], and Zhou and Boocock [11], all of whom described the reaction mixture during this stage as a pseudo-single-phase emulsion. The final reaction stage could be attributed to the two-phase reversion as glycerol (GL) concentration is increased, the ineffectiveness of the catalyst which dissolves in the produced polar phase, and the depletion of the glyceride reactants.

The main assumption of the simulation was to homogenize the mixing of the two reactants (TG and AL) at the reactor inlet, assuming that they would be pumped/injected together via a peristaltic or diaphragm pump at the desired velocity to ensure proper and homogenized mixings. This assumption made it possible to carry the analysis in a single phase flow of multiple reacting species. The simulation was carried out in an upright, dual annular reactor in which the flow was injected tangentially at values much higher than the laminar regime (Reynolds number $\approx$ 6000). Results of the kinetic studies done elsewhere by the authors were favorably compared in the literature [2]. The values of the rate constants and activation energies for the eight elementary reactions used in this study are shown in Table 1. It should be noted that the adopted kinetic model of these inferred values was pseudo-first-order that combined with a shunt-reaction scheme, as was also stated by the work of Freedman et al. [12]. This work was intended to show the feasibility of carrying high fidelity reactive flow simulations to the transesterification under different operational conditions in an attempt to use the model as a conceptual design tool at an early stage of reactor design.

Governing Equations and Model Development

System of Equations

Modeling reactive flow requires the application of flow continuity, momentum, energy equations, additional species transport equations, and turbulence scalars. The transesterification starts as soon as the reactant components enter the

Table 1. Kinetic Data of Transesterification, Reaction Constant, and Activation Energy

Reaction rate constant	K_1	K_2	K_3	K_4	K_5	K_6	K_7	K_8
WCO at 50°C Janajreh [2]	0.0356	0.1718	0.0688	0.2786	0.0866	0.0025	4.3E-9	0.0031
WCO at 60°C Janajreh [2]	0.0414	0.1579	0.0846	0.3054	0.1126	0.005	1.9E-8	0.0014
Noureddini and Zhu [1]	0.049	0.102	0.218	1.280	0.239	0.007	7.84E-5	1.5E-5
Activation EnergykJ/kmol	E_1	E_2	E_3	E_4	E_5	E_6	E_7	E_8
WCO at 50-60 °C Janajreh [2]	0.1319	0.0737	0.1823	0.0807	0.2304	0.6342	3.0544	0.6825
Noureddini and Zhu [1]	0.0632	0.0477	0.0955	0.0704	0.0308	0.0461	-	-

reactor. The flow is governed by the Navier-Stokes equations associated with temporal, convective, viscous, and source terms and is described in Equation (1):

$$\frac{\partial}{\partial t}(\phi) + \frac{\partial}{\partial x_i}(u_i\phi) = \frac{\partial}{\partial x_i}(\Gamma_\phi \frac{\partial \phi}{\partial x_i}) + S_\phi \quad (1)$$

Time Rate *Advective* *Diffusion* *Source*

where, u_i is the i^{th} (x, y, and z) velocity component and S_ϕ is the source term, due to the dispersed phase interaction. Also, ϕ is the dependent variable corresponding to density, ρ, constituting the continuity; the velocity density multiple, ρu_i, constituting the momentum; and, the temperature, T, governs the energy. The turbulent scalars, turbulent kinetic energy, k, and the turbulent dissipation rate, ϵ, can also be represented by ϕ.

These two equations in the steady state flow regime are written here as Equations (2) and (3):

$$\rho u_i \frac{\partial k}{\partial x_i} = \mu_t \left(\frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j} \right) \frac{\partial u_j}{\partial x_i} + - \frac{\partial}{\partial x_i} \left(\frac{\mu_t}{\sigma_k} \frac{\partial k}{\partial x_i} \right) - \rho \varepsilon \quad (2)$$

$$\rho u_i \frac{\partial \varepsilon}{\partial x_i} = C_{1\varepsilon} \frac{\mu_t \varepsilon}{k} \left(\frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j} \right) \frac{\partial u_j}{\partial x_i} + \frac{\partial}{\partial x_i} \left(\frac{\mu_t}{\sigma_\varepsilon} \frac{\partial \varepsilon}{\partial x_i} \right) - C_{2\varepsilon} \frac{\rho \varepsilon^2}{k} \quad (3)$$

The right-hand terms represent the diffusion, the generation, and the destruction/dissipation of the turbulent quantities, respectively. In these equations, μ_t is the eddy viscosity, which is an order of magnitude higher than the laminar viscosity, and can be written as Equation (4):

$$\mu_t = f_\mu C_\mu k^2 / \varepsilon \quad (4)$$

where, f and C are user-defined constants and $C_{1\varepsilon}$, $C_{2\varepsilon}$, σ_k, and σ_ε are empirical constants. The species m_i transport equation is given here as Equation (5):

$$\frac{\partial}{\partial t}(\rho m_i) + \frac{\partial}{\partial x_i}(\rho u_i m_i) = \frac{\partial}{\partial x_i}\left(\rho D_{i,m} + \mu_t / Sc_t\right) \frac{\partial m_i}{\partial x_i} + R_i + S_i \quad (5)$$

where, $D_{i,m}$ is the diffusion coefficient of m_i specie; Sc_t is the turbulent Schmidt number and is defined as the ratio of the eddy viscosity, μ_t, to the eddy diffusivity, $D_{i,m}$.

These transport equations incorporate an additional reaction source term, R_i, that accounts for species reaction and are governed by the stoichiometric reaction according to Equation (6) and its Arrhenius rate of Equation (7):

$$\sum_{i=1}^{N} v'_{i,r} S_i \underset{k_{b,r}}{\overset{k_{f,r}}{\Leftrightarrow}} \sum_{i=1}^{N} v''_{i,r} S_i \quad (6)$$

$$k_{f\,or\,b} = A T^{\alpha} e^{-E/RT} \quad (7)$$

where, v represents the stoichiometric coefficients; A is the pre-constant rate; T is the temperature raised to its dependent exponent, α; E is the activation energy; and, R is the universal gas constant.

The reaction rate is proportional to the concentration of the reaction products raised to the specified power coefficients. That is, the i^{th} species production/destruction, due to the reaction r, is written as shown in Equation (8):

$$R_{i,r} = M_{i,r}(v''_{i,r} - v'_{i,r})\left(k_f \prod_{j=1}^{N} C_{j,r}^{\eta^*_{rj}} - k_b \prod_{j=1}^{N} C_{j,r}^{v''_{j,r}} \right) \quad (8)$$

where, k_f and k_b are the forward and backward reaction constants, based on Arrhenius Equation (7); C_j is the molar concentration of the j^{th} specie raised to the stoichiometric coefficients v and reaction order η; and, M_i is the molecular weight of species i [13].

The four reversible volumetric reactions were considered, which are described by Equations (9)-(12) and referred to as TG reaction, DG reaction, MG reaction, and Shunt reaction, respectively.

$$Triglyceride + Alcohol \xleftarrow{catalyst} \rightleftharpoons FAME + Diglyceride \quad (9)$$

$$Diglyceride + Alcohol \xleftarrow{catalyst} FAME + Monoglyceride \quad (10)$$

$$Monoglyceride + Alcohol \xleftarrow{catalyst} FAME + Glycerol \quad (11)$$

$$Triglyceride + 3Alcohol \xleftarrow{catalyst} 3FAME + Glycerol \quad (12)$$

Model Assumptions and Development

The assumption of instantaneous equilibrium and equal diffusivity is too generic for numerical conversion of lipid within the annular reactor conditions. Hence, the numerical investigation in this transesterification study was based on eight global reactions and their kinetics in order to more accurately describe the multi-physics, multi-scale, multi-component model. The model was based on Eulerian reactive flow that considers the four reversible reactions of Equations (9)-(12). The injection of the well-mixed reaction components transversally into the first and outer annular chambers of the reactor marks the onset of these homogeneous reactions. As the swirl intensity decreases along the reactor, the flow exits the first annular chamber and is introduced again to the second inner annular chamber, thereby rejuvenating the swirl and doubling the reaction time.

Four mesh types were made for the reactor by surface revolution of the 2D quadratic cross-sectional surfaces. The first 3D mesh was used to validate the model against experimental measurements of the material species, while the other three 3D meshes were utilized for the mesh sensitivity study. The fluid was modeled with the Eulerian concept, where the residuals over the entire control volume were being monitored. The conservation of mass, momentum, energy, and species were ensured, based on Equations (1)-(5). As a turbulent flow was considered, the closure problem for the Reynolds stresses in the momentum equation was solved according to Equation (4) and the standard k-ε model leading to the inclusion of the two coupled transport equations for k and ε of Equation (2) and (3). The kinetics for the homogeneous reactions and their chemical equations are described in Table 1.

The reactive flow at different component settings of stoichiometric and over stoichiometric of the methanol and lipid (WCO) concentrations, flow conditions, and temperatures was prescribed at the inlet, atmospheric conditions at the flow exit, and no-slip velocity and adiabatic temperature at the chambers walls. These analyses and reactions were developed within the framework of Fluent/Ansys commercial code [14] following steady flow and semi-implicit pressure-velocity couplings (SIMPLE). A second-order upwind discretization scheme was used for the advective terms, and a second-order central scheme was used for the diffusion terms. The solution was obtained when a residual of 1E-5 in all scalar equations was reached—i.e., mass, velocities, k, e, and the five transport equations: methanol, FAME (Fatty Acid Methyl Ester), triglyceride, monoglycerides, and glycerol (no lipid transport as it constituted the balance).

The two reactant species were entering the inner, exiting, then entering the outer chambers of the upright reactor at an inlet Reynolds number (Re) near 6000 (based on the bottom inlet diameter of 4 mm) and 1-to-6 triglyceride-methanol molar ratio. The atmospheric pressure outlet boundary condition was assigned at the outer tubular chamber located at the top. The no-slip wall velocities were imposed on all of the surfaces of the reactor. Descriptions of the six species are summarized in Table 2, following the work of Narvaez et al. [15] and Goodwin [16]. The mixing law was used to determine the overall properties of the mixture, as needed, and is given in Equation (13):

$$\varphi_m = \sum_{i=1}^{n} Y_i \varphi_i \quad (13)$$

where, φ is the property and the subscripts i and m signify the i^{th} species and mixture, respectively; Y_i is the mass fraction of the i^{th} specie.

Model Validation

A simple cylindrical flow model was used and validated against experimental data of the species concentrations that were taken from a well-stirred reactor in the work by Noureddini and Zhu [1]. This model consisted of a 3D, asymmetrical 15-degree wedge, with a radius of 1 cm and a length of 100 cm. The domain was discretized with a high-quality hexagonal mesh with one cell per degree circumfer-

Table 2. Summary of Species Properties and Molecular Weights

Species	Chemical formula	Molecular weight	Viscosity (kg/m.s)	Cp (J/kg.^{o}C)	Density Kg/m^3
Methanol	CH_4O	32	3.96e-4	1.470e3	791.8
Waste oil or Triglyceride	$C_{54}H_{104}O_6$	848	1.61e-2	2.2e3	883.3
Diglyceride	$C_{37}H_{72}O_5$	596	-	-	880
Monoglyceride	$C_{20}H_{40}O_4$	344	-	-	875
Biodiesel	$C_{18}H_{36}O_6$	284	1.12e-3	1.187e3	870
Glycerol	$C_3H_9O_3$	93	1.412e0	0238.6	1261

entially by 15 cells radially by 400 in the length-wise direction, the latter directions have equally spaced divisions. The mixture was injected axially at 0.001 m/s into the reactor at 50 ^{o}C. A steady-state solution was sought as the solution reached the desired residuals of pf 1E-8 in all scalar equations: mass, momentum, energy, and transport. Experimentally, in the waste-to-energy group at Masdar, WCO was used as a feedstock, which was collected from the institute's canteen at an estimated generating rate of nearly 60 liter/ week. The oil was typically properly handled and kept away from water mixing for third-party collection. The collected sample was water dried, filtered down to 15 mm by means of filtering papers, and titrated, suggesting low free-acid contents making it suitable for transesterification. Palm oil and sunflower oil were the common types of oil used locally, which can be represented by the average molecular formula of $C_{54}H_{104}O_6$.

Measurements of the concentration of oil (triglyceride), FAME, glycerol, and alcohol were conducted by Thermoscientific GC/MS that was equipped with an FID detector. A standard FAME column that was calibrated against several standard biodiesel samples was used to set the calibration curve for qualitative and quantitative species measurements. Results of the model and experimental data are plotted in Figure 1. It should be noted that the time in the model corresponds to the local transfer time, which was equated to the specific downstream station divided by the inflow duct velocity. Measurements are those obtained from the well-stirred batch reactor at 3600 rpm and 50 ^{o}C. Model results show some noticeable deviations in the first 20 minutes for the biodiesel, glycerol, and alcohol, yet the trends were followed fairly well. A fairly good model and experimental match was shown to dominate the remaining time in all of the four species.

Continuous Reactor Configuration and Setup

An illustration of the dual annular chamber transesterification reactor is depicted in Figure 2. For detailed drawings of the chambers and their exact dimensions, please refer to the work by Abd Rabu et al. [17] and Abd Rabu [18]. The tubular reactor consisted of two coinciding and separated chambers with many features including: compactness, low pressure drop, ease of temperature control, isothermal reaction condition, and modular configuration for easy scaling up and scaling down. The modularity can be achieved through multiple-reactor stacking or simply by using longer tubing fitted on the same end-caps. The reactants were introduced circumferentially and, hence, minimized the pressure drop and increased the residence flow time during its swirling trajectory.

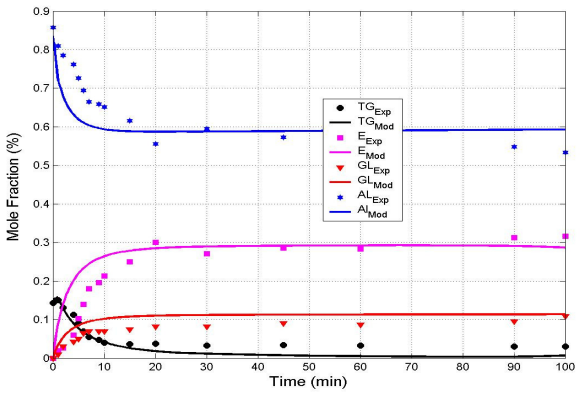

Figure 1. Validation of Model Data and Experimental Measurements of Triglyceride, Ester, Glycerol, and Methanol

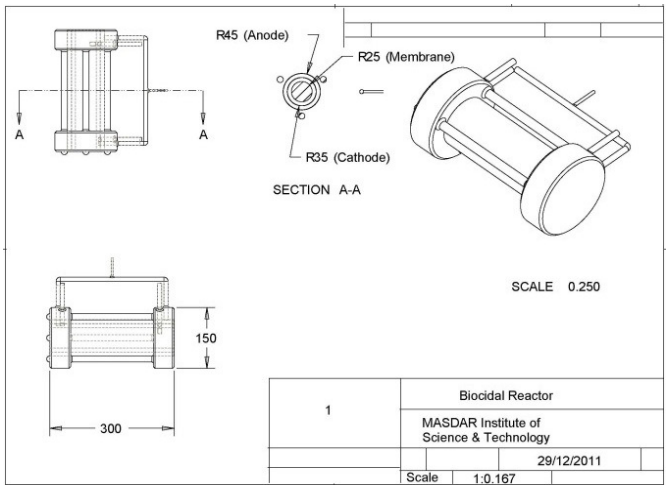

Figure 2. Dual Annular Chamber Reactor Configuration (dimensions in mm)

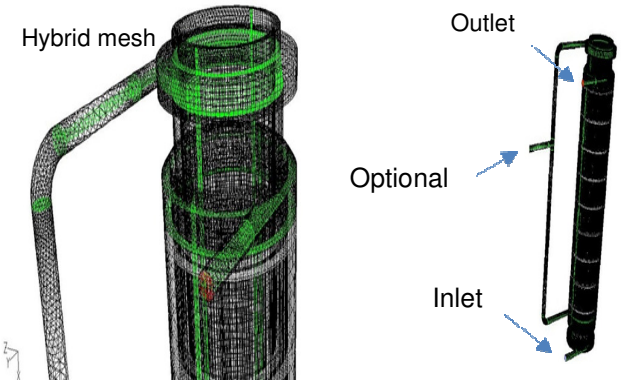

Figure 3. Reactor Tube Configuration and the Discretized Mesh (dimensions are mm)

A hybrid hexagonal and trigonal mesh was carefully generated to capture the near-wall, steep-flow transition. It was comprised of 275,000 cells for the two chambers, in addition to the connecting tubing between the exit of the outer 1st chamber and inlet of the inner 2nd chamber, as shown in Figure 3.

Mesh Sensitivity Analysis

The mesh sensitivity study was performed on a non-reactive flow over three levels of mesh refinements in order to assess the solution mesh dependency. The fine, baseline, and coarse meshes were comprised of 2200K, 275K, and 34K cells, respectively. Flow quantify, such as total pressure drop, was in close proximity of the three meshes (8 Pas, 12 Pas, and 12.75 Pas). The total resident time was used as one of the important flow parameters that the reaction also depended upon. An order of 150 seconds was observed in the first chamber and 160 seconds in the second. The finer mesh led to slightly lower residence time than the baseline mesh, suggesting the feasibility of the latter. A summary of the generated meshes and their flow data are listed in Table 3. The flow was injected at velocity beyond the laminar regime, ensuring the required homogeneous mixing of the reactants and avoidance of any mass transfer limitation. The mixture was entrained following a swirl and slow trajectory leading to the desired residence time.

Table 3. Summary of the Mesh Sensitivity Results

Mesh level	Number of cells	Pressure Drop (Pas) and Rel. Error	Total Residence time: inlet to outlet (s) and Rel. Error.
Coarse	34,375	12.69; 34.3%	30.21; 12.4%
Baseline	275,000	9.93; 5.01%	27.24; 1.4%
Fine Mesh	2,200,000	9.45;---	26.87;---

Simulation Results

The flows simulated an ideal mixture of the two reacting species (methanol and biodiesel). Figure 3 shows the path-line trajectory, colored by the residence time and its velocity contours in the outer flow chamber. For the given reactor, the total particle transfer time of 0.75 m/s (Re ≈ 500) was near 30 seconds, as depicted in Figure 4. This value was an order and a half of a magnitude higher than if the flow would have entered the reactor axially along the z direction (nearly 1 m effective length). A lower velocity flow entered the laminar regime leading to an inhomogeneous reactant mixture and two-phase flow of the immiscible reactants. Therefore, avoidance of the mass transfer limitation was only possible at high velocities, which ensured turbulence entry flow. As the mixture was entrained in a swirling and long trajectory, the reactivity was also enhanced, due to the longer residence time. Stoichiometry reactive flow was simulated at a corresponding three moles of methanol to one mole of WCO (mass ratio of 1:8.745).

The kinetic rate of these reactions in kmol/m^3-s is depicted in Figures 5 and 6. In general, the forward reactions dominated the backward reactions and favored the production of biodiesel. The most pronounced was the conversion of the TG to DG and FAME, which took place as soon the reactants were injected into the reactor and maintained within the entire reactor volume. The reverse reaction gained strength as the forward reaction slowed down but remained at an inferior rate compared with the weakest forward values, which were located near the outlet in the top reactor region. The DG reaction seemed to be more pronounced near the outlet region in both forward and backward reaction kinetics and the forward reaction was dominated volu-

metrically by attaining a point nearly half an order of magnitude higher than its backward precursor. The MG reaction was more pronounced in the forward direction and reached its maximum value near midstream. The backward reaction seemed to attack the outlet region, where higher concentrations of GL and FAME were formed.

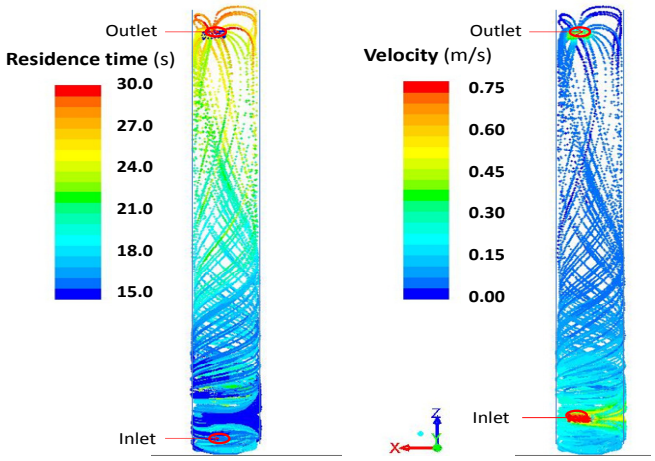

Figure 4. Flow Trajectory Colored by the Resident Time (right) and its Velocty (left)

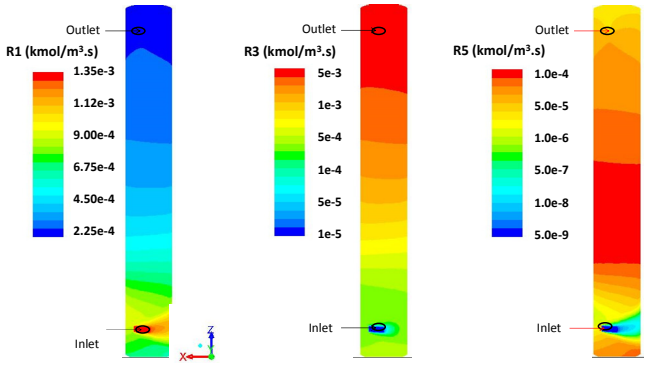

Figure 5. Colored Contours of Reaction Kinetics of the Forward Reactions (TG→DG, DG→MG, MG→GL)

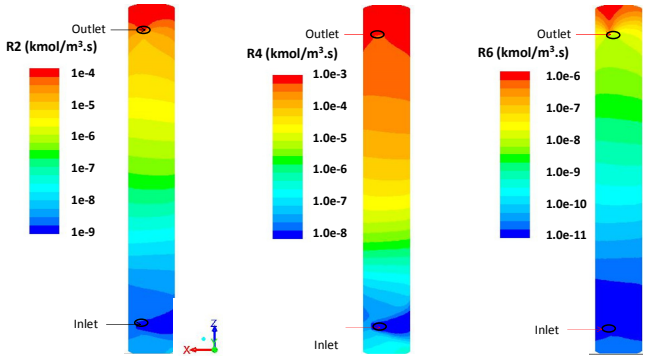

Figure 6. Colored Contours of Reaction Kinetics of the Backward Reactions (DG→TG, MG→DG, GL→MG)

The mole fractions of the six species are depicted in the colored contour plots in Figures 7 and 8 within the inner flow chamber, based on a once-through flow without recirculating the flow and the stoichiometric methanol triglyceride (3:1) ratio. The reaction was far from ideal, as neither the methanol nor the TG were used up and only 0.36 mole fraction was attained in FAME, as depicted in Figure 7. The formation of DG and MG intermediates remained near the reactor outlet at the measurable values presented here, as shown in Figure 8. Glycerol formation in trend along the reactor was appropriate, as depicted in Figure 8, but that was also incomplete, as it only reached 0.175 mole fraction compared to 0.25, which was the ideal conversion value.

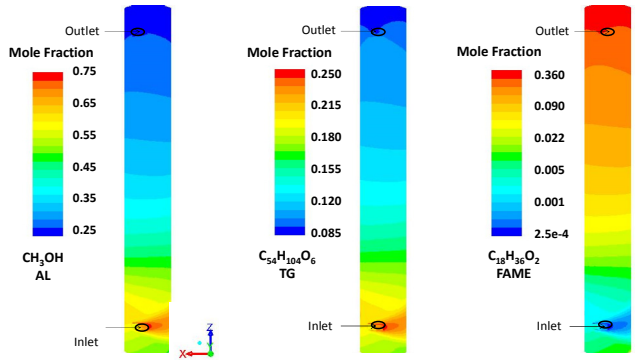

Figure 7. Colored Contours of the mol Fraction Species of Alcohol, TG, and FAME

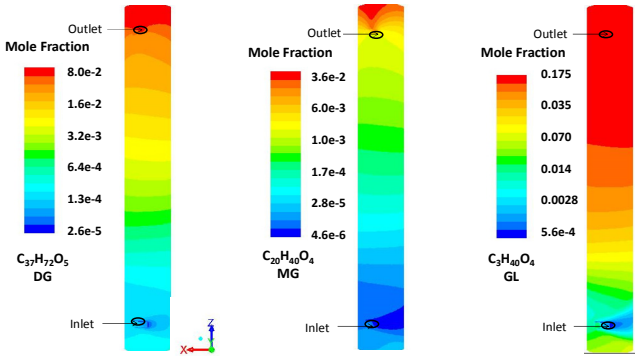

Figure 8. Colored Contours of the mol Fraction Species of DG, MG, and GL

Table 3 summarizes the resulting mole fraction at the reactor outlet. Defining the conversion was the relative change in the molar fraction from the initial concentration or the ideal final concentration. Despite the relatively short once-through pass, the TG conversion and its formation to Ester showed good conversions of 54% and 55%, respectively. Methanol depletion was fairly high at 62% and glycerol conversion mostly lagged at 30% because it occurred at the end of the reaction chain. While the conversion discussed here was incomplete, it was still impartially high,

considering the incoming flow velocity (0.75 m/s) and the short length of the reactor chamber (0.30 m). Batch transesterification processes typically take two hours of processing time with proper stirring, as shown in Figure 1. In the work by Noureddini and Zhu [1], the batch reactor reaction took several minutes to reach roughly 30% conversion and nearly 100 minutes to near completion, while pushing the reaction forward to a higher methanol TG molar ratio of 1:6.

Table 3. Species Molar Fraction under Ideal Conditions

Port and ratio/species	$C_{54}H_{104}O_6$ (WCO)	CH_4O	$C_{18}H_{36}O_2$ (FAME)	$C_3H_9O_2$
inlet	0.25	0.75	0	0
outlet	0.115	0.287	0.340	0.174
conversion*	54%	62%	55%	30%

*TG Conversion = (0.25-0.18)/0.25

Sensitivity Study

Modularity of the annular reactor facilitated sensitivity studies over the different configurations and imposed flow conditions. This included the effects of molar ratio, inlet velocity and temperature, and reactor size. Results for the influence of the increase in the methanol-triglyceride ratio and increase in the inlet flow velocity are shown in Figure 9. As the molar ration increased, the rate of the forward reaction became more dominant over the reverse reaction. This was in accordance with the *le chatelier's principle*, which steers the reaction to consume the additional reactants and towards more favorable equilibrium, thus more methanol and TG conversions took place and resulted in higher FAME yield. The velocity trend, however, was just the opposite, as expected: higher velocity reduced the residence time, thereby lowering the conversion and the biodiesel yield.

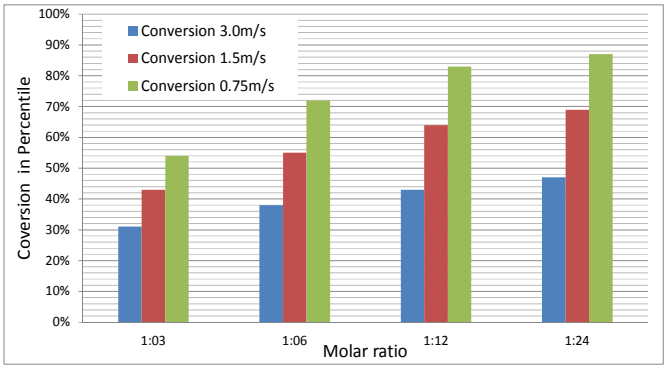

Figure 9. Influence of Molar Ratio and Velocity on the Conversion of TG into FAME

The influence of inlet reactant temperature on the reaction and the yield is depicted in Figure 10. As these reactions were neither strongly exothermic nor endothermic, temperature had little effect. Also, no sensible enthalpy was generated, thus temperature reflected a unified trend and was also less pronounced. It also may appear counter intuitive, but the increase in temperature resulted in slightly lower conversions. This was due to the reversibility in these reactions, which also fed both ways on the temperature according to their activation energy. In general, irreversible reaction rates increased with the increase in temperature, while reversible reactions required an optimal value for the conversion and the desired yield.

Results of the influence of reactor size are depicted in Figure 11. It clearly shows the pronounced effect of reactor length, which dominated the increase in the annular diameter or width. This was due to the effective residence time, which was greatly increased as the reactor was stretched axially and continued to maintain the swirling momentum, which was washed-out for wider-diameter reactors.

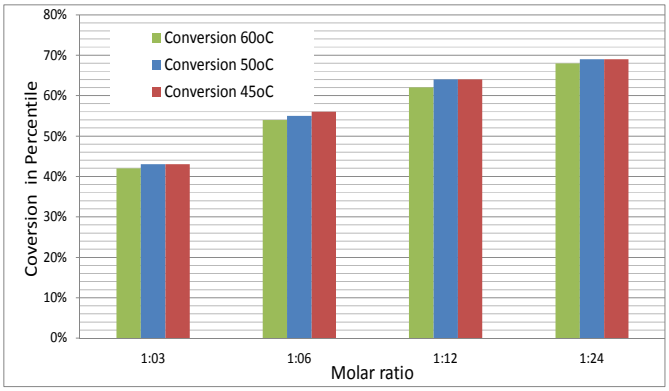

Figure 10. Influence of the Temperature on the Conversion of TG into FAME

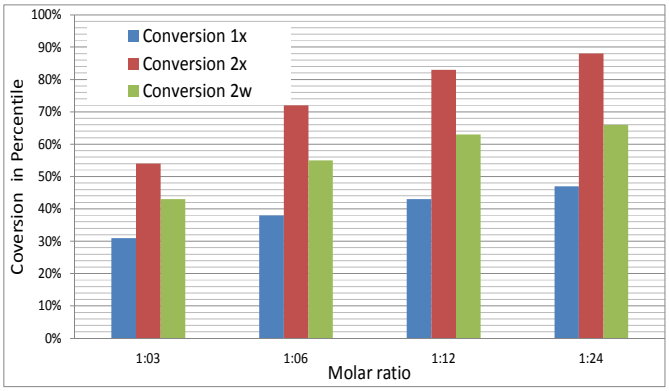

Figure 11. Influence of Reactor Length and Width on the Conversion of TG into FAME under Fixed-inlet Conditions (velocity and temperature)

Conclusions

A high fidelity reactive flow model based on CFD for transesterification was developed. The model was validated against experimental data for alcohol, TG, FAME, and glycerol yields obtained by Analytical GC/MS measurements. The model simulated the flow within a continuous upright annular reactor. Chemical kinetics of eight elementary reactions were used and the flow was assumed to be a single homogenous phase of multiple species representing the two reactants (methanol and triglyceride/lipid), the intermediates (di-glyceride and mono-glyceride), and products (biodiesel/ FAME and glycerol). As the flow was injected circumferentially into the reactor's annular chambers, it followed a swirling trajectory and resulted in an order of magnitude increase in residence time, compared to the tube-length transfer time.

A sensitivity study was conducted and showed that the conversion was also in favor of the excess amount of methanol as well as lower inlet velocity, as this was attributed to the longer reaction time. Temperature influence seemed counterintuitive, due to the inclusion of the reversible reactions; however, the influence was also less significant than other parameters such as molar ration and reactor size. The influence of reactor length was significant in enhancing the conversion of TG and the yield of FAME, as it also contributed to the increase in the reaction time. It was, however, more dominant than the increase in the reactor diameter/ width. This work emphasized the role that each parameter (mixing of the flow, the excess methanol molar amount, geometry configuration, and temperature) played on the conversion of the TG and yield of biodiesel. Results of the simulation emphasized the advantage of efficient mixing of the flow, which reduced the additional methanol concentration, relaxed the downstream separation, and improved process economy.

References

[1] Noureddini, H., & Zhu, D. (1997). Kinetics of Transesterification of Soybean Oil. *Journal of the American Oil Chemists Society, 74*(11), 1457-1463.

[2] Janajreh, I., Almusharekh, M., & Ghenai, C. (2014). Transesterification Process of Waste Cooking Oil: Catalyst Synthesis, Kinetic Study, and Modeling Sensitivity. *International Conference on Sustainable Solid Waste Management*. Athens, Greece.

[3] Janajreh, I., & Al Shrah, M. (2013). Numerical Simulation of Multiple Step Transesterification of Waste Oil in Tubular Reactor. *Journal of Infrastructure Systems (ASCE)*, *22*(2), 127-143.

[4] Janajreh, I., & Al Shrah, M. (2013). Numerical and Experimental Investigation of Downdraft Gasification of Wood Chips. *Energy Conversion and Management*, *65*, 783-792.

[5] Ghenai, C., & Janajreh, I. (2010). CFD Analysis of the Effects of Co-firing Biomass with Coa. *Energy Conversion and Management*, *51*(8), 1694-1701.

[6] Stamenković, O. S., Lazić, M. L., Todorović, Z. B., Veljković, V. B., & Skala, D. U. (2007). The Effect of Agitation Intensity on Alkali-catalyzed Methanolysis of Sunflower Oil. *Bioresource Technology*, *98*, 2688-2699.

[7] Stamenković, O. S., Todorović, Z. B., Lazić, M. L., Veljković, V. B., & Skala, D. U. (2008). Kinetics of Sunflower Oil Methanolysis at Low Temperatures. *Bioresource Technology*, *99*, 1131-1140.

[8] De Boer, K., & Bahri, P. A. (2009). Investigation of Liquid-liquid Two Phase Flow in Biodiesel Production. *7th International Conference on CFD in the Minerals and Process Industries*. Melbourne, Australia.

[9] Boocock, D. G. B., Konar, S. K., Moa, V., Lee, C., & Buligan, S. (1998). Fast Formation of High-purity Methyl Esters from Vegetable Oils. *Journal of the American Oil Chemists Society*, *75*(9), 1167-1172.

[10] Ma, F., Clements, L. D., & Hanna, M. A. (1999). The Effect of Mixing on Transesterification of Beef Tallow. *Bioresource Technology*, *69*(3), 289-293.

[11] Zhou, W., & Boocock, D. G. B. (2006). Phase Behavior of the Base Catalyzed Transesterification of Soybean Oil. *Journal of the American Oil Chemists Society*, *83*(12), 1041-1045.

[12] Freedman, B., Butterfield, R. O., & Pryde, E. H. (1986). Transesterification Kinetics of Soybean Oil. *Journal of Infrastructure Systems (ASCE), 63*(10), 1375-1380.

[13] Laidler, K. (1984). The Development of the Arrhenius Equation. *Journal of Chemical Education, 61*(6), 494.

[14] Fluent Inc, USA. (n.d.). FLUENT, A. 6.3. In *Theory Manual.* Retrieved from http://www.ansys.com/

[15] Narvaez, P. C., Rincon, S. M., Castaneda, L. Z., & Sanchez, F. J. (2008). Determination of Some Physical and Transport Properties of Palm Oil and of Its Methyl Esters. *Latin American Applied Research, 38*, 1-6.

[16] Goodwin, D. R. (1987). Methanol Thermodynamic Properties From 176 to 673 at Pressures to 700 Bar. *Journal of Physical and Chemistry Reference Data*, *16*, 799-891.

[17] Abd Rabu, R., Janajreh, I., & Honnery, D. (2013). Transesterification of Waste Cooking Oil: Process Optimization and Conversion Metrics. *Energy Conversion and Management*, *65*, 764-769.

[18] Abd Rabu, R. (2013). A Thesis on Transesterification of Waste Cooking Oil. Unpublished Master's thesis, Masdar Institute Thesis Data.

Biographies

ISAM JANAJREH is currently an Associate Mechanical Engineering Professor and the director of the waste to energy lab at the Masdar Institute. He is an internationally recognized expert in the area of waste to energy, particularly in the feedstock characterization thermochemical pathways, including gasification and pyrolysis. He has authored more than 80 publications on the subject and presented at over 80 conferences. Dr. Janajreh may be reached at ijanajreh@masdar.ac.ae

MOHAMED ALMUSHARREKH is currently a Ph.D. candidate at Masdar Institute. He earned his M.S. in Mechanical Engineering in 2014 from the Masdar Institute. Mohammed is part of the waste management group headed by Dr. Janajreh and is well-versed in both the experimentation and the simulation of feedstock transesterification. Mr. Almusharrekh may be reached at malmusharrekh@masdar.ac.ae

FROM PIT TO LONG LIE: A FALL-DETECTION ALGORITHM FOR SMARTPHONES

Jeffry A. Pulcifer-Stump, Central Michigan University; Patrick Seeling, Central Michigan University;
Janet L. Koch, Garden City High School; Steven Kettler, Alma High School; Tolga Kaya, Central Michigan University

Abstract

The elderly population of the world continues to grow and, therefore, creates a need for heightened safety measures. Smartphones have been broadly adopted throughout age categories and mobile applications are extremely useful during all types of daily activities; popular applications today not only target recreational uses, but also health tracking and awareness. As men and women over the age of 65 continue to lead active lives, a smartphone application that can detect falling incidents would be very useful. In this research project, a fall-detection application was developed that utilized the acceleration sensor embedded in most mobile devices. The application continuously monitored a person's movement and checked multiple threshold points for a fall impact. Upon impact, phone settings were utilized to communicate with a contact for immediate assistance. The algorithm was successfully implemented, based on the Android platform and its accuracy was tested with eight subjects, who performed 135 fall experiments overall. The measured sensitivity was calculated to be 92% and the measured specificity was 100%.

Introduction and Related Works

The life expectancy of the elderly population, specifically regarding those over the age of 65, continues to increase, thanks to better healthcare technology and available treatment options [1]. The median age increased significantly from 29.5 to 37.2 in the time period from 1960 to 2010, resulting in a 13% increase of the overall elderly population in the U.S. [2]. Therefore, different challenges for the elderly population were introduced to health professionals; these are commonly related to fatigue and incidents in addition to particular illnesses [1], [3], [4]. Among these identified incidents, falls represent one of the most frequent cases that challenges the elderly population. Although it is difficult to define a specific fall event and collect the fall-related reports, various studies have shown that almost 30% of elderly persons fall at least once a year [1], [3], a number that corresponds to more than 10 million elderly in the U.S., who would experience a fall incident within a year [2]. The consequences from falling can be fatal in extreme cases or can lead to serious health problems [1], [3]. Thus, several studies have focused on the causes, consequences, and prevention methods of falls in the elderly population [5-7].

On the other hand, fall detection methods have been explored increasingly during the last two decades. Although several techniques are commonly used to detect a fall incident, studies are often classified into two main groups: accelerometer/gyroscope-based and video-based [8-11]. While accelerometer-based sensors typically have to be carried by the subject on certain locations of the body, such as one's waist, knees, or head, video-based systems suffer from complicated installation procedures and privacy issues that stem from monitoring [8-12]. In this paper, the authors focus on the application of accelerometer-based sensors, which rely on fluctuations in tilt and motion to detect a fall incident.

The majority of accelerometer-based fall detectors rely on 2D or 3D accelerometers and/or gyroscopes. In most cases, the location of the sensing device determines the accuracy of the system. Typically, multiple sensors in different regions of the body were employed in past studies and threshold-based fall detection algorithms were utilized [11], [13-15]. However, these prior works relied on the sensor to be fixed on the waist, which limited the practical use of the system. Commercial sensor boards were also implemented as wearable fall and posture detectors [16], [17]. Although all these systems reached high sensitivity and specificity levels, they either utilized more than one worn sensor or a limited, fixed location of the sensor itself.

Healthcare companies have also been working on efficient and affordable solutions that can help with a fall incident. Most of the products are simply push-button-activated emergency call systems in which users carry a wireless-enabled button, usually worn as a wristband, pendant, or belt clip, requiring button activation when the fall occurs. This signal simply activates a phone call to the healthcare unit and a 2-way telephone communication is established [18-20]. Sensitive floor mats have also been used to track a person and determine if he/she reached the bedroom door while traveling from the bed [21]. Some of the most common systems currently on the market are not phone-based applications and, hence, require a separate sensing system and charge a monthly fee. For example, The Brickhouse Alert Fall Monitoring System relies on a CST (Custom Sensors and Technology) sensing system with 24-hr service. Similarly, Link to Life Fall Detector, CST-L TL, and Prime Medical Alert Amber Select use the same technology [22], [23]. Each relies on a tilt sensor to detect a fall and has a device attached to the person, most commonly a pendant-

style device. One of the most complete devices is Halo Monitoring, which uses an accelerometer sensor similar to this smartphone application (App). It is worn on either the chest or on a belt clip and tracks vital signs, sleeping patterns, and activity levels. Caregivers are notified by user preference but the systems require a landline or computer for tracking.

While there are successful prototypes and products available today, almost all of the systems require significant attention by the user and feature rigorous installation procedures. An affordable and easy-to-use point-of-care solution for fall detection and notification has attracted significant interest. With recent advances in smartphone technology, researchers have been taking advantage of using embedded sensors that are readily available in the phone and have been developing algorithms for fall detection [24-27]. However, most of the algorithms currently require that the phone be kept in a certain orientation. Smartphone Apps such as T3Lab Fall Detection and Fade are free and utilize the accelerometer sensor built into the phone. The T3Lab App, which continuously runs in the background of the smartphone, allows an alarm to sound when a fall is detected. However, the App is very sensitive and can easily be false-triggered. The Fade App requires the user to turn it on and off and has a recovery feature that allows for movement beyond a potential fall impact.

With an aging population that features increasing numbers of adults over the age of 65 that remain independently living in their homes, the need for reliable fall detection is apparent. The convenience of a fall-detection App downloaded to one's smartphone creates a sense of security, as daily activities are carried out without the need for any extra equipment. In this paper, the authors propose an Android-based smartphone as a platform for fall detection. The proposed algorithm is based on multiple thresholds including free fall, impact, orientation cross, and a long lie. The algorithm is able to successfully detect an occurring fall, regardless of the device's orientation and placement. The proposed approach enables reliable and accurate fall incident monitoring through the utilization of smartphone applications by the elderly, which allows them to live independently in their own homes. Furthermore, the authors focus on the materials and methods of the fall-detection algorithm and test protocol.

Fall Detection Algorithm

The proposed algorithm was developed and tested using an HTC 4G LTE smartphone, which utilizes the Android 4.1.1 ("Jelly Bean") operating system. The phone itself represents the current mid-level range of smartphones available to users, with an approximate price of $200 with carrier subventions. The program's offline data storage system allows gathering of data in real time during program execution, with data processing available at a later time. A more in-depth description of the general Android application environment, which also features a simple interface suitable for a target population group and safeguards for real-life scenarios (e.g., setting a personal calling number instead of 911) can be found in the study by Oner et al. [27].

Acceleration data were collected using the device's embedded accelerometer sensor in the fastest sampling mode, which was 50 samples per second (i.e., one sample every 20 ms). The overall geometric average of the acceleration points, A, in each direction was calculated using Equation (1):

$$A = \sqrt{A_x^2 + A_y^2 + A_z^2} \quad (1)$$

where,

A_x = acceleration in the –x direction in m/s^2
A_y = acceleration in the –y direction in m/s^2
A_z = acceleration in the –z direction in m/s^2

An average was calculated for every 25 data points (approximately 500 ms). As a new data point was gathered, the oldest data point was discarded and a new average was calculated, implemented as a moving average. For example, at a point in time, where the *data(i)* was recorded, averaging for *data(i-24)* to *data(i)* was performed. The subsequent average calculation starting with the new data point *i+1* would, hence, consist of the window of values from *data(i-23)* to *data(i+1)*.

Increasing the window size would further smooth the data, but alter the information needed in order to detect the fall with the proposed algorithm. In addition to the sliding window that was used by the fall detection algorithm, a second long-term average was created to increase the algorithm's accuracy. This average value of the overall data was generated and reset every 30 seconds (approximately every 1500 samples).

The fall detection algorithm itself was primarily based on three different thresholds that were evaluated successively in order to identify different fall characteristics, as illustrated in Figure 1. These thresholds were based on the magnitude of changes found in the data over a fixed time period, in combination with phone orientation changes. As an additional precaution, the algorithm also evaluated the impacts that would indicate a fall (i.e., hitting a surface trailed by a period of lying). It should be noted that several of the threshold values that were part of the algorithm were experimentally derived.

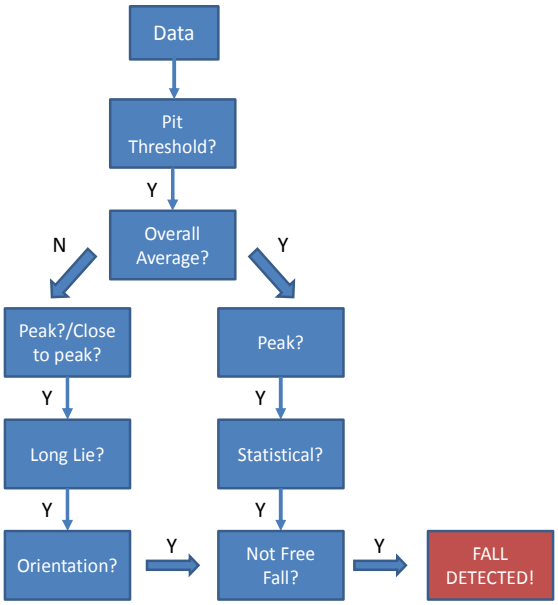

Figure 1. Proposed Algorithm

The initially filtered data were first evaluated against the long-term average data for a continuous decline, which was termed as the "Pit." The Pit was set when the initially filtered data were 3% below the moving average data. The pit threshold was reached if the real-time data went below 75% of the pit's value, which was the first indication of the fall. While checking for this threshold, an assessment of whether the overall average had increased over the overall average at the time the pit was set within one second from the time that the pit threshold was triggered. The algorithm checked for the impact that constituted a typical fall, once both previous triggers had been set. As soon as the person hit the floor surface, acceleration reached a maximum value, due to the impact. If this impact's reported short-term moving average data were larger than 1.925 times the overall average and within 1.2 seconds after the pit threshold was triggered, the algorithm would trigger the Peak detection switch (see Figure 2). These values were experimentally derived and found to be accurate, as described in the results section.

Upon the trigger of the pit being set, a statistical analysis was performed. Once the data were below 75% of the overall average, the first point, *P1*, was stored. Then, the next point, *P2*, was stored when the data exceeded 75% of the data. Once both points were set and the data were greater than the overall average, the standard deviation and the percentage of data between the two stored points was calculated. The standard deviation was calculated using Equation (2):

$$S = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}\left(x_i - \bar{x}\right)^2} \qquad (2)$$

where,

N = sample size

x_i = the i^{th} data value from the combined acceleration value

$\bar{x}$ = the mean of the data gathered between the pit set and pit release points

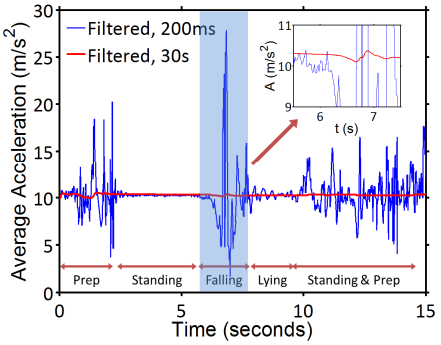

Figure 2. A Representative Fall Event

The percent of data to each point is calculated as the statistical data if it reaches a critical value of 0.12, which triggers the statistical switch. If the algorithm detects an impact, where the statistical threshold is exceeded, a fall detection signal is generated. Another way of ensuring that the fall is detected is to evaluate the orientation of the phone or the sharpness of the free-fall signal. If a person falls, the phone would most likely be pointing in the z direction. The orientation of the phone would be checked to determine if the acceleration in the z direction is close to 1 g, which would trigger the orientation switch. On the other hand, the derivative of the signal is calculated after the pit is set. If the derivative of the acceleration in the z- direction, $\partial A_z/\partial t$, is greater than 11 times the amount of the overall average, the rate switch is triggered.

If an impact after falling is identified or the data are determined to be close to the threshold, the algorithm evaluates whether a long-lie case is detectable. The data are considered close to the upper threshold if it rises above 1.75 times the overall average. In case of a longer time period after a fall, where a person lies, the acceleration magnitude becomes relatively stagnant (within 10% of a deviation from the overall average) for at least four seconds from the time of when the data breach the impact threshold or are close to that threshold (though other time periods could be facilitated in a straightforward manner). During the four seconds of evaluating the data, if the amount of data outside the 10% deviation is less than 20% then the long-lie trigger is activated. If the algorithm detects a time period of stagnant data, indicating a long lie, the orientation, and the rate of change correlates to a fall incident then a fall detection signal is generated (see Figure 2).

An idealistic representation of the typical data obtained and a representative sample of real-world data are illustrated in Figure 2. This test was performed, where the subject threw himself on a low bed, while the phone was in his pants' pocket. The overall average and the moving average data were plotted. Corresponding threshold points were marked as the pit threshold, peak (impact), and long lie. The subset shows the axes crossing. A recovery algorithm was also implemented. Once the fall was detected, and if the original data stayed within 15% of the moving average, the subject was considered to be lying on the floor. However, if more than 35% of the data points for a certain time interval (the default was set to five seconds) were above or below the 15% range then the recovery switch was turned on. Depending on user preference, a fall with a recovery could be reported immediately or saved for future diagnosis.

Figure 3 shows a snapshot of the Android App that was used for the tests. Two sensors were implemented in the algorithm: a fall sensor and a pedometer. The pedometer was used to test the functionality of the phone's embedded accelerometer [27]. Each threshold was prompted at the App screen to monitor the program's progress. When a fall was detected, the App gave a pop-up message that a fall had been detected. The App would be simplified significantly for the end-user.

Figure 3. A Snapshot of the Application at the Time of the Detection of a Fall Event

Human Subject Tests

It was a big challenge to evaluate a fall detector in real-world settings. Since it is ethically wrong to perform tests on elderly subjects, most studies focused on younger adults for experimental trials. Furthermore, there are no standard test scenarios that would be reasonable approximations for real-life simulations. Thankfully, some research groups have focused on developing test standards and evaluation metrics taking these considerations into account. Noury et al. [12], [28] developed a good set of scenarios that would generate both positive and negative fall events.

Simulated real falls were also studied to evaluate different algorithms in terms of the sensitivity and specificity settings [29], [30]. Using these definitions, the specificity (ability to detect only a fall) and the sensitivity (ability to detect a fall) of the fall detection device were evaluated. In evaluating falls, there could be four different decisions that could occur [12], [28]:

1. True positive (*TP*): A fall event occurs and the device detects it.
2. False positive (*FP*): No fall occurs but the device gives a fall detection signal.
3. True negative (*TN*): No fall occurs and the device does not give a fall detection signal.
4. False Negative (*FN*): A fall event occurs but the device does not detect it.

Sensitivity and specificity can then be defined as $TN/(TN+FP)$ and $TP/(TP+FN)$, respectively [12], [28]. Test scenarios adapted from the work by Noury et al. [12], [28] were used to assess the proposed fall detection system's accuracy. The following scenarios were expected to trigger a fall event:

1. Backward fall
   a) ending sitting
   b) ending lying
   c) ending in a lateral position
2. Forward fall
   a) on the knees
   b) with forward arm protection
   c) ending lying flat
   d) with rotation, ending in the lateral right position
   e) with rotation, ending in the lateral left position
3. Lateral fall
   a) ending lying flat to the right
   b) ending lying flat to the left
4. Real-world backward fall
   a) assisted falling without letting the patient know the exact timing

The following scenarios were not expected to trigger a fall event:

1. Syncope
   a) a vertical slipping against a wall finishing in a sitting position
2. Neutral
   a) sit down on a chair then stand
   b) lie down on the bed then rise up
   c) walk a few meters
   d) bend down, catch something on the floor then rise up
   e) cough or sneeze

Real-world backward fall tests were performed using the method implemented by Klenk et al. [30]. In this test, subjects were held inclined backwards about 30 to 40 degrees and allowed to fall onto a mattress on the floor with the instruction of "try not to fall." Subjects did not know when the fall would be initiated. This way, real-world backward fall situations were simulated and the results were used to optimize the fall detection algorithm. Subjects were chosen from healthy adults (a health survey was conducted) ranging from 20 to 40 years old. Consent forms were given to the subjects and experiments were initiated after they signed. The University's Institutional Review Board (IRB) approved the experimental procedures. Each fall scenario was first demonstrated and then the subjects were asked to perform those scenarios by putting the phone into their front pants' pocket. It should be noted that placement of the phone in different locations could yield different results. Real-world backward fall tests were only performed with the subjects that felt comfortable executing the scenario. Each subject was asked to perform the 11 positive and six negative fall executions described above. Data from each fall were recorded at the event and subsequently analyzed.

Eight subjects (six male and two female) evaluated the fall scenarios. The average age of subjects was 26.9 years with a standard deviation of 6.7. The average body weight of the subjects was 162.3 lb. with a standard deviation of 30.5. A total of 87 fall and 48 no-fall events were recorded. A set of fall event examples are given in Figure 4. More than 90% of falls were detected with the trigger of upper and lower thresholds and the statistical switches. Each data collection session started with a few seconds of preparation (clicking the button to start recording, putting the phone into the pocket, and waiting for a few seconds) and was followed by the execution of the event. Exemplary data captured from backward, forward, and lateral fall events are shown in Figures 4(a), 4(b), and 4(c), respectively. Fluctuations after the fall were due to the bouncing from the bed mattress that was positioned on a carpeted floor in order to limit the impact of the falls, as required by the experimental protocol that was approved by the IRB. These fluctuations were more prominent for forward and lateral falls. The steady portion of the data right after the fall was the long lie, where the subjects lie on the mattress for a few seconds. Figure 4 (d) shows the data for bending, where a decline in the moving average occurs that triggers the pit threshold. However, the impact threshold was not exceeded. Similarly, sitting down did not trigger the impact threshold, even though the pit threshold was set; see Figure 4(e). The walking event had the periodical dips and hills that could be associated with pit threshold and impact switched; see Figure 4(f). However, since the impact was not high enough or the moving average did not go below a certain threshold, fall events were not recorded.

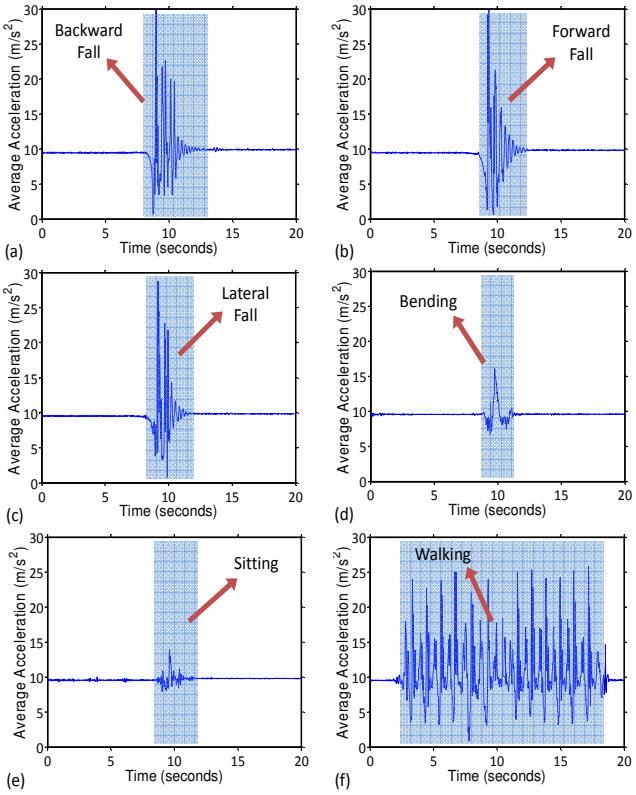

Figure 4. A Representative Fall Event

A summary of the success rate for the tests is given in Table 1. The overall sensitivity was calculated as 92% and the specificity was 100%. Sensitivity is defined as the ability to detect a fall, whereas specificity is defined as only detecting a fall [12], [28]. Therefore, specificity can only be calculated for no-fall events. On the other hand, sensitivity was calculated for actual fall events. Table 1 shows "N/D" for undefined cases.

Table 1. Success Rates

Tests	Success rate	
	Sensitivity	Specificity
Backwards	94%	N/D
Forward	90%	N/D
Lateral	88%	N/D
Bending	N/D	100%
Walking	N/D	100%
Sitting	N/D	100%
Lying	N/D	100%

While being able to avoid false positives would be the result of using a smartphone during activities of a person's daily life, some falls were not detected by the algorithm. Overall, a contributing factor was the mattress that was required by the IRB to limit the impact of the experimental falls for the participants. While softening the fall for the volunteers, it also limited the maximum values that were obtained in the data gathering process from the accelerometer. Secondly, the volunteers themselves likely altered their "natural" fall behavior through event anticipation and appropriate precautious activity as well as for conducting multiple experiments in a single setting. For the lateral fall scenarios, for example the lowest detection rate, the volunteers likely cushioned their fall before turning to their sides, which could have had a significant impact on the sensor readings obtained. It is highly anticipated that employing the algorithm in future studies with more volunteers will significantly increase the already high success rate.

Conclusions and Future Works

In this study, a new fall detection algorithm was proposed that was implemented on an Android-based smartphone. Through various experiments, the algorithm was found to result in a sensitivity of 92% and a specificity of 100%. The algorithm was, in turn, very successful at not detecting false falls that could be derived from daily activities (such as sitting, walking, and bending). On the other hand, a high rate of success was achieved in the detection of real falls, while undetected falls were only the result of experimental limitations. As smart sensors and phone-enabled technologies are on the rise, the authors believe that this proposed work is timely and relevant [31-33]. The authors also believe that the algorithm presented will contribute to the use of smartphones as medical monitoring devices, specifically fall events. Current commercial devices are either very expensive or require a response from the patient in order to invoke the alarm system. Although a simple button can serve this purpose, in the case of a user of these systems losing consciousness or not being willing to seek help, the system will not be functional. Therefore, having a smartphone in their pockets can continuously monitor their movements and the response method can easily be adjusted on the phone. Response methods considered were to call 911 automatically, send a text, email, or initiate a phone call to either a local health unit or a relative.

Due to ethical and practical restrictions, the experiment was limited to a younger population than ultimately targeted. The distribution of the developed algorithm in the form of an Android App will be part of future works. The distributed App will afford crowd-source human subject trials without restrictions and provide further data for refinements of the algorithm. Other future works are directed to the evaluation of different phone positions and daily life scenarios, which could trigger a detected fall.

References

[1] Tideiksaar, R. (1997). *Falling in Old Age*. Springer.

[2] United States Census Bureau, 2010 Census Briefs (2011). Retrieved September 11, 2014, from http://www.census.gov/prod/cen2010/briefs/c2010br-03.pdf

[3] Wild, D., Nayak, U. S., & Isaacs, B. (1981). How Dangerous Are Falls in Old People at Home? *British Medical Journal (Clinical research ed.)*, *282*(6260), 266.

[4] Tinetti, M. E., Speechley, M., & Ginter, S. F. (1988). Risk Factors for Falls among Elderly Persons Living in the Community. *New England Journal of Medicine*, *319*(26), 1701-1707.

[5] Kiel, D. P. (1991). Falls. *Rhode Island Medical Journal*, *74*(2), 75-79.

[6] Salvà, A., Bolíbar, I., Pera, G., & Arias, C. (2004). Incidence and Consequences of Falls among Elderly People Living in the Community. *Medicina Clinica*, *122*(5), 172-176.

[7] Ryynänen, O. P., Kivelä, S. L., Honkanen, R., & Laippala, P. (1991). Falls and Lying Helpless in the Elderly. *Zeitschrift fur Gerontologie*, *25*(4), 278-282.

[8] Luštrek, M., & Kaluža, B. (2009). Fall Detection and Activity Recognition with Machine Learning. *Informatica*, *33*(2), 197-204.

[9] Auvinet, E., Multon, F., Saint-Arnaud, A., Rousseau, J., & Meunier, J. (2011). Fall Detection with Multiple Cameras: An Occlusion-resistant Method Based on 3 -d Silhouette Vertical Distribution. *IEEE Transactions on Information Technology in Biomedicine*, *15* (2), 290-300.

[10] Sixsmith, A., & Johnson, N. (2004). A Smart Sensor

to Detect the Falls of the Elderly. *IEEE Pervasive Computing*, *3*(2), 42-47.

[11] Bourke, A. K., O'Brien, J. V., & Lyons, G. M. (2007). Evaluation of a Threshold-based Tri-axial Accelerometer Fall Detection Algorithm. *Gait & Posture*, *26*(2), 194-199.

[12] Noury, N., Fleury, A., Rumeau, P., Bourke, A. K., Laighin, G. O., Rialle, V., et al. (2007). Fall Detection-Principles and Methods. *29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society,* (pp. 1663-1666).

[13] Bourke, A. K., & Lyons, G. M. (2008). A Threshold-based Fall-detection Algorithm using a Bi-axial Gyroscope Sensor. *Medical Engineering & Physics*, *30* (1), 84-90.

[14] Kangas, M., Vikman, I., Wiklander, J., Lindgren, P., Nyberg, L., & Jämsä, T. (2009). Sensitivity and Specificity of Fall Detection in People Aged 40 Years and Over. *Gait & Posture*, *29*(4), 571-574.

[15] Kangas, M., Konttila, A., Lindgren, P., Winblad, I., & Jämsä, T. (2008). Comparison of Low-complexity Fall Detection Algorithms for Body Attached Accelerometers. *Gait & Posture*, *28*(2), 285-291.

[16] Chen, J., Kwong, K., Chang, D., Luk, J., & Bajcsy, R. (2006). Wearable Sensors for Reliable Fall Detection. *27th Annual International Conference of the Engineering in Medicine and Biology Society,* (pp. 3551-3554).

[17] Li, Q., Stankovic, J. A., Hanson, M. A., Barth, A. T., Lach, J., & Zhou, G. (2009). Accurate, Fast Fall Detection using Gyroscopes and Accelerometer-derived Posture Information. *Sixth International Workshop on Wearable and Implantable Body Sensor Networks,* (pp. 138-143).

[18] Brickhouse Alert Home Systems. (n.d.). Fall Monitoring and Panic Button System. Retrieved September 11, 2014, from http://www.brickhousealert.com/personal-emergency-medical-alarm.html

[19] Alert 1 Medical Alert Systems (n.d.). Retrieved September 11, 2014, from http://www.alert-1.com/system/medical-alert-system/504

[20] Vivago. (n.d.). Alarm Buttons. Retrieved September 11, 2014, from http://www.vivago.com/pro/products/

[21] Betterbuysrus. (n.d.). Cordless Fall Guard Monitoring Systems. Retrieved September 11, 2014, from http://www.betterbuys-r-us.com/Fall-Guard-Monitor-Cordless-Ststem.htm

[22] Pioneer Emergency Medical Alarms. (n.d.). Auto Fall Guard. Retrieved September 11, 2014, from http://www.pioneeremergency.com/product/auto-fall-guard/

[23] Tunstall. (n.d.). Fall Detectors. Retrieved September 11, 2014, from http://www.tunstall.com/solutions/fall-detectors

[24] Dai, J., Bai, X., Yang, Z., Shen, Z., & Xuan, D. (2010). Mobile Phone-based Pervasive Fall Detection. *Personal and Ubiquitous Computing*, *14*(7), 633-643.

[25] Sposaro, F., & Tyson, G. (2009). iFall: An Android Application for Fall Monitoring and Response. *Annual International Conference of the IEEE Engineering in Medicine and Biology Society,* (pp. 6119-6122).

[26] Zhang, T., Wang, J., Liu, P., & Hou, J. (2006). Fall Detection by Embedding an Accelerometer in Cellphone and Using KFD Algorithm. *International Journal of Computer Science and Network Security*, *6*(10), 277-284.

[27] Oner, M., Pulcifer-Stump, J. A., Seeling, P., & Kaya, T. (2012). Towards the Run and Walk Activity Classification through Step Detection-An Android Application. *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society,* (pp. 1980-1983).

[28] Noury, N., Rumeau, P., Bourke, A. K., O'Laighin, G., & Lundy, J. E. (2008). A Proposal for the Classification and Evaluation of Fall Detectors. *IRBM*, *29* (6), 340-349.

[29] Bagalà, F., Klenk, J., Chiari, L., Cappello, A., & Becker, C. (2012). Accelerometer-based Algorithms on Real Falls Database. *Gait & Posture*, *35*, S28.

[30] Klenk, J., Becker, C., Lieken, F., Nicolai, S., Maetzler, W., Alt, W., et al. (2011). Comparison of Acceleration Signals of Simulated and Real-world Backward Falls. *Medical Engineering & Physics*, *33*(3), 368-373.

[31] White, T. S., Petsch, K., & Kaya, T. (2015). Evaluation of the Silicon-on-Glass Microfabrication Process for MEMS Accelerometers. *International Journal of Modern Engineering*, *15*(1), 36-45.

[32] Winter, C., Soylemez, N., Trivedi, J., Pickens, M., Craig, C., & Vaidyanathan, V. (2006). Design of a Sensor Based Smart Sprinkler System. *International Journal of Modern Engineering*, *7*(1).

[33] Lee, Y., Zhu, X., Pandya, A., & Hsu, S. (2010). Interactive Data Visualization and Analysis for Mobile-Phone Performance Evaluation. *International Journal of Modern Engineering, 11*(1), 82-94.

Biographies

JEFFRY A. PULCIFER-STUMP is a recent graduate from the Computer Science Undergraduate program of Central Michigan University. Mr. Pulcifer-Stump may be reached at pulci1ja@cmich.edu

PATRICK SEELING is currently an Assistant Professor

in the Department of Computer Science at Central Michigan University. He received his Dipl.-Ing. Degree in Industrial Engineering and Management from the Technical University of Berlin (Berlin, Germany) in 2002 and his Ph.D. in Electrical Engineering from Arizona State University in 2005. He was a Faculty Research Associate and Associated Faculty member with the Department of Electrical Engineering at Arizona State University from 2005 to 2007. From 2008 to 2011, he was an Assistant Professor in the Department of Computing and New Media Technologies at the University of Wisconsin-Stevens Point. He currently leads the Distributed Internetworked Systems and Content (DISC) lab at Central Michigan University, reachable at DISClab.org. His research interests are in networking (with a focus on multimedia and energy optimizations), distributed and mobile systems, assistive technologies, and computer-mediated education, which are typically embodied in smart-device implementations. Dr. Seeling may be reached at seeli1p@cmich.edu

JANET L. KOCH is currently a high school science teacher at Garden City High School. She has been involved in the National Science Foundation granted Research Experiences for Teachers program at Central Michigan University during the summers of 2012 and 2013. Ms. Koch served as the president of the CMU chapter of the National Science Teachers Association for several years. Ms. Koch may be reached at koch1j@cmich.edu

STEVEN KETTLER is currently a high school science and mathematics teacher at Alma High School. He has been involved in the National Science Foundation granted Research Experiences for Teachers program at Central Michigan University during the summers of 2012 and 2013. Mr. Kettler has been actively involved in Science Olympiads and robotics projects as a mentor. Mr. Kettler may be reached at skettler@almaschools.net

TOLGA KAYA is currently an Assistant Professor at the School of Engineering and Technology of Central Michigan University (CMU). Prior to joining CMU, Dr. Kaya worked as a post-doctorate associate at Yale University's Department of Electrical Engineering. He received his Ph.D. degree in Electronics Engineering in 2007 from Istanbul Technical University, where he also worked as a graduate and teaching assistant for eight years. Dr. Kaya also worked as a technology consultant at Brigthwell Corporation in 2006 and a senior circuit designer at Microelectronics Circuit Design in Turkey from 1999 to 2006. His research interests include the design, fabrication, and testing of micro-scale sensors and microfluidic-based cell manipulation. Dr. Kaya is also exploring efficient ways of incorporating education technology and active learning strategies in his classrooms. Dr. Kaya may be reached at kaya2t@cmich.edu

OPTIMALITY IDENTIFICATION IN EPIDEMIOLOGY USING NEURAL NETWORKS

Samir Talssi, Hassan II University; Noura Yousfi, Hassan II University

Abstract

An optimal control methodology of drug therapy can produce a drug dosing strategy to reduce the cost of treatment, making patient healing faster or stabilizing his/her case. Some models of the controlled treatment of diseases such as HIV infection were proposed in different works. In this study, a neural network with a single hidden layer was used to identify the optimality of drug treatment of HIV infection of CD4+T-cells. The choice of a single hidden layer was justified. The training set was generated by a numerical simulation model of treatment through a semi-implicit finite difference method. The back-propagation algorithm was introduced to perform and demonstrate the effectiveness of this neural approach. The result of this study was a neural network able to mimic and predict the interaction between HIV and the immune system under drug therapy.

Introduction

The human immunodeficiency virus (HIV) is a lentivirus that attacks the immune system and causes the acquired immune deficiency syndrome AIDS. Under a VIH infection, the immune system weakens progressively to become paralyzed against any fatal opportunistic infections. CD4+T-cells are a type of white blood cells that play an essential role in the human immune system; unfortunately, they are an HIV replication key. Indeed, HIV is a retrovirus that requires cells from a "host" in order to replicate itself. CD4+T -cells are the hosts of HIV. HIV attaches to the CD4+T-cells, allowing the virus to get into the CD4+T-cells and then infect and damage them. As CD4+T-cells operate, the immune system becomes less effective and the human body becomes vulnerable to infections and diseases [1].

Some antiretroviral drugs are available nowadays that help the immune system in the prevention of HIV infection, but it is not possible to cure it. The use of reverse transcriptase inhibitors is one of the chemotherapies that opposes the conversion of RNA of the virus to DNA—reverse transcription—and leads to the minimization of viral population. Today, some antiretroviral drugs exist that help the immune system in the prevention of HIV infection. The use of reverse transcriptase inhibitors is one of the chemotherapies that opposes the conversion of RNA to DNA of the virus and leads to the minimization of the viral population, allowing the number of CD4+T-cells to remain high. Another antiretroviral drug is the protease inhibitor that prevents the generation of viruses from the infected CD4+T Actively-cells. This kind of therapy has prolonged life of the infected Individuals; however, the cost of the drug for this kind of treatment is high [2] and the side effects can be potentially severe. Therefore, the question is whether there exists a drug treatment schedule that can sustain a low viral load and a healthy immune system, while minimizing the amount of drugs used. This suggests an optimal control approach to treatment scheduling.

Significant efforts have been evidenced in the research literature on modeling of physiological and immunological response of HIV in individuals [3], [4]. Some models consider the dynamics of the CD4+T-cells and virus populations as well as the effects of drug therapy [5-7]; there are also some models that include an intracellular delay [8], [9]. Two optimal treatments of an HIV infection model were proposed [10]: one optimal control represents the efficiency of drug treatment in inhibiting viral production, while the other controls the efficiency in preventing new infections. Neural network with a single hidden layer was used to identify the optimality of this drug treatment.

Identification using neural network modeling permits a nonlinear multivariable process. The result of this identification seems a good compromise for systems that are difficult to model by conventional methods. A neural network has the ability to learn sophisticated nonlinear relationships and provide an ideal means of modeling complicated nonlinear systems [11-13]. In this paper, the authors describe the controlled mathematical model of HIV infection with two control terms [10], present the neural network model for identification, and discuss the identification result.

Mathematical Model of HIV Infection with Two Control Terms

An optimal therapy was presented by Hattaf and Yousfi [10] for minimizing the cost of treatment, reducing the viral load, and improving the immune response. The model incorporated the cure of infected cells [14]. Two controls that measure the efficiency of reverse transcriptase and protease inhibitors were used and two types of virus particles were included into the model to examine the effect of protease inhibitors.

Let (x), (y), (v_I), and (v_{NI}) denote the concentration in mm^{-3} of uninfected CD4+T-cells, infected cells, infectious virus, and noninfectious virus, respectively. The model is given by the nonlinear system of differential equations, presented in Equation (1):

$$\begin{cases} \dot{x} = s - dx - (1 - u_1(t))\beta v_I x + ry \\ \dot{y} = (1 - u_1(t))\beta v_I x - (a + r)y \\ \dot{v}_I = (1 - u_2(t))ky + \mu v_I \\ \dot{v}_{NI} = u_2(t)ky - \mu v_{NI} \end{cases} \quad (1)$$

where,

s = uninfected CD4+T-cells production rate
d = uninfected CD4+T-cells death rate
β = rate CD4+T-cells become infected by virus
a = death rate of infected CD4+T-cells

and where, $u_1(t)$ and $u_2(t)$ and are the control functions; $u_1(t)$ represents the efficiency of drug therapy in blocking new infection, so that infection rate in the presence of drug is $(1 - u_1(t))\beta$; $u_2(t)$ represents the efficiency of drug therapy in inhibiting viral production; and, $(1 - u_2(t))k$ is the virion production rate under therapy.

Free virus production rate by infected cells at a rate k and cleared at a rate μ. Some infected cells may also revert to the uninfected state by loss of all cccDNA from their nucleus at a rate r. The existence of an optimal control pair (u_1,u_2) was demonstrated by Hattaf and Yousfi [10], [14] using a result by Fleming and Rishel [15].

Neural Network for Identification

Several types of neural networks exist; the difference between them is their size; and, in the case of layered neural network architectures, the number of layers in a network, the number of nodes per layer, and the number of connections. Neural networks can identify a non-linear correspondence $y = F(x)$ during a learning phase; they can also learn how to correctly associate output patterns, y, to input patterns, x. It was proven by Fleming and Rishel [15] and Brattka [17], based on classic mathematical result of Kolmogorov [16], that for any continuous mapping there must exist in a three-layer neural network having an input layer with n processing elements, a hidden layer with (2n + 1) processing elements, and an output layer with m processing elements that exactly implements f.

$$f : [0,1]^n \subset R^n \to R^m$$

This result gave hope that neural networks would turn out to be able to approximate any function arising in the real world. Cybenko [18] demonstrated that a three-layer neural network can approximate any continuous and multivariate function. Hornik et al. [19] continued this work. Therefore, a feed-forward neural network with one hidden layer (three layers) was implemented in order to identify the system (see Figure 1). The number of processing elements per layers was fixed using the Hecht-Nielsen theorem [20].

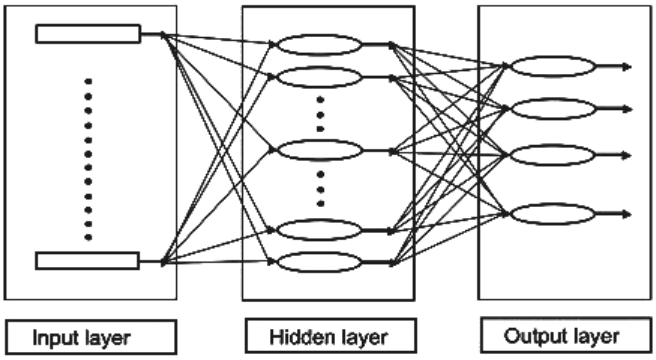

Figure 1. One Hidden-layer Feed-forward Network Architecture with Sigmoidal Activities

Neural Approach

The goal of this approach was to identify, using neural networks, the solutions of a controlled system (see Figure 1) with variables x, y, v_I, and v_{NI}. The semi-implicit finite difference method was used by Hattaf and Yousfi [10] as a numerical algorithm and found interesting results, where were reproduced to construct a learning-basis network. A neural network was used for identification according to Figure 2.

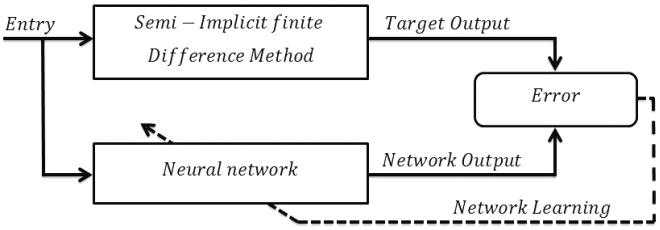

Figure 2. Identification Process using a Neural Network

The following steps describe the method used in this current study:

a) Generation and normalization of training data
b) Network generation
c) Network learning using the back-propagation method and training data
d) Validation of results

Generation and Normalization of Training Data

Network learning using standardized data generated efficient results. Different methods can be used for data normalization, as shown by Shanker et al. [21], who represented two effective methods: linear transformation and statistical standards. Linear transformation was used in this current approach, with a slight modification. The principle of this method was simple: first, the minimum and maximum data were determined and then the data were converted using Equation (2):

$$x \leftarrow \big((x - min)\big) / \big((max - min)\big) \tag{2}$$

The following changes in Equation (3) allow values between *HI* (= 0.9) and *LO* (= 0.1):

$$x \leftarrow \big((x - min) / (max - \min)\big) * (HI - LO) + LO \tag{3}$$

Network Generation

The neural network used here was a feed-forward network with one hidden layer of sigmoidal nonlinearities. The results by Hecht-Nielson [20] were used to fix the number of neurons in the hidden layer and adjust this number in the simulation. The hidden layer was a full interlayer connected to both input and output layers, each neuron in the hidden layer was connected with all entries, and each network output was connected with all hidden neurons. Figure 3 shows a feed-forward network with one hidden layer of sigmoidal nonlinearities. In the input layer, the neurons have a linear activation function. In the output layer, the activations are sigmoid. All thresholds, θ_i, and weights, w, of the network were randomly initialized to small values between 0 and 1.

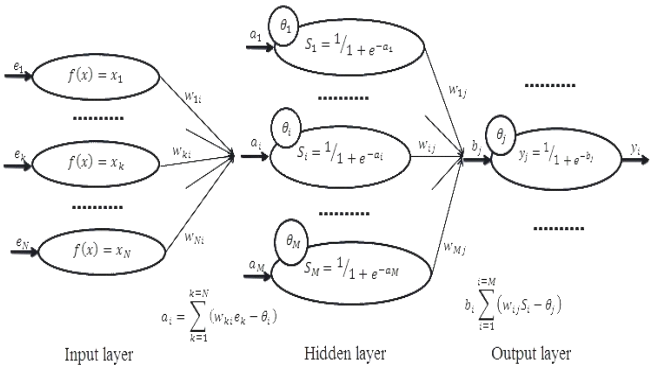

Figure 3. Propagation Process in a Feed-forward Neural Network

Network Learning using a Backpropagation Method and Training Data

The network's learning was a stage that posed difficulties most of the time. The back-propagation algorithm was used in many cases to perform the network's learning [22]; information or entries was propagated through the network from the input layer to the output layer. The modified back-propagation method presented by Vogl et al. [22] was employed in the training process to ensure and accelerate the algorithm's convergence. In this version of back-propagation, the network weights are not updated after each pattern is presented. Rather, the weights are updated only after all training input patterns have been presented.

Changes in weight were calculated using Equation (4):

$$\Delta w_{ji}(t+1) = \eta \delta_{pj} a_{pi} + \alpha . \Delta w_{ji}(t-1) \tag{4}$$

where, w_{ji} is the weight on the connection from input i to neuron j; η is the learning rate; α is the momentum factor; and, δ is the error signal. The hidden neuron error signal δ_{pj} is given by Equation (5):

$$\delta_{pj} = a_{pj}(1 - a_{pj}) \sum_{x} \delta_{px} w_{xj} \tag{5}$$

where, δ_{px} is the signal error in the next neuron, x; w_{xj} is the weight of the connection from hidden neuron j to the next neuron, x; and, δ_{pj} is the signal error in the output neuron, j, given by Equation (6):

$$\delta_{pj} = a_{pj}(1 - a_{pj})(t_{pj} - a_{pj}) \tag{6}$$

where, t_{pj} is the target value of output neuron, j, for pattern p, and a_{pj} is the actual output value of output neuron, j, for pattern p.

Figure 4 shows a schematic of the back-propagation algorithm used. The total error for all training patterns was calculated at any iteration. If the total error was reduced then the learning rate, η, would be multiplied by a factor $\varphi > 1$ for the next iteration. Otherwise, if the error was more than a few percent above the previous value, all changes to the weights would be rejected, η would be multiplied by a factor $\phi < 1$, and α would be reduced to zero [22].

Simulation Study

The therapy can be described by four variables: x, y, v_I, and v_{NI}, denoting the concentration in mm^{-3} of uninfected CD4+T-Cells, infected cells, infectious virus, and noninfectious virus, respectively. First, a workforce of 2500 data

points (x, y, v_I, and v_{NI}) was generated using the semi-implicit finite difference method [10] and normalized. Of those, 2000 were used for learning the network and 500 to test network performance training. Finally, 1000 new data points were generated using the obtained network in order to validate and compare the identification results. The back-propagation algorithm was applied several times (in the best case, 25 times). Each time, the network settings were randomly reinitialized to give a new stimulus to the calculations. Figures 5-8 show the evolution of the concentration in mm^{-3} under controlled therapy.

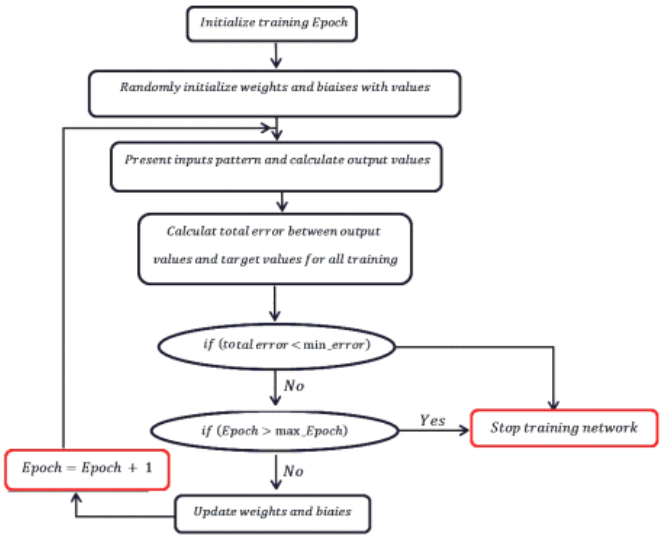

Figure 4. Backpropagation Algorithm

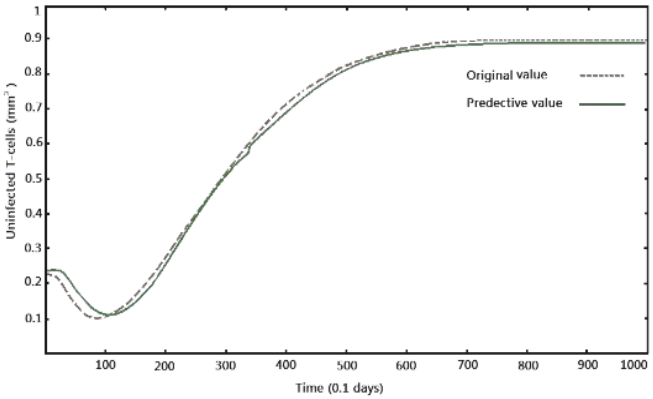

Figure 5. Uninfected CD4+T-cells

The graphs show the predicted and originals values of x, y, v_I, and v_{NI} obtained with the feed-forward neural network. Graphically, it is clear that the predictive values have the same behavior as the original; the small perturbation noticed was generated at a random initialization of the network parameters.

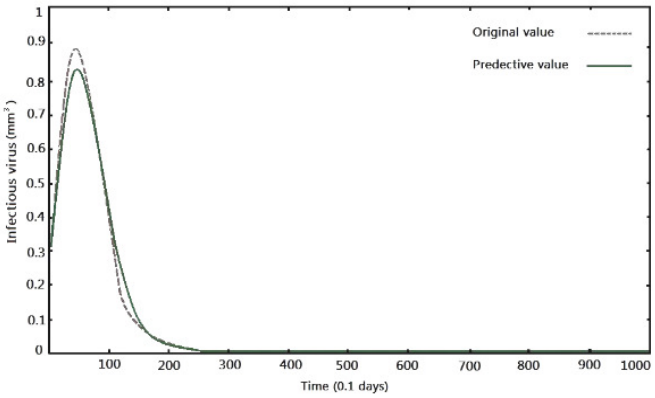

Figure 6. Infected Cells

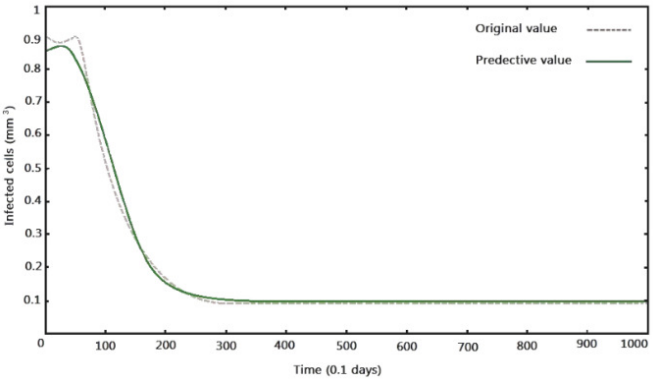

Figure 7. Infectious Virus

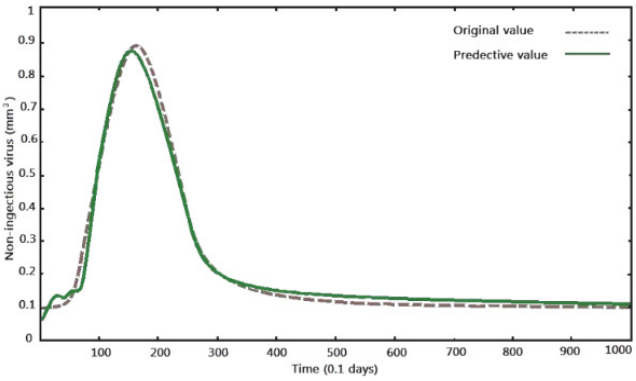

Figure 8. Non-infectious Virus

Conclusion

The life expectancy of patients infected with HIV has increased dramatically with the advent of antiviral treatment. In this paper, the authors presented a development of a new approach for the identification and prediction of optimal treatment of HIV drugs in numerical models of the en-

vironment using neural network techniques. The training data were generated using the data from Hattaf and Yousfi [10], and were normalized to converge back-propagation. This approach provides an independent perspective mechanism of ideas specific to the therapy and allows rapid understanding of the behavior of HIV during treatment; this speed will help us to make good decisions with the aim of reducing the cost of treatment and avoiding critical phases of the disease. There is no effective cure for HIV infection; existing treatments can only block the evolution of the virus in the body and maintain the balance between virus and defense system.

Among the identified risk factors, tobacco consumption is of paramount importance because it can be modified at the individual level. It turns out, however, that its control is difficult because of the existence of these psychosocial factors in persons living with HIV. This constraint, tobacco, will be added to the constraints of the approach in the authors' next study on the control treatment of HIV stress from smoking.

References

[1] Bushman, F. D., Nabel, G. J., & Swanstrom, R. (Eds.). (2011). *HIV: From Biology to Prevention and Treatment*. Cold Spring Harbor Laboratory Press.

[2] Gilks, C. F., Crowley, S., Ekpini, R., Gove, S., Perriens, J., Souteyrand, Y., et al. (2006). The WHO Public-Health Approach to Antiretroviral Treatment against HIV in Resource-Limited Settings. *The Lancet*, *368*(9534), 505-510.

[3] Culshaw, R. V. (2004). Review of HIV Models: The Role of the Natural Immune Response and Implication for Treatment. *Journal of Biological Systems, 12* (2), 123-136.

[4] Callaway, D. S., & Perelson, A. S. (2002). HIV-1 Infection and Low Steady State Viral Loads. *Bulletin of Mathematical Biology*, *64*(1), 29-64.

[5] Perelson, A. S., & Nelson, P. W. (1999). Mathematical Analysis of HIV-1 Dynamics in Vivo. *SIAM Review*, *41*(1), 3-44.

[6] Andersen, R. M., & May, R. M. (1988). Epidemiological Parameters of HIV Transmission. *Nature*, *333* (6173), 514-519.

[7] Bonhoeffer, S., May, R. M., Shaw, G. M., & Nowak, M. A. (1997). Virus Dynamics and Drug Therapy. *Proceedings of the National Academy of Sciences*, *94* (13), 6971-6976.

[8] Dumrongpokaphan, T., Lenbury, Y., Ouncharoen, R., & Xu, Y. (2007). An Intracellular Delay-Differential Equation Model of the HIV Infection and Immune Control. *Mathematical Modelling of Natural Phenomena*, *2*(1), 84-112.

[9] Nelson, P. W., Murray, J. D., Perelson, A. S. (2000). A Model of HIV-1 Pathogenesis that Includes an Intracellular Delay. *Mathematical Biosciences, 163* (2), 201-215.

[10] Hattaf, K., & Yousfi, N. (2012). Two Optimal Treatments of HIV Infection Model. *World Journal of Modelling and Simulation*, *8*(1), 27-35.

[11] Chu, S. R., Shoureshi, R., & Tenorio, M. (1990). Neural Networks for System Identification. *Control Systems Magazine, 10*(3), 31-35.

[12] Barron, A. (1991). Universal Approximation Bounds for Superposition of a Sigmoidal Function. *IEEE Transactions on Information Theory, 39*(3), 930-945.

[13] Chen, S., Billings, S. A., & Grant, P. M. (1990). Non -Linear System Identification Using Neural Networks. *International Journal of Control*, *51*(6), 1191-1214.

[14] Hattaf, K., & Yousfi, N. (2011). Dynamics of HIV Infection Model with Therapy and Cure Rate. *International Journal of Tomography & Simulation*, *15* (W11), 74-80.

[15] Fleming, W. H., & Rishel, R. W. (1976). Deterministic and Stochastic Optimal Control. *Bulletin of the American Mathematical Society*, *82*, 869-870.

[16] Kůrková, V. (1992). Kolmogorov's Theorem and Multilayer Neural Networks. *Neural Networks*, *5*(3), 501-506.

[17] Brattka, V. (2003). A Computable Kolmogorov Superposition Theorem. *Computability and Complexity in Analysis*, *272*, 7-22.

[18] Cybenko, G. (1989). Approximation by Superpositions of a Sigmoidal Function. *Mathematics of Control, Signals, and Systems*, *2*(4), 303-314.

[19] Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer Feedforward Networks are Universal Approximators. *Neural Networks*, *2*(5), 359-366.

[20] Hecht-Nielsen, R. (1989). Theory of the Backpropagation Neural Network. *International Joint Conference on Neural Networks,* (pp. 593-605).

[21] Shanker, M., Hu, M. Y., & Hung, M. S. (1996). Effect of Data Standardization on Neural Network Training. *Omega*, *24*(4), 385-397.

[22] Vogl, T. P., Mangis, J. K., Rigler, A. K., Zink, W. T., & Alkon, D. L. (1988). Accelerating the Convergence of the Back-Propagation Method. *Biological Cybernetics*, *59*(4-5), 257-263.

Biographies

SAMIR TALSSI is a Professor at higher institutes of applied engineering IGA-Casablanca. His interest area is artificial intelligence and its applications in epidemiology.

Dr. Talssi was a Member in the High Council Education in Morocco and has over 10 years of experience as manager and educator. Professor Talssi may be reached at samirtalssi@gmail.com

NOURA YOUSFI is currently a Professor in the Department of Mathematics and Computer Science in Mohammedia Hassan II University. She is director of several doctoral student research projects. Dr. Yousfi has over 20 years' of experience as a scientist, manager, and teacher. She is also a biomathematics specialist. Dr. Yousfi may be reached at nourayousfi@hotmail.com

TUNING OF CASCADE CONTROL STRUCTURES SUBJECT TO PROCESS CONSTRAINTS

Constantine Tzouanas, Clear Lake High School; Minh Le, University of Houston-Downtown;
Vassilios Tzouanas, University of Houston-Downtown

Abstract

In this study, the authors developed a method for the simultaneous tuning of proportional-integral-derivative (PID) controllers in cascade control structures, based on the concept of co-simulation. The uniqueness of the proposed method is the simultaneous design/tuning of the controllers in a cascade structure. The method is applicable to series as well as parallel cascade control. The controllers can be simultaneously tuned, while a performance criterion is optimized subject to process constraints. Important constraints include a controlled variable, a manipulated variable, and rate-of-change constraints [1]. The tuning method is independent of the PID controller type and accommodates controllers with proportional and/or derivative action on the error or process variable. It accounts for robustness in response to modeling errors by including constraints on the maximum sensitivity function. The performance of the proposed method was demonstrated using simulation and experimental runs.

Introduction

Cascade control has been extensively used in the process industries and is one of the most successful methods for enhancing single-loop control performance [2]. The first reported use of cascade control structures dates back to 1956 [3]. That application, a distillation column tray temperature controller (primary) cascaded to a steam flow controller (secondary), was termed as what is now known as series cascade control. A disturbance, or pressure drop on the steam supply line, affects the steam flow (secondary variable), which then affects the tray temperature (primary variable). However, there are applications where the disturbance affects both secondary and primary variables [4]. An example is the simultaneous (parallel) effect of a feed composition change on a tray temperature and product composition in a distillation column. In a two-level cascade control structure, there is an additional controlled variable and an additional controller, compared to a single-loop application. In spite of the added complexity and instrumentation costs, it may be beneficial to use cascade control. Krishnaswamy et al. [5] developed recommendations for when to use cascade control. Using first-order plus time-delay (FOPTD) models, they developed charts to show the extent of improvement possible with cascade control relative to single-loop control.

The design/tuning of the two controllers (typically of the PID type) is done in a sequential manner. With the primary (or master) control loop open, the secondary (or slave) controller is tuned. Then, with the secondary loop closed, the primary controller is tuned. By following this approach, a number of PID tuning methods have been used, ranging from the famous Ziegler and Nichols method to the most recent ones, which are based on the internal model control principle. Setpoint tracking and/or disturbance rejection could be defined as control objectives. For instance, parallel cascade control structures have been designed for disturbance rejection by Yu [6]. The design of parallel control structures capable of minimizing the interactions between the primary and secondary loops has also been considered by Brambilla and Semino [7] and Lee et al. [8]. From a servo-performance viewpoint, PID controllers have been designed by Lee and Park [9] to obtain a desired closed-loop response for the primary variable. Recently, Azar and Serrano [10] proposed an internal model control plus proportional/integral/derivative (IMC-PID) tuning procedure, which is based on gain and phase margin specifications of the two loops.

All of the previous methods consider the tuning issue in a sequential manner and are analytic in nature. Even though gain and margin specifications are considered in some cases, none of the aforementioned methods explicitly considers process constraints on the controlled and manipulated variables and the range of tuning parameters. Process constraints are important and meaningful to the practicing engineer and board specialists, who are responsible for the safe, reliable, and profitable operation of a process. In recent work, tuning of PID controllers was done subject to process constraints and the success of such an approach was demonstrated using simulation [1] and experimental studies [11]. In this current study, the issue of simultaneous design of controllers, subject to process constraints in a cascade structure, was addressed. The resulting tuning method was not analytic. Rather, it was model-based, independent of the type of PID controller, and could accommodate controllers with proportional and/or derivative action on the control error or process variable, and calculate tuning parameters by optimizing a performance criterion subject to constraints.

Block Diagram of Cascade Control Structures

The block diagram representation of a series cascade control structure is shown in Figure 1. The secondary controlled variable affects the primary controlled variable. The objective of this structure is to eliminate the effect of the disturbance, which affects the secondary controlled variable before its effect is propagated to the primary variable. Typically, the secondary loop is tuned for disturbance rejection and the primary loop for setpoint tracking [6]. Figure 2 shows a block diagram representation of a parallel control structure. In such a structure, the manipulated variable and the disturbance affect the primary and secondary controlled variables through parallel paths [12]. For instance, the reflux (manipulated variable) affects both the tray temperature (secondary) and product composition (primary) controlled variables in the typical distillation column process example [4]. Likewise, through parallel paths, feed composition (disturbance) affects both the tray temperature and product composition.

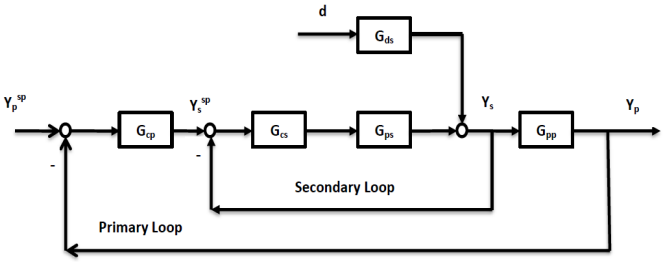

Figure 1. Schematic of a Series Cascade Structure

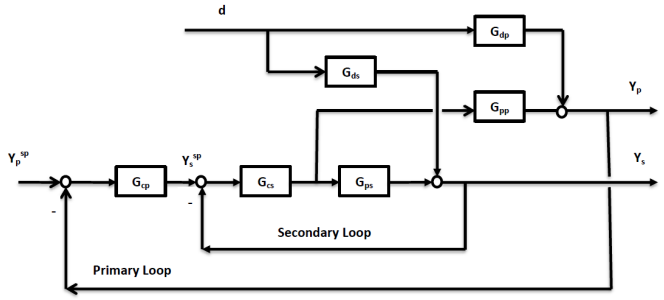

Figure 2. Schematic of a Parallel Cascade Structure

In this current study, tuning of the secondary and primary controllers was done simultaneously, subject to constraints. In practice, robustness in response to modeling errors is of significant importance. The sensitivity function is a good measure for assessing closed-loop system robustness [13]. Thus, the sensitivity functions of the two loops will be among the constraints considered. Typically, tuning is done for maximum sensitivity values less than 2.0. Depending on what structure (series or parallel) is involved, the sensitivity functions are given by Equations (1)-(5). The controllers of the cascade structures are of the proportional/integral/derivative (PID) form. The tuning parameters of these two controllers (up to a total of six) are calculated by optimizing a performance criterion, subject to constraints based on the concept of co-simulation.

Secondary Loop (series or parallel):

$$M_{s,s} = \max_{0<\omega<\infty} \left| \frac{1}{1 + G_{cs}(i\omega) \cdot G_{ps}(i\omega)} \right| \tag{1}$$

Primary Loop (series):

$$M_{s,p} = \max_{0<\omega<\infty} \left| \frac{1}{1 + G_{cp}(i\omega) \cdot G_{pp}(i\omega) \cdot G_{sp}(i\omega)} \right| \tag{2}$$

where,

$$G_{sp} = \frac{G_{cs} \cdot G_{ps}}{1 + G_{cs} \cdot G_{ps}} \tag{3}$$

Primary Loop (parallel):

$$M_{s,p} = \max_{0<\omega<\infty} \left| \frac{1}{1 + G_{cp}(i\omega) \cdot G_{pp}(i\omega) \cdot G_{sp}(i\omega)} \right| \tag{4}$$

where,

$$G_{sp}(i\omega) = \frac{G_{cs}}{1 + G_{cs} \cdot G_{ps}} \tag{5}$$

A Co-simulation Approach

For the sake of simplicity and by considering just a single loop, evaluation of control performance and tuning of the PID controller are done simultaneously over a desired time horizon, while a performance criterion is optimized and process constraints, meaningful to the practicing engineer, are satisfied. In simple terms, the co-simulation approach follows the steps shown in Figure 3. A process model is controlled using a PID controller. Initial tuning parameters are provided and the closed-loop system performance is evaluated over a desired time horizon for the expected setpoint and/or load changes. For instance, the Integral Absolute Error (IAE) could be used to measure system performance. Then, using an optimization algorithm and process constraints, the tuning parameters of the PID controller are updated and the process is repeated until the performance

criterion reaches a minimum value. Thus, by following this co-simulation-based approach, tuning parameters are obtained, which help ensure that process constraints will be satisfied prior to implementing them in the real process. This co-simulation approach has been used for the tuning of single-loop PID controllers subject to constraints [1], [11]. The same approach is now extended to simultaneously tune PID controllers in a cascade control structure.

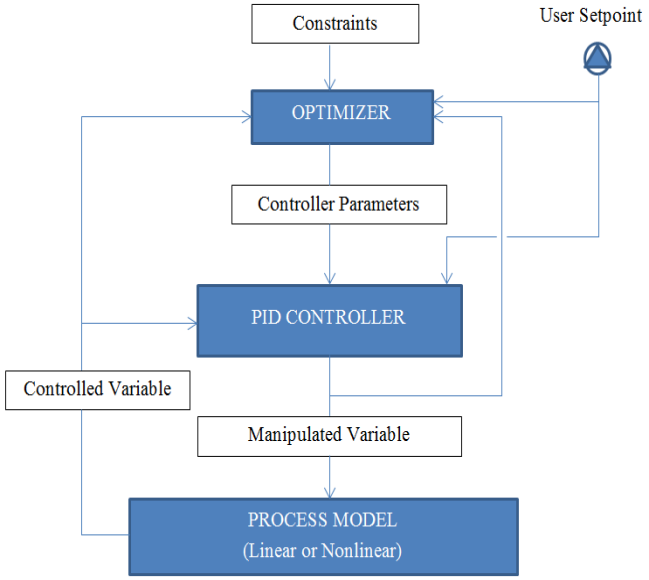

Figure 3. PID Controller Design Using Co-simulation

Tuning of a Cascade Structure

The objective is not to replace but complement previously proposed analytic tuning approaches by directly considering important process constraints such as manipulated variable size and rate-of-change constraints, as well as constraints on the controlled variable and the tuning parameters. Manipulated variable constraints are meant to reflect the inherent capacity of the process to cause and/or reject change in the case of setpoint response or load disturbances, respectively. Controlled variable constraints are meant to meet desired objectives such as product quality constraints. Constraints on the tuning parameters are meant to limit the search space by utilizing experiential knowledge or analytical knowledge [1], [11].

Estimation of the tuning parameters is done by optimizing either the integral absolute error (IAE) or the integral square error (ISE) over a desired time horizon, t_f. Other performance measures could also be considered. The performance criteria and the various constraints are mentioned next. Since there are two loops involved, there is a duplicate set of process constraints, one for the secondary loop and another one for the primary loop. The performance criteria used in this study, integral absolute error (IAE) and integral square error (ISE), are given by Equations (6) and (7), respectively:

$$IAE = \int_0^{t_f} |e(t)| dt \tag{6}$$

$$ISE = \int_0^{t_f} e(t)^2 dt \tag{7}$$

The controlled variable constraints are positional only and are shown in Equation (8):

$$y_{LL} \le y(t) \le y_{UL} \tag{8}$$

Positional and rate of change constraints for the manipulated variables were considered. They are shown in Equations (9) and (10):

$$m_{LL} \le m(t) \le m_{UL} \tag{9}$$

$$\Delta m_{LL} \le \Delta m(t) \le \Delta m_{UL} \tag{10}$$

To limit the search space for appropriate tuning parameters and speed up convergence of the optimization algorithm, tuning parameter constraints were considered, as shown in Equations (11) through (13):

$$K_{C,LL} \le K_C \le K_{C,UL} \tag{11}$$

$$\tau_{I,LL} \le \tau_I \le \tau_{I,UL} \tag{12}$$

$$\tau_{D,LL} \le \tau_D \le \tau_{D,UL} \tag{13}$$

To account for robustness against modeling errors, maximum sensitivity function constraints were considered, as shown in Equation (14):

$$M_S \le M_{S,UL} \tag{14}$$

The subscripts $_{LL}$ and $_{UL}$ stand for lower limit and upper limit constraints.

From an implementation viewpoint, the process models for the secondary and primary models were assumed to be first-order plus time delay (FOPTD). The Euler integration method was used to solve the resulting ordinary differential equations. For computer implementation, the discrete version of a velocity-type PID controller was used. The computer platform was Microsoft Excel and the Solver function with the GRG (gradient) optimization algorithm was used. The upper/lower limits for controlled and manipulated variables were process specific. The upper/lower limits for tuning parameters were set as multiple/fraction of the tuning parameters obtained using the IMC method [14].

Simulation Results

In this section, a number of simulation examples illustrate the proposed method. Series and parallel cascade structures were considered.

Example 1: Series Cascade

This process was also discussed by Lee and Park [9]. The FOPTD transfer functions are given by Equations (15)-(17):

$$G_{pm(}(s) = \frac{10.2e^{-61.71s}}{66.49s+1} \quad (15)$$

$$G_{ps}(s) = \frac{2.988e^{-3.66s}}{13.98s+1} \quad (16)$$

$$G_{ds}(s) = \frac{1}{100s+1} \quad (17)$$

PID controllers were used in both loops. A number of constraints were imposed on the controlled and manipulated variables, as shown in Table 1. Table 2 shows the range of the tuning parameters. The max/min values were set as a multiple/fraction of the tuning parameters obtained using the IMC methodology. The maximum sensitivity was set at 1.8 for both loops.

Table 1. Process Constraints

Variable	Lower Limit (LL)	Min Value	Max Value	Upper Limit (UL)
Primary Controlled Variable	-0.20	-0.05	0.08	0.20
Primary Manipulated Variable	-0.14	-0.01	0.00	0.14
Primary Manipulated Variable ROC	-0.14	0.00	0.00	0.14
Secondary Controlled Variable	-0.14	-0.01	0.05	0.14
Secondary Manipulated Variable	-0.19	-0.33	0.00	0.19
Secondary Manipulated Variable ROC	-0.19	-0.33	0.01	0.19
Primary Loop M_s	-	-	1.06	1.8
Secondary Loop M_s	-	-	1.80	1.8

Table 2. Tuning Parameter Range

Loop	Parameter	Min	Max
Primary	K_C	0.01	0.25
	τ_i	13.30	332.45
	τ_D	1.05	42.15
Secondary	K_C	0.05	1.31
	τ_i	2.66	66.40
	τ_D	0.08	3.22

Using the previously described methodology, the primary and secondary PID controllers were tuned for a unit step change in the disturbance. PID controllers were used in both loops to obtain a direct comparison with the results obtained by Lee and Park [9]. The tuning parameters for the primary loop were: K_c=0.08, τ_I=68.56, and τ_D=3.40. For the secondary loop, they were: K_c=0.93, τ_I=4.53, and τ_D=2.02. Figure 4 shows a comparison of the system response to that obtained using the tuning parameters reported by Lee and Park [9]. These tuning parameters were: K_c=0.09, τ_I=90.53, and τ_D=18.2 (primary loop) and K_c=0.883, τ_I=14.5, and τ_D=1.117 (secondary loop).

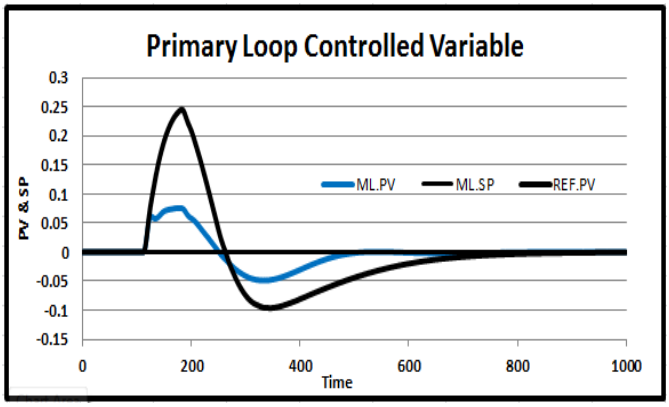

Figure 4. Example 1: Primary-loop Response to a Disturbance Step Change

In Figure 4, the blue line (ML.PV) corresponds to the new tuning method, while the black line (REF.PV) corresponds to the tuning method reported by Lee and Park [9]. By simultaneously tuning the secondary and primary loops using the proposed method, the control system was capable of rejecting the disturbance; its performance also compared very well to that obtained using the tuning reported by Lee and Park [9]. However, when tuning a cascade control structure, the behavior of both loops must be considered. Figures 5 and 6 show how the manipulated variable of the primary

loop (setpoint to secondary) changes and how the secondary loop is capable of following this setpoint, respectively. Based on Figures 5 and 6, the proposed method required less movement in the manipulated variable and yielded a smoother response relative to the tuning reported by Lee and Park [9].

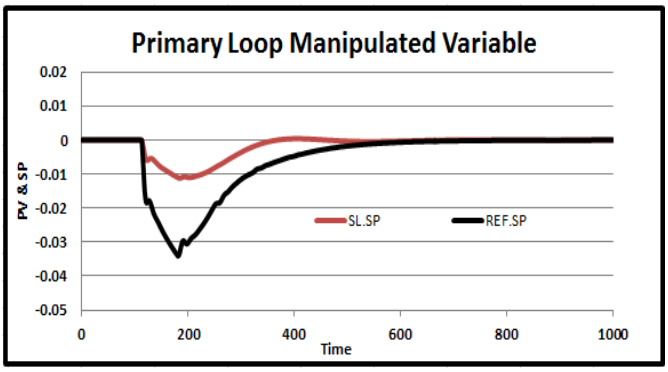

Figure 5. Example 1: Primary-loop Manipulated Variable Response to a Disturbance Step Change

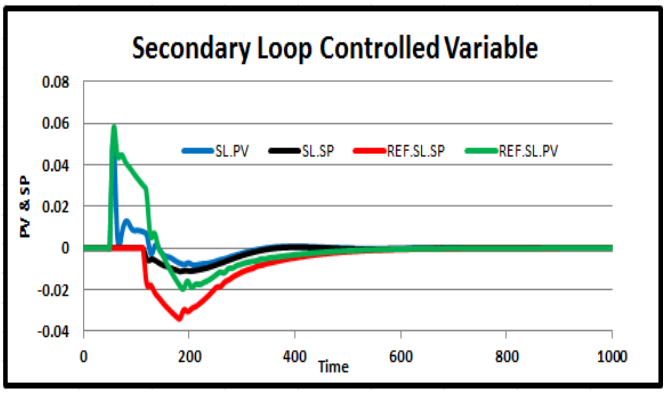

Figure 6. Example 1: Secondary-loop Performance in Response to a Disturbance Step Change

The proposed method allowed tuning for disturbance rejection, setpoint tracking, or a combination of the two. For instance, the primary loop could be tuned for setpoint tracking and the secondary for disturbance rejection. Figure 7 shows the system response for a unit setpoint change in the primary loop and a unit disturbance change that enters through the secondary loop. In Figure 7, the setpoint was changed at time equal to 10, while the disturbance was changed at time equal to 500. The red line was the manipulated variable of the primary loop (setpoint to secondary) and is plotted against the secondary axis. The tuning parameters for the primary loop were: K_c=0.071, τ_I=87.36, and τ_D=1.053. For the secondary loop, they were: K_c=0.415, τ_I=5.635, and τ_D=0.08. Again, the system performance was quite satisfactory.

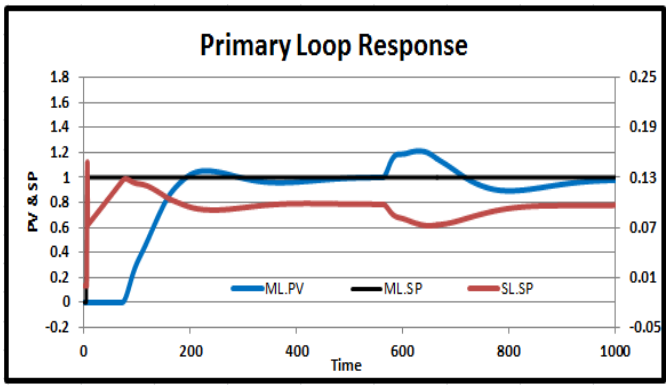

Figure 7. Example 1: Primary-loop Response to Setpoint and Disturbance Step Changes

Example 2: Parallel Cascade

This process was also discussed by Yu [6]. The FOPTD transfer functions are given by Equations (18) and (19):

$$G_{pm}(s) = G_{dm}(s) = \frac{1 \cdot e^{-4s}}{20 \cdot s + 1} \tag{18}$$

$$G_{ps}(s) = G_{ds}(s) = \frac{1}{10 \cdot s + 1} \tag{19}$$

By optimizing the IAE subject to constraints, the tuning parameters for the primary loop were: K_c=0.429, τ_I=4.00, and τ_D=0.091. For the secondary loop, they were: K_c=4.373, τ_I=2.320, and τ_D=0.0. The active process constraint was the maximum rate of change of the secondary controller output, which was set equal to 6.0. Figure 8 shows the system response for a unit step change in setpoint at time equal 10 and a simultaneous unit step change in both disturbances at time equal to 125. This closed-loop performance (blue line, ML.PV) compared favorably with that reported by Yu [6] and reproduced in the same figure (red line, REF.PV). Figure 9 shows the manipulated variable of the primary control loop. The blue line corresponds to that obtained using the tuning reported by Yu [6]. Again, the new method required less movement in the manipulated variable and yielded a smoother response.

Experimental Results

The proposed tuning method was applied to the twin water tank experimental system reported by Tzouanas and Tzouanas [11]. Referring to Figure 10, the objective was to control the water level in Tank 2 using a series cascade configuration. The secondary loop controlled the level in Tank 1 by adjusting the voltage to the pump.

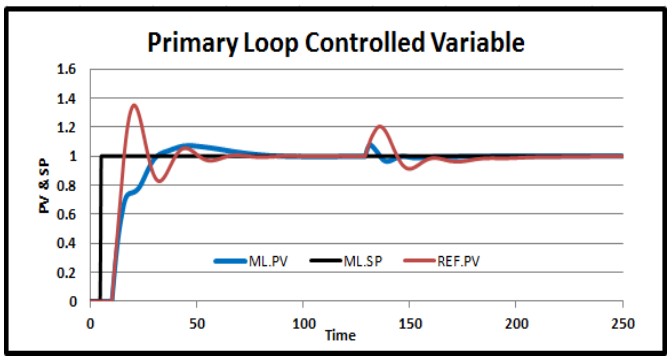

Figure 8. Example 2: Primary-loop Controlled Variable Response to Setpoint and Disturbance Step Changes

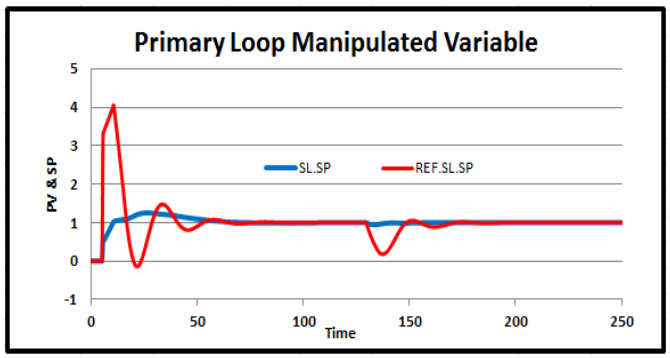

Figure 9. Example 2: Primary-loop Manipulated Variable Response to Setpoint and Disturbance Step Changes

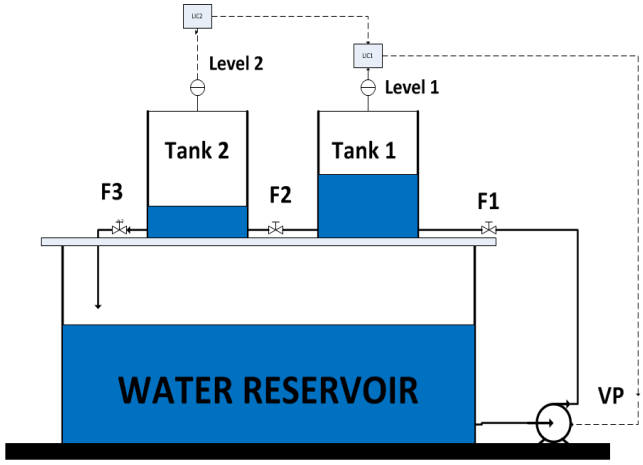

Figure 10. Experimental Twin Water Tank Process

The transfer functions were obtained using step tests and are as follows:

Primary loop (tank 2 level):

$$G_{pm}(s) = \frac{0.66e^{-7.08s}}{7.68s+1} \tag{20}$$

Secondary loop (tank 1 level):

$$G_{ps}(s) = \frac{11.8e^{-4.22s}}{48.12s+1} \tag{21}$$

The units of levels are in cm, while those of voltage in VDC. The time is in seconds. The system was tuned for level 2 setpoint tracking. The tuning parameters of the primary and secondary PI-only controllers were obtained by minimizing the IAE. The active constraint was the maximum rate of change for the pump voltage, which was set at 0.17 VDC/s. The resulting tuning parameters for the primary loop were: K_c=0.2015 and τ_I=8.575. For the secondary loop, they were: K_c=0.145 and τ_I=22.603. The simulated closed-loop performance of the two loops is shown in Figure 11 (primary loop) and Figure 12 (secondary loop).

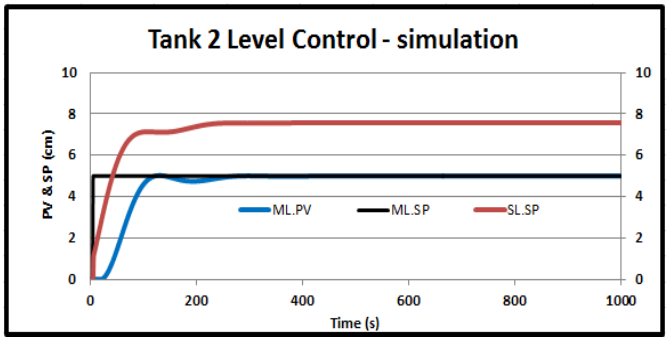

Figure 11. Primary-loop Response to Setpoint Step Changes

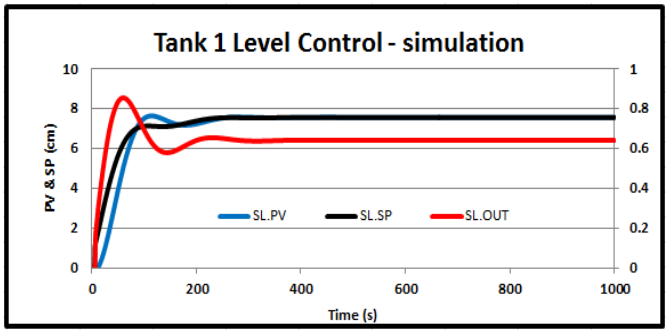

Figure 12. Secondary-loop Response to Setpoint Step Changes

The performance of the PI controllers was also tested on the experimental system. The closed-loop behavior of the primary loop (tank 2 level) is shown in Figure 13. The size of the setpoint step change was 3 cm. The controlled variable (L2.PV) followed the setpoint (L2.SP) very closely. Figure 14 shows the performance of the secondary loop. Again, the controlled variable (L1.PV) closely followed the setpoint (L1.SP), which was determined by the primary controller. The manipulated variable [(L1.OUT (VDC)] was the voltage to the pump (plotted against the secondary axis). Based on the simulation and experimental runs for this twin

tank system, the proposed tuning method resulted in very good closed-loop performance; but, most importantly, it allowed for the simultaneous tuning of the controllers and accounted for process constraints. Accounting for process constraints is important because it has a significant impact on the long-term reliability of real-life control systems.

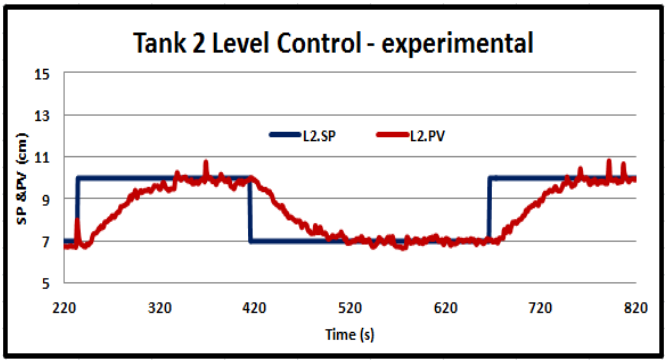

Figure 13. Primary-loop Response to Setpoint Step Changes (experimental data)

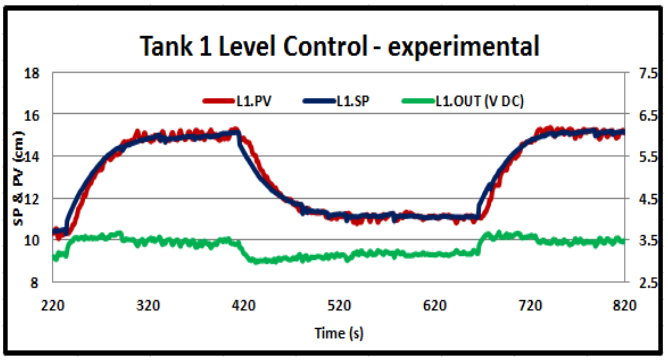

Figure 14. Secondary-loop Response to Setpoint Step Changes (experimental data)

Conclusions

In this study, a method was developed for the simultaneous tuning of cascade control structures subject to constraints. The simultaneous design of such controllers is a unique contribution of this work. The new method followed the concept of co-simulation. The controller design was done by optimizing a performance criterion subject to process, robustness, and tuning parameter range constraints. The new method was applied to series as well parallel cascade control and its performance was demonstrated using simulation and experimental runs. Based on simulation runs, it could be concluded that the performance of the proposed method compared favorably to that of other widely used tuning methods.

References

[1] Tzouanas, C., & Tzouanas, V. (2013). Optimal Tuning of PID Controllers Subject to Process Constraints. *International Journal of Modern Engineering, 14*(1), 70-80.

[2] Marlin, T. E. (2000*). Process Control.* (2nd ed.). McGraw-Hill.

[3] Franks, R. G., & Worley, C. W. (1956). Quantitative Analysis of Cascade Control. *Industrial and Engineering Chemistry*, *48*(6), 1074–1079.

[4] Luyben, W. L. (1973). Parallel Cascade Control. *Industrial & Engineering Chemical Fundamentals, 12* (4), 463-467.

[5] Krishnaswamy, P. R., Rangaiah, G. P., Jha, R. K., & Deshpande, P. B. (1990). When to Use Cascade Control. *Industrial & Engineering Chemistry Research*, *29*(10), 2163-2166.

[6] Yu, C. C. (1988). Design of Parallel Cascade Control for Disturbance Rejection. *AIChE Journal*, *34*(11), 1833-1838.

[7] Brambilla, A., & Semino, D. (1992). Nonlinear Filter in Cascade Control Schemes. *Industrial & Engineering Chemistry Research*, *31*(12), 2694-2699.

[8] Lee, Y., Skliar, M., & Lee, M. (2006). Analytical Method of PID Controller Design for Parallel Cascade Control. *Journal of Process Control, 16*(8), 809-818.

[9] Lee, Y., & Park, S. (1998). PID Controller Tuning to Obtain Desired Closed Loop Responses for Cascade Control Systems. *Industrial & Engineering Chemistry Research*, *37*(5), 1859-1865.

[10] Azar, A. T., & Serrano, F. E. (2014). Robust IMC-PID Tuning for Cascade Control Systems with Gain and Phase Margin Specifications. *Neural Computing & Applications, 25*(5), 983-995.

[11] Tzouanas, C., & Tzouanas, V. (2014). Optimal Tuning of PI Controllers subject to Process Constraints – Experimental Evaluation. *International Journal of Modern Engineering, 14*(2), 15-21.

[12] Semino, D., & Brambilla, A. (1996). An Efficient Structure for Parallel Cascade Control. *Industrial & Engineering Chemistry Research*, *35*(6), 1845-1852.

[13] Åström, K. J., Panagopoulos, H., & Hägglund, T. (1998). Design of PI Controllers Based on Non-Convex Optimization. *Automatica, 34*(5), 585–601.

[14] Rivera, D. E., Morari, M., & Skogestad, S. (1986). Internal Model Control: PID Controller Design. *Industrial & Engineering Chemistry Process Design and Development, 25*(1), 252–265.

Biographies

CONSTANTINE TZOUANAS is currently a senior at Clear Lake High School, Houston, TX. His interests focus on economics and engineering.

MINH LE is a graduate of the Control and Instrumentation Engineering Technology program at the University of Houston – Downtown. His interests are in the areas of process control and instrumentation.

VASSILIOS TZOUANAS is an Assistant Professor of Process Control and Instrumentation at the University of Houston – Downtown. Dr. Tzouanas earned a Diploma in Chemical Engineering from Aristotle University, the Master of Science degree in Chemical Engineering/Process Control from the University of Alberta, and the Doctor of Philosophy degree in Chemical Engineering/Process Control from Lehigh University. His professional experience includes technical and management positions with major operating companies. His research interests focus on process control systems, process modeling, and simulation. He is a member of AIChE and ASEE. Dr. Tzouanas may be reached at tzouanasv@uhd.edu

MARKOV DECISION PROCESS-BASED STRUCTURAL HEALTH MONITORING MODEL FOR CIVIL INFRASTRUCTURE SYSTEMS USING ENERGY-EFFICIENT WIRELESS SMART-SENSOR NETWORKS

Abayomi M. Ajofoyinbo, Texas Southern University; David O. Olowokere, Texas Southern University

Abstract

In this paper, the authors present a structural health monitoring (SHM) model for civil infrastructure systems (CIS) using energy-efficient wireless smart-sensor networks (WSSN), based on the Markov Decision Process (MDP). To further improve energy efficiency, the operation of the SHM system was characterized as a discounted, continuous-time MDP consisting of three states: *sleep* (or *sensing*), *processing*, and *transmitting* (or *receiving*). Moreover, to monitor the point in time of a state's transition, this continuous-time MDP model was transformed into a discrete-time MDP model. Optimal results, in terms of the average optimal value of the sum of the expected total discounted energy consumption, was subsequently achieved. The conclusion was reached through an optimal analysis that this MDP-based SHM model would increase energy efficiency and may prolong the life of SHM applications.

Introduction

Structural health monitoring systems for civil infrastructure systems can be designed to collect real-time data on a continuous basis or at discrete points in time. By analyzing collected data with respect to dynamic response to excitations, the future performance of civil infrastructure systems can be predicted. A wireless smart-sensor network (WSSN) may be described as a wireless network consisting of independent smart-sensor nodes deployed spatially to monitor environmental or physical conditions. Since wireless smart-sensor nodes are typically battery-powered, energy conservation is vital for long-term monitoring of civil infrastructure systems. Even though several techniques for power harvesting have been presented in the literature, power consumption must still be managed to prolong the life of SHM systems. A wireless smart-sensor (WSS) node is made up of subsystems for power, computing, communication, and the physical sensors. The communication subsystem consumes more energy than the other subsystems. An important energy-efficient design goal is to reduce communications among wireless sensor nodes.

Researchers have investigated challenges of SHM systems with different research goals. For instance, Sim et al. [1] developed an efficient means of autonomous, long-term monitoring of cable tension using Imote2 Smart Sensors. A new displacement sensing system was developed by Park et al. [2] by incorporating wireless sensor technology into a multi-metric data-based algorithm. He et al. [3] presented an inter-encoding multi-swarm particle-swarm optimization (IMPSO) algorithm to optimally place multi-axial sensors onto large structures for modal identification. Moreover, Park et al. [4] presented a WSSN-based decentralized processing scheme for damage detection of building structures. This paper adopted a Damaged Induced Inter-story Deflection (DI-ID) model proposed by Koo et al. [5] and extended for use in a decentralized computing environment in the WSSN. Li et al. [6] developed a post-sensing time synchronization scheme to reduce the latency of data collection, while maintaining high synchronization accuracy of the collected data. A multi-hop bulk-data transfer approach was also implemented in order to achieve high data throughput. In an interesting contribution, Sim et al. [7] and Sim et al. [8] investigated a decentralized random decrement technique (RDT) for efficient data aggregation and system identification in wireless smart-sensor networks. A new decentralized data aggregation approach for system identification, based on the random decrement technique (RDT), was also presented in those papers.

Rice et al. [9] developed an opened-source framework for structural health monitoring using the design principles of service-oriented architecture. This technique provides a suite of services implementing key middleware infrastructure necessary to provide high-quality sensor data. Moreover, Jo et al. [10] presented the development of a hybrid wireless smart-sensor network to achieve a full-scale SHM system for civil infrastructure monitoring. This hybrid system provides power harvesting for sensor nodes, improved sensing application, decentralized data aggregation, and environmental monitoring. Cho et al. [11] analyzed the data collected from the 2nd Jindo Bridge, a cable-stayed bridge in Korea, and found that the WSSN-based SHM systems performed effectively in giving direct access to the physical

status of the cable-stayed bridge. According to Rice et al. [12] and Jang et al. [13], SHM systems that are based on wireless smart sensors offered many advantages over the traditional wired-sensor systems. An example of the features that can be used to characterize a civil infrastructure system is the acceleration time history measured by accelerometers mounted on the structure. An SHM system is a decision system that has a knowledge base at the back-end and wired/wireless sensors at the front-end.

Markov Decision Process (MDP) models offer sequential decision making when outcomes are uncertain. In MDP models, choosing an action in a current state generates rewards and determines the state at the next decision epoch through a transition probability function. Moreover, a policy is a prescription for which action to choose at every future decision epoch, and decision epochs are points in time when a system executes action(s). The Semi-Markov Decision Process (SMDP) is an example of a continuous-time MDP. In SMDP models, decision epochs follow each state transition and the times between decision epochs are exponentially distributed. Researchers have characterized systems process as MDP or SMDP and investigated optimal policies in different problem domains. For example, Ajofoyinbo and Olowokere [14] investigated the problem of energy utilization in structural health monitoring and control of civil infrastructures. Ni et al. [15] investigated an optimal call admission control policy for non-priority schemes and reserved channel schemes in wireless mobile networks using an SMDP model.

Related Research

In an interesting contribution, Chen and Casciati [16] presented the development of a high-performance wireless data acquisition system designed for sensors used in structural health monitoring applications. A scheme for bridge construction safety and health monitoring based on wireless sensor networks was presented by Li and Yuan [17]. The use of an energy harvester to power wireless nodes for purposes of structural health monitoring was investigated by Cahill et al. [18]. Furthermore, Gu et al. [19] presented energy synchronized communications as a transparent middleware between the network layer and MAC (media access control) layer that controls the amount and timing of the radio frequency activity at receiving nodes. A comprehensive review of the recent literature on the various possible energy harvesting technologies from ambient environment for the wireless sensor networks was presented by Zhou et al. [20]. Fu et al. [21] investigated the problem of finding locations of sensor nodes, where there can be reliable diagnosis of the health of a structure, while consuming minimum energy during data collection. Recent technology developments in the field of bridge health monitoring using wireless sensor networks was introduced by Zhou and Yi [22]. Moreover, a power saving and energy optimization technique for wireless sensor networks was presented by Sendra et al. [23].

In another contribution, Zhou et al. [24] presented a method for processing data on-board, thereby saving significant power through reduced wireless transmission time. In an interesting contribution, Nagayama et al. [25] presented two complementary, reliable multi-hop communication solutions for monitoring civil infrastructure, namely: the general-purpose multi-hop, and the single-sink multi-hop. The problem of data congestion and excessive use of power while transmitting large amounts of data generated by large arrays of wireless smart sensors, due to limited bandwidth of wireless communication, was investigated by Jo et al. [26]. This paper presented decentralized approaches for reducing the information on individual nodes using the sensor node's onboard computational ability.

In another interesting contribution, Nagayama et al. [27] presented the use of a limited number of high-sensitivity reference sensors to decrease the effect of noise in the estimation of cross-correlation functions. Basic properties of vibration-based structural health monitoring are that changes of structural properties will affect the vibrational response of the structure. The two aspects of vibration-based damage detection are: identification and extraction of vibration-related features, and correlation of features to the structural properties [28]. Two different states were compared: the baseline (or "undamaged") state, and all subsequent states. All subsequent states were compared to the baseline state in order to ascertain the state of health of a civil infrastructure system. Such comparisons can disclose changes in the features of the civil infrastructure systems, which may reveal the occurrence, location, type, and relative severity of damage.

The development of an Imote2 sensor board with onboard signal processing designed for the demands of Structural Health Monitoring (SHM) applications was investigated by Rice and Spencer [29]. The paper presented the design and testing of the SHM Accelerometer (SHM-A) board, which interfaces with the Imote2 wireless sensor platform. Moreover, the design, development, and large-scale deployment and testing of scalable wireless sensor networks for structural health monitoring was presented by Pakzad et al. [30]. This new approach introduced key technology innovations, which included maximizing the effective network bandwidth with a large number of communication hops, and reliable command dissemination and data transfer for high-frequency sampling. Two discriminating factors in structur-

al health monitoring (SHM) were identified by Kim et al. [31], namely: the time-scale of the change and the severity of the change. This paper described a platform for indirect detection of structural state through measurement and interpretation of ambient vibration and strong motion.

Motivation and Problem Definition

Motivation for the Research

The research for this paper was motivated by the following factors:

a) The need to determine dissimilarity between values of baseline dynamic responses and subsequent measured dynamic responses of civil infrastructure systems to excitations.
b) The need to use (a) as a basis for decision making to reduce wireless data transmission(s), and thereby ensure efficient utilization of limited on-board energy resources.
c) The need to monitor the point in time of state transition in the MDP-based model, as indicators of underlying partial failure(s) in the wireless smart-sensor node(s), which may ultimately lead to energy waste.

Objectives of the Research

The objectives of the research were:

a) Formulate a decision framework for SHM systems.
b) Characterize SHM systems as SMDP, and subsequently transform this SMDP model into a discrete-time MDP in order to incorporate the on-board capability for detecting partial failures in wireless sensor nodes.
c) Combine (a) and (b) to achieve energy-efficient data transmissions.
d) Test the efficacy of the MDP-based SHM model.

Problem Definition

The notations (or abbreviations) used in this paper are presented in Table 1.

Sample Deployment of Wireless Sensor Nodes

An example of civil infrastructure systems considered in this study was a 2-way bridge on which wireless smart-sensor nodes were mounted to measure vibration (see Figure 1). Gateway nodes were installed on the bridge in a way that all wireless smart-sensor nodes could transmit in single-hop mode to the nearest gateway node. With the implementation of an energy-efficient routing protocol [32], however, the framework presented in this paper can be applied to general WSSN topologies and multi-hop transmissions. It should be noted that the communication range of a MICA2 wireless smart-sensor node is between 150 m and 300 m [33]. It is therefore expected that WSS nodes would be deployed in a manner that would provide adequate coverage of civil infrastructure systems, since structural vibration can occur at any location.

Table 1. Notations / Abbreviations

Notation	Description
MDP	Markov Decision Process
SMDP	Semi-Markov Decision Process
SHM	Structural Health Monitoring
CIS	Civil Infrastructure Systems
WSS	Wireless Smart Sensor
WSSN	Wireless Smart-sensor Networks
CREST	Center for Research Excellence in Science and Technology

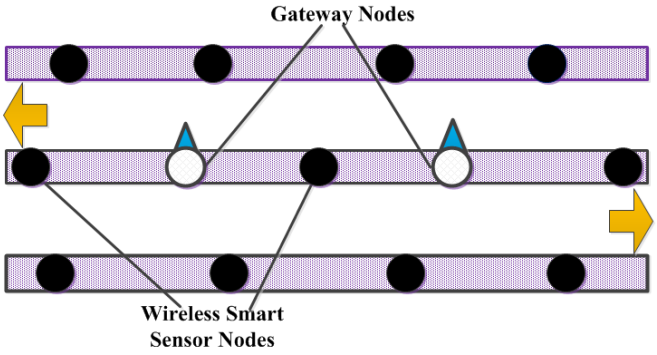

Figure 1. Sample Deployment of Wireless Smart Sensors on a 2-way Bridge

Wireless smart-sensor nodes carry-out local data processing to reduce data transmission to the gateway nodes. In order to save battery life, gateway nodes are configured to be in sleep mode most of the time, and periodically check their hardware interrupts to determine occurrence of external events, such as incoming packets, that require attention. The watchdog and asynchronous interrupts logic remain powered-on when the gateway node is in sleep mode. This feature enables gateway nodes to wake and receive incoming packets from neighboring WSS nodes for onward transmission to a base-station (or control room).

Structural Vibration Measurements

Vibration represents wasted energy, which sometimes translates into movements that may cause damage to civil infrastructure systems. Structural vibration can be measured by accelerometers, which convert vibration into electrical signals. WSS nodes use integrated accelerometers to measure vibration-based dynamic responses of the bridge to excitations. Vibration-measurable characteristics include velocity and acceleration. Moreover, the vibration signal amplitude indicates the severity of the cause of vibration. Let *b(t)* represent the value of the baseline measured dynamic response of the bridge to excitation. Sensor measurements may come with both useful and noise components. For the baseline response, a useful component is denoted as $a_1(t)$ and a noise component as $c_1(t)$. Moreover, let *s(t)* represents subsequent measured values of dynamic responses, which may also consist of a useful component $a_2(t)$ and its associated noise component $c_2(t)$. It was assumed that both *b(t)* and *s(t)* signals were measured on the same frequency range and scale factor. Equations for baseline and subsequent measured responses are defined as shown in Equations (1) and (2):

$$b(t) = a_1(t) + c_1(t) \tag{1}$$

$$s(t) = a_2(t) + c_2(t) \tag{2}$$

Two separate signals were compared in cross-correlation. Cross-correlation between two signals, *b(t)* and *s(t)*, is a measure of similarity between these two signals. Thus, the average cross-correlation between the two signals measured over interval T is given by Equation (3):

$$x_{12}(\tau) = \lim_{T\to\infty} \frac{1}{T} \int_0^T b(t)\, s(t-\tau) dt \tag{3}$$

where, $x_{12}(\tau)$ is the correlation function and a measure of the similarity between signals *b(t)* and *s(t)*.

This measure of correlation is a function of a new variable, τ. This new variable represents a time delay (or time shift) between the two signals, *b(t)* and *s(t)*. Energy represents the size of a signal. The energies of time-delayed (or time-advanced) signals must be same as that of the original signals, since time advanced (or time delayed) cannot change the energy of a signal. Correlation co-efficient, C_c, is given by Equation (4) [34], [35]:

$$C_c = \frac{x_{12}(\tau)}{\sqrt{x_{11}(0)\, x_{22}(0)}} \tag{4}$$

where, $x_{11}(0)$ and $x_{22}(0)$ denote the mean square values of the signals *b(t)* and *s(t)*, respectively, with C_c in the range of -1 to +1.

The closer C_c is to -1, the stronger the negative correlation; the closer C_c is to +1, the stronger the positive correlation.

Notes on Criterion for Data Handling and Signal Correlation

In this study, the signals correlation analysis was used for pre-data handling in order to ascertain the need for further processing of measured dynamic responses. This analysis was necessary for energy preservation in the SHM model. On the one hand, if results of the cross-correlation analysis indicate that the measured dynamic response is not similar to the baseline dynamic response, then there is a need to invoke the data handling criterion in order to ascertain whether actual data transmission is necessary.

On the other hand, if results of the cross-correlation analysis indicate that the measured dynamic response is similar to the baseline dynamic response, then invoking the data handling criterion is not necessary. From Equation (4) above, values of C_c in the range of $-1<C_c<0$ or $0<C_c<1$ indicates that *b(t)* is not similar to *s(t)*. In particular, $C_c=0$ indicates that *b(t)* is totally different from *s(t)*. Dissimilarity can, therefore, vary from minimum to maximum values. Thus, depending on the level of dissimilarity, decisions are made in the SHM model regarding whether to discard the measured data or transmit the data.

Data-handling Criterion

The value of a signal can be modeled by the energy contained in that signal. The total-energy-over-time interval $k_1 \le t \le k_2$ in continuous-time signals *b(t)* and *s(t)*, denoted as M_1 and M_2, respectively, are given by Equations (5) and (6):

$$M_1 = \int_{k_1}^{k_2} |b(t)|^2\, dt \tag{5}$$

$$M_2 = \int_{k_1}^{k_2} |s(t)|^2\, dt \tag{6}$$

The average power dissipated by signals *b(t)* and *s(t)* during the time interval $k_1 \le t \le k_2$ are given by Equations (7) and (8):

$$y_1 = \frac{1}{(k_2 - k_1)} \int_{k_1}^{k_2} |b(t)|^2 \, dt \tag{7}$$

$$y_2 = \frac{1}{(k_2 - k_1)} \int_{k_1}^{k_2} |s(t)|^2 \, dt \tag{8}$$

where, y_1 and y_2 denote the average power dissipated by signals $b(t)$ and $s(t)$, respectively, as shown in Equation (9):

$$W = y_2 - y_1 \text{ and } W > 0 \tag{9}$$

where, W represents the net average power dissipated by signals $b(t)$ and $s(t)$.

A pre-data handling analysis on the signals was subsequently based on the criterion in Equation (10):

$$W = \begin{cases} p_1 \le W \le p_2 : \textit{discard measured data} \\ p_2 < W \le p_3 : \textit{transmit measured data} \end{cases} \tag{10}$$

where, p_1, p_2, and p_3 define ranges for values of W. Moreover, p_1, p_2, and p_3 are specified during initial characterization of SHM systems.

In this study, the output of the accelerometer was assumed to be in the range of $0.01(g)$ to ξ(g), where 0.01 and ξ denote the lower and upper bounds of the accelerometer's output range, respectively. For purposes of decision making, the ranges of values of W were associated with the accelerometer's output, as follows:

$p_1 = 0.01$, $p_2 = 0.25\xi$, and $p_3 = \xi$.

SMDP Model

An SMDP model typically consists of the following five elements: decision epochs, state space, action space, transition probability, and rewards. SMDP can be described mathematically via Equation (11) [36], [37]:

$$SMDP = \{T, S, A_s, p_t(j \mid s, a), z_t(s, a)\}$$
$$t \in T, s \in S, a \in A_s \tag{11}$$

The characterization of the SHM system, as a discounted SMDP, is presented in Table 2. The corresponding transition-rate diagram for this model is presented in Figure 2. In this model, λ and μ denote positive-rate parameters for the sojourn time distribution in states during the "forward-sequence" and "return-sequence", respectively. The sojourn time in the current state was assumed to be exponentially distributed, with the distribution function given as $F(t|s,a)=1-e^{-\alpha(s,a)t}$ and where α(s,a) was a positive-rate parameter. $F(t|s,a)$ was the probability that the next decision epoch would occur within t time units of the current decision epoch, given that the WSS node chose action a in state s at the current decision epoch. State transitions occur at the end of sojourn period in the current state. The SMDP generated a sequence of rewards as it made its transition from state to state. A lump-sum reward was received in the current state upon choosing an action, and the process thereafter accumulated rewards at a continuous reward rate.

Table 2. Characterization of the SHM System as a Discounted SMDP

Elements	Characterization
Decision epochs	Decision epochs are points in time when the SHM system executes action(s). That is, $T = \{ t_1, t_2, t_3, \ldots, t_\infty \}$
State space	A system occupies a state when it is totally described by the values of variables that define the state. Thus, the *states* are: state 1 (*sleep or sensing*), state 2 (*processing*), state 3 (*transmitting or receiving*). That is, $s = \{s_1, s_2, s_3\}$.
Action space	In every state s and decision epoch t, action a among members of the action set A is chosen. $A=\{a_{i,j}, a_{i,h}\}$; where $a_{i,j}$ represents action for transition from states s_i to s_j, and $a_{i,h}$ represents action for transition from states s_i to s_h.
Transition probability	Choosing an action a in current state s at current decision epoch t, the system state j at the next decision epoch is determined by the probability distribution $p(j \mid s, a)$.
Rewards	The immediate discounted rewards function, $z(s,a)$, is given by $z(s,a) = x(s,a) + \int_0^\infty \sum_{j \in S} \left[\int_0^v e^{-\eta t} c(j', s, a) \, . \, p(j' \mid t, s, a) dt \right] F(v \mid s, a) \, dv.$

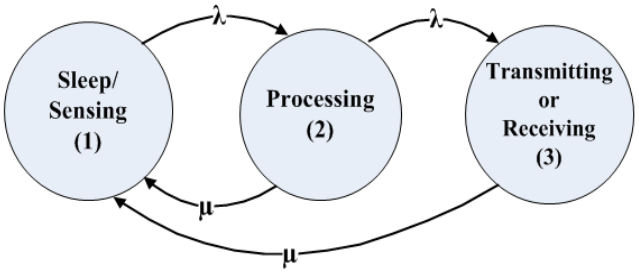

Figure 2. Transition-rate Diagram for an SMDP-based SHM Model

Problem Solution

The probability structure of the sojourn time in a state and the state transition for the SHM model consisted of two processes: the natural process and the Semi-Markov Decision Process [36]. Whereas the natural process models the state's evolution of the SHM system as if it were observed continuously at all times, the Semi-Markov Decision Process models the evolution of the system only at decision epochs. This is shown by Equation (12):

$$r(t, j \mid s,a) = p(j \mid s,a)F(t \mid s,a) \tag{12}$$

where, *r(t,j|s,a)* is the joint probability that the state at the next decision epoch equals *j* and that the next decision epoch occurs at or before time *t* when action *a* is chosen in state *s* at the present decision epoch. In addition, *p(j|s,a)* is the probability that the embedded Markov Decision Process occupies state *j* at next decision epoch, given that action *a* was chosen in state *s* at the current decision epoch t. *F(t|s,a)* is subsequently defined by Equation (13):

$$F(t \mid s,a) \le 1-\gamma; \quad 0 \le \gamma < 1, t > 0 \tag{13}$$

Rewards

For this study, rewards represented earnings from energy-efficient data transmission by wireless sensor nodes. The immediate rewards received in a current state consisted of lump-sum rewards and accumulated rewards at continuous reward rates. When the SHM's system process chose *action*, it either received a lump-sum reward or paid a lump-sum cost. Moreover, the system accrued rewards (or incurred costs) at a continuous rate, as long as the natural process occupied state *j'*, and *action a* was chosen in state *s* at the preceding decision epoch. For purposes of illustration, the unit of reward was μJ. For example, a reward of +1 represented earnings of $1\mu J$ (i.e., $1\mu J$ earned for efficient energy consumption for data transmission) and a reward of -1 represented a loss of $1\mu J$. It should be noted that negative rewards indicated an energy loss, due to inefficient utilization of limited energy resources in wireless smart-sensor networks. Rewards were subsequently specified as follows.

Rewards when the SHM system's process efficiently utilized energy resources. A lump-sum reward associated with the *sleep/sensing* state was equal to 2, and the system's process accumulated rewards between decision epochs at a continuous reward rate of 10. A lump-sum reward associated with the *processing* state was equal to 2, and the system's process accumulated rewards between decision epochs at a continuous reward rate of 10. A lump-sum reward associated with the *transmitting/receiving* state was equal to 5, and the system's process accumulated rewards between decision epochs at a continuous reward rate of 10.

Rewards when the SHM system's process inefficiently utilized energy resources**.** A lump-sum reward associated with the *sleep/sensing* state was equal to -2, the system's process accumulated rewards in the current state at a continuous reward rate of -2, and also accumulated rewards between decision epochs at a continuous reward rate of 10. A lump-sum reward associated with the *processing* state was equal to -2, and the system's process accumulated rewards in the current state at a continuous reward rate of -10, and also accumulated rewards between decision epochs at a continuous reward rate of 10. A lump-sum reward associated with the *transmitting/receiving* state was equal to -5, and the system's process accumulated rewards in the current state at a continuous reward rate of -15, and also accumulated rewards between decision epochs at a continuous reward rate of 10.

Consequently, the total discounted reward in the current state was computed using Equation (14):

$$\begin{aligned} z(s,a) &= x(s,a) \\ &+ \int_0^{\infty} \sum_{j \in S} \left[\int_0^{v} e^{-\eta t} c\left(j', s, a\right) p(j' \mid t, s, a) dt \right] F\left(v \mid s, a\right) dv \end{aligned} \tag{14}$$

where, *z(s,a)*, *x(s,a)*, *c(j',s,a)*, *p(j'|t,s,a)*, and η denote immediate rewards received in the current state, a lump-sum reward, accumulated reward rates, the transition probability of the natural process, and the discount rate, respectively. Since the natural process in the SMDP model does not change state until the next decision epoch, then *p(j'|t,s,a)* was set to 1. Equation (14) can then be expressed as shown in Equations (15)-(17):

$$\begin{aligned} z(s,a) &= x(s,a) \\ &+ c\left(j', s, a\right) \int_0^{\infty} \sum_{j \in S} \int_0^{v} e^{-\eta t} dt\, F\left(v \mid s, a\right) dv \end{aligned} \tag{15}$$

where, for example:

$$\begin{aligned} z(s,a_{i,h}) &= x(s,a_{i,h}) \\ &+ c\left(j', s, a_{i,h}\right) \int_0^{\infty} \int_0^{v} e^{-\eta t} dt\, F\left(v \mid s, a_{i,h}\right) dv \end{aligned} \tag{16}$$

and:

$$\begin{aligned} z(s,a_{i,j}) &= x(s,a_{i,j}) \\ &+ c\left(j', s, a_{i,j}\right) \int_0^{\infty} \int_0^{v} e^{-\eta t} dt\, F\left(v \mid s, a_{i,j}\right) dv \end{aligned} \tag{17}$$

In Equations (14)-(17), the integration is over the domain of the given sojourn time in the current states. The infinite horizon sum of the expected total discounted reward in the current states, starting from state 1 under stationary policy π, is then given by Equation (18):

$$v_\eta^\pi(s) = z(s,a) + \sum_{j\in S}\int_0^\infty e^{-\eta t}\, r(t, j \mid s,a)\, dt\, v_\eta^\pi(j) \tag{18}$$

which, by invoking Equation (12), yields Equation (19):

$$v_\eta^\pi(s) = z(s,a) + \sum_{j\in S} p(j \mid s,a)\int_0^\infty e^{-\eta t}\, F(t \mid s,a)\, dt\, v_\eta^\pi(j) \tag{19}$$

where, $v_\eta^\pi(s)$ and $v_\eta^\pi(j)$ denote the sum of the expected total discounted rewards in the current and next states (starting from state j), respectively.

Relationship between Continuous-time and Discrete-time MDP

Incidents that cause irregular vibrations occurring at random times and the sojourn time in each state of the SHM model—a random variable that is assumed to be exponentially distributed—enable us to characterize the SHM system's process as SMDP. In order to monitor the occurrence of partial failures in the wireless smart-sensor nodes, which may untimely lead to transitions between the SHM's system states, a need arises for transformation of the SMDP model into a discrete-time MDP model. The approach by Puterman [36] was consequently adopted in order to achieve this transformation. The discounted state transition function $\gamma(j|s,a)$, for $j \in s$ and $a \in A_s$, is given by Equation (20):

$$\gamma(j \mid s,a) = p(j \mid s,a)\int_0^\infty e^{-\eta t} F(t \mid s,a)\, dt \tag{20}$$

The integration in Equation (20) is over the domain of the given sojourn time in the current states. Moreover, Equations (19) and (20) transform the discounted SMDP model into a discounted discrete-time MDP model. The state-dependent and action-dependent discount rate can be expressed by Equation (21):

$$\beta(s,a) \equiv \sum_{j\in S}\gamma(j \mid s,a) \tag{21}$$

Thus,

$$v_\eta^\pi(s) = z(s,a) + \sum_{j\in S}\gamma\left(j \mid s,a\right) v_\eta^\pi(j) \tag{22}$$

This discounted discrete-time MDP version enables the SHM model to monitor the point in time of state transitions in relation to the completion of state-dependent activity, which is indicated by expiration of the sojourn time in a current state. The delay in the system's process transition out of the current state into a next state, in spite of the expiration of the sojourn time in the current state, may be caused by a partial failure in either the software or hardware of the wireless smart-sensor nodes. The SHM model is, therefore, capable of self-detection of partial failures in smart-sensor nodes, which may ultimately lead to inefficient energy utilization. The state transition diagram of the discounted discrete-time MDP version is given in Figure 3.

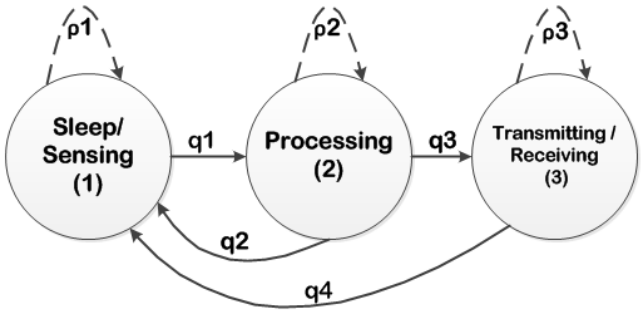

Figure 3. State Transition Diagram of the Transformed Model (discrete-time MDP version)

The transition probabilities of the embedded MDP process are defined as $\rho_1=p(s_1|s_1,a_{1,2})$, $\rho_2=p(s_2|s_2,a_{2,3})$, $\rho_3=p(s_3|s_3,a_{3,1})$, $q_1=p(s_2|s_1,a_{1,2})$, $q_2=p(s_1|s_2,a_{2,1})$, $q_3=p(s_3|s_2,a_{2,3})$, and $q_4=p(s_1|s_3,a_{3,1})$. It should be noted that ρ_1, ρ_2, and ρ_3 represent transition probabilities for the delayed transition out of current states 1, 2, and 3, respectively.

The optimal policy is one that maximizes rewards from efficient energy utilization over an infinite horizon. An infinite horizon implies that the SHM system's process runs infinitely. The optimality equation, after transforming the discounted SMDP model into a discounted discrete-time MDP model, is given by Equation (23):

$$v_\eta^*(s) = \underbrace{\max}_{a\in A_s}\left\{ z(s,a) + \sum_{j\in S}\gamma(j \mid s,a)\; v_\eta(j) \right\} \tag{23}$$

The average optimal value of the sum of the expected total discounted reward over an infinite horizon is given by Equation (24):

$$v_{aov}^{\pi}(s) = \underbrace{\lim}_{n\to\infty}\left\{\frac{1}{n}\sum_{i=1}^{n}\left(v_{\eta}^{*}(s)\right)_i\right\} \quad (24)$$

where, i denotes the indexing variable and $v_{aov}^{\pi}(s)$ denotes the average optimal value of following policy π over an infinite horizon in a current state. It should be noted that the SHM system's process transits from state to state and continues to run in this manner infinitely. There is, therefore, a need to prove the convergence of $v_{\eta}^{*}(s)$ over the infinite horizon.

Theorem. If the system's process runs over an infinite horizon consisting of i cycles, where, $i = 1,2,\ldots,\infty$, then the average optimal value of the sum of the expected total discounted rewards in the current states starting from state 1 is given by Equation (25):

$$v_{aov}^{\pi}(s) = v_{\eta}^{*}(s) \quad (25)$$

Proof. Let i denote the indexing variable for a number of cycles and $v_{aov}^{\pi}(s)$ denote the average optimal value of the sum of the expected total discounted rewards in the current states starting from state one, as given by Equations (26)-(28):

$$v_{aov}^{\pi}(s) = \underbrace{\lim}_{n\to\infty}\left\{\frac{1}{n}\sum_{i=1}^{n}\left(v_{\eta}^{*}(s)\right)_i\right\} \quad (26)$$

$$v_{aov}^{\pi}(s) = \underbrace{\lim}_{n\to\infty}\left\{\frac{n\left(v_{\eta}^{*}(s)\right)}{n}\right\} \quad (27)$$

hence,

$$v_{aov}^{\pi}(s) = v_{\eta}^{*}(s) \quad (28)$$

Numerical Analysis and Discussion of Results

The initial characterization of civil infrastructure systems provides a baseline model of the systems in an undamaged state. In order to achieve continuous monitoring, large amounts of data are normally generated by wireless sensor nodes deployed on civil infrastructure systems. For purposes of illustration, the maximum output of the accelerometers was 5g (ξ=5), and the SHM system could realistically estimate sojourn time in each state. The SHM model was tested using two options: *Options A* and *B*.

Option A: The SHM system wakes from sleep and starts sensing excitations at decision epoch 1 then transits to the next state (*processing*). Depending on the outcome of the processing in state 2 at decision epoch 2, the system's process can discard the measured data and transit back to the *sleep* (or *sensing*) state, or transit to state 3 (*transmitting* or *receiving*) for data transmission. If the system's process transits to state 3 then, upon expiration of the sojourn time at decision epoch 3 (in state 3), the system's process transits back to the *sleep* (or *sensing*) state. This is the scenario in which there is no delayed transition out of the current states. For this case, energy resources of the wireless smart-sensor nodes are efficiently utilized for data transmissions.

Option B: The SHM system wakes from sleep and starts sensing excitations at decision epoch 1. Upon expiration of the sojourn time in this current state, the system's process should transit to a next state. However, the system's process may fail to change state in a timely manner, due to partial failures in the sensor node. Similarly, at decision epoch 2 in state 2 (*processing)*, the system may fail to change state in a timely manner, due to partial failures in the sensor node. Depending on the outcome of the processing in this state, the system's process can either discard the measured data and transit back to the *sleep* (or *sensing*) state or transit to the *transmitting* (or *receiving*) state for data transmission. If the SHM system transits to state 3 then, upon expiration of the sojourn time in state 3 at decision epoch 3, the system's process may transit or may fail to transit in a timely manner to the next state (*sleep* or *sensing*). This is the scenario in which there is a delayed transition out of the current states. For this case, energy resources of the wireless smart-sensor nodes are inefficiently utilized.

Results of the numerical analysis are presented in Tables 3 and 4, as well as Figures 4 and 5. The optimal values represent the maximum value of the sum of the expected total discounted rewards in each state, starting at state 1. The expected total discounted rewards in each state and the sum of the expected total discounted rewards starting from state 1 are presented in Tables 3 and 4.

Table 3. Total Discounted Rewards in States at Decision Epochs [example case: W = 1.0 (values in μJ)]

v(s)	Options	1	2	3	Total	Optimal value
$v_1(s)$	A	56.4	6.1	0	62.5	62.5
	B	-5.9	-0.9	0	-6.8	
$v_2(s)$	A		6.1	0	6.1	6.1
	B		-0.9	0	-0.9	
$v_3(s)$	A			0	0	0
	B			0	0	

Table 4. Total Discounted Rewards in States at Decision Epochs [example case: W = 3.5 (values in μJ)]

v(s)	Options	1	2	3	Total	Optimal value
$v_1(s)$	A	1126.0	134.8	53.1	1313.9	1313.9
	B	-9.2	-1.7	0.5	-10.4	
$v_2(s)$	A		134.8	53.1	187.9	187.9
	B		-1.7	0.5	-1.2	
$v_3(s)$	A			53.1	53.1	53.1
	B			0.5	0.5	

It should be noted that the results presented in Table 3 apply to all values of W in the range of 0.01<=W<=0.25ξ, where ξ=5. Similarly, the results presented in Table 4 apply to all values of W in the range of 0.25ξ< W<=5.

In particular, Table 3 shows the results for the case where the net average power dissipated was W=1.0. Whereas Option A earned an expected total discounted reward of 56.4μJ starting from state 1, Option B earned -5.9μJ. Similarly, Option A earned higher expected total discounted rewards of 6.1μJ, when the analysis started from state 2, while option B earned -0.9μJ. These higher rewards implied higher energy efficiency in the SHM model. It should further be noted that W=1.0 falls in the range in which the measured value of the dynamic response was not significant and the data should be discarded (i.e., no data transmission).

Moreover, Table 4 shows the results for the case where the net average power dissipated was W = 3.5. Whereas Option A earned an expected total discounted reward of 1126.0μJ starting from state 1, Option B earned -9.2μJ. Similarly, Option A earned higher expected total discounted rewards of 134.8μJ, when the analysis started from state 2, while option B earned -1.7μJ. Moreover, when the analysis started from state 3, Option A earned 53.1μJ, while Option B earned 0.5μJ. These higher rewards implied higher energy-efficient data transmissions by the SHM model. It should again be noted that W=3.5 falls in the range in which the measured value of dynamic response was significant; hence, the sensor node transmitted the data to a gateway node.

Figure 4 graphically presents the sum of the expected total discounted rewards in all states at the decision epochs starting from state 1. Whereas Option A earned 62.5μJ in state 1 for W = 1, Option B earned -6.8μJ. Option A earned a higher sum of expected total discounted rewards equal to 6.1μJ in state 2, while Option B earned -0.9μJ.

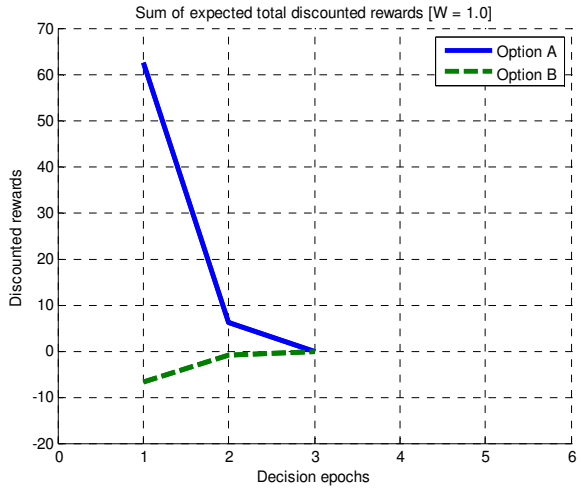

Figure 4. Sum of Expected Total Discounted Rewards Starting from State 1 (W =1.0)

In Figure 5, the graph shows the sum of the expected total discounted rewards in all states at decision epochs starting from state 1. Whereas Options A earned a sum of the expected total discounted rewards equal to 1313.9μJ in state 1, Option B earned -10.4μJ. When the analysis was based on the system's process starting from state 2, Option A earned 187.9μJ in rewards, while Option B earned -1.2μJ. Moreover, Option A earned 53.1μJ, while Option B earned 0.5μJ, when the analysis started from state 3.

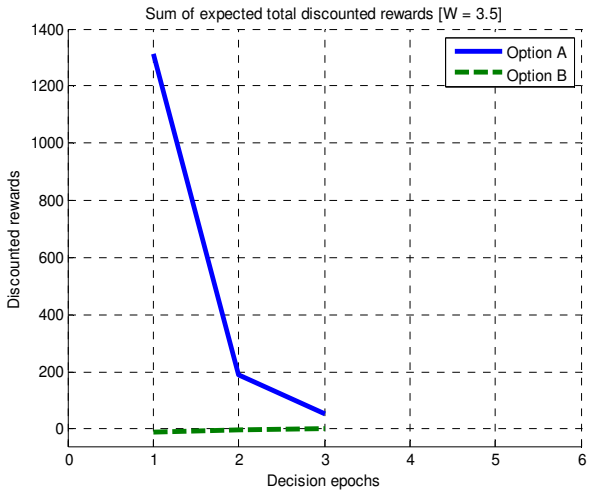

Figure 5. Sum of Expected Total Discounted Rewards Starting from State 1 (W = 3.5)

The results further showed that Option A yielded optimal values of 1313.9μJ, 187.9μJ, and 53.1μJ when the SHM system's process started from states 1, 2, and 3, respectively.

The contributions of this study can be summarized as follows:

- A decision framework for the design and implementation of SHM systems using wireless smart-sensor networks that can offer energy-efficient data transmissions and, thereby, prolong the life of SHM applications.
- The SHM model presented in this paper is capable of detecting the occurrence of partial failures in wireless smart-sensor nodes that may systematically lead to inefficient utilization of limited energy resources. This detection mechanism may be utilized by designers of SHM applications to plan appropriate maintenance strategies for the SHM systems.

Conclusion

In this study, the authors investigated the problem of inefficient energy utilization for data transmissions in structural health monitoring (SHM) applications for civil infrastructure systems using wireless smart-sensor networks. The aim of the study was to improve energy efficiency in SHM applications. For this reason, a novel SMDP-based model was constructed for an SHM system. The SHM system was characterized as a discounted SMDP consisting of three states: *sleep* (or *sensing*), *processing*, and *transmitting* (or *receiving*).

In order to monitor the point in time of state transitions, indicated by the expiration of the sojourn time in the current states, the SMDP-based model was transformed into a discounted discrete-time MDP-based model. The efficacy of the MDP-based SHM model was subsequently tested via numerical analysis, which was implemented in MATLAB software. Optimal results in terms of the average optimal value of the sum of the expected total discounted energy consumption for data transmissions were obtained. An optimal analysis yielded the conclusion that the SHM model provided a decision framework for the design and implementation of SHM systems using wireless smart-sensor networks that can offer energy-efficient data transmissions.

In addition, the SHM model also provided the capability for detecting the occurrence of partial failures in wireless smart-sensor nodes that may systematically lead to inefficient utilization of limited energy resources. It was expected that the results may provide guidelines for the design and implementation of energy-efficient structural health monitoring systems for civil infrastructure systems.

Acknowledgments

This research was supported in part by the National Science Foundation (NSF) under Grants NSF-1137732 and NSF-1241626 (CREST─Center for Research Excellence in Science and Technology).

References

[1] Sim, S.-H., Li, J., Jo, H., Park, J.-W., Cho, S., Spencer, B. F., Jr., et al. (2014). A Wireless Smart Sensor Network for Automated Monitoring of Cable Tension. *Smart Materials and Structures*, *23*(2), 25006. doi.10.1088/0964-1726/23/2/025006

[2] Park, J.-W., Sim, S.-H., & Jung, H.-J. (2014). Wireless Displacement Sensing System for Bridges using Multi-sensor Fusion. *Smart Materials and Structures*, *23*(4), 045022. doi.10.1088-1726/23/4/045022

[3] He, L., Lian, J., Ma, B., & Wang, H. (2014). Optimal Multi-axial Sensor Placement for Modal Identification of Large Structures. *Structural Control and Health Monitoring, 21*(1), 61-79.

[4] Park, J.-W., Sim, S.-H., Jung, H.-J. (2013). Wireless Sensor Network for Decentralized Damage Detection of Building Structures. *Journal of Smart Structures and Systems*, *12*(3-4), 399-414.

[5] Koo, K. Y., Sung, S. H., & Jung, H. J. (2011). Damage Quantification of Sheer Building Using Deflections Obtained by Modal Flexibility. *Smart Materials and Structures*, *20*(4), 045010. doi.10.1088/0964-1726/20/4/045010

[6] Li, J., Nagayama, T., Mechitov, K. A., & Spencer, B. F., Jr. (2012). Efficient Campaign-type Structural Health Monitoring Using Wireless Smart Sensors. *Proceedings of SPIE 8345, Sensors and Smart Structures for Civil, Mechanical and Aerospace Systems*. San Diego, California, USA. doi:10.1117/12.914860

[7] Sim, S., Carbonell-Marquez, J., Spencer, B. F., Jr., & Jo, H. (2011). Decentralized Random Decrement Technique for Efficient Data Aggregation and System Identification in Wireless Smart Sensor Networks. *Probabilistic Engineering Mechanics*, *26*(1), 81-91.

[8] Sim, S.-H., Spencer, B. F., Jr., Zhang, M., & Xie, H. (2010). Automated Decentralised Modal Analysis using Smart Sensors. *Journal of Structural Control and Health Monitoring*. doi:10.1002/stc.348

[9] Rice, J. A., Mechitov, K. A., Sim, S. H., & Spencer, B. F., Jr. (2011). Enabling Framework for Structural Health Monitoring using Smart Sensors. *Journal of Structural Control and Health Monitoring*, *28*(5), 574-587.

[10] Jo, H., Sim, S., Mechitov, K. A., Kim, R., Li, J., Moinzadeh, P., et al. (2011). Hybrid Wireless Smart Sensor Network for Full-scale Structural Health Monitoring of a Cable-stayed Bridge. *Proceedings of SPIE 7981, Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems*. doi:10.1117/12.880513

[11] Cho, S., Jo, H., Jang, S., Park, J., Jung, H.-J., Yun, C. -B., et al. (2010). Structural Health Monitoring of Cable-stayed Bridge using Wireless Sensor Technology: Data Analysis. *Journal of Smart Structures and Systems*, *6*(5-6), 461-480.

[12] Rice, J. A., Mechitov, K., Sim, S. H., Nagayama, T., Jang, S., Kim, R., et al. (2010). Flexible Smart Sensor Framework for Autonomous Structural Health Monitoring. *Journal of Smart Structures and Systems, 6*(5-6), 423-438.

[13] Jang, S., Jo, H., Cho, S., Mechitov, K. A., Rice, J. A., Sim, S. H., et al. (2010). Structural Health Monitoring of a Cable-stayed Bridge Using Smart Sensor Technology: Deployment and Evaluation. *Smart Structures and Systems*, *6*(5-6), 439-459.

[14] Ajofoyinbo, A. M., & Olowokere, D. O. (2014). Health Monitoring and Control of Civil Infrastructures using Wireless Smart Sensors. *Bulletin of Electrical Engineering and Informatics*, *3*(3), 201-212.

[15] Ni, W., Li., W., & Alam, M. (2009). Determination of Optimal Call Admission Control Policy in Wireless Networks. *IEEE Transactions on Wireless Communications*, *8*(2), 1038-1044.

[16] Chen, Z., & Casciati, F. (2014). A Low-noise, Real-time Wireless Data Acquisition System for Structural Monitoring Applications. *Structural Control and Health Monitoring*, *21*(7), 1118-1136. doi:10.1002/stc.1636

[17] Li, X., & Yuan, J. (2014). Safety and Health Monitoring of Bridge Construction Based on Wireless Sensor Network. *International Journal of Multimedia and Ubiquitous Engineering*, *9*(3), 257-268.

[18] Cahill, P., O'Keeffe, R., Jackson, N., Mathewson, A., & Pakrashi, V. (2014). Structural Health Monitoring of Reinforced Concrete Beam using Piezoelectric Energy Harvesting System. *7th European Workshop on Structural Health Monitoring*, (pp. 190-196). La Cite, Nantes, France.

[19] Gu, Y., He, L., Zhu, T., & He, T. (2014). Achieving Energy-synchronized Communication in Energy-harvesting Wireless Sensor Networks. *ACM Transactions on Embedded Computing Systems*, *13*(2), 68:1-68:26.

[20] Zhou, G., Huang, L., Li, W., & Zhu, Z. (2014). Harvesting Ambient Environment Energy for Wireless Sensor Networks: A Survey. *Journal of Sensors*. doi.10.1155/2014/815467

[21] Fu, T. S., Gosh, A., Johnson, E. A., & Krishnamachari, B. (2013). Energy-efficient Deployment Strategies in Structural Health Monitoring using Wireless Sensor Network. *Structural Control and Health Monitoring*, *20*(6), 971-986.

[22] Zhou, G.-D., & Yi, T.-H. (2013). Recent Developments on Wireless Sensor Networks Technology for Bridge Health Monitoring. *Mathematical Problems in Engineering*. doi.10.1155/2013/947867

[23] Sendra, S., Lloret, J., Garcia, M., & Toledo, J. F. (2011). Power Saving and Energy Optimization Techniques for Wireless Sensor Networks. *Journal of Communications*, *6*(6), 439-459.

[24] Zhou, D., Ha, D. S., & Inman, D. J. (2010). Ultra-power Active Wireless Sensor for Structural Health Monitoring. *Smart Structures and Systems, 6*(5-6), 675-687.

[25] Nagayama, T., Moinzadeh, P., Mechitov, K., Ushita, M., Makihata, N., Ieiri, M., et al. (2010). Reliable Multi-hop Communication for Structural Health Monitoring. *Smart Structures and Systems*, *6*(5-6), 481-504.

[26] Jo, H., Sim, S., Nagayama, T., & Spencer, B. F., Jr. (2010). Decentralized Stochastic Model Identification using High Sensitivity Wireless Smart Sensors. *Proceedings of 5th World Conference on Structural Control and Monitoring, Paper 5WCSC M-10466.* Tokyo, Japan.

[27] Nagayama, T., Ushita, M., Fujino, Y., Ieiri, M., & Makihata, N. (2010). The Combined Use of Low-cost Smart Sensors and High Accuracy Sensors Apprehend Structural Behavior. *Proceedings of SPIE Smart Structures/NDE,* (pp. 7647-7642). San Diego, California, USA.

[28] Karbhari, V. M., Guan, H., & Sikorsky, C. (2009). Operational Modal Analysis for Vibration-based Structural Health Monitoring of Civil Structures. *Structural Health Monitoring of Civil Infrastructure Systems*, 213-259.

[29] Rice, J. A., & Spencer, B. F., Jr. (2008). Structural Health Monitoring Sensor Development for the Imote2 Platform. *Proceedings SPIE 6932, Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems*, *Paper 693234*. doi:10.1117/12.776695

[30] Pakzad, S. N., Fenves, G. L., Kim, S., & Culler, D. E. (2008). Design and Implementation of Scalable Wireless Sensor Network for Structural Monitoring. *Journal of Infrastructure System*, *14*(1), 89-101.

[31] Kim, S., Pakzad, S., Culler, D., Demmel, J., Fenves, G., Glaser, S., et al. (2007). Health Monitoring of

Civil Infrastructures Using Wireless Sensor Networks. *Proceedings of 6th International Symposium on Information Processing in Sensor Networks,* (pp. 254-263), Cambridge, MA.

[32] Pantazis, N. A., Nikolidaskis, S. A., & Vergados, D. D. (2013). Energy-efficient Routing Protocols in Wireless Sensor Networks: A Survey. *Communications Surveys & Tutorials*, *15*(2), 551-591.

[33] Crossbow Technology. (n.d.). Mica2 Wireless Measurement System Datasheet. Retrieved from http://www.xbow.co

[34] Ghosh, M. K., Sen, S., & Mukhopadhyay, S. (2008). *Measurement and Instrumentation: Trends and Applications*. CRC Press.

[35] Hsu, H. P. (2011). *Schaum's Outline of Probability, Random Variables, and Random Processes.* McGraw-Hill.

[36] Puterman, M. L. (2005). *Markov Decision Process: Discrete Stochastic Dynamic Programming.* Wiley-Interscience.

[37] Howard, R. A. (1960). *Dynamic Programming and Markov Process.* New York, NY: The Massachusetts Institute of Technology and John Wiley & Sons, Inc.

Biographies

DAVID O. OLOWOKERE received his Ph.D. from the State University of New York. He currently heads the engineering programs at Texas Southern University and directs the TSU Aviation and Flight programs. In addition, he has been the principal investigator for a University–Industry partnership providing engineering support for the Safety and Mission Assurance Program at the Johnson Space Center in Houston, and the NASA Marshall Center in Huntsville, Alabama. He had served as principal investigator for research grants from several organizations including the U.S. National Science Foundation, NASA, the U.S. Department of Energy, and the U.S. Department of Defense. Dr. Olowokere had previously been a faculty member at the University of Alabama, University of Detroit, Wayne State University, and Bucknell University. His publications have appeared in several peer-reviewed journals including the ASCE Journal, Journal of Computer and Structures, Journal of Constructional Steel Research, and he has made presentations in numerous conferences, workshops, and symposia worldwide in the general area of renewable energy technology, engineering practice, and engineering management. Dr. Olowokere may be reached at olowokeredo@tsu.edu

ABAYOMI AJOFOYINBO received his Ph.D. in Systems Engineering from the University of Lagos, Nigeria, in 2008. Dr. Ajofoyinbo is a Visiting Assistant Professor in the Department of Engineering at Texas Southern University (TSU). He is currently in charge of assessment for the engineering programs at TSU and teaches courses in electrical/electronic and computer engineering. Between September 2013 and August 2014, Dr. Ajofoyinbo worked for TSU as a Postdoctoral Fellow at the National Science Foundation's CREST Centre for Research on Complex Networks. He is a Senior Lecturer in the Department of Systems Engineering at the University of Lagos, Nigeria. His research interests include intelligent control, embedded systems, wireless communications, wireless sensor networks, and engineering systems modeling analysis. He has published peer-reviewed papers in referred journals and international conferences. Dr. Ajofoyinbo may be reached at ajofoyinboa@tsu.edu

MODELING AND ANALYSIS OF VARIABLES AFFECTING PERFORMANCE OF A ROBOTIC MANUFACTURING CELL: AN INDUSTRIAL CASE STUDY

Julie Zhang, University of Northern Iowa; Ali Kashef, University of Northern Iowa; Mark Ecker, University of Northern Iowa; Elvis Alicic, University of Northern Iowa

Abstract

Automated robotic manufacturing cells are widely used in manufacturing companies, due to their many benefits. In this paper, the authors present an industrial case study that assessed the performance of a robotic manufacturing cell consisting of one FANUC robot centered between four CNC machines and a washer, and machining five different parts in a part family. Using two approaches, a specific computer language simulation using Simulation Language Extensible (SLX) and statistical regression modeling, the throughput of the manufacturing cell was predicted. Both of the approaches used six independent variables: machine downtime, off-plan time, setup time, weekly schedule requirements, scrap rate, and preceding operation output.

The comparison of the two prediction results found that the statistical regression model generated a more accurate performance assessment, while the SXL-based simulation had higher requirements on data, thereby leading to less accurate results. Additionally, the most significant factors impacting robotic manufacturing cell performance identified through the regression model were off-plan time and setup time. The goal of this study was to better understand and utilize the robotic manufacturing cell. Also, an integrated methodology should be proposed for analyzing the output performance of robotic manufacturing cells in general.

Introduction

Maintaining a sustainable manufacturing industry not only leads to quality jobs and societal prosperity, but also ensures a competitive economy and national security [1]. In the past few decades, advanced technological innovations have been implemented in manufacturing companies, allowing them to become more competitive in today's global environment. Robotics is an advanced manufacturing technology, where automated machines replace human labor in the manufacturing process to conduct work according to predesigned routes. Robots today are found not only in automotive factories but also in industries such as food and beverages, pharmaceuticals, metals, and medical devices [2].

When a robot is used in hardware production, it is not uncommon for it to be organized and synchronized to work with other nearby machine tools. Often arranged in circular or linear fashion, called automated cellular manufacturing or ACM, robots can perform material handling, machine tending, loading, unloading, and inspection. ACM provides many benefits including reduced setup times, more efficient material handling, monitoring of work-in-process (WIP) inventory, reduced market response time, and better efficiency and quality in production [3]. These benefits, along with the robots' high repeatability, reliability, and flexibility, have drawn many organizations to adopt robots into their processes, with the tendency that ACM is becoming increasingly more sophisticated [4].

Robots in machining operations tend to be highly dynamic and unique, due to differences in cells. This dynamic environment and uniqueness allow for multiple alternatives and possibilities from a planning and layout standpoint. These machines range from being set up in single or multiple operations to an entire family of parts. Planning and implementation of automated cells is difficult and requires extensive knowledge of the process characteristics that may or may not exist in manufacturing situations. Due to the dynamic nature of automated machining cells that use robots and perform various part manufacturing tasks, it is difficult to determine the factors that impact robots, which can result in lost time or productivity. A study of determining factors and evaluating robotic cell performance is needed to provide guidelines for planning layout configurations of automated machining cells. In this paper, the authors present a case study on the assessment of the performance of a robotic manufacturing cell. The goal was to provide insights into which factors are critical for modeling and planning robotic machining cells, leading to quicker delivery times, more flexibility, and better utilization of machine tools.

Literature Review

The cellular manufacturing system (CMS) has drawn a lot of research attention since it was first introduced onto the manufacturing floor. Manufacturing companies that use cellular manufacturing modules to increase productivity by reducing throughput time and WIP inventories must have an

accurate way of modeling and validating their project plans. Simulation of manufacturing cells has gained popularity in research for its ability to dynamically model complex systems. Simulations can help identify manufacturing constraints, such as bottlenecks, and help predict manufacturing cell performance. The result is an increased understanding of the process potential, better utilization of resources, and, accordingly, improved adaptation to changes in the process that positively affects productivity, flexibility, and efficiency. And, when used accurately and appropriately, simulations can greatly assist in the design and improvement of the new manufacturing cells and help validate methods and plans before they are implemented on the floor [5-12].

The design problem of the cellular manufacturing system was investigated by Ahkioon et al. [3] through the introduction of routing flexibility and alternate contingency routing, as well as alternate main routings. In addition, this study considered the trade-offs between increased flexibility and additional cost as a result of the contingency routings showing no significant increase in system cost with the additional routing flexibilities. Das and Abdul-Kader [5] proposed a dynamic multi-objective model of integer programming for designing a cellular manufacturing system that considered machine reliability and part demand changes over time, allowing alternative part processing routes that would maximize machine system reliability and minimize system costs.

Yazici [6] conducted a simulation-based study to determine the influence of volume, mix, routing, and labor flexibilities with respect to continuously changing demand in cellular manufacturing and job shops. The results of the study indicated that added routing flexibility leads to significantly shorter lead times (above 70 percent) with both low- and high-volume flexibility. In a study conducted by Djassemi [7], the discrepancy between flexibility and uneven machine utilization in manufacturing cells was examined through simulation modeling of a variable demand system and a flexible workforce environment. The results of the study indicated that use of flexible cross-trained operators can minimize load imbalance present in dedicated machines in manufacturing cells and improve flexibility with respect to unstable demand. By developing a mathematic model, Chang [8] incorporated multiple attributes of routing efficiency, versatility, and variety in order to measure routing flexibility.

Other studies looked at robotic cell scheduling and balancing of robotic movements in order to guide cell layout and improve process efficiency and flexibility. For example, Gultekin et al. [9] investigated a scheduling problem in a robotic cell with identical parts and a given number of machines. Their findings in the process flexibility of robot move cycles indicated that an optimal number of machines must be identified in order to reduce cycle time and increase throughput. The robotic assembly line balancing problem often impacts cell productivity, which is typically measured by throughput and cycle time. In certain robotic applications, robots are key factors in the process (such as the semiconductor industry) impacting productivity and where robotic programming methods become very important. Geismar and Pinedo [10] investigated throughput of a robotic cell used in a microlithography process that had a stochastic processing time for a semiconductor manufacturing industry. The distribution function of the robot's sequence time was found and verified with simulation results and scheduling schemes were presented to improve throughput. Abd et al. [11] presented an algorithm that involved scheduling the robotic cell for assembly of multiple products. Using cycle time, total transportation time, utilization rate, and workload rate as performance measures, the study created a mathematical model for their industrial case study to effectively manage the scheduling of a robotic cell.

Using a robotic cell composed of three machines and a robot in the center of the cell, Shafiei-Monafared et al. [12] conducted a study that uniquely developed a flow-graph knowledge-based model to analyze cycle times when the part processing time was stochastic. Developing an effective knowledge-based system relies heavily on the quality of the domain knowledge of manufacturing. Today, capturing this domain knowledge remains a challenge to knowledge-based system developers. Different approaches were used in the simulations of the robotic manufacturing cell performance in the aforementioned studies. Although the evaluation methods involving the advanced algorithm were effective, they may have difficulties in implementation, due to the fact that industrial robotic cell performance evaluation is a complex problem. Using a standard, effective approach that can be easily carried out is preferred for most industrial applications. The following case study demonstrated how a standard method, a statistical linear regression model, could be used to identify the significant factors that impact manufacturing cell performance.

Case Study: Performance Evaluation of a Robotic Machining Cell

The aim of this study was to evaluate the performance of a robotic manufacturing cell by comparing the computer simulation modeling and statistical regression modeling results with the actual cell performance data on the performance measure of throughput. In addition, this study determined which factors were significantly affecting the accuracy of modeling in a robotic manufacturing cell. Lastly, based on the results, a practical methodology for modeling a robotic machining cell performance is proposed.

Robot-centered Machining Cell Layout

The manufacturing cell analyzed in this study consisted of one robot centered between four CNC machines and a washer, as shown in Figure 1. The cell was designed to produce a specific part family of roughly 12 different parts. In this study, the data from five different parts were used for computer simulation and statistical regression modeling. The parts were introduced into the cell through a pallet system holding a maximum batch of 36 parts. Likewise, the parts were released from the cell through a pallet system holding a maximum batch of 36 parts.

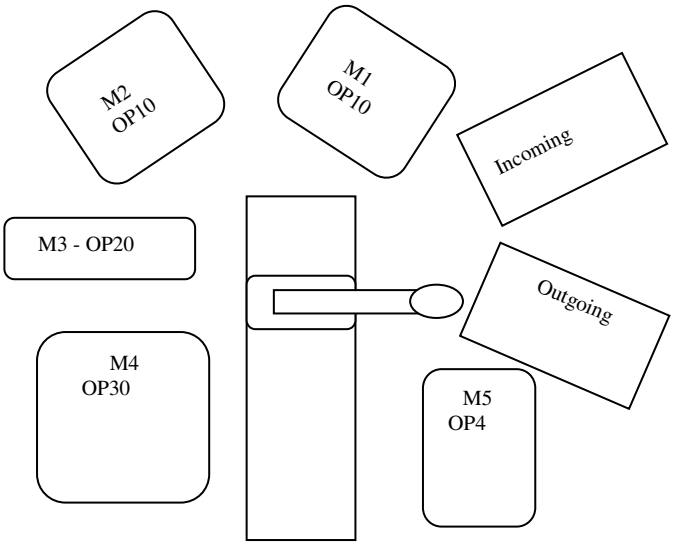

Figure 1. Cell Layout

Parts Processing

The part was processed through four different operations in sequential order, as follows: operation 10 (OP10), operation 20 (OP20), operation 30 (OP30), and operation 40 (OP40). In order to balance the cycle times in the cell, machine 1 (M1) and machine 2 (M2) were identical and performed the same operation, OP10. The cell was capable of processing only one part number at a time. One operator was in charge of operating the cell and his responsibility included setting up machines, changing machine and robotic tooling, trucking in and out materials via the pallet system, performing inspection on all required features per designated frequency, and maintaining a clean and safe cell. For the parts studied, the processing times are presented in Table 1. The highlighted cells in Table 1 came from the constraint machine for the specific part because the processing time at these stations was the longest when parts A, B, C, D, and E were routed in the cell.

Table 1. Part Processing Times per Machine (minutes)

PART #	M1	M2	M3	M4	M5
A	7.415*	7.415*	2.65	1.52	1.96
B	7.6*	7.6*	2.65	1.52	1.96
C	7.82	7.82	2.53	4.493	1.96
D	8.57	8.57	2.65	4.51	1.96
E	7.9	7.9	2.65	4.564	1.96

* M1 and M2 were two identical machines; therefore, the real processing time should be half of the value in the able.

Robot Characteristics and Sequencing

The robot used in this cell was a FANUC R2000iB 165F. This is a standard, floor-mounted industrial robot suitable for material handling applications up to 165 kg in weight. This robot had six individual axes and was mounted on an external track, which gave it seven axes in total. Each axis was powered by a servo motor and the robot was connected to a controller. The end-of-arm tooling was a dual gripper design, capable of handling up to two parts at one time. The grippers were designed to handle all of the parts in the specific part family.

The robot programming capabilities were quite extensive and could change priority in real-time. The sequence had to be followed according to the order of operations. For M1 and M2, the robot would serve whichever machine was requesting a part and whichever machine the operator selected the robot to serve. If both machines requested and selected simultaneously, priority was given to M1. After the robot completed a full cycle with M1 being selected, the robot would serve M2. The main concern from a programming standpoint was to make sure the robot was not the bottleneck in the system.

Table 2 shows the robot sequencing in detail for startup of an empty cell. Once a steady state was achieved, the sequence of looping continuously from steps 27-48 remained. The sequence shows that there were actually seven parts in the system before one part was placed on the outgoing pallet. The capability of the robot programming allowed for this complexity in order to maximize the use of the machines and create a much more efficient cell. The use of dual grippers (GP1 and GP2) allowed for more flexibility and efficiency.

Table 2. Detailed List of Robot Sequence – Startup of Empty Cell

STEP	ROBOT TASK
1	Unload Part 1 From Inc Pallet GP1
2	Load Part 1 in M1 GP1
3	Unload Part 2 From Inc Pallet GP1
4	Load Part 2 in M2 GP1
5	Unload Part 3 From Inc Pallet GP1
6	Unload Part 1 From M1 GP2
7	Load Part 3 in M1 GP1
8	Load Part 1 on Drip Stand1 GP2
9	Unload Part 4 From Inc Pallet GP1
10	Unload Part 2 From M2 GP2
11	Load Part 4 in M2 GP1
12	Load Part 2 on Drip Stand2 GP2
13	Unload Part 5 From Inc Pallet GP1
14	Unload Part 3 From M1 GP2
15	Load Part 5 in M1 GP1
16	Unload Part 1 From Drip Stand1 GP1
17	Load Part 3 on Drip Stand1 GP2
18	Load Part 1 in M3 GP1
19	Unload Part 6 From Inc Pallet GP1
20	Unload Part 4 From M2 GP2
21	Load Part 6 in M2 GP1
22	Unload Part 2 From Drip Stand2 GP1
23	Load Part 4 on Drip Stand2 GP2
24	Unload Part 1 From M3 GP2
25	Load Part 2 in M3 GP1
26	Load Part 1 in M4 GP2
27	Unload Part 7 From Inc Pallet GP1
28	Unload Part 5 From M1 GP2
29	Load Part 7 in M1 GP1
30	Unload Part 3 From Drip Stand1 GP1
31	Load Part 5 on Drip Stand 1 GP2
32	Unload Part 2 From M3 GP1
33	Load Part 3 in M3 GP2
34	Unload Part 1 From M4 GP2
35	Load Part 2 in M4 GP1

Table 2. continued

STEP	ROBOT TASK
36	Load Part 1 in M5
37	Unload Part 8 From Inc Pallet GP1
38	Unload Part 6 From M2 GP2
39	Load Part 8 in M2 GP1
40	Unload Part 4 From Drip Stand2 GP1
41	Load Part 6 on Drip Stand2 GP2
42	Unload Part 3 From M3 GP2
43	Load Part 4 in M3 GP1
44	Unload Part 2 From M4 GP1
45	Load Part 3 in M4 GP2
46	Load Part 2 in M5 GP1
47	Unload Part 1 From M5 GP1
48	Load Part 1 on Out Pallet GP1
	STEADY STATE ACHIEVED

Building of the Simulation Model

The first phase of this study consisted of building a simulation model and assessing its predictability by comparing its results to actual production data. The simulations were executed for longer periods of time (greater than 20 weeks) across five different part numbers.

The simulation model was built using an internally developed interface specific to machining operations in the organization. The programming language used to run the simulation was called Simulation Language Extensible (SLX), a product of the Wolverine Software Corporation. SLX is a general-purpose simulation language similar to the C programming language and is known for its capability and extensibility. The way SLX conducts the simulation is through tracking of discrete events, based on the input data, and outputting statistical summaries. If a simulation model has no programming errors and does not predict actual cell performance well, then most likely the input data are either incorrect or lacking critical variables. It is very difficult to accurately simulate processes without accurate input data. In many cases, the data do not exist, due to a lack of system knowledge regarding data collected over longer periods of time. To determine critical factors affecting the performance of a cell, a cause-and-effect diagram was developed (see Figure 2). The following six independent variables, abstracted from Figure 2, were included as the input data in the simulation model to predict throughput performance, due to their relevance and availability.

- Machine Downtime
- Off-plan Time – (time spent doing things other than running the cell, such as meetings)
- Quality/Scrap Rate
- Total Setup Time/Number of Setups
- Weekly Schedule Requirements
- Preceding Operation Output

The data collection for these variables came from the corporate system software. Queries from the data system and certain departmental reports were utilized in acquiring the data for roughly two months, producing a sample size of 39. From the output of the simulation model, using the SLX platform, cell throughput for parts A, B, C, D, and E was compared with the actual production data in order to assess the accuracy of the simulation model. The comparison showed that the simulation model did not effectively predict robotic performance in terms of throughput (see Table 3). For example, the average daily throughput predicted from the simulation model for Part A, average number of parts unloaded per day, was 163.4. This estimated throughput was relatively close to the average daily throughput expectation of 220 that was based on the actual cycle time standard (ACTS) of 16.5, but the simulation results for the average daily throughput achieved, 77, were very different. The simulation results from SXL for part A is summarized in Table 4.

Table 3. Summary of Throughput for Parts A through E

PART #	ACTS	AVG Daily Throughput Expected	AVG Daily Throughput Simulated	AVG Daily Throughput Achieved
A	15.6	220	163.4	77.0
B	16.0	220	156.5	94.3
C	10.5	180	140.8	75.4
D	11.2	180	124.1	58.6
E	10.1	180	134.6	68.9

Table 4. Simulation Model Results for Part A

General	
Simulation Weeks	50
Total Demand (Parts)	500000
Loads	
Total Number of Parts Loaded	57221
Part Type	Parts Loaded
A	57221
Average Number of Parts Per day	163.49
Unloads	
Total Number of Parts Unloaded	57204
Discarded Parts	0
Part Type	Parts Unloaded
A	57204
Average Number of Parts Per day	163.44

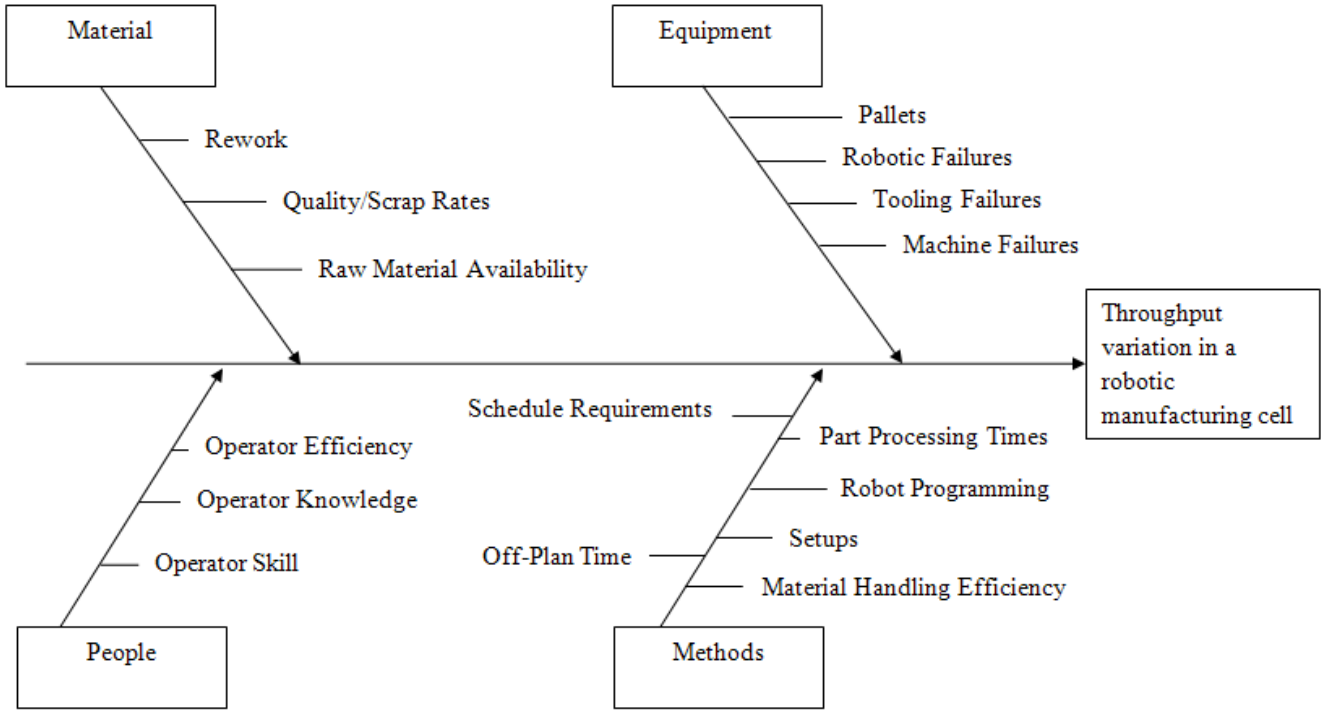

Figure 2. Cause-and-effect Diagram

Building of the Statistical Regression Model

The second approach in this study was to predict cell performance using a multiple regression model. Throughput was modeled as the dependent variable together with six independent variables: machine downtime, off-plan time, scrap rate, setup time, weekly schedule requirements, and preceding operation throughput. Before the regression model was built, redundancy checks were performed to ensure no multicollinearity existed. Multicollinearity is normally checked through the calculation of tolerance or variance inflation factor (VIF), and the two checks will lead to the same conclusion—see Equations (1) and (2). A high tolerance (or low VIF) indicates little or no multicollinearity, with the cutoff value of tolerance higher than 0.20 or 0.1 and a VIF lower than 5 or 10 [13].

$$Tolerance = 1 - R^2 \quad (1)$$

$$VIF = \frac{1}{Tolerance} \quad (2)$$

The results of the checks are shown in Table 5. All of the tolerance numbers in Table 5 are higher than the cutoff value 0.2, which indicated that little or no multicollinearity existed. The predictor variable that was most highly correlated with other predictor variables was preceding operation output, with a tolerance of 0.78. This was expected because this was another measure of output. The preceding operation was a manual manufacturing cell that fed parts to the robotic manufacturing cell in this study. The output produced in the preceding operation could be a critical factor of throughput of the robotic manufacturing cell, thus justifying its inclusion in the study.

Table 5. Results of the Redundancy Check

Variables	Tolerance = 1-R^2	R^2	Partial	Semipart
Off-plan time (hours)	1.0000	0.0000	-0.6448	-0.6448
Weekly schedule requirements (hours)	0.9948	0.0052	0.0728	0.0557
Scrap ($) last two years	0.9545	0.0455	-0.0024	-0.0018
Machine downtime (hours)	0.9776	0.0224	-0.1475	-0.1127
Preceding operation output	0.7847	0.2153	0.0297	0.0227
Setup time (hours)	0.9379	0.0621	-0.3430	-0.2622

Initially, a model with all of the variable effects combined was built. This model, shown in Table 6, only had two significant variables—setup time and off-plan time—besides the intercept, because all three items had small P-values. To ensure that the assumption of normality was valid, a normal probability plot was constructed to show the data distribution (see Figure 3). The plot shows no alarming concerns. Additionally, a predicted versus residuals plot (see Figure 4) was generated to check for constant variance. The graph shows the distribution of points to be generally spread equally and randomly around zero. Therefore, the assumption of constant variance (homoscedasticity) was valid.

Table 6. Multiple Regression Model – Results of All Effects (all input data included)

	SS	DF	MS	F	P-Value
Intercept	49762.81	1	49762.81	32.60159	0.000003
Scrap ($) last two years	4.34	1	4.34	0.00284	0.957811
Off-plan time (hour)	31417.45	1	31417.45	20.58281	0.000076
Setup time (hour)	5340.92	1	5340.92	3.49905	0.070574
Machine downtime (hour)	94.53	1	94.53	0.06193	0.805062
Weekly schedule requirements (hour)	0.09	1	0.09	0.00006	0.993827
Preceding operation output	367.86	1	367.86	0.24100	0.626835
Error	48844.55	32	1526.39		

As this multiple regression model with all variables only has two significant variables (setup time and off-plan time) included, the backward and forward stepwise regression approaches were performed using a P-value of 0.15 as a cutoff value to include or remove a factor. The backward stepwise and forward stepwise procedures are shown in Tables 7 and 8. The regression model results from the two stepwise approaches indicated there were only two out of the six variables, setup time and off-plan time, significantly affecting the throughput of the robotic manufacturing cell.

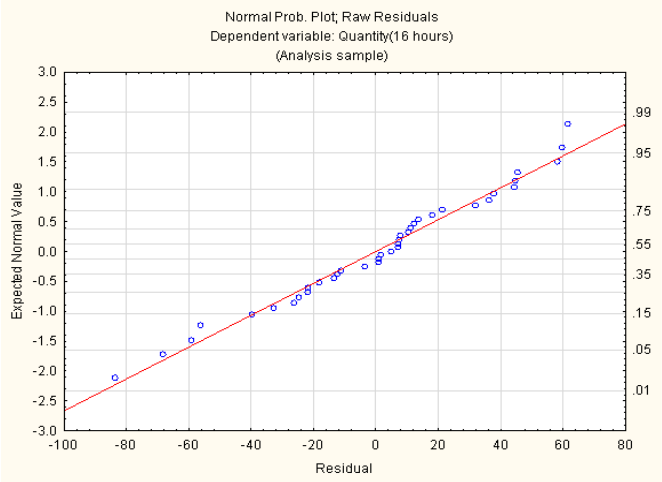

Figure 3. Normal Probability Plot

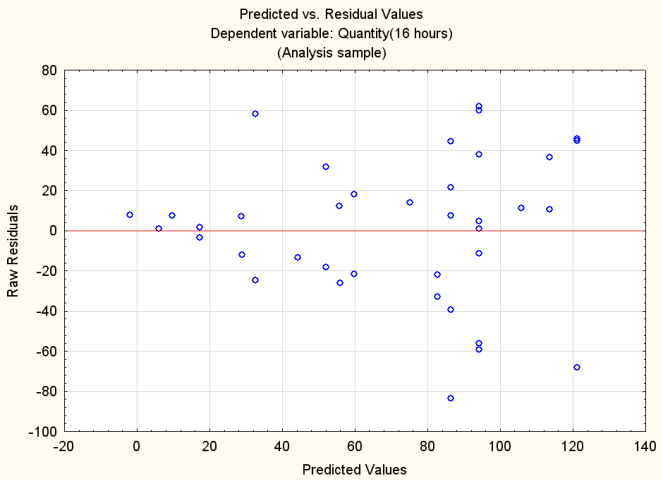

Figure 4. Predicted versus Residuals Plot

Table 7. Multiple Regression Model – Backward Stepwise Results

	Steps	DF	F to	P to Remove	F to	P to Enter	Effect
Scrap ($)	1	1	0.0028	0.9578			In
Off-Plan Time		1	20.5828	0.0001			In
Setup Time		1	3.4991	0.0706			In
Machine Downtime		1	0.0619	0.8051			In
Schedule Requirements		1	0.0001	0.9938			Removed
Preceding OP Output		1	0.2410	0.6268			In
Scrap ($)	2	1	0.0032	0.9553			Removed
Off-Plan Time		1	21.7460	0.0000			In
Setup Time		1	3.8388	0.0586			In
Machine Downtime		1	0.0657	0.7993			In
Preceding OP Output		1	0.2634	0.6112			In
Schedule Requirements		1			0.0001	0.9938	Out
Preceding OP Output	3	1	0.2799	0.6002			In
Off-Plan Time		1	24.9175	0.0000			In
Setup Time		1	4.0699	0.0516			In
Machine Downtime		1	0.0662	0.7985			Removed
Scrap ($)		1			0.0032	0.9553	Out
Schedule Requirements		1			0.0003	0.9857	Out
Preceding OP Output	4	1	0.3071	0.5830			Removed
Off-Plan Time		1	27.6465	0.0000			In
Setup Time		1	4.9802	0.0321			In
Machine Downtime		1			0.0662	0.7985	Out
Scrap ($)		1			0.0018	0.9663	Out
Schedule Requirements		1			0.0014	0.9699	Out
Setup Time	5	1	4.8007	0.0350			In
Off-Plan Time		1	33.2228	0.0000			In
Preceding OP Output		1			0.3071	0.5830	Out
Machine Downtime		1			0.0870	0.7698	Out
Scrap ($)		1			0.0000	0.9953	Out
Schedule Requirements		1			0.0080	0.9293	Out

Furthermore, the final regression model, including only the two significant input variables (off-plan time and setup time) as the independent variables, was built. The final regression model statistics are displayed in Table 9, where F(2, 36) = 16.915 and a very small P-value (<.00001), concluding that the model was significant in predicting the throughput of the robotic manufacturing cell. The multiple regression model also produced an R-squared value of 0.485, which showed that about half of the total variation in throughput could be attributed to these two variables, making them very good predictors of throughput quantity.

Table 9. Regression Statistics for the Final Regression Model

Statistics	Results
F	16.915
P value	<0.00001
Multiple R	0.696458604
R Square	0.485054588
Adjusted R Square	0.455629135
Standard Error	37.50863699
Observations	38

The coefficients of the two predictors and their statistical tests are displayed in Table 10. The variables with statistically significant P-values (less than the cutoff 0.05) are: setup time and off-plan time. They were significant factors in predicting the actual throughout of the robotic manufacturing cell. The fitted regression equation is given in Equation (3):

$$Throughput = 121.5 - 7.74T_{offplan} - 6.66T_{setup} \qquad (3)$$

Table 10. Regression Coefficients

	Coefficients	Standard Error	t Stat	P-value
Intercept	121.5252	12.0949	10.0477	0.0000
Off-Plan Time	-7.7380	1.3618	-5.6823	0.0000
Setup Time	-6.6603	3.1448	-2.1179	0.0350

The equation suggests that both off-plan time and setup time have a negative effect on throughput. The throughput was expected to decrease by 7.74 units with every unit increase in off-plan time, assuming that all other variables were constant. Likewise, throughput was expected to decrease by 6.66 units with every unit increase in setup time, assuming that all other variables were held constant. The prediction results of the regression model are listed in Table 11, which shows that the use of multiple regression modeling was much more accurate predicting throughput performance in a robotic manufacturing cell when compared to computer program simulation modeling. Table 11 also summarizes the performance between different modeling techniques.

Summary and Conclusion

Capacity and production planning can be greatly improved with accurate assessments of performance of robotic manufacturing cells. The ability to predict the performance of a cell is highly beneficial in project planning for new cells and integration of automated manufacturing cells. An inaccurate prediction used in decision making for imple-

Table 8. Multiple Regression Model – Forward Stepwise Results

	Steps	DF	F to	P to Remove	F to	P to Enter	Effect
Scrap	1	1			0.7328	0.3975	Out
Off-Plan Time		1			26.3260	0.0000	Entered
Setup Time		1			0.3251	0.5720	Out
Machine Downtime		1			1.6727	0.2039	Out
Schedule Requirements		1			0.0030	0.9565	Out
Preceding OP Output		1			3.1250	0.0853	Out
Off-Plan Time	2	1	26.3260	0.0000			In
Scrap ($)		1			0.0002	0.9888	Out
Setup Time		1			4.8007	0.0350	Entered
Machine Downtime		1			0.8005	0.3769	Out
Schedule Requirements		1			0.1920	0.6639	Out
Preceding OP Output		1			0.0317	0.8597	Out
Off-Plan Time	3	1	33.2228	0.0000			In
Setup Time		1	4.8007	0.0350			In
Scrap ($)		1			0.0000	0.9953	Out
Machine Downtime		1			0.0870	0.7698	Out
Schedule Requirements		1			0.0080	0.9293	Out
Preceding OP Output		1			0.3071	0.5830	Out

mentation of new cells can cause organizations to lose a lot of time, resources, and money. This study focused on assessing the performance of a robotic manufacturing cell through computer language-based simulation modeling and general regression model building in an effort to determine the most significant variables that affect the performance of the cell, as measured by throughput.

- The comparison showed that the computer language-based simulation model was not predictive of actual output by consistently predicting much higher output amounts, but the statistical regression model performed better in terms of generating approximately close results.
- Investigations were performed using forward and backward stepwise approaches in order to determine the most significant factors affecting the throughput of a robotic manufacturing cell, which turned out to be off-plan time and machine setup time.
- It was also hypothesized that weekly schedule requirements are a significant factor of throughput. However, the results of the study showed otherwise.
- The practical implications of this study suggest that a lot of time is being spent doing setups and off-plan activities such as meetings, projects, and working in other areas, which negatively impact the productivity of manufacturing cells. These activities should be reduced as much as possible.

Table 11. Summary of Regression Prediction on the Throughput

PART #	Avg Daily Throughput Expected	Avg Daily Throughput Simulated	Avg Reg Model Prediction	Avg Daily Throughput Achieved
A	220	163.4	88.5	77
B	220	156.5	88.5	94.3
C	180	140.8	88.5	75.4
D	180	124.1	88.5	58.6
E	180	134.6	88.5	68.9

In a highly dynamic automated manufacturing cell, simulation modeling can be greatly beneficial for effectively and efficiently pinpointing bottlenecks and constraints in the process. However, the success of simulation modeling is highly dependent on the quality of input data. The majority of the time, when data do not exist, simulation modeling can be restricted. Therefore, simulation modeling presenting best-case scenarios and predictions may not be applied as a universal tool, when the lack of input data cannot be obtained or simulated. In the presented case study, the use of real data to build statistical models provided much more accurate results. As a result, an integrated methodology of simulation modeling and statistical regression modeling, that can provide more accurate predictions for planning future cells, is suggested.

Future work is recommended to include the study of the same group of variables, as well as additional quantifiable variables, over longer periods of time. Duplication of this study in other organizations and environments is encouraged. In addition, the use of an expert or knowledge-based system can be used to acquire accurate input data that may be used to create a more predictive simulation model.

References

[1] Popkin, J., & Kobe, K. (2010). Manufacturing Resurgence — A Must for U.S. Prosperity. Retrieved August 18, 2013, from http://www.nam.org/~/media/F36EC9F57BFF4DA4AEBAFAAB4B009B92/Popkin_Report.pdf

[2] Brown, A. S. (2008). Robots near the Million Mark. *Mechanical Engineering-CIME*, *130*(8), 11.

[3] Ahkioon, S., Bulgak, A. A., & Bektas, T. (2009). Cellular Manufacturing Systems Design with Routing Flexibility, Machine Procurement, Production Planning and Dynamic System Reconfiguration. *International Journal of Production Research, 47*(6), 1573-1600.

[4] Bangert, M. (2009). Robots Inspect It All. *Quality Magazine, 48*(5), 36-41.

[5] Das, K., & Abdul-Kader, W. (2011). Consideration of Dynamic Changes in Machine Reliability and Part Demand: A Cellular Manufacturing Systems Design Model. *International Journal of Production Research, 49*(7), 2123-2142.

[6] Yazici, H. J. (2005). Influence of Flexibilities on Manufacturing Cells for Faster Delivery Using Simulation. *Journal of Manufacturing Technology Management, 16*(8), 825-841.

[7] Djassemi, M. (2005). A Simulation Analysis of Factors Influencing the Flexibility of Cellular Manufacturing. *International Journal of Production Research, 43*(10), 2101-2111.

[8] Chang, A. Y. (2007). On the Measurement of Routing Flexibility: A Multiple Attribute Approach. *International Journal of Production Economics, 109*(1–2), 122-136.

[9] Gultekin, H., Aktruk, M. S., & Karasan, O. E. (2008). Scheduling in Robotic Cells: Process Flexibility and Cell Layout. *International Journal of Production Research, 46*(8), 2105-2121.

[10] Geismar, N. H., & Pinedo, M. (2010). Robotic Cells with Stochastic Processing Times. *IIE Transactions, 42*, 897-914.
[11] Abd, K., Abhary, K., & Marian, R. (2012). Scheduling and Performance Evaluation of Robotic Flexible Assembly Cells under Different Dispatching Rules. *Advances in Mechanical Engineering, 2*(1), 30-39.
[12] Shafiei-Monafared, S., Salehi-Gilani, K., & Jenab, K. (2009). Productivity Analysis in a Robotic Cell. *International Journal of Production Research, 47*(23), 6651-6662.
[13] O'Brien, R. M. (2007). A Caution Regarding Rules of Thumb for Variance Inflation Factors. *Quality & Quantity*, *42*(5), 673-690.

Biographies

JULIE Z. ZHANG is an Associate Professor in the Technology Department at the University of Northern Iowa. She teaches courses in manufacturing processes, materials, automation, statistical quality control, total quality management, and CAD/CAM applications. Her research includes green manufacturing and evaluation of bio-based cutting fluids, real-time cutting tool condition monitoring, and adaptive control for automated machines. Dr. Zhang may be reached at juli.zhang@uni.edu

ALI E. KASHEF is a Professor in the Technology Department at the University of Northern Iowa. He is the coordinator for the Technology Management program. He teaches in the manufacturing and technology management programs. His research interests are in the areas of quality, environmental waste, lean manufacturing, and management. Dr. Kashef may be reached at Kashef@uni.edu

MARK ECKER is a Professor in the Mathematics Department at the University of Northern Iowa, where he teaches hands-on data analytic techniques. His expertise is in statistical analysis. His research involves creating methodologies for spatial correlation often present in environmental and economic data; often requiring significant computer programming and run times to fit complex models to real datasets. Dr. Ecker may be reached at mark.ecker@uni.edu

ELVIS ALICIC obtained his M.S. in Industrial Technology from the University of Northern Iowa in 2012 and a B.A. in Computer Information Systems from Wartburg College. He has provided automation technology, programming support, and robotics solutions across multiple factories in a manufacturing corporation, and became one of the lead engineers in the design and implementation of many robotic machining cells. He has expertise in robotic vision systems and dynamic process simulation.

NEURAL NETWORKS FOR CONTROL OF HEPATITIS B VIRUS INFECTION

Samir Talssi, Hassan II University, Mohammedia-Casablanca; Noura Yousfi, Hassan II University, Mohammedia-Casablanca

Abstract

According to the World Health Organization (WHO), hepatitis B kills more than 780,000 people each year. Another two and a half billion people are infected with the virus and 370 million have long-term chronic infections. In this study, the authors developed a new approach for optimal control of hepatitis B drug treatment using neural networks. A neural network was used as a tool for decision support that had to be taken at each step during treatment processing.

Mathematical models of hepatitis B drug treatment exist in literature; one of these models was used to implement this new approach and test its effectiveness. In inverse control models, network training is performed using preprocessed data generated through a semi-implicit finite difference method. The error back-propagation algorithm was used in the learning step in order to adjust parameters of the neural networks. Simulation provided interesting results, presented in the form of curves for easier comparison.

Introduction

The VHB (hepatitis B virus) is a highly contagious virus; 10 times more than hepatitis C and 100 times more than HIV. Hepatitis B is the 10th leading cause of death in the world. It is a viral infection caused by the VHB virus that attacks the liver and destroys liver cells with the appearance of cirrhosis or liver cancer [1-4]. Several mathematical models have been developed to understand the dynamics of VHB and its behavior facing a drug therapy [5-9].

In recent years, new techniques have been developed to identify and control the dynamic systems. Artificial neural networks (ANNs) is one of these techniques that has gained great attention in control applications. Its ability to identify a nonlinear system has been exploited in the synthesis of nonlinear controllers and to make a variety of control strategies. In this paper, the authors present a quick overview of neural networks. The approach used in this study consisted of the control of infection therapy, based on the support provided by neural networks. An optimal hepatitis B drug treatment model introduced by Hattaf et al. [7] was used as an example for testing the effectiveness of this approach in simulation studies.

Multilayer Neural Networks

Several types of neural networks have been introduced in the literature, but multilayer networks are most widely used in control applications. In a multilayer neural network (see Figure 1), the neurons are arranged in distributed layers between an input layer and an output layer.

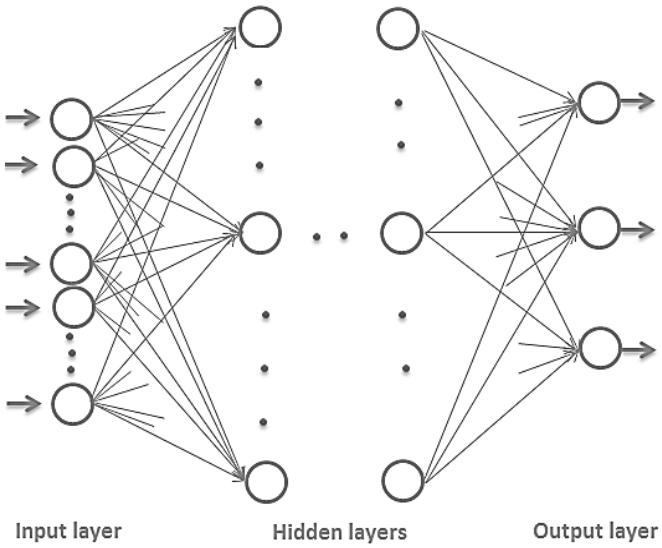

Figure 1. Multi-layer Neural Network

Neural networks are universal approximate, which was proven by Kùrková [10] and Brattka [11] using the classic mathematical result of Kolmogorov, which states that for any continuous mapping there must exist a three-layer neural network that implements f exactly:

$$f : [0,1]^n \subset R^n \rightarrow R^m$$

Cybenko [12] and Hornik et al. [13] proved that a single hidden layer feedforward neural network with a non-linear activation in the hidden layer and a linear output activation can approximate any continuous functions within the meaning of the uniform norm on compact; Barron [14] indicated the required number of neurons to approach certain classes of functions with a fixed precision. The use of an ANN in systems control was introduced by Narendra and Parthasarathy [15] and Werbos [16]. Neural networks have been grouped into two families in control frameworks:

- The use of neural networks as an approximate to be surrounded by a control strategy [17], [18].
- The use of neural networks as controllers and classic controllers [19-22].

The Inverse Model of a Nonlinear System of Differential Equations

The inverse model using a neural network was trained to approximate a system inverse model, which provided the necessary control parameters to bring the system into a desired state. A dynamic system can be described by Equation (1) connecting its inputs to outputs:

$$y(n+1) = f\left(y(n),\ldots,y(n-l+1),u(n),\ldots,u(n-m+1)\right) \quad (1)$$

The system output, y(n + 1), depends on the previous *l* values outputs and *m* previous values of the inputs. Equation (2) presents the form of the corresponding system inverse model:

$$u(n) = f^{-1}\left(y(n+1), y(n),\ldots, y(n-l+1), u(n-1), u(n-m+1)\right) (2)$$

where, y(n + 1) is unknown, but when substituted by a desired value such as $y_d(n + 1)$, Equation (2) can be rewritten as Equation (3):

$$u(n) = f^{-1}\left(y_d(n+1), y(n),\ldots, y(n-l+1), u(n-1), u(n-m+1)\right) (3)$$

Control Based on a Recurrent Neural Network Inverse Model

Several types of neural networks can be used to present the inverse model [15], [23]. The static multi-layer perceptron networks offer the simplest solution; however, the representation of the dynamic aspect of the system remains a problem. The solution is presented by the application of delay to the input layer to overcome this gap. The solution allows the application of the traditional gradient backpropagation algorithm for multilayer network learning. Generally, there are two methods of feedback control topologies: direct architecture (see Figure 2) and indirect architecture (see Figure 3). Solid lines denote control signal flow loops, while dashed lines denote tuning loops.

In indirect architecture control (see Figure 2), there are two functions. In the identifier block, the neural network is tuned to learn the dynamics of the unknown plant; the controller block uses this information to control the system. In the tuning loops, the weight adjustment is made using an error gradient backpropagation algorithm.

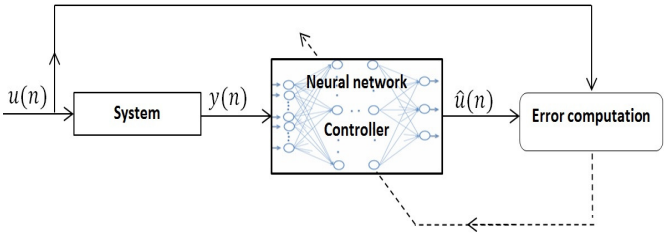

Figure 2. Direct Architecture

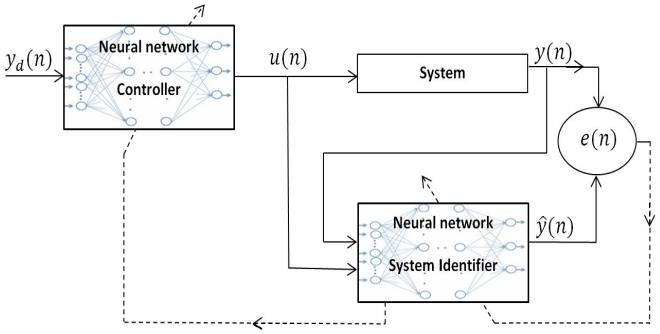

Figure 3. Direct Architecture

Learning Algorithm: Error Backpropagation Gradient

The modified back-propagation method was first formalized in 1974 by Werbos [16] and later by Parker [24] and Rumelhart et al. [25]. It has since been modified in order to adapt to a given situation or to accelerate its convergence [18], [26]. The modified back-propagation method presented by Talssi and Yousfi [18] and Rumelhart et al. [25] was used to ensure and accelerate the algorithm's convergence in the training step of the presented approach. In this version of back-propagation, the neural network weights were not updated after each pattern was presented. Rather, the weights were modified only after all training input patterns were presented (or after presenting some of the data); the change was made using the calculated error, as shown in Figure 4.

Changes in weight were calculated using Equation (4):

$$\Delta w_{ji}(t+1) = \eta\delta_{pj}a_{pi} + \alpha.\Delta w_{ji}(t-1) \quad (4)$$

where, w_{ji} is the weight on the connection from input *i* to neuron *j*; η is the learning rate; α is the momentum factor; and, δ is the error signal. The hidden neuron error signal, δ_{pj}, was computed using Equation (5):

$$\delta_{pj} = a_{pj}\left(1-a_{pj}\right)\sum_{x}\delta_{px}w_{xj} \quad (5)$$

where, signal δ_{px} is the signal error in the next neuron x and w_{xj} is the weight of the connection from the hidden neuron, j, to the next neuron, x. In the output neuron, j, signal error, δ_{pj}, was computed using Equation (6):

$$\delta_{pj} = a_{pj}\left(1 - a_{pj}\right)\left(t_{pj} - a_{pj}\right) \tag{6}$$

where, t_{pj} is the target value of output neuron j for pattern p and a_{pj} is the actual output value of output neuron j for pattern p.

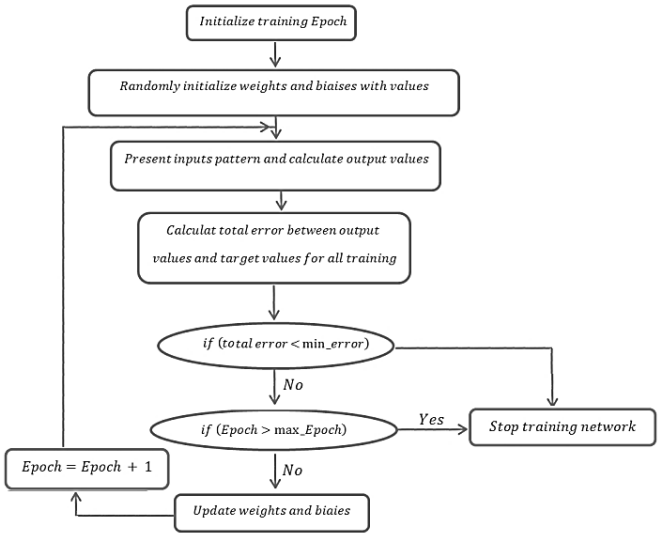

Figure 4. Error Backpropagation Algorithm

Neural Network Approach

In the direct architecture of inverse control using neural networks (see Figure 2), the inverse control of drug therapy of hepatitis B is identified next. The inverse neural model's identification begins by fixing the input vector (the number of delays in inputs and outputs). In a second step, the architecture of the network is selected (the number of hidden layers and the number of neurons per layer) using the results presented by Vogl et al. [26]. The network parameters were performed during the learning steps using the error backpropagation algorithm (see Figure. 4). The objective was to control the state of the system, Equation (7), in order to reduce the cost of treatment and maximize the benefit represented by an objective function [8].

Simulation Example

To implement the proposed method, an example of a mathematical model for optimal hepatitis B treatment was introduced by Hattaf et al. [7]. This mathematical model contained the following variables: uninfected target cells (T), infected cells (I), and free virions (V). The model is given by the nonlinear system of differential equations shown in Equation (7):

$$\begin{cases} \dfrac{dT}{dt} = \lambda - dT - \left(1 - u_1(t)\right)\beta VT, \\ \dfrac{dI}{dt} = \left(1 - u_1(t)\right)\beta VT - \delta I, \\ \dfrac{dV}{dt} = \left(1 - u_2(t)\right)pI \end{cases} \tag{7}$$

$$\left(T(0) = T_0, I(0) = I_0 \; and \; V(0) = V_0 \; are \; given\right)$$

The control $u_2(t)$ represents the efficiency of drug therapy in inhibiting viral production; the control $u_1(t)$ represents the efficiency of drug therapy in blocking new infections. The model parameters are: death rate of target cells (d), death rate of infected cells (δ), clearance rate of free virions (c), rate of production of virions per infected cell (p), rate of infection of new target cells (β), rate of production of new target cells (λ).

Simulation Results

A Semi-Implicit Finite Difference Method (SIFDM) was described by Hattaf et al. [7] for obtaining the optimal control of the model in Equation (7). In this simulation, the SIFDM was used to generate learning data; the data were pre-processed, similar to how it was done in the study by Talssi and Yousfi [18], before starting the learning procedure. The training examples were the quintuplets (T, I, V, u_1, u_2), where T represented uninfected target cells, I represented infected cells, and V was a free virus. The triplets (T, I, V) were presented in the entries of the learning process and (u_1, u_2), the parameter of control, represented the target output.

The SIFDM and the neural network approach were implemented using C++. With 2000 training examples, the gradient backpropagation algorithm was successfully adjusted to the parameters of the network after several attempts. At each of these attempts, the network's settings were randomly reinitialized to give a new stimulus to the calculations. Finally, the learning process was successful in achieving an acceptable accuracy. Using the semi-implicit finite-difference method, 300 training examples were generated for testing the network training performance. To validate and visualize the results of this approach, 1000 examples for (T, I, V) were generated by the SIFDM. Another 1000 training examples for (u_1, u_2) were generated using the obtained network. The results are compared in Figures 5 and 6.

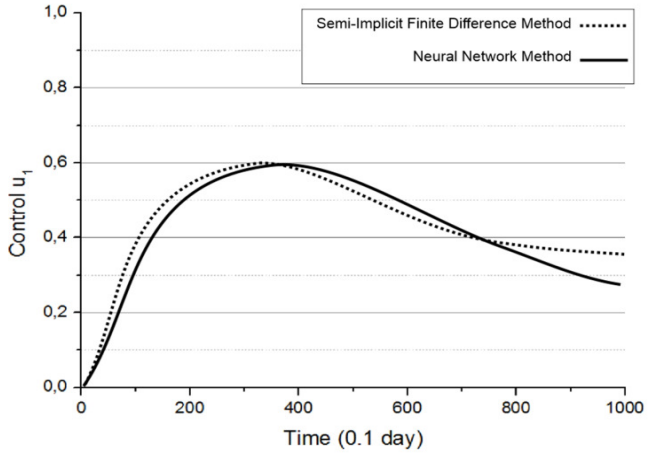

Figure 5. The Control μ_1

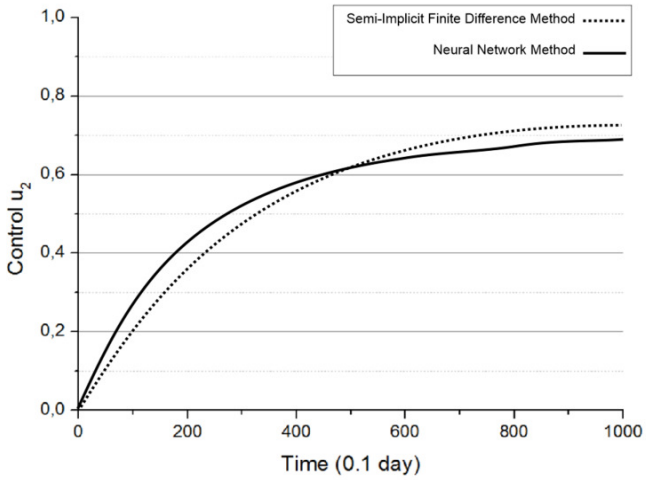

Figure 6. The Control μ_2

With 1000 generated examples for (u_1, u_2) using the networks, 1000 examples for (T, I, V) were computed using the SIFDM. The results are given in Figures 7-9.

The hatched curves represent results obtained using the SIFDM method; the other continuous curves represent the results obtained using the ANN. The identification of the SIFDM using the ANN allowed for curves having the same behaviour in all of the figures. An important point is that the control parameters in this study represented the cost of treatment and it is clear that the curves in Figure 5 and 6 were generated by the ANN parameters representing the controls, which are often located below the curves generated by SIFDM. This means that the ANN can be a good tool for making decisions during treatment to reduce the cost of treatment.

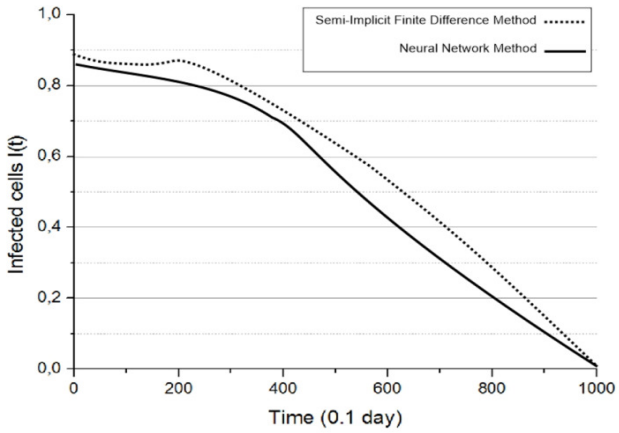

Figure 7. Function I

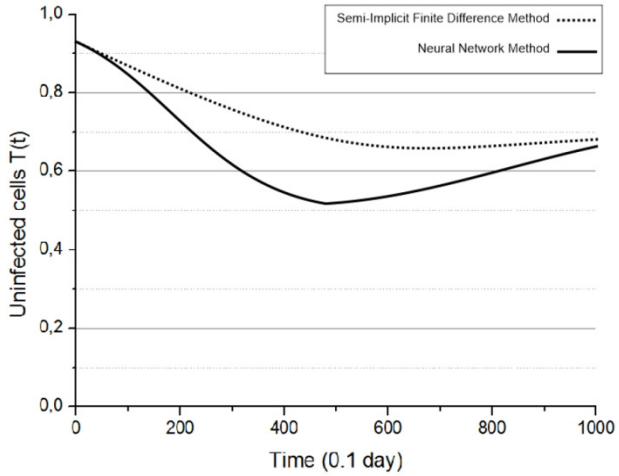

Figure 8. Function T

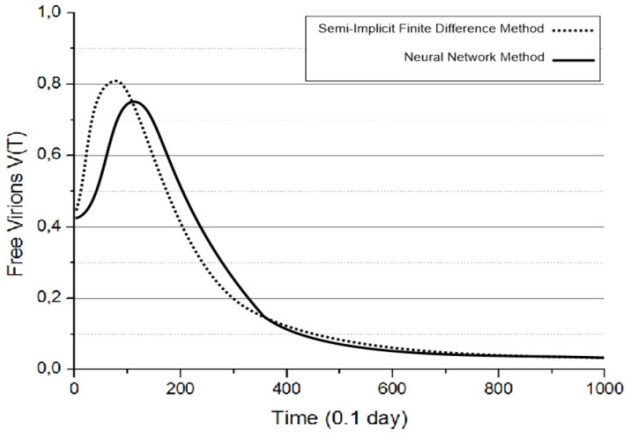

Figure 9. Function V

The results shown in Figures 7-9 illustrate the neural network's capability to identify and provide the right control parameters, which lead to the desired results. Figure 7 represents the rate of the infected cells and Figure 9 represents the rate of free virions; the decay of those curves over time illustrates the effectiveness of treatment. One of the drawbacks encountered in the implementation of this approach was the duration of the network's learning phase; the convergence of the backpropagation algorithm was not always guaranteed, otherwise good modification of algorithm parameters and preprocessing training data [18], [27], [28] would reduce the learning time.

Conclusions

An inverse control approach based on neural networks was developed in this study to optimize hepatitis B drug therapy. The results of the proposed neural network approach were compared with an SIFDM simulation; no such study was found in the literature. The statistics calculated during treatments were approximate, though not exact, and the neural network was able to overcome the inaccuracy or the lack of data. Generally, most VHB-related deaths result from chronic infections and infections acquired during periods of perinatal and early childhood. Inclusion of the vaccine against hepatitis B in the national infant immunization programs could prevent up to 80% of VHB-related deaths. Until that time comes, an optimized drug treatment plan would be of great importance, and the integration of an approach like the one presented here is necessary. This approach can be used as a tool for fast decision making to optimize the cost and treatment time, which are due to the inaccuracy or the lack of data.

References

[1] Blumberg, B. S. (2002). *Hepatitis B: The Hunt for a Killer Virus*. Princeton University Press.

[2] Ganem, D., & Prince, A. M. (2004). Hepatitis B Virus Infection-Natural History and Clinical Consequences. *New England Journal of Medicine*, *350*(11), 1118-1129.

[3] Ly, K. N., Xing, J., Klevens, R. M., Jiles, R. B., & Holmberg, S. D. (2014). Causes of Death and Characteristics of Decedents with Viral Hepatitis, United States, 2010. *Clinical Infectious Diseases*, *58*(1), 40-49.

[4] World Health Organization. (2012). *Guidance on Prevention of Viral Hepatitis B and C Among People Who Inject Drugs*. World Health Organization.

[5] Goldstein, S. T., Zhou, F., Hadler, S. C., Bell, B. P., Mast, E. E., & Margolis, H. S. (2005). A Mathematical Model to Estimate Global Hepatitis B Disease Burden and Vaccination Impact. *International Journal of Epidemiology*, *34*(6), 1329-1339.

[6] Gourley, S. A., Kuang, Y., & Nagy, J. D. (2008). Dynamics of a Delay Differential Equation Model of Hepatitis B Virus Infection. *Journal of Biological Dynamics*, *2*(2), 140-153.

[7] Hattaf, K., Rachik, M., Saadi, S., & Yousfi, N. (2009). Optimal Control of Treatment in a Basic Virus Infection Model. *Applied Mathematical Sciences*, *3*(17-20), 949-958.

[8] Min, L., Su, Y., & Kuang, Y. (2008). Mathematical Analysis of a Basic Virus Infection Model with Application to HBV infection. *Journal of Mathematics*, *38*(5).

[9] Thornley, S., Bullen, C., & Roberts, M. (2008). Hepatitis B in a High Prevalence New Zealand Population: A Mathematical Model Applied to Infection Control Policy. *Journal of Theoretical Biology*, *254* (3), 599-603.

[10] Kùrková, V. (1992). Kolmogorov's Theorem and Multilayer Neural Networks. *Neural Networks*, *5*(3), 501-506.

[11] Brattka, V. (2003). A Computable Kolmogorov Superposition Theorem. *Computability and Complexity in Analysis. Informatik Berichte*, *272*, 7-22.

[12] Cybenko, G. (1989). Approximation by Superpositions of a Sigmoidal Function. *Mathematics of Control, Signals and Systems*, *2*(4), 303-314.

[13] Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer Feedforward Networks are Universal Approximators. *Neural Networks*, *2*(5), 359-366.

[14] Barron, A. R. (1993). Universal Approximation Bounds for Superpositions of a Sigmoidal Function. *IEEE Transactions on Information Theory*, *39*(3), 930-945.

[15] Narendra, K. S., & Parthasarathy, K. (1990). Identification and Control of Dynamical Systems using Neural Networks. *IEEE Transactions on Neural Networks, 1*(1), 4-27.

[16] Werbos, P. (1974). Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences. Unpublished doctoral dissertation, Harvard University.

[17] Jin-quan, H., & Lewis, F. L. (2003). Neural-Network Predictive Control for Nonlinear Dynamic Systems with Time-Delay. *IEEE Transactions on Neural Networks*, *14*(2), 377-389.

[18] Talssi, S., & Yousfi, N. (2014). Optimality Identification in Epidemiology Using Neural Networks. *Proceedings of the 2014 IAJC-ISAM International Conference.* Orlando, Florida.

[19] Irwin, G. W., Warwick, K., & Hunt, K. J. (Eds.). (1995). *Neural Network Applications in Control* (No. 53).
[20] Miller, W. T., Werbos, P. J., & Sutton, R. S. (Eds.). (1995). *Neural Networks for Control*. MIT Press.
[21] Hunt, K. J., Sbarbaro, D., Żbikowski, R., & Gawthrop, P. J. (1992). Neural Networks for Control Systems - A Survey. *Automatica*, *28*(6), 1083-1112.
[22] Hunt, K. J., & Sbarbaro, D. (1991). Neural Networks for Nonlinear Internal Model Control. *IEE Proceedings of Control Theory and Applications*, *138*(5), 431 -438.
[23] Wolpert, D. M., & Kawato, M. (1998). Multiple Paired Forward and Inverse Models for Motor Control. *Neural Networks*, *11*(7), 1317-1329.
[24] Parker, D. B. (1987). Optimal Algorithms for Adaptive Networks: Second Order Back Propagation, Second Order Direct Propagation, and Second Order Hebbian Learning. *Proceedings of the IEEE International Conference on Neural Networks* (pp. 593-600).
[25] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1985). *Learning Internal Representations by Error Propagation.* (No. ICS-8506).
[26] Vogl, T. P., Mangis, J. K., Rigler, A. K., Zink, W. T., & Alkon, D. L. (1988). Accelerating the Convergence of the Back-Propagation Method. *Biological Cybernetics*, *59*(4-5), 257-263.
[27] Hecht-Nielsen, R. (1989). Theory of the Back-Propagation Neural Network. *International Joint Conference on Neural Networks,* (pp. 593-605).
[28] Shanker, M., Hu, M. Y., & Hung, M. S. (1996). Effect of Data Standardization on Neural Network Training. *Omega*, *24*(4), 385-397.

Biographies

SAMIR TALSSI is a Professor in the higher institutes of applied engineering, IGA-Casablanca. His interest area is artificial intelligence and its application in epidemiology. Dr. Talssi was a member of the High Council Education in Morocco and has over 10 years of experience as a manager and educator. Professor Talssi may be reached at samirtalssi@gmail.com

NOURA YOUSFI is currently a Professor in the Department of Mathematics and Computer Science in Mohammedia Hassan II University. She is the director of several doctoral student research projects. Dr. Yousfi has over 20 years of experience as a scientist, manager, and teacher. She is also a biomathematics specialist. Dr. Yousfi may be reached at nourayoussfi@hotmail.com

INSTRUCTIONS FOR AUTHORS: MANUSCRIPT REQUIREMENTS

The INTERNATIONAL JOURNAL OF MODERN ENGINEERING is an online/print publication, designed for Engineering, Engineering Technology, and Industrial Technology professionals. All submissions to this journal, submission of manuscripts, peer-reviews of submitted documents, requested editing changes, notification of acceptance or rejection, and final publication of accepted manuscripts will be handled electronically. The only exception is the submission of separate high-quality image files that are too large to send electronically.

All manuscript submissions must be prepared in Microsoft Word (.doc or .docx) and contain all figures, images and/or pictures embedded where you want them and appropriately captioned. Also, for all accepted manuscripts, each figure, image or picture that was imported into your Word document must be saved individually as a **300dpi or higher JPEG (.jpg)** file and submitted separately; your manuscript and figure numbers must be used in the title of the file (e.g., **M13-F-18 Figure 4**); that means one additional file for each image imported into your manuscript. These 300dpi images do NOT need to be embedded in your manuscript. For tables or graphs created directly in Word, you do not need to submit them as separate files.

Included below is a summary of the formatting instructions. You should, however, review the sample Word document on our website (www.ijme.us/formatting_guidelines) for details on how to correctly format your manuscript. The editorial staff reserves the right to edit and reformat any submitted document in order to meet publication standards of the journal.

The references included in the References section of your manuscript must follow APA-formatting guidelines. In order to help you, the sample Word document also includes numerous examples of how to format a variety of scenarios. Keep in mind that an incorrectly formatted manuscript will be returned to you, a delay that may cause it (if accepted) to be moved to a subsequent issue of the journal.

1. **Word Document Page Setup**: Two columns with ¼" spacing between columns; Top of page = ¾"; Bottom of page = 1" (from the top of the footer to bottom of page); Left margin = ¾"; Right margin = ¾".

2. **Paper Title**: Centered at the top of the first page with a 22-point Times New Roman (Bold), Small-Caps font.

3. **Page Breaks**: Do not use page breaks.

4. **Body Fonts**: Use 10-point Times New Roman (TNR) for body text throughout (1/8" paragraph indention); 9-point TNR for author names/affiliations under the paper title; 16-point TNR for major section titles; 14-point TNR for minor section titles; 9-point TNR BOLD for caption titles for tables and figures; other font sizes as noted in the sample document.

5. **In-text Referencing**: List and number each reference when referring to them in the body of your document (e.g., [1]). The first entry must be [1] followed by [2], [3], etc., continuing in numerical order to the final entry in your References section. Again, see the sample Word document for specifics. Do not use the End-Page Reference utility in Microsoft Word. You must manually place references in the body of the text.

6. **Tables and Figures**: Center all tables and figures. Captions for tables must be above the table, while captions for figures are below; all captions are left-justified.

7. **Page Limit**: Manuscripts should not be more than 15 pages (single-spaced, 2-column format).

8. **Page Numbering**: Do not use page numbers.

9. **Publication Charges:** Manuscripts accepted for publication are subject to mandatory publication charges.

10. **Copyright Agreement:** A Copyright Transfer Form must be signed and submitted by all authors on a given paper before that paper will be published.

11. **Submissions:** All manuscripts and associated files must be submitted electronically.

    MANUSCRIPTS should be submitted to Dr. Philip D. Weinsier, manuscript editor, at philipw@bgsu.edu along with a copy to editor@ijme.us.

    FILES containing your high-quality images should ONLY be submitted to philipw@bgsu.edu.

CUTTING EDGE JOURNAL OF RESEARCH AND INNOVATION IN ENGINEERING

Print ISSN: 2152-4157
Online ISSN: 2152-4165

www.ijeri.org

www.iajc.org

INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH AND INNOVATION

ABOUT IJERI:

- IJERI is the second official journal of the International Association of Journals and Conferences (IAJC).
- IJERI is a high-quality, independent journal steered by a distinguished board of directors and supported by an international review board representing many well-known universities, colleges, and corporations in the U.S. and abroad.
- IJERI has an impact factor of **1.58**, placing it among an elite group of most-cited engineering journals worldwide.

OTHER IAJC JOURNALS:

- The International Journal of Modern Engineering (IJME) For more information visit www.ijme.us
- The Technology Interface International Journal (TIIJ) For more information visit www.tiij.org

IJERI SUBMISSIONS:

- Manuscripts should be sent electronically to the manuscript editor, Dr. Philip Weinsier, at philipw@bgsu.edu.

  For submission guidelines visit www.ijeri.org/submissions

TO JOIN THE REVIEW BOARD:

- Contact the chair of the International Review Board, Dr. Philip Weinsier, at philipw@bgsu.edu.

  For more information visit www.ijeri.org/editorial

INDEXING ORGANIZATIONS:

- IJERI is currently indexed by 16 agencies. For a complete listing, please visit us at www.ijeri.org.

Contact us:

Mark Rajai, Ph.D.

Editor-in-Chief
California State University-Northridge
College of Engineering and Computer Science
Room: JD 4510
Northridge, CA 91330
Office: (818) 677-5003
Email: mrajai@csun.edu

www.tiij.org

www.ijme.us

THE LEADING JOURNAL OF ENGINEERING, APPLIED SCIENCE AND TECHNOLOGY

The latest impact factor (IF) calculation (Google Scholar method) for IJME of 3.0 moves it even higher in its march towards the top 10 engineering journals.

IJME IS THE OFFICAL AND FLAGSHIP JOURNAL OF THE INTERNATIONAL ASSOCATION OF JOURNALS AND CONFERENCE (IAJC)
www.iajc.org

The International Journal of Modern Engineering (IJME) is a highly-selective, peer-reviewed journal covering topics that appeal to a broad readership of various branches of engineering and related technologies. IJME is steered by the IAJC distinguished board of directors and is supported by an international review board consisting of prominent individuals representing many well-known universities, colleges, and corporations in the United States and abroad.

IJME Contact Information

General questions or inquiries about sponsorship of the journal should be directed to:

Mark Rajai, Ph.D.
Editor-in-Chief
Office: (818) 677-5003
Email: editor@ijme.us
Department of Manufacturing Systems Engineering & Management
California State University-Northridge
18111 Nordhoff St.
Northridge, CA 91330