

AUTOMATING THE DEVELOPMENT OF SHIPYARD MANUFACTURING MODELS

Gabriel A. Burnett
D. J. Medeiros

Industrial & Manufacturing Engineering
The Pennsylvania State University
University Park, PA, 16802, U.S.A.

Daniel A. Finke
Mark T. Traband

Applied Research Laboratory
The Pennsylvania State University
State College, PA 16804, U.S.A.

ABSTRACT

Simulation results are often needed within a short time frame, while the development of simulation models can be time consuming. We develop a methodology to facilitate rapid generation of simulation models from an enterprise database. Data is communicated between Product Lifecycle Management (PLM) software and Flexsim using a standard Microsoft Excel format. We have developed a custom Flexsim interface and software-specific model generator that creates a discrete event simulation model from the PLM input data. Preliminary results show that the methodology can reduce the cost of simulation model generation while simultaneously improving the accuracy of generated models. This work highlights the benefits of automatic model generation techniques, describes a shipbuilding implementation of the methodology, and provides direction for future work.

1 INTRODUCTION

1.1 Problem Statement

The major objective was to develop a methodology for automating the creation of discrete event simulation models. The goal was to produce an easy to use tool for small shipyards, which frequently do not have expertise in simulation. Specific goals for the simulation were that it allow for easy user interaction, that it use commercially available simulation software, and that it produce realistic graphics. Although the research is geared towards shipyards or other low volume high value manufacturing systems, the concept is relevant to a wide variety of applications.

1.2 Target Environment

The design process was guided by explicit assumptions regarding target environment and expected simulation model use. The target environment is that of a small shipyard producing complex, low-volume assemblies according to a

known schedule. Characteristics of the target environment are as follows:

- The system manufactures complex assemblies comprising discrete parts.
- Processing occurs in work centers with varying capabilities.
- Parts are stored in finite capacity buffers.
- Material handling is provided by human operators, forklifts, and cranes.
- Information concerning the capabilities of operators and transporters (i.e. what tasks they are capable of, how fast they move, and what locations they serve) is known.
- Part arrival times, routings, process requirements, and processing times are known.

Simulation models are driven by input data. These data have varying levels of detail; therefore it is necessary to determine an appropriate level that is consistent throughout the model. The implementation and resultant data requirements depend on the desired use of the model, ours was designed to answer a specific class of questions focused on system performance in meeting due dates.

The system generates models appropriate for evaluating schedules and analyzing expected system performance (e.g. closeness to due date). Note that the assumption of known processing times implies deterministic processing times. Deterministic times were used because we are focused on high level scheduling and information concerning random distributions does not exist for the target systems.

1.3 Shipyard Manufacturing Simulation

Discrete Event Simulation (DES) has been increasingly used in the US to analyze manufacturing systems unique to the shipbuilding industry. DES is an enabler for the evaluation of production schedules, resource utilization, material and work flow, and capacity (e.g. Kiran et al. 2001; Medeiros et al. 2000; Vacante and Costa 1996). This

technology has also proven to be useful in the justification and evaluation of proposed capital investments (e.g. Krause et al. 2004) as well as the assessment of increased workloads and sequence changes.

Although some simulation modeling is being performed in shipyards, further research is justified. According to Leong et al. (2006), “the National Research Council (NRC) has repeatedly identified simulation and modeling as a high priority area...one of two breakthrough-technologies that will accelerate progress in addressing the grand challenges facing manufacturing in 2020”.

1.4 Automatic Model Generators

One drawback to the use of simulation is the time needed to construct accurate models. Generating models is “knowledge intensive, time consuming, and error-prone” (Lee and Kang 1996). Medeiros et al. (2000) identify “the length of time required to construct models” as a barrier to the use of discrete event simulation in shipyards.

One way to address these weaknesses is through the use of automatic programming techniques. Madden et al. (2005) remark that “the use of auto-generated models will permit the simulation tool to find more widespread usage among the project managers.”

Simulation generators have been researched for some time. Amongst the earliest examples is Oldfather et al. (1966) which describes a method for programming by questionnaire. The definition of an automatic model generator varies; it is dependent on current technology. Mathewson (1984) states that “a program generator is a tool to aid in the production of computer-coded representations of a logical model.” He defines a generator as software that translates the logic of the model into the code of a simulation model. Older definitions suggest that having a graphical user interface to “drag-and-drop” model elements into the environment would constitute autogeneration. This definition would include many modern simulation packages, e.g. Flexsim, Arena, and Quest, as autogenerators.

Son (2000) describes a variety of methods for defining a simulation model: natural language interface, graphical interface, interactive dialogue interface, and the use of the existing resource and process models. Models can be generated from data defined using any of the aforementioned methods, or any combination thereof.

Recent work has demonstrated improved technologies for integrating existing data models with simulation models. Madden et al. (2005) develop a strategy for autogenerating space shuttle ground processing simulation models where data for the models comes from existing scheduling tools. Kiran et al. (2001) integrate a hierarchical model of a shipyard with an external scheduling application. Kronberger et al. (2006) develop a semi-automated simulation

model generator for scheduler-benchmarking that integrates with SAP/ERP in the modeling process.

In this research we generate simulation models based on existing resource and process models, as well as additional user-defined data, such as model execution data.

1.5 Data Exchange Standards

McLean et al. (2003) observe that “the development of neutral, vendor-independent data formats for storing simulation models and transferring data could greatly improve the accessibility of simulation technology to industry. Simulation standards for these models could help to accelerate the modeling process and reduce modeling costs.”

To date, there has not emerged a widely accepted standard format for representing manufacturing systems for simulation purposes (McLean et al. 2003, Röhl and Uhrmacher 2005). There is a recent trend towards eXtensible Markup Language (XML) formats. Roberto Lu (as quoted in McLean 2003) observes that “XML is the most logical choice of protocol since it is platform independent and widely accepted by almost all software packages”. Röhl and Uhrmacher (2005) remark that XML “has become the quasi-standard for storing semi-structured data.”

One such XML format is Simulation Data Exchange (SDX). SDX files were originally written in a plain ASCII text format, but are now saved in an XML format. SDX files contain system information and some run control information. A number of simulation packages can read, and in some cases create, SDX files, e.g. AutoMod®, WITNESS®, Simul8® (Sly and Moorthy 2001). SDX files are not, however, compatible with Flexsim.

Researchers at the Manufacturing Engineering Laboratory at the U. S. National Institute of Standards and Technology have conducted research in the areas of standard data interfaces for manufacturing simulators. NIST has coordinated projects to explore the use of XML as a standard for manufacturing simulation (e.g. McLean et al. 2003). SDX, for example, was “recently managed and coordinated by the National Institute of Standards and Technology (NIST)” (Sly and Moorthy 2001).

Harward and Harrell (2006) evaluated the shop data model being developed by NIST in the context of ProModel simulation software. They evaluated the NIST model in terms of its viability as a neutral file format for the discrete-event simulation of manufacturing systems. They found that while the NIST SDM only supports 28% of ProModel data elements, future enhancements to the SDM should allow the information model to serve as a foundation upon which a common information model and neutral file formats for the discrete event simulation industry could be built.

In this project we use Microsoft Excel to store and represent system data. The focus of this work is getting and using data, not data exchange format. The requirement was

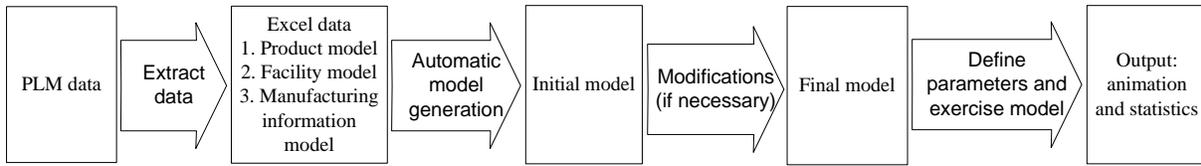


Figure 1: Overview of Methodology

that the chosen format facilitate effective data transfer. We used Excel rather than XML as it provided for a simpler interface more appropriate to client needs. It is easy to read and write data to and from Excel files, and it is easy to view and manually modify data in Excel worksheets.

Manually adding data to an Excel file is easier than manually adding data to an XML file. While much of the data used by our methodology to generate models is stored in PLM software, some of it must be manually entered by the user. Thus, a user-friendly interface is preferred. Excel allows users easy and consolidated access to input data.

Additionally, an Excel interface facilitates a template-based representation of a manufacturing system. Template based representations have proven effective in modeling systems (e.g. Guru and Savory 2004).

2 METHODOLOGY

2.1 Overview

There are two stages to be automated for model generation: system specification and model construction (Murry and Sheppard 1988). System specification requires the determination of data necessary to build a valid simulation model. We analyzed the manufacturing systems of interest to determine the necessary data.

In this research, system specification is provided by existing resource and process data as well as production schedules and (potentially) additional user input. Resource and process data comes from Product Lifecycle Management (PLM) software, additional user input comes through an interactive interface.

Figure 1 provides an overview of the methodology. Data extracted from the PLM system is used to create a neutral model description in an Excel format. The simulation model representation is based on Product, Facility, and

Manufacturing Information models. A Flexsim-specific model generator is used to automatically create a simulation model from the input data. The model can be manually modified if necessary. The model is then executed according to user-defined execution parameters. The individual components of the diagram are discussed in the following sections.

2.2 Input Data Conceptual Models

We developed an Excel interface based on the information needed to specify a model. Inputs are based on a Product Model, Facility Model, and Manufacturing Information Model. These are conceptual models; they do not necessarily correspond to the manner in which data is stored. The input models exhibit modularity; they are defined such that one could, for example, swap one facility model for another. (Obviously this assumes the appropriateness of the second facility model. One would not, for example, expect good results when swapping a printed circuit board facility model for a shipyard facility model.) Brief descriptions of the standard models are provided in the following sections.

The Excel format is designed with worksheets corresponding to different aspects of the model. The default Excel file contains the following worksheets: general info, workcenters, storage locations, operators, transport, cranes, part arrival schedule/production schedule, work schedules, sources, and sinks. Figure 2 provides a section of sample data used for model development. The data shown describes some of the workcenters in a manufacturing facility.

2.2.1 Product Model

The Product Model describes what is being produced by the manufacturing system. The model describes part attrib-

	A	B	C	E	F	G	H	I	J	K	L	N	O	P	Q
1	Workcenter Index	Name	Group01	X	Y	Z	Size X	Size Y	Size Z	Geometry	Time Table	Process1	Process2	Process3	Process4
2	1	2	2	1	1	1	1	1	1	2	2	2	2	2	2
12	10	SteelFab10	SteelFab	240	150	0	50	50	1	Default	Shift1	SteelFit	SteelWeld	PipeFit	PipeWeld
13	11	BlastPaint01	BlastPaint	350	100	0	50	50	1	Default	Shift1	Blast	Paint	Dry	NONE
14	12	BlastPaint02	BlastPaint	350	50	0	50	50	1	Default	Shift1	Blast	Paint	Dry	NONE
15	17	FinalOutfit01	FinalOutfit	450	100	0	50	50	1	Default	Shift1	Electrical	Insulation	SheetMetal	MechOutfitting
16	18	FinalOutfit02	FinalOutfit	550	100	0	50	50	1	Default	Shift1	Electrical	Insulation	SheetMetal	MechOutfitting

Figure 2: Screenshot of the Excel worksheet that describes workcenters

sutes, including geometries, as well as product assembly structures.

2.2.2 Facility Model

The Facility Model describes the facility, including equipment and stationary objects. Descriptions include geometry, location, capabilities, and processes. Entities defined in the Facility Model include the following:

- Workcenters
- Operators
- Overhead cranes
- Transport equipment (such as forklifts)
- Storage locations

2.2.3 Manufacturing Information Model

The Manufacturing Information Model contains information defining how the system works. The Manufacturing Information Model includes information such as the following:

- Work schedules, i.e. labor shift schedules
- Manufacturing schedule, i.e. production schedule
- Process times
- Part routings
- Processing requirements, e.g. machine and operator requirements

2.2.4 Execution Parameters

Execution parameters detail how the model will be run, such as warm-up period, run length, termination criteria, and number of replications. In this implementation termination criterion are based on data in the production schedule from the PLM system.

2.3 Model Generation

The model generator builds a discrete event simulation model from the information stored in the Excel file. The generator effectively serves to translate the data into a working model. Error checking is conducted during model generation; the user is notified if required data is missing.

The user interacts with Flexsim via a custom graphical user interface (GUI). The interface has buttons for the most common operations: clearing the model, importing and generating a new model, running the model, and outputting relevant statistics. Figure 3 provides a screen-shot of the GUI.

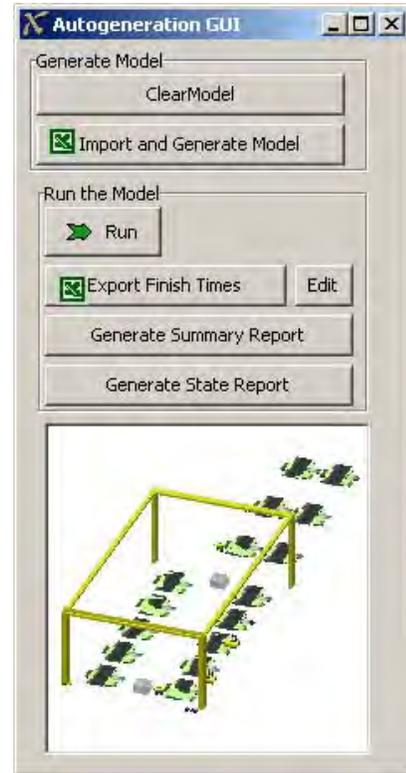


Figure 3: Custom Flexsim GUI

Data is imported to Flexsim using the Multiple Table Excel Import. Flexsim creates an internal table corresponding to each Excel worksheet. The internal tables are later referenced when generating the model.

Work schedules are created when the corresponding Excel worksheet is read into Flexsim. When model entities are created they can be associated with specific work schedules. The part arrival schedule is also generated when the corresponding worksheet is read into Flexsim. The part arrival schedule will be associated with the part source.

The simulation model is created by executing the model generation code. The code traverses the tables, generating the model entities and creating their associated attributes. The units for time and distance are defined in the Excel file (specifically in the General Info worksheet). Flexsim provides support for a wide variety of modeling constructs. Work centers are represented as processors with multiple capabilities. Storage locations, overhead cranes, human operators, and transport vehicles are represented by the corresponding Flexsim constructs.

In addition, we create a series of default, program specific entities: source, sink, and dispatcher. These are constructs that exist in the simulation model, but not in the real-world system – modeling constructs that you cannot get from outside systems. They are examples of simulation package specific implementations. The source, sink, and dispatcher concepts are not unique to Flexsim; many simu-

lation packages implement similar constructs (Quest, for example, supports sources, sinks, and dispatchers).

We create a single dispatcher for the entire model. All mobile resources (transport vehicles such as forklifts and cranes, as well as operators) are connected to the dispatcher. Mobile resources have attributes describing all processes they can perform, as well as attributes describing all locations from which they can transport parts. The dispatcher is also connected to everything that may request transport or an operator. All requests for transport and operators go to the dispatcher. The dispatcher then assigns the task appropriately.

Models are constructed such that all part routings are possible. Basically, everything is connected to everything. This method enables generality and flexibility in the model, but requires that routings be specified for every part (or specified using a routing table).

We use the Flexsim library of standard built-in modeling constructs as much as possible to reduce redundant work. We also developed a library of custom modeling constructs. Model code or pointers to custom model code can be specified in the Excel file. Alternatively, the logic associated with specific model entities can be modified after generating the model.

After generating the model the user has the capability to run it either using the custom GUI or through the use of standard Flexsim run controls. Results of the simulation model runs can then be obtained. Flexsim provides built-in functionality for a series of output reports (full report, summary report, and state report). We added buttons to the custom interface to generate these reports. The time at which a part is both created and completed is automatically recorded. The custom interface has a button that outputs the creation and completion times to an Excel or text file.

If desired, a user can specify routings to specific sinks. This enhances system analysis, as certain statistics are recorded at the sinks. If a sink is not specified in a product's routing, then after completion of all specified steps the part will be routed to the default sink.

3 SUMMARY AND CONCLUSIONS

This paper details the development of a methodology for automating much of the work associated with simulation model development. We developed a language-neutral Excel format for communicating system data and simulation parameters and a Flexsim-specific model generator. Our research was conducted in the context of large shipbuilding manufacturing centers. The methodology lends itself to different domains, but is biased towards industries that do similar things. Our choices concerning appropriate levels of abstraction were guided by assumptions regarding both the type of system to be modeled as well as the type of decisions the model will support.

Preliminary results indicate that the system enables users to create a model with less time and effort – the system automates a large fraction of the work of model creation. Modeling errors are reduced as data is automatically transferred. The system is easily extendible and allows non-expert use. Our experience supports the findings of Guru and Savory (2004) who state that the template-based methodology requires less expert knowledge of simulation model development than the standard method of manually building a model.

The Excel interface is user friendly. Even if information in the Excel file has to be manually entered, users report that it presents a faster method for model building than the standard Flexsim graphical dialogue interface. The Excel file provides a single location for a wide variety of system information and simulation parameters. In addition, experimentation using the simulation model is simplified, especially regarding changes to parameters such as location.

The methodology is most useful for complicated systems. There is some overhead in setting up the system, and it may be faster to build simple models using the standard interface. There is some threshold where it becomes worthwhile to incur the overhead in setting up the autogeneration system. The benefits of the system become more apparent when modeling complicated systems, such as those with many processing centers with varying process capabilities.

4 FUTURE WORK

We did not set out to develop a fully generic methodology and input data has been described in terms of specific PLM software. There is significant research to support the development of a unified XML standard, and future work may include support for XML data formats. Although the work described in this paper is an automatic model generator for a specific class of manufacturing systems, the work is relevant to a wide variety of systems. Future work will include expanding the generator to other domains. Also, the methodology still requires significant manual intervention. Future work could include automating some of those steps. Finally, this work involved deterministic systems; we were concerned with high-level scheduling and data concerning processing time variability did not exist for the systems under consideration. Future work will involve incorporating variability into the models.

Some of the future work described requires little work to implement, for example, expanding the generator to incorporate variability in the models. The challenge lies in gathering data from which to develop accurate models.

REFERENCES

- Guru, A., and P. Savory. 2004. A Template-Based Conceptual Modeling Infrastructure for Simulation of Physical Security Systems. In *Proceeding of the 2004 Winter Simulation Conference*, ed. J.S. Smith, B.A. Peters, R. Ingalls, and M.D. Rosetti, 866-873. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.
- Harward, G., and C. Harrell. 2006. Assessment of the NIST Shop Data Model as a Neutral File Format. In *Proceedings of the 2006 Winter Simulation Conference*, ed. L.F. Perrone, F.P. Wieland, J. Liu, B.G. Lawson, D.M. Nicol, and R.M. Fujimoto, 941-946. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.
- Kiran, A.S., T. Cetinkaya, and J. Cabrera. 2001. Hierarchical Modeling of a Shipyard Integrated with an External Scheduling Application. In *Proceedings of the 2001 Winter Simulation Conference*, ed. M.W. Rohrer, D.J. Medeiros, B.A. Peters, and J.S. Smith, 877-881. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.
- Krause, M., F. Roland, D. Steinhauer, and M. Heinemann. 2004. Discrete Event Simulation: An Efficient Tool to Assist Shipyard Investment and Production Planning. *Journal of Ship Production* 20 (3):176-182.
- Kronberger, G., A. Weidenhiller, B. Kerschbaumer, and H. Jodlbauer. 2006. Automated Simulation Model Generation for Scheduler-Benchmarking in Manufacturing. In *Proceedings of the International Mediterranean Modelling Multiconference (I3M 2006)*, 45-50.
- Lee, J., and S. Kang. 1996. Efficient Simulation Model Generation Using Automatic Programming Techniques. In *Proceedings of the 1996 Winter Simulation Conference*, ed. D.T. Brunner, J.W. Swain, J.M. Charnes, and D.J. Morrice, 708-723. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.
- Leong, S., Y.T. Lee, and F. Riddick. 2006. A Core Manufacturing Simulation Data Information Model for Manufacturing Applications. In *Proceedings of the Systems Interoperability Standards Organization 2006 Fall Simulation Interoperability Workshop*. Paper 06F-SIW-028.
- Madden, M. G., R. Wyrick, and D. E. O'Neill. 2005. A Strategy for Autogeneration of Space Shuttle Ground Processing Simulation Models for Project Makespan Estimation. In *Proceedings of the 2005 Winter Simulation Conference*, ed. M.E. Kuhl, N.M. Steiger, F.B. Armstrong, and J.A. Jones, 1251-1259. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.
- Mathewson, S.C. 1984. The Application of Program Generator Software and Its Extensions to Discrete-Event Simulation Modeling. *IIE Transactions* 16 (1):3-18.
- McLean, C., W. Leong, C. Harrell, P.M. Zimmerman, and R.F. Lu. 2003. Simulation Standards: Current Status, Needs, and Future Directions. In *Proceedings of the 2003 Winter Simulation Conference*, ed. D. Ferrin, D.J. Morrice, S. Chick, and P. Sanchez, 2019-2026. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.
- Medeiros, D., M. Traband, A. Tribble, R. Lepro, K. Fast, and D. Williams. 2000. Simulation Based Design for a Shipyard Manufacturing Process. In *Proceedings of the 2000 Winter Simulation Conference*, ed. P.A. Fishwick, K. Kang, J.A. Joines, and R.R. Barton, 1411-1414. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.
- Murray, K., and S. Sheppard. 1988. Knowledge-based simulation model specification. *Simulation* 50(3):112-119.
- Oldfather, P.M, A.S. Ginsberg, and H.M. Markowitz. 1966. Programming by Questionnaire: How to Construct a Program Generator. Rand Report, RM-5129-PR.
- Rohl, M., and A.M. Uhrmacher. 2005. Flexible integration of XML into modeling and simulation systems. In *Proceedings of the 2005 Winter Simulation Conference*, ed. M.E. Kuhl, N.M. Steiger, F.B. Armstrong, and J.A. Jones, 1813-1820. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.
- Sly, D., and S. Moorthy. 2001. Simulation Data Exchange (SDX) Implementation and Use. In *Proceedings of the 2001 Winter Simulation Conference*, ed. M.W. Rohrer, D.J. Medeiros, B.A. Peters, and J.S. Smith, 1473-1477. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.
- Son, Y.-J. 2000. *Simulation Based Shop Floor Control: Automatic Model Generation and Control Interface*. PhD Thesis, Penn State University.
- Vacante, S., and P.A. Costa. 1996. Application of Production Simulation Tools in the Shipyards of Fincantieri Naval Shipbuilding Division. In *Proceedings European Simulation Symposium 1996: Simulation in Industry*.

AUTHOR BIOGRAPHIES

GABRIEL BURNETT is a Research Assistant at Penn State's Applied Research Laboratory. He is currently pursuing Ph.D. in Industrial Engineering and Operations Research from Penn State University. He holds an M.S.I.E. & O.R. from Penn State and a B.S.I.E. from Arizona State. His recent work has focused on the use of simulation and optimization to support process improvements in the shipbuilding industry. His email address is gab192@psu.edu.

DANIEL A. FINKE is an Associate Research Engineer at Penn State's Applied Research Laboratory. He holds an M.S. in Industrial Engineering and Operations Research from the Pennsylvania State University and a B.S. in Industrial Engineering from New Mexico State University. He has been a primary team member on several simulation and modeling projects supporting process improvements in the shipbuilding industry. His current research interests include simulation based optimization and scheduling, resource allocation optimization and decision support. His email address is daf903@psu.edu .

D. J. MEDEIROS is Associate Professor of Industrial Engineering at Penn State University. She holds a Ph.D. and M.S.I.E from Purdue University and a B.S.I.E. from the University of Massachusetts at Amherst. She has served as track coordinator, Proceedings Editor, and Program Chair for WSC. Her research interests include manufacturing systems control, healthcare process modeling, and CAD/CAM. Her e-mail address is djm3@psu.edu.

MARK T. TRABAND has been employed as a Research Associate at the Applied Research Laboratory, The Pennsylvania State University since 1990. He is currently the head of the Manufacturing Systems Division. Dr. Traband received a B.S. degree in Industrial Engineering from Virginia Polytechnic Institute and State University in 1985. He was selected as an Office of Naval Research Graduate Fellow in 1985. He received M.S. and Ph.D. degrees in Industrial Engineering from The Pennsylvania State University in 1987 and 1995. His email address is mtt1@psu.edu.