
Electronic Theses and Dissertations, 2004-2019

2012

A Framework For Process Data Collection, Analysis, And Visualization In Construction Projects

Reza Akhavian
University of Central Florida



Part of the [Civil Engineering Commons](#)

Find similar works at: <https://stars.library.ucf.edu/etd>

University of Central Florida Libraries <http://library.ucf.edu>

This Masters Thesis (Open Access) is brought to you for free and open access by STARS. It has been accepted for inclusion in Electronic Theses and Dissertations, 2004-2019 by an authorized administrator of STARS. For more information, please contact STARS@ucf.edu.

STARS Citation

Akhavian, Reza, "A Framework For Process Data Collection, Analysis, And Visualization In Construction Projects" (2012). *Electronic Theses and Dissertations, 2004-2019*. 2177.

<https://stars.library.ucf.edu/etd/2177>

A FRAMEWORK FOR PROCESS DATA COLLECTION, ANALYSIS, AND
VISUALIZATION IN CONSTRUCTION PROJECTS

by

REZA AKHAVIAN
B.S. University of Tehran, 2010

A thesis submitted in partial fulfillment of the requirements
for the degree of Master of Science
in the Department of Civil, Environmental, and Construction Engineering
in the College of Engineering and Computer Sciences
at the University of Central Florida
Orlando, Florida

Summer Term
2012

© 2012 Reza Akhavian

ABSTRACT

Automated data collection, simulation and visualization can substantially enhance the process of designing, analysis, planning, and control of many engineering processes. In particular, managing processes that are dynamic in nature can significantly benefit from such techniques. Construction projects are good examples of such processes where a variety of equipment and resources constantly interact inside an evolving environment. Management of such settings requires a platform capable of providing decision-makers with updated information about the status of project entities and assisting site personnel making critical decisions under uncertainty. To this end, the current practice of using historical data or expert judgments as static inputs to create empirical formulations, bar chart schedules, and simulation networks to study project activities, resource operations, and the environment under which a project is taking place does not seem to offer reliable results.

The presented research investigates the requirements and applicability of a data-driven modeling framework capable of collecting and analyzing real time field data from construction equipment. In the developed data collection scheme, a stream of real time data is continuously transferred to a data analysis module to calculate the input parameters required to create dynamic 3D visualizations of ongoing engineering activities, and update the contents of a discrete event simulation (DES) model representing the real engineering process. The generated data-driven simulation model is

an effective tool for projecting future progress based on existing performance. Ultimately, the developed framework can be used by project decision-makers for short-term project planning and control since the resulting simulation and visualization are completely based on the latest status of project entities.

ACKNOWLEDGMENTS

The present document is my final Master's Thesis. This Thesis describes my graduate research which began in Fall 2010 and concluded in Summer 2012.

I would like to thank many people in my life who have contributed to my academic journey. Above all, my sincere appreciation is due to my advisor Dr. Amir H. Behzadan for his continuous support, endless patience and unstinting encouragement during my Master's studies and in the course of this research. He is truly a dedicated teacher and incredible advisor. Thank you for believing in me, giving me the opportunity to succeed, and helping me to promote from an ordinary student to a curious and motivated researcher.

My special thanks to my other committee members, Dr. Oloufa and Dr. Tatari who offered guidance and support. I am always indebted to Dr. Oloufa for his motivational lessons about construction engineering and management during the very first semesters I began my studies in the United States.

There are some other people that I am deeply grateful to them for their presence in my life; my lovely parents Hossein Akhavian and Maryam Batebi who were both teachers and I have been and will be always their student. Their love and patience have been always a source of support for me, particularly during the days I am far away from them. I would also like to appreciate the presence of Negin Alimohammadi, my love, wonderful

friend, and great source of emotional support and motivation during my Master's studies.

She never left my hand during hard days.

I am also thankful to all my friends, especially Saeed Hadian and Sina Zel Taat for their help and friendship and all nice experience we had during these years.

TABLE OF CONTENTS

LIST OF FIGURE.....	x
LIST OF TABLES.....	xiii
CHAPTER 1: INTRODUCTION.....	1
1.1 Thesis Statement.....	1
1.2 Research Motivation.....	1
1.3 Research Contributions.....	3
1.4 Research Objective and Project Tasks.....	5
1.5 Organization of the Thesis.....	6
CHAPTER 2: LITERATURE REVIEW.....	8
2.1 Automated Data Collection.....	8
2.2 Simulation in Construction.....	13
2.3 Real Time Simulation.....	14
2.4 Visualization in Construction.....	16
CHAPTER 3: DYNAMIC DATA-DRIVEN APPLICATION SYSTEM (DDDAS).....	18
3.1 Overview.....	18
3.2 General Applications.....	20
3.3 Developed Framework Based on DDDAS.....	22

CHAPTER 4: VISUALIZATION WITH OPENSCENEGRAPH (OSG).....	25
4.1 Overview	25
4.2 Implementation in This Research.....	31
CHAPTER 5: DEVELOPED FRAMEWORK.....	33
5.1 System Architecture	33
5.2 Real Time Data Collection.....	34
5.2.1 Serial Port Communication in .NET Environment	36
5.2.2 Data Collection Using LabVIEW	40
5.3 Automated Data Classification and Analysis.....	45
5.3.1 VI Structure for Data Analysis Module	49
5.4 Data-Driven Simulation	50
5.5 Pre- and post-processed Animations	52
5.6 What-If Analysis	55
5.7 Decision-Making and Dynamic Feedback	56
CHAPTER 6: Laboratory Scale Experiments and Results	58
6.1 Preliminary Results	58
6.1.1 Experiment Tools and Peripheral Devices.....	58
6.1.2 Single Object Data Collection and Visualization	61
6.1.3 Double Object Data Collection and Visualization.....	64

6.2	Comprehensive Example: Data-Driven Simulation.....	69
CHAPTER 7: CONCLUSIONS AND FUTURE WORK.....		78
7.1	Conclusions	78
7.2	Future Work	81
APPENDIX A: C++ ALGORITHMS FLOWCHARTS		83
APPENDIX B: LabVIEW GRAPHICAL PROGRAMMING AND ALGORITHMS		90
REFERENCES		98

LIST OF FIGURE

Figure 3.1: Basic Concepts of DDDAS	19
Figure 3.2: Mandel et al. Developed DDDAS for real time modeling of Wildfire	21
Figure 3.3: Developed DDDAS in the Presented Research.....	24
Figure 4.1: A Sample Scene Graph Hierarchy.....	27
Figure 4.2. Hierarchical Scene Graph and Relationships between Different Nodes	29
Figure 5.1: System Architecture of the Developed DDDAS Framework	34
Figure 5.2: Datagram Structure of the PNI TCM Prime Orientation Tracker	37
Figure 5.3: C++ Orientation Tracker Serial Communication Class	39
Figure 5.4: Flowchart of 3D Orientation Tracker Serial Communication Process.....	40
Figure 5.5: A VI Consists of a Front Panel and a Block Diagram.....	42
Figure 5.6: Developed VISA Interface Structure	43
Figure 5.7: Simplified Layout of an Earthmoving Operation.....	46
Figure 5.8: Activity Durations Based on the Variation of Equipment Body Orientation with Respect to Time (RB = Raise Bucket, LT = Load Truck, LB = Lower Bucket, RTB = Raise Truck Bed, P = Put, LTB = Lower Truck Bed).....	47
Figure 5.9: Single VI Containing Data Acquisition and Data Analysis Functions	50
Figure 5.10: DES Model of a Typical Earthmoving Operation.....	51
Figure 6.1: Overall Arrangement of Experiment Tools and Devices	59

Figure 6.2: A Prime 3D Orientation Tracker Mounted on a Model Excavator with Definitions of Yaw, Pitch and Roll Angles	60
Figure 6.3: Orientation Tracker Mounted on a Loader's Boom.....	61
Figure 6.4: Front Panel of Data Collection VI for a Single Loader.....	62
Figure 6.5: Real Time Display of Loader's Boom Movements and Corresponding 3D Animation Generated in Real Time	63
Figure 6.6: Orientation Trackers Mounted on a Loader's Boom and a Truck's Bed.....	65
Figure 6.7: Front Panel of Data Collection VI for a Double Object Experiment	66
Figure 6.8: Partial View of the Extensive Block Diagram Developed in this Research ..	67
Figure 6.9: Real Time Display of Loader's Boom and Truck's Bed Movements and Corresponding Animations	68
Figure 6.10: Experiment Layout of a Model Dam Construction Scenario	69
Figure 6.11: DES Model of Rock Hauling Activity	70
Figure 6.12: STROBOSCOPE Simulation Input File	71
Figure 6.13: STROBOSCOPE Simulation Output File.....	72
Figure 6.14: Developed VI for Data Collection and Analysis for Rock Hauling Example	74
Figure 6.15: STROBOSCOPE Simulation Input File Containing Updated Activity Durations.....	75
Figure 6.16: STROBOSCOPE Simulation Output File Based on the Updated Durations	76
Figure A. 1: CreateAnimationPath() Function Flowchart	86
Figure A. 2: CreateMovingModel() Function Flowchart	87

Figure A. 3: CreateModel() Function Flowchart.....	88
Figure A. 4: Main() Function Flowchart	89
Figure B. 1: A Customized VI - The Upper Window is the Front Panel and the Bottom Window is the Block Diagram.....	92
Figure B. 2: <i>VISA Open</i> Opens the Specified Port by the <i>VISA Resource Name</i>	93
Figure B. 3: <i>VISA Write</i> Writes Data to the Specified Port by the <i>VISA Resource Name</i>	94
Figure B. 4: <i>VISA Read</i> Reads Data from the Specified Port by the <i>VISA Resource Name</i>	94
Figure B. 5: <i>VISA Close</i> Closes the Specified Port by the <i>VISA Resource Name</i>	94
Figure B. 6: A Series of VISA Functions and Their Connections as Used in this Research	94
Figure B. 7: Requested Data Classified from Cluster of Real Time Orientation Data.....	95
Figure B. 8: <i>Unbundled By Name</i> Function that Returns Cluster Elements Whose Names Have Been Specified.....	95
Figure B. 9: <i>Greater?</i> Function Returns True If x Is Greater than Y - This Function Was Used to Detect Data Exceeding a Specified Threshold	96
Figure B. 10: <i>Tick Count</i> Function That Returns the Value of a Timer – This Function Was Used to Measure the Duration of Each Activity.....	96
Figure B. 11: <i>Build Array</i> Function to Store Activity Durations in a Numerical Array...	96
Figure B. 12: <i>Statistics</i> Tool Returns the Specified Statistical Characteristics of Input Arrays.....	97

LIST OF TABLES

Table 2.1: Previous research on remote data collection applications in construction jobsites	12
Table 6.1: Manufacturer’s Specifications of Prime 3D Orientation Tracker.....	60
Table 6.2: Comparison between Estimated Durations and Actual Durations Based on Real Time data	77

CHAPTER 1: INTRODUCTION

1.1 Thesis Statement

The efficiency of various construction tasks including the planning and control of equipment operations can significantly increase if adequate operational data is collected in real time, analyzed, and effectively integrated into the decision-making process. This real time filed data stream can be used as a reliable source to modify project plans, validate and improve existing control metrics, and update the underlying parameters of computer models (e.g. simulation and visualization) describing the interactions between different project resources, all in an effort to assist project personnel in predicting the future performance given the current conditions of resources on the ground.

1.2 Research Motivation

Resource planning and control at the operations level are critical components of managing the performance of ongoing activities in a construction site [1]. A comprehensive operations level plan can help project decision-makers and site personnel foresee potential problems such as spatial conflicts and resource underutilization even before the actual operation takes place. This will also help save effort that would have otherwise been put on reworks, resolving conflicts, and performing change orders, which

will ultimately translate into significant savings in project time and cost. For example, Cox et al. [2] suggested that rework is typically responsible for 6-12% of the overall expenditure for a typical construction project. Construction Industry Dispute Avoidance Task Force (DART) reported that annually, more than \$60 billion was spent on change orders in the United States [3]. Also, according to the Federal Facilities Council (FFC), in 10-30% of all construction projects serious disputes are estimated to arise with a total cost of resolution between \$4-12 billion each year [4]. One of the major impediments of effective project planning is managing a large volume of information including inputs from alternative designs, material properties, labor productivity, equipment specifications, and work schedules. This will become even more sophisticated when the dynamics of the construction project creates several layers of uncertainty that can range from internal factors (e.g. project time and cost variations, equipment breakdowns, contractor claims) to external events (e.g. weather conditions, financial market stability). Computer applications have thus evolved during the past several years to facilitate the process of project planning by providing a convenient and reliable means for modeling, simulating, and visualizing project activities [5, 6, 7, 8, 9, 10, 11]. In order to create reliable computer models of a future construction project during the planning stage, one needs to carefully examine every detail of the operations within that project, and identify major events and processes that will potentially impact the outcome of each operation. Once such events and processes are identified, attributes such as resource consumption levels and activity durations should be determined. For a small operation, this can be done in a relatively short period of time using existing numerical tools and statistical data

from past projects. However, as the size of the operation increases and with the introduction of more resources and activities, creating a simulation model that realistically represents the actual operation becomes a tedious if not an impossible task [12]. This is mainly due to the fact that collecting accurate and reliable field data from ongoing activities and resource operations, and integrating the collected data into the planning process turns into a challenging task. In addition, the uncertainties caused by unforeseen site conditions, equipment breakdowns, work delays, and the evolving nature of a construction project may slow down or interrupt the progress of data collection. Even if all such data is collected, handling a large volume of information in a single platform can prove to be time and labor intensive. As a result, it is very likely that the modeler uses strict rules, simplifying assumptions, and rigid design parameters inside the model to streamline the modeling process. These may seriously impact the accuracy of the model in representing the dynamics of the project which will ultimately be detrimental to the reliability of the model for verification and validation purposes [13].

1.3 Research Contributions

Traditional simulation paradigms employ static data and information available from similar projects and operate under a given set of system design parameters (e.g. activity precedence relationships, duration distributions) [14]. In the absence of a methodology that facilitates real time field data collection, most project decision-makers rely on readily available project information and subjective personal judgments when evaluating uncertainties and forecasting future project performance [12]. Recently, advances in

automation and information technology resulted in new approaches for collecting and managing construction work data. In particular, automated tracking systems have been evolved to collect necessary information about the position of construction resources for different purposes [15, 16, 17]. Timely use of field data to determine the location and status of resources (e.g. construction equipment and personnel) helps in describing the context surrounding the operations and therefore is valuable for monitoring the workflow of activities during these operations. Also, field data supports operational decisions and helps predict the performance of a construction system based on the latest project status.

Another valuable implication of field data acquisition is the application of the collected data in creating visual representations at different levels of detail corresponding to various operations on a construction site. Visualizing field data has been demonstrated to have many applications such as maintenance crew training [18], safety management [19], and damage prevention [20]. But from the point of view of planning, monitoring, and control, 3D visualization not only does offer a convenient tool for decision-makers to get a real insight of what is exactly happening in a jobsite (particularly for operations that are hard to quantify or represent in a parametric model), but also is a of substantial value for verification and validation of the underlying simulation model(s). This is especially important because decision-makers often do not have the time and knowledge to confirm the accuracy and validity of simulation models and thus do not usually rely on the results obtained from such models [10]. In addition, visualization assists in investigating events that are hard to be quantified in a definitive manner, but yet can affect the final outcome. Examples of such events include work zone overcrowding due to simultaneous execution

of different trades in building construction, safety problems, and potential for physical collisions.

The benefits of construction field data collection, simulation, and visualization have been investigated in isolated cases in the past. However, the potential of these three promising techniques when integrated in a single framework that facilitates the process of short-term planning and control of construction projects in operations level has not yet been explored [21]. Hence, the presented research is mainly motivated by this need and is aimed to fill this gap by investigating the requirements and applicability of an integrated framework that uses the paradigm of dynamic data-driven simulation to address the problem of short-term operational level planning and control. The underlying concepts and applications of dynamic data-driven simulation, which is also referred to as dynamic data-driven application system (DDDAS) are introduced in Chapter 3.

1.4 Research Objective and Project Tasks

The overall objective of this study is to *design a framework for integrating field data collection, data analysis, visualization and simulation for short-term decision-making in construction projects*. In order to achieve this objective, the following research tasks were identified and successfully completed:

- Investigate the requirements and design a functional system to collect real time data from equipment involved in different construction processes.

- Build data classification and analysis methods to provide orderly data and link them to specific activities describing the status of construction equipment.
- Develop an algorithm for creating 3D pre-processed visualizations of concurrent construction equipment activities.
- Conduct statistical analysis on data to obtain and update the probabilistic distributions describing the duration of individual field activities within a simulation model corresponding to the actual operations.

1.5 Organization of the Thesis

The following Chapters of this Thesis are shaped around the concepts, details, and implementation of the research tasks listed above. This Thesis is divided into seven Chapters. In particular:

- Chapter 1: Introduction – This Chapter contains the Thesis statement, identified gaps that motivated this research, the novel approach that this study adopts to address the identified gaps, and the overall objective and tasks defined and accomplished in this project.
- Chapter 2: Literature Review – A review of previous related research and state-of-the-art studies in the realm of automated data collection, simulation in construction, visualization in construction, and using the advantages of real time simulation in construction projects is presented in this Chapter.

- Chapter 3: Dynamic data-driven application system (DDDAS) – This Chapter introduces the basic concept of a relatively new paradigm for data-driven simulation and outlines its application in various field of science and engineering and emphasis on its application in the context of presented study.
- Chapter 4: Visualization with OpenSceneGraph (OSG) – Detailed description and technical aspects of the visualization toolkit that has been used in this research is presented in this Chapter and it is shown that how the proposed methodology benefits from employing these concepts.
- Chapter 5: Developed Framework – The overall system architecture of the developed framework is introduced in this Chapter and individual components of the system and their tasks are discussed in detail.
- Chapter 6: Laboratory Scale Experiments and Results – This Chapter demonstrates the validity and applicability of the presented methodology by presenting preliminary laboratory scale experiments and resulted outcomes.
- Chapter 7: Conclusions and Future Work – A discussion about the identified gaps in knowledge and the developed research methodology for addressing these gaps is presented in this Chapter and future research for further development of the presented framework is described.

CHAPTER 2: LITERATURE REVIEW

In Chapter 1, a general introduction to the research was presented and the motivation, potential contributions, research objective, and project tasks were described in details. The presented research aims to address the gaps identified in the current body of knowledge (as described in Chapter 1) through investigating the potentials and opportunities provided by emerging innovations in engineering instrumentation and computation. In this Chapter, a comprehensive review of recent research efforts and current demands in the areas of automated data collection, and visualization and simulation within the construction engineering and management domain will be conducted, in an effort to put the presented work into context and demonstrate its potentials in addressing some of the longstanding challenges faced by the construction research community.

2.1 Automated Data Collection

Collecting accurate and reliable data is one of the most critical components of every decision support system. Data captured manually using traditional onsite data collection techniques can be outdated, inaccurate, or missing certain pieces [22, 23, 24, 25, 26]. McCullouch indicated that field supervisory personnel spend on average 30%-50% of their time on recording and analyzing filed generated data[23]. Saidi et al. [27] stated that despite the recent advancements in construction measurement and sensing technologies, having accurate and updated information about the status of construction operations

remains an issue in the construction industry. As a result, automated data collection and resource location tracking techniques have received credibility over the past several years, as they facilitate processes including but not limited to resource management, productivity analysis, quality control, and monitoring workflow processes. To this end, work still needs to be done in order to take advantage of such technologies when planning activities at early stages of a project where the scope of the work and the dynamics of the project environment are still evolving.

Automated resource (personnel, equipment, materials) tracking has been the subject of many studies in construction and facility management [15, 28, 29, 30, 31, 32]. Resource location tracking applications use different techniques for indoor and outdoor environments. A variety of outdoor and indoor location tracking technologies exist with significantly different characteristics, infrastructure, and device requirements [16]. Radio Frequency Identification (RFID), for example, has been increasingly used for tracking purposes in construction jobsites. RFID systems use tags and a reader which sends radio frequency signals to read data from the tags. One of the early attempts in using of RFID in construction industry was made by Jaselskis et al. [33]. They proposed RFID for tracking high-valued materials on construction jobsites. Song et al. [28] used RFID to automate the task of tracking the delivery and receipt of fabricated pipe spools in lay down yards and under shipping portals. Since RFID readers and tags do not require line-of-sight, the readers can detect several tags at a time, and the tags can function properly in harsh conditions. However, the short reading range which mostly is a function of the communication frequency can be an obstacle for the use of RFID systems in large

construction sites [34]. Researchers have also used the Global Positioning System (GPS) for its capability in tracking construction labor and equipment in outdoor environments and construction sites [16, 35, 36]. GPS is an outdoor satellite-based worldwide radio-navigation system formed by a constellation of 24 satellites, ground control stations, and end users [37]. To address the challenge faced by equipment operators who have limited field view and depth perception when they control equipment remotely with video cameras, Oloufa et al. [35] developed a system for collision detection and vehicle tracking by using differential GPS, wireless, and web-based technologies. The most important impediment in using GPS is that its functionality is to the most extent, limited to outdoor environments since a clear line-of-sight between the satellites and the GPS receiver is always needed. More recently, there have also been some attempts in combining RFID with GPS technology [31, 38]. Jang et al. [15] introduced an Automated Material Tracking (AMTRACK) system based on ZigBee localization technology to overcome the drawbacks of GPS and RFID systems in terms of accuracy and cost. Another technology that has been studied for automated tracking is Ultra Wide Band (UWB). Teizer et al. [39] developed an UWB data collection tool for work zone safety management and location tracking. In an indoor environment, where Global Navigation Satellite System (GNSS) data is not available, indoor positioning technologies are used. RFID and UWB can be used in both indoor and outdoor environments. GPS, as stated before, has generally developed only for outdoor environments. However, another technology called indoor GPS has recently emerged which is not satellite-based [32]. Wireless Local Area Network (WLAN) is another technology used for indoor tracking

and localizations [32]. Inertial Navigation Systems (INS) such as accelerometers and other systems such as Bluetooth, Infrared, and Ultrasonic are other examples of indoor localization technologies [40].

Another category of localization and tracking technologies is motion-based. Generally, motion-based devices sense motion and its attributes such as velocity, acceleration, and heading directions. For position sensing, inertial navigation systems (INS) or inertial measurement unit (IMU) are constructed using a combination of gyroscopes, accelerometers, and magnetometers [41]. Using IMU the current state of the target in terms of location, speed, and heading direction can be determined by using state estimation algorithms using the information provided by gyroscopes and accelerometers. Behzadan et al. [42] developed an augmented reality (AR) hardware framework in which they used orientation trackers capable of measuring compass heading using magnetic field sensors.

In addition to the sensor-based data acquisition technologies described above, vision-based tracking has lately started to gain credibility among researchers. In a recent study, Brilakis et al. [43] presented an automated framework for vision based tracking using two cameras. Although this method seems to overcome the disadvantages of existing sensor-based techniques such as limited coverage area and dependence on preinstalled tags on the objects, it is still much costly and requires a more involved maintenance and calibration. Table 1 summarizes the existing tracking and localization techniques. As Table 1 suggests and to the author's best knowledge, the application of real time data

collection for the purpose of planning and monitoring of equipment motions has not yet been investigated.

Table 2.1:

Previous research on remote data collection applications in construction jobsites

Study	Application	Technology
Ergen et al. 2007	Material Tracking	RFID
Song et al. 2006a	Material Tracking	RFID
Behzadan et al. 2008	Personnel Tracking	GPS
Caldas et al. 2006	Material Tracking	GPS
Grau and Caldas 2009	Material Tracking	RFID + GPS
Ergen et al. 2007	Material Tracking	RFID + GPS
Jang et al. 2007	Material Tracking	ZigBee
Teizer et al. 2007	Safety	UWB
Khoury and Kamat 2009	Tracking Mobile Users	UWB/Indoor GPS/WLAN/
Behzadan et al. 2008b	Mobile AR Hardware	IMU
Brilakis et al. 2010	Project Entities	Vision Based

2.2 Simulation in Construction

Simulation is a valuable tool for effective construction planning and management mainly due to the presence of operational and decision-making uncertainties in most construction processes. However, a large amount of research previously conducted in construction simulation have one thing in common: almost all of them assume that when the simulation model is created, sufficient data with adequate level of detail is readily available mainly in form of historical records from similar projects or expert thoughts and judgments (which may prove to be subjective). It is clear that providing such input, there is almost no guarantee that the generated output reliably reflects the expected performance of project entities, since the bulk of the data do not particularly belong to that project.

Among several existing methods for modeling construction operations, discrete-event simulation (DES) has gained a lot of interest by researchers since almost every construction operation can be effectively broken down to and modeled as a network of discrete activities, each consuming resources (personnel, material, and equipment) to be completed [44]. DES models provide an effective means to establish logical relationships between activities within a project which compete over and make use of available and often scarce resources. The introduction of CYCLONE [45], marked the beginning of a new era in modern construction simulation research. CYCLONE aimed to simplify the modeling of processes that are cyclic in nature. Subsequently, many attempts were made to develop different simulation systems based on CYCLONE. Examples include

INSIGHT [6] that enabled videotaping of field operations, and extracting and analyzing videotaped data to obtain estimated values for the productivity of the system and its components. Further studies explored the applicability of object-oriented and modular programming in developing simulation systems. Examples of such efforts include MODSIM [46] capable of translating a simulation code to the C language for compilation and linking. In another effort, STROBOSCOPE [47] an extensible programmable system capable of modeling complex construction operations was introduced. Later, an activity-based construction modeling and simulation method called ABC [9] was developed.

A DES system called COOPS was introduced by Liu and Ioannou [7] which used object-oriented design for simulation. Martinez and Ioannou [44] examined DES systems based on three characteristics: application breadth (general or special purpose), modeling paradigm (process interaction versus activity scanning), and flexibility (i.e. programmability). Also, a new simplified DES approach or SDESA was developed by Lu [48] for planning construction operations which can be used as a general-purpose construction planning tool to track the performance of individual resources and handle cyclic or looped processes.

2.3 Real Time Simulation

Real time simulation has been explored by researchers in several engineering and scientific fields. For example, Hunter et al. [49] developed a simulation model based on inflow data aggregated over a short time interval to create an accurate estimate of the evolving state of transportation systems. In another example, Tavakoli et al. [50]

suggested a generic simulation platform for real time DES modeling in healthcare and manufacturing applications. Also, a yard crane dispatching algorithm based on real time data driven simulation was proposed by Guo et al. [51] to solve the problem of yard crane job sequencing by minimizing the average vehicle waiting time. In the construction domain, however, despite previous work in real time data collection and processing, very limited amount of research has been done in effectively incorporating field data into an existing simulation model for short-term planning and control of the same operations. Chung et al. [52] suggested using Bayesian techniques to update the distributions of input parameters for tunnel simulation by “manually” collecting project data from a tunneling project on a bi-weekly basis and using the collected data to improve simulation input models. Also, Song et al. [12] described a framework of real time simulation for short-term scheduling of heavy construction operations and developed a prototype system for asphalt hauling and paving projects.

To this date, only a limited number of previous projects investigated the planning and control of engineering systems through real time simulation using the latest changes in activity patterns and interactions. In the absence of a simulation system that is not using an accurate input data, the resulting output should be evaluated with prudence. Abourizk et al. [53] discussed that random input tends to propagate to the output of the simulation model. They warned of using improper molding of input data through demonstration of the sensitivity of the output parameters as well as resource utilization to the input model utilized.

2.4 Visualization in Construction

The role of visualization in construction engineering and management has been generally limited to the design of construction products using 3D CAD modeling or the demonstration of how an entity evolves over time using 4D CAD applications. Visualizing the actual interactions between resources (including personnel, equipment, and materials) that result in a constructed facility has received a very little attention [54, 55]. Almost all of the efforts in this area concentrated on visualization of “simulated” construction operations. Schematic modeling such as DISCO, iconic animation [56], and 2D system visualizations such as PROOF [57] are some of the first generation systems intended for visualizing simulated construction operations. More recently, Kamat and Martinez [10] presented VITASCOPE as a general-purpose, user-extensible 3D animation system for visualizing simulated processes in smooth, continuous, 3D virtual worlds. Behzadan and Kamat [11] designed and implemented ARVISCOPE, an augmented reality (AR)-based mobile visualization system that allowed dynamic visualization of simulated operations in outdoor environments using an external scripting language.

Confirming the veracity and validity of the simulated construction operation is a major goal in creating post-processed visualization systems [54]. Nevertheless, verification and validation of the simulation model can be conveniently performed if a similar, yet pre-processed animation representing the actual ongoing activities exists. Having both pre-

and post-processed animations in a similar visualization environment side by side, facilitates the comparison between the real world systems and the model.

CHAPTER 3: DYNAMIC DATA-DRIVEN APPLICATION SYSTEM (DDDAS)

3.1 Overview

As described in previous Chapters, a major requirement of a robust decision support system capable of offering real time analysis of concurrent construction operations is the ability to provide decision-makers with a reliable basis to predict upcoming system performance by using the incoming data streams to simulate the actual operations. To achieve this, the concept of a relatively new simulation paradigm often referred to as dynamic data-driven application system (DDDAS) and its potentials in the realm of construction engineering and management was investigated in this research.

A DDDAS model is sought to dynamically measure site data in form of a new information layer, integrate the collected data with the corresponding simulation model to constantly adapt the model to the dynamics of the construction system, and constantly update it based on the latest collected operational data [58]. Although the dynamic nature of many complex systems such as those in construction requires simultaneous injection of collected data into the simulation model in response to the evolving conditions, many computational models used to date only allow fixed data inputs while the simulation is launched [14].

Initially, DDDAS was conceived by the National Science Foundation (NSF) in 2000 following two catastrophic events. The first was the missed prediction of the track and

magnitude of a storm that blanketed a number of cities from South Carolina to New England in January 2000, and the other was the failure of a simulation model to predict the propagation and behaviors of a fire near Los Alamos National Laboratory in May 2000 mainly due to the changing nature of fire and consequently, the inability of emergency response agencies to take appropriate actions to limit its propagation [59]. Scientists believed that such miscalculations were due to computer simulation models that were unable to incorporate real time changing conditions on the ground [59].

Recently, advances in computational technologies for data collection, analysis, and modeling provided the necessary tools for accurate measurement and injection of necessary data into corresponding simulation models and enabled the development of the DDDAS. Figure 3.1 is a schematic diagram showing the basic components of a DDDAS (as introduced by the NSF) consisting of the following modules: data acquisition tools, simulation model, dynamic data control and acquisition, and visualization and human interface.

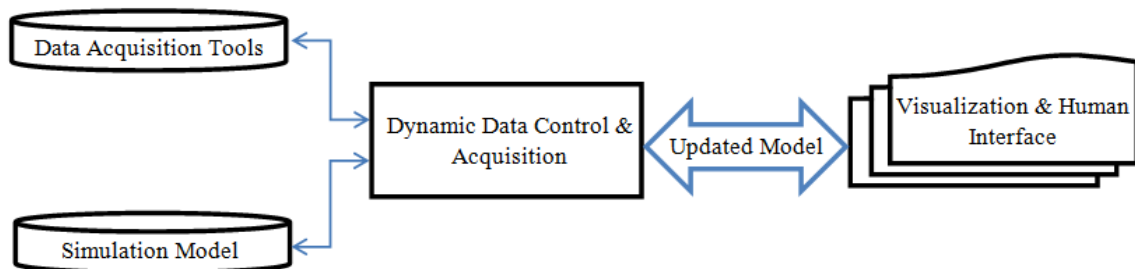


Figure 3.1: Basic Concepts of DDDAS

Data acquisition tools refer to field equipment used for remote data collection such as wireless sensors and instruments. Simulation model represents those models that need to be updated based on the stream of the incoming data. Dynamic data control and acquisition includes algorithms for data analysis used to prepare data for representation and input modeling. Finally, visualization and human interface refers to the human expert interaction to steer the model (if needed) and determine answers to critical decision-making problems based on the simulation results. These components and their interactions, as stated before, symbolize a rudimentary representation of the DDDAS concept and most of the platforms, including what was developed in this research are built upon this basic premise.

3.2 General Applications

As an emerging and promising area of research, DDDAS is gaining credibility among scientists and researchers in various fields of study while posing challenges in mathematical algorithms, systems software and data collection. Nevertheless, engineering problem solving in general and construction engineering and management in particular, are yet to benefit from the opportunity offered through employing this concept.

In a research aimed at forecasting the wildfire behavior, Mandel et al. [60] proposed a DDDAS that included coupled weather and fire numerical models, an automated data acquisition and control module, visualization and user interface module, and a communication infrastructure. In their developed system, data acquisition and control module directs data to the numerical models where multiple simulations are running.

Synchronously, the simulation inputs are adjusted based on the actual measurements of the field. Also, simulation results are presented through visualization and user interface module to the user in order to determine alternative firefighting scenarios. In this example, data collection was performed using wireless network sensors and cameras mounted on airplanes. Personal digital assistant (PDA) devices were also used as convenient visualization and user interface tools while numerical model ran on a remote supercomputer. Figure 3.2 shows how this particular application has been built upon the basic DDDAS concept previously shown in Figure 3.1.

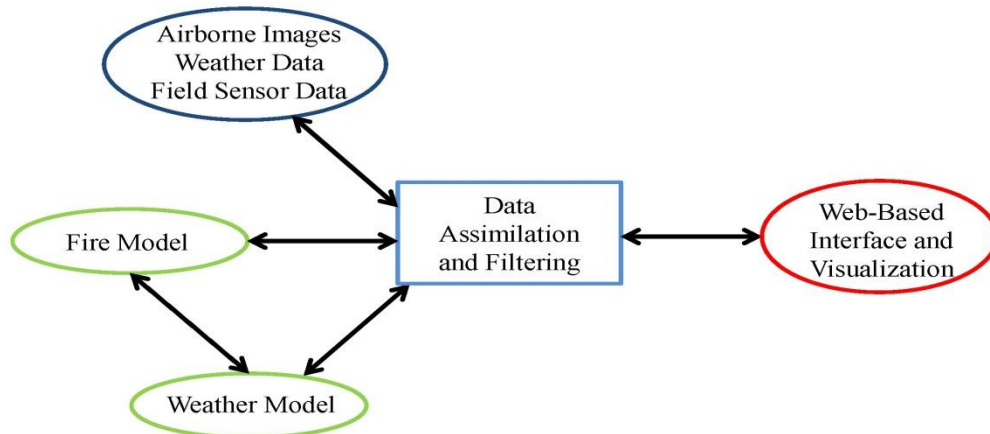


Figure 3.2: Mandel et al. Developed DDDAS for real time modeling of Wildfire

In another research, Douglas et al. [61] investigated the application of a DDDAS in an environmental engineering set up. They considered the case of a contaminant spill occurring near a clear water aquifer. Sensors were used to measure where the contaminant was, how and in what direction it was moving, and to monitor the environmental impacts of the spill. Numerical simulation procedures for multi-scale interpolation were used in order to map sensor data and to continuously update the

simulation model. The study demonstrated that frequent updating of the sensor data in the simulation considerably improved the prediction results.

Gaylor et al. [62] indicated that in case of a crisis, management should make decisions in order to react to dynamic uncertain conditions. In this regard, having access to real time data in a format, that can be readily understood and acted upon, is critical. Therefore, they applied the concept of DDDAS to support emergency medical treatment decisions in crisis conditions. Their complex dynamic environment fed and responded to a stream of real time data coming including positional data coming from GPS trackers mounted on ambulances and vital signs sensors mounted on patient body.

The NSF has also proposed some applications in workshops held for introducing DDDAS. An interesting DDDAS application is traffic light control, since there are always two significant variants: whether the plan is to minimize or to maximize the number of red lights encountered. As stated by NSF 2000 [59], the ultimate goal should be to continuously optimize the timing of the traffic lights. Using DDDAS and based on a sophisticated model, data generated by sensors embedded under streets and also other factors such as weather conditions can assist in predicting and optimizing the flow of vehicle movements [59].

3.3 Developed Framework Based on DDDAS

Unlike several other scientific fields, the idea of DDDAS has been given very little attention in engineering simulation in general, and has not been widely applied to

construction research in particular. DDDAS enables a more accurate prediction of how a dynamic construction system will behave in the future based on the current status of its constituents (i.e. resources). Therefore, construction projects can benefit from this novel paradigm if necessary infrastructure, algorithms, and tools to launch robust DDDAS platforms are effectively designed and implemented.

Traditionally, there has been a major disconnect between DES modeling (which is mainly conducted at the planning stage) and the actual site dynamics (during the construction phase). Incorporating the concept of DDDAS into the modeling process can help significantly improve conventional DES modeling. For example, more realistic activity parameters (e.g. probabilistic duration distributions, dependencies) can be obtained by measuring data collected from different pieces of equipment involved in that activity. In short, DDDAS facilitates the process of tailoring an existing DES model to better meet the evolving conditions of the real system using the latest data as input to the corresponding simulation model.

The DDDAS technique designed and implemented in this research captures sensor-based real time data from resources on a jobsite, classifies and analyzes the collected data to a meaningful format for the following modules, incorporates the analyzed data to update the corresponding DES model, and creates an exact dynamic 3D visualization of the ongoing operations using the collected data, all in an effort to assist project decision-makers in short-term operations planning and control [63]. Figure 3.3 illustrates a simplified diagram of the developed DDDAS in this research.

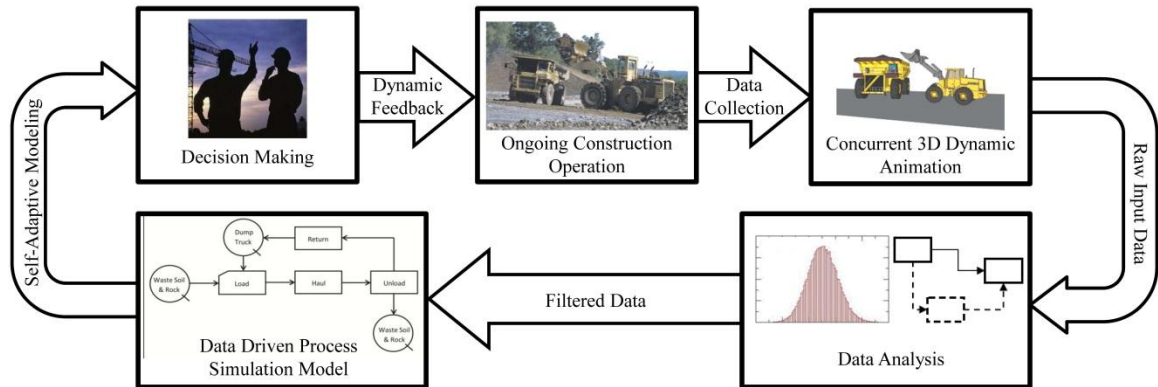


Figure 3.3: Developed DDDAS in the Presented Research

As shown in this Figure, the framework built upon the general concept of DDDAS. Real time collected data from ongoing construction operation move through a data analysis module to provide required information for updating the data-driven simulation model. Also, a visualization system providing a concurrent 3D animation of ongoing activities serves as the human interface module. Detailed description and system architecture of the developed framework can be found in Chapter 5.

CHAPTER 4: VISUALIZATION WITH OPENSCENEGRAPH (OSG)

4.1 Overview

In this research, OpenSceneGraph (OSG) which is built upon the industry standard OpenGL graphics library is used inside the .NET environment to create pre-processed animations of ongoing equipment activities and to link each and every object motion inside the animation to the collected field data that represent the actual motion of that object. This Chapter provides technical details about the algorithms developed to create 3D animations using CAD models of construction equipment.

A scene assembled from discrete components that can be dynamically manipulated, provides essential means for creating a contextual animation. To facilitate the creation and manipulation (i.e. positioning, orienting, and scaling) of objects in an assembled scenes, the concept of scene graphs were implemented in this research. Generally, a scene graph is a hierarchical organization of shapes, groups of shapes, and groups of groups that together construct a scene [64]. Computer graphics implementations build upon the concept of scene graphs release the end user from implementing and optimizing low level graphical programming and complexities involved in rendering process of 3D objects in a scene [41]. The scene graph application programming interface (API) provides a means for constructing scenes that follow a hierarchical data structure of objects.

OSG is a collection of open-source libraries that provide scene management and graphics rendering optimization functionality to applications. It has been written in ANSI C++ and uses the industry standard OpenGL low-level graphics API [65]. Although there are a few other scene graph-based libraries such as Performer, Open Inventor, and Java3D, this research used OSG due to the fact that it is capable of reading various image file formats which supports the prospect of designing a more generalizable visualization platform. At the same time, OSG provides the functionalities required to describe a complex scene using an object-oriented representation which releases the user from implementing and optimizing low level graphical programming and facilitates rapid development of graphic applications.

In OSG terminology, a *node* is an object that can be part of or entirely comprise a scene graph. Each node as a collection of one or more values and methods compresses what is required to be drawn. *Root node* is the highest level node to which all the elements of a scene graph (directly or indirectly) are connected [66]. Each scene graph comprised of nodes in a graph structure that are connected together via individual child-parent relationships. The edges that connect the nodes describe a meaningful relationship that exists between them. The *root node* is usually connected to intermediate grouping nodes called *internal* or *group nodes*. These nodes commonly are responsible for 3D transformations performing positioning (translation), orientation (rotation) and size (scaling). *Leaf nodes* are the lowest level nodes that contain the geometrical description of the components and are located at the terminus of a branch [67]. Figure 4.1 shows a sample scene graph in which *Jobsite* is the root node. Scene sub-graphs are created and

attached to the root node to complete the scene structure by encapsulating the entire jobsite. In Figure 4.1, sub-graphs *Truck*, *Excavator* and *Terrain* are all child nodes of *Jobsite*. Also, nodes *Excavator* and *Truck* have their own child nodes at the lowest level of the hierarchy.

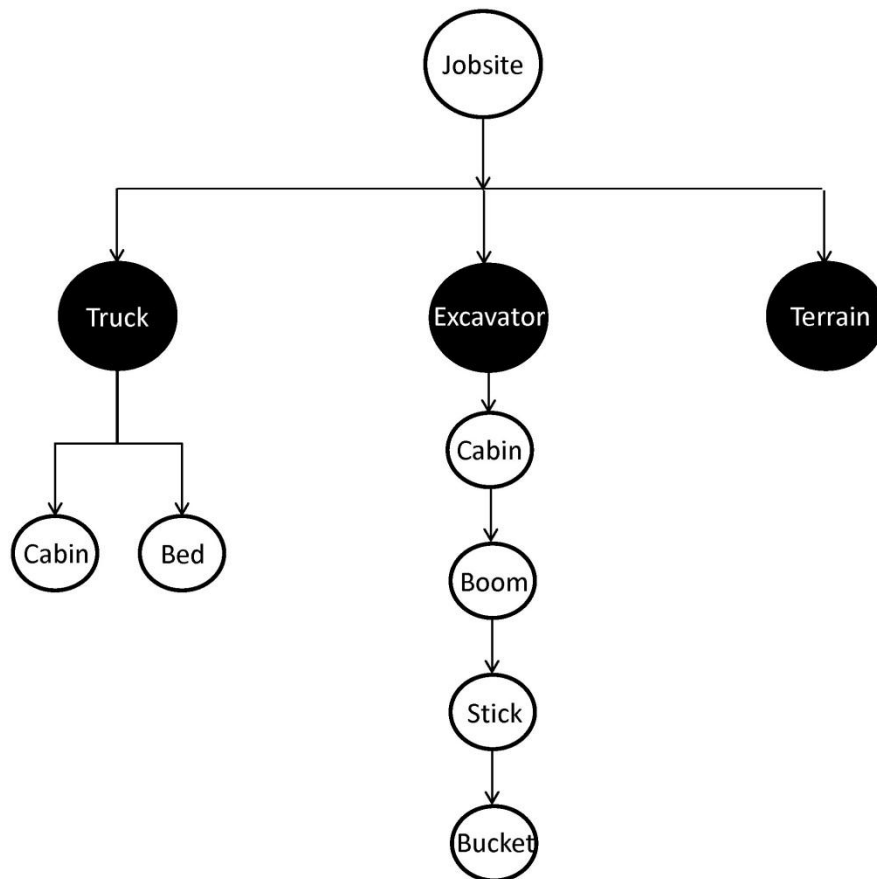


Figure 4.1: A Sample Scene Graph Hierarchy

Using transformation nodes, each geometrical model is created in its own local coordinate frame, stored as a leaf node in the scene graph, and appropriate placement of the model in terms of position and orientation will be made inside the coordinate frame of

its parent node. Scene graph developers can manipulate the translation, rotation, and (scale) of different nodes using transformation nodes.

The overall transformation of a child object relative to its parent node is obtained by multiplying the individual matrices as follows:

$$T_{Child}^{Parent} = \begin{bmatrix} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & T_y \\ 0 & 0 & 1 & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ \cos \gamma & -\sin \gamma & 0 & 0 \\ \sin \gamma & \cos \gamma & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \alpha & 0 & \sin \alpha & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \alpha & 0 & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \beta & -\sin \beta & 0 & 0 \\ \sin \beta & \cos \beta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Where the first matrix shows the transformation of the child node with respect to its parent node, the second, third, and fourth matrices, are the rotation about the local X, Y, and Z axes, respectively, and the fifth matrix is the scale matrix. Considering a scene consisting of a loader and a truck, Figure 4.2 shows the hierarchical scene graph and relationships between different nodes.

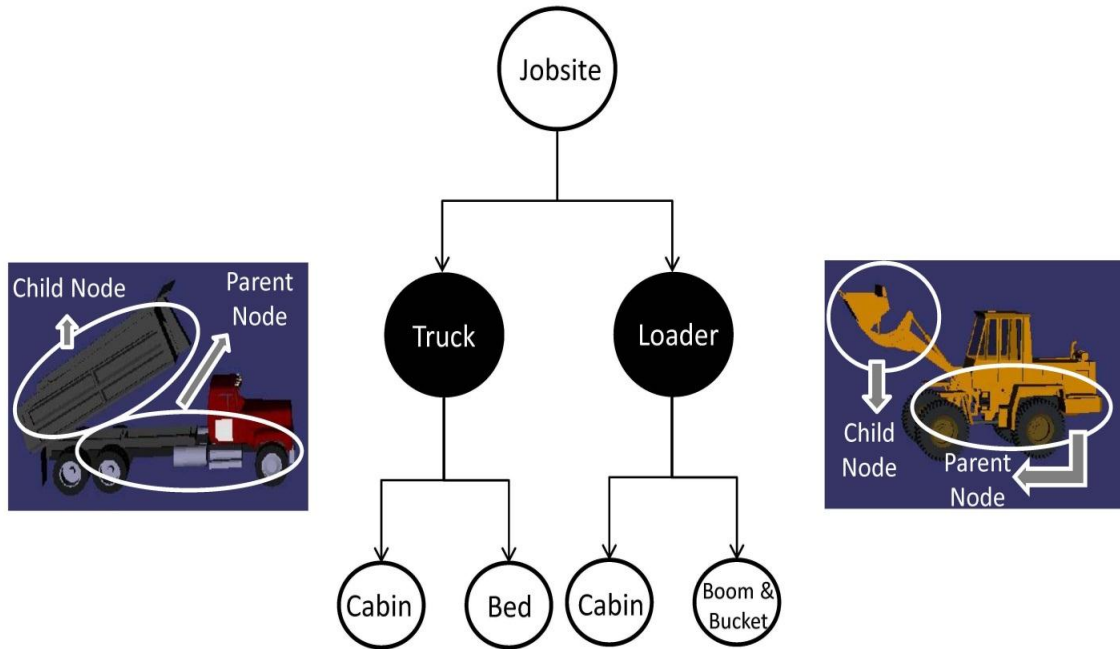


Figure 4.2. Hierarchical Scene Graph and Relationships between Different Nodes

Using the concept of scene graphs, if the angle of rotation of a child node about the X axis of its parent node is γ , the default value for this angular motion can be set to zero to represent the initial rotation matrix of the child node relative to the parent node, as follows,

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \gamma & -\sin \gamma & 0 \\ 0 & \sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \xrightarrow{\gamma=0} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

If a motion sensor capable of detecting angular motions is connected to the real object being represented by the child node in the scene graph, as soon as the rotation angle about the local X axis (also called the pitch angle) changes due to a change in the real

object's orientation, the sensor determines the change, the collected value is used to update the value of γ , and consequently the above rotation matrix is updated. For example, the truck bed shown in Figure 4.2 is rotated upward by 45° from its initial orientation. When this change is detected, the new pitch angle is used to update the corresponding rotation matrix as follows,

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \gamma & -\sin \gamma & 0 \\ 0 & \sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \xrightarrow{\gamma=45^\circ} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos 45^\circ & -\sin 45^\circ & 0 \\ 0 & \sin 45^\circ & \cos 45^\circ & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

This new rotation matrix will be then used to update the overall transformation matrix of the child (bed) node relative to its parent (truck) node (T_{Child}^{Parent}). The animation is updated in each frame according to the overall transformation matrices of all the objects exist in the scene.

One of the important requirements of any visualization system is providing a suitable view of the described scene. OSG provides several utilities to arrange desired viewpoints from which the viewer can watch the scene. The position and orientation of a viewpoint can be manipulated while the scene is being displayed to achieve the desired view of a scene graph or different views of the same scene graph. Moreover, it is possible to set several viewpoints with different positions and angles to have various views of the same scene graph as if depicting a single scene with different cameras installed in distinct

places [68]. In OSG, viewpoint definition is independent from the actual scene graph representation and as such, there is no nodal representation for viewpoints.

Creating an animation of the scene objects requires that a dynamic relationship between scene graph components is first established. Such relationship can be obtained through dynamically manipulating the values of the transformation matrices in the scene graph. Although the increments by which values in a transformation matrix change is discrete over time, a realistic animation showing exact movements of real world objects need a smooth transition between discrete points with the passage of animation time. OSG provides complex mechanisms to achieve this goal through constant monitoring and updating of all moving objects using frame updating algorithms [66]. Technical details of these algorithms are beyond the scope of this study but can be found in [64, 67].

4.2 Implementation in This Research

Since OSG is a free open source toolkit, it allows access to code internals, thus providing the opportunity to manipulate and modify the original content of the code to supplement the rest of the framework developed in the .NET environment for the specific purposes of this research.

Each of the articulated components depicted in the scene consists of separate parts (nodes) created within different modeling packages such as 3D Studio™ (.3ds), AutoCAD™ (.dxf), MicroStation™ (.dwg), and VRML (.wrl) that are stored in the user computer. By connecting these nodes through assigning a special coordination relative to

the main origin of the scene, and also creating a meaningful child-parent relationship between separate parts, each single component of the scene is created as a standalone model that can be moved inside the animation as necessary. Also, the animation speed which is the rate of dynamic increase/decrease of the values of transformation matrices, and a desired coordination for the viewpoint should be specified.

Using the real time positional and orientation data from the sensors mounted on the target objects (in real world), the developed algorithm stores the data in form of vectors as the animation path for each solid, yet articulated entity in the scene. Thus, the animated scene is capable of showing the actual movements of every real object using real time data representing the translation and/or orientation of that object's articulated parts, or the object as a whole.

CHAPTER 5: DEVELOPED FRAMEWORK

5.1 System Architecture

In Chapters 3 and 4, the technologies, concepts, and tools necessary to address the challenges described in Chapter 1 were discussed in detail. The need for the presented research was further justified by an extensive review of previously conducted studies in Chapter 2. This Chapter outlines a detailed account of the individual components of the framework and their interconnection in the context of the developed research methodology.

Figure 5.1 depicts the higher level system architecture of the developed framework in which the relationship between major building elements, as well as an overall view of how raw operational data flows through the system, is eventually transformed into a meaningful format, and used in different processes are illustrated. As previously described in Chapter 3, the framework is built around the concept of dynamic data-driven application system (DDDAS) and thus contains major components (modules) that were previously illustrated in Figure 3.3. The following Subsections provide more details about these components.

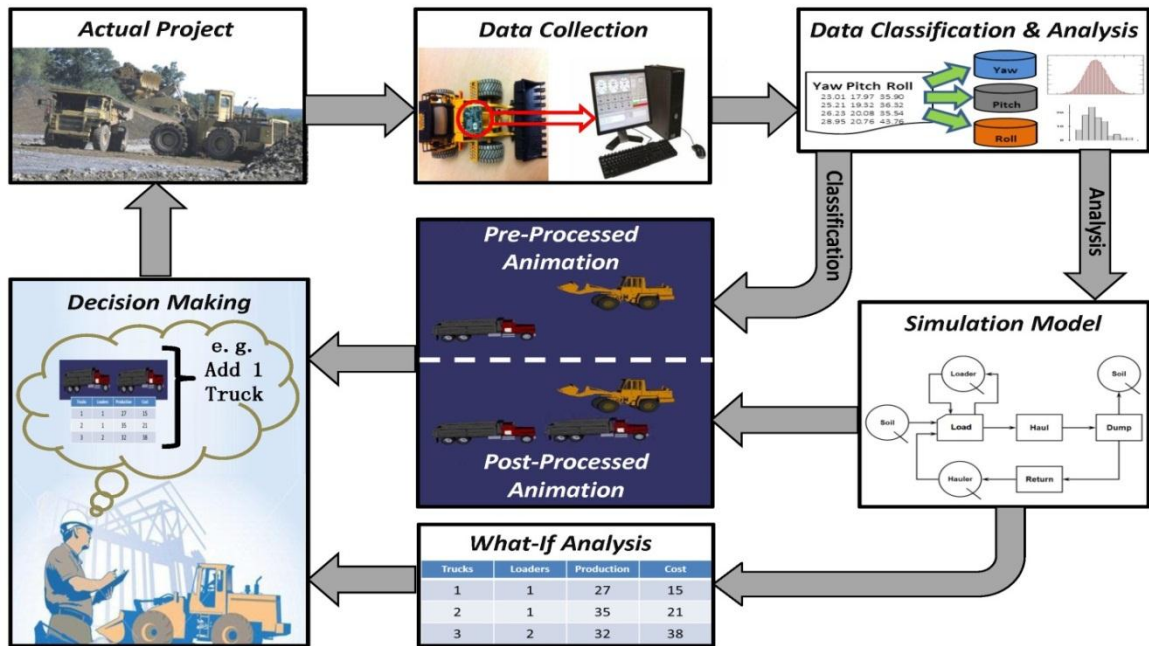


Figure 5.1: System Architecture of the Developed DDDAS Framework

5.2 Real Time Data Collection

Banks [14] summed up the simulation environment from a data collection point of view by indicating that data are rarely readily available and data collection is one of the most important and difficult problems in simulation modeling. Given a dynamic simulation modeling system, the problem could even get more complicated since the system requires real time field data collection and integration. As a result, data acquisition is one of the most challenging and computing intensive parts of a DDDAS given that it is almost impossible to manually collect real time data in large projects. Depending on the extent and complexity of a project, designing and implementing a reliable means to acquire, communicate, and synchronize data from multiple sources may itself be a major challenge. Real time data is used not only for updating and fine-tuning the model with the

latest changes occurring in the real system, but it also serves as the basis for model validation and verification. Since the model needs to be continually updated, an uninterrupted flow of input data is needed to reflect the latest changes in the status of activities and resources. Therefore, developing and implementing a robust and automated data collection infrastructure including sensing and communication technologies is necessary.

In many construction projects, resources are in constant motion. Examples include dump trucks transferring soil from a cut area to a fill location, crews laying reinforced concrete rebars on a floor slab, and a tower crane lifting steel sections from a flatbed truck. As a result, from a modeling perspective, capturing these changes in resource (e.g. equipment, personnel, and material) positions is necessary. In addition to the positional data, most construction equipment (e.g. cranes, excavators, shovels, loaders) have hinged moving parts and thus, collecting the angles of orientation for these parts is also essential in order to describe their motions. Such data can be acquired using orientation sensors that capture three angles of rotation (i.e. yaw, pitch, and roll). In the presented research, orientation data are captured and transmitted to the system in order to simulate and animate the body configuration of construction equipment in real time.

In the course of this study, the data collection procedure was developed in two different environments. Since both the manufacturer sample algorithms for the data collection device and the open source code visualization toolkit, OpenSceneGraph (OSG), were written in a .NET environment, a data collection system was initially designed by

creating an object-oriented platform in .NET environment. Later, due to the flexibility of LabVIEW graphical programming for more sophisticated data analysis and processing required in this research, a more efficient data collection procedure was developed in LabVIEW using almost the same principles and algorithms originally created in .NET environment. In this Section, first the overall functions and classes of the developed data collection system in C++ are described. Subsequently, a general description of implementing these functions in the graphical programming environment of LabVIEW will be presented.

5.2.1 Serial Port Communication in .NET Environment

Serial is a standard device communication protocol used for transferring data to or from a peripheral devices via computer serial ports [69]. In order to communicate with the data collection devices used in this research, a serial port communication algorithm was developed. A major factor in designing this algorithm was generalizability which in the context of this research, is defined as the ability of the framework to communicate with a variety of data collection devices without the need to significantly modify the communication algorithms. Although the developed algorithm is to certain extents, unique and has been tested with the PNI TCM Prime 3D orientation tracker, it benefits from a generic structure that can be easily used to communicate with other data collection devices that transmit data using the RS-232 protocol. RS-232 is a specification for serial communication and is one of the most popular for sensor connections [69]. Since the collected tracker data is in a binary format, the developed serial port communication

algorithm contains methods to decode the transmitted data and convert them to a computer interpretable format.

The developed algorithm uses the advantages of object oriented programming in Microsoft Visual C++ .NET environment. Using serial communication libraries, the initial communication with the port is established, the port is opened, data (i.e. three orientation angles) is received through the port, and the port is closed at the end of the experiment. The orientation data coming through different brands of orientation trackers follow different data transmission standards. The orientation tracker sensor, PNI TCM Prime module, utilizes a binary data transmission protocol to obtain and extract the tracker data that is transmitted over an RS-232 interface. Each data packet contains a component called Frame Type ID that describes the content of the packet. Based on this ID, the packet may contain each of the 3D rotational angles as well as the current temperature (ranging from -40 °C to +85 °C). These values are stored in the packet Payload [70]. The datagram structure of the PNI TCM Prime module is shown in Figure 5.2.

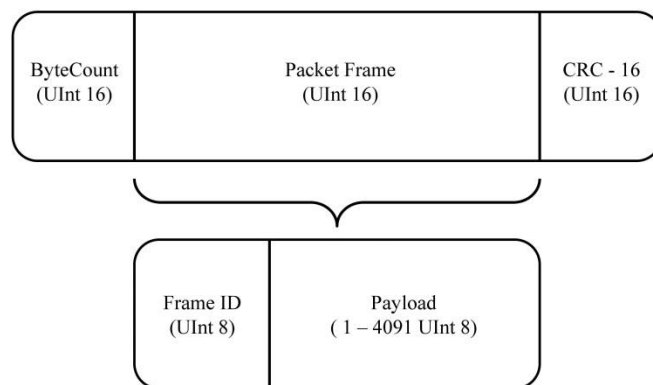


Figure 5.2: Datagram Structure of the PNI TCM Prime Orientation Tracker

Using the binary data provides the system with the advantage of fast data transmission. However, this will in turn make the communication very sensitive to data corruption. As a result, a mathematical transformation method called the Cyclic Redundancy Check (CRC) is used to separate useful and corrupted binary data packets. CRC is applied to a series of bytes and produces an integer result that can be used for error detection. After data is received from the orientation tracker, the tracking application computes the CRC value using the existing contents of the data packet and compares this value to the one originally calculated when the packet was being constructed prior to transmission. If the two values are not identical, the packet is considered as corrupted and will be disregarded and the application waits for the next data packet. If the two values are equal, the data is safe to be used and extracted into its components. Using a set of binary data manipulation statements provided in the application programming interface (API) of the tracker device, the numerical values for each of the orientation angles are obtained. The main functionalities of the managed C++ class developed for acquiring orientation tracker data through a serial port is shown in Figure 5.3.

```

__gc class PRIME
{
    Public:
        Initialize ();
        Control ();
        ReceivePacket ();
        CRC ();
        ParsePrimePacket ();
        Shutdown ();
};

```

Figure 5.3: C++ Orientation Tracker Serial Communication Class

The *PRIME::Initialize()* function is called first to open the serial port and set up the port properties (e.g. baud rate, data bits). Then, *PRIME::Control()* extracts the Payload piece by piece. The number of requested angles (up to three) should be defined in this function. Based on the number of requested data pieces, this function will be called consecutive times and each time sends a control command to the tracker. In response, the tracker sends a single packet containing binary values of the requested angles. For example, if all three orientation angles (yaw, pitch, and roll) are required, the function will be called three times and in return, the tracker sends binary values of three angles. Next, *PRIME::ReceivePacket()* is called to receive the binary data packet. This is followed by a call to *PRIME::CRC()* to check if the received data is error-free. If the data is not corrupted, the contents of the packet will be extracted by *PRIME::ParsePrimePacket()*. This function stores the numerical value of the required angles in numerical variables which will later be used to construct and display the real time animation of moving parts. Finally, the *PRIME::Shutdown()* class will close the port. The flowchart in Figure 5.4 shows major steps in acquiring orientation data using the PRIME class introduced in

Figure 5.3. More information about all other functions and classes developed using Microsoft Visual C++ can be found in Appendix A.

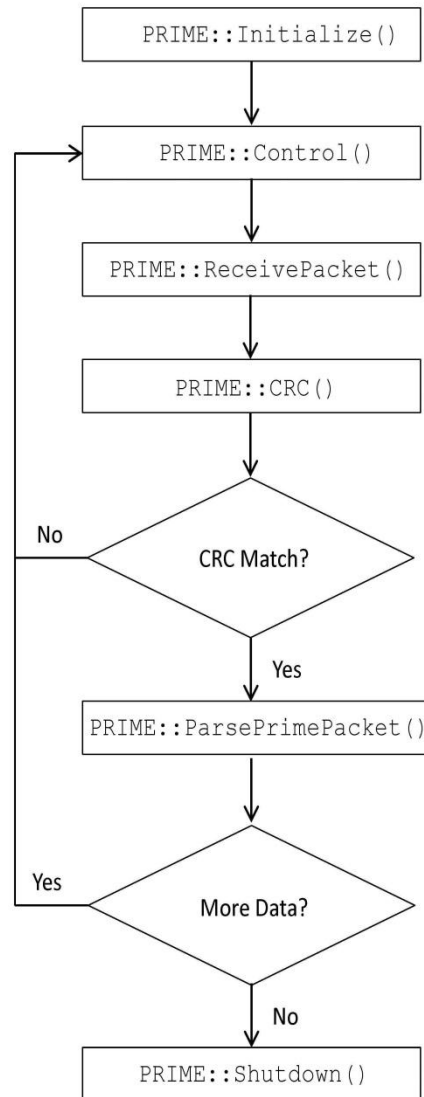


Figure 5.4: Flowchart of 3D Orientation Tracker Serial Communication Process

5.2.2 Data Collection Using LabVIEW

The presented methodology for data collection takes advantages of LabVIEW graphical programming environment developed by the National Instruments (NI). This essentially

enables the creation of a standalone data collection and analysis framework that uses the same functionalities employed in C++ but in a more efficient manner. LabVIEW, in essence, is a system design platform that enables automating data collection and measurements supporting a wide variety of sensors. As a powerful data collection tool, LabVIEW abstracts much of the administrative complexity of computer programming such as memory allocation and language syntax [71]. It was used in this research in order to create a single platform that provides more control and flexibility as far as data collection and analysis, and displaying the results in a highly interactive (i.e. visual) environment are concerned. Figure 5.5 shows a sample snapshot of the LabVIEW graphical environment. Each program written in LabVIEW is called a Virtual Instrument (VI) which consists of a graphical user interface (i.e. Front Panel) and a graphical code (i.e. Block Diagram). Each node in a Block Diagram performs a specific task and is connected to other nodes via wires. More information about LabVIEW graphical programming can be found in Appendix B.

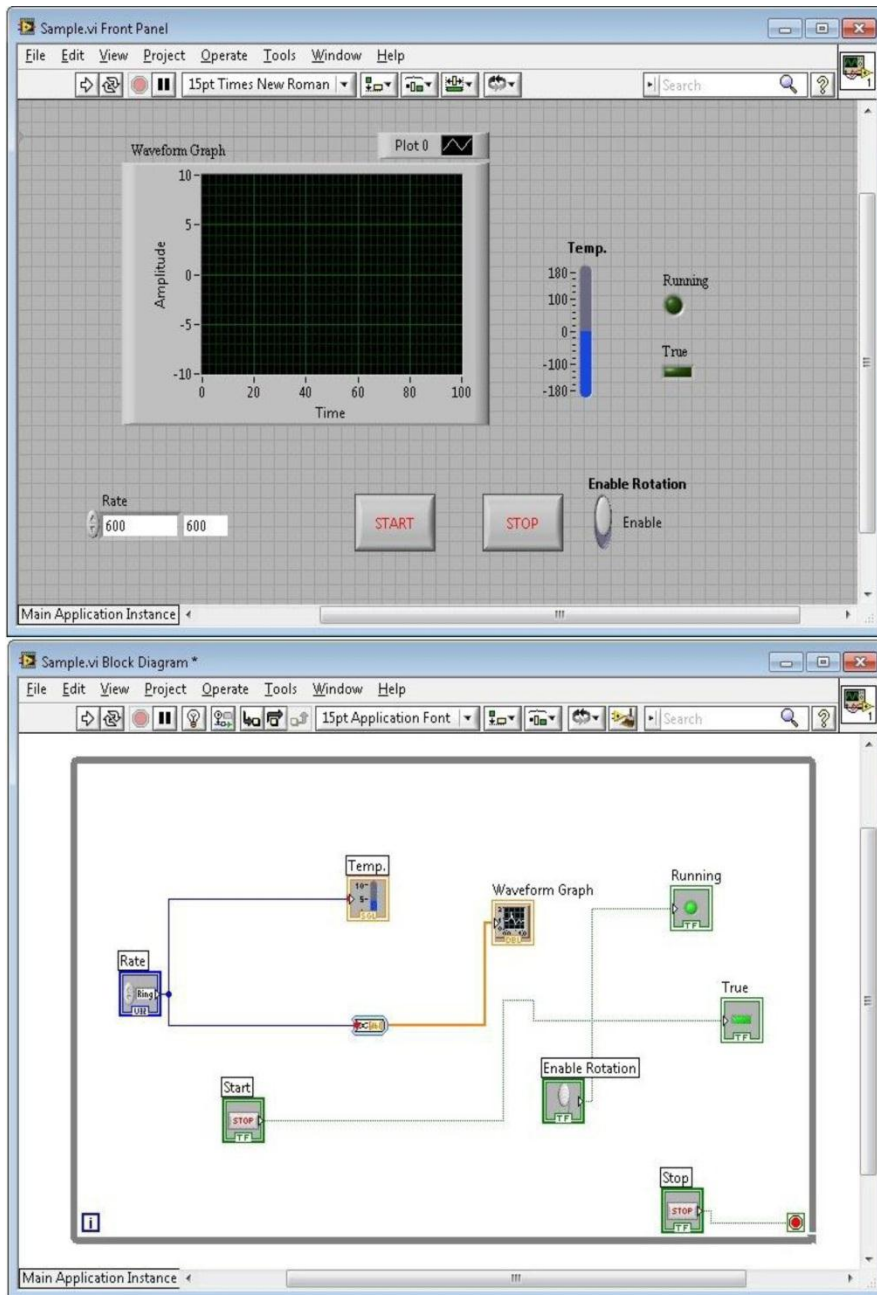


Figure 5.5: A VI Consists of a Front Panel and a Block Diagram

In this research, a real time data acquisition VI was designed and implemented to customize and append an instrument driver for the PNI TCM orientation tracker.. To create an interface between the instrument driver and the data collection device, the NI Virtual Instrumentation Software Architecture (VISA) API was used for serial communication. VISA, basically provides users with the ability to open, configure (i.e. setting baud rate, flow control, parity), write to and read from, and close any type of interfaces such as GPIB, TCP/IP, Ethernet/LAN, IEEE 1394, USB, and serial, and handle errors in a fast and easier way in comparison with the same functions developed in a text-based programming environment (e.g. C++). Figure 5.6 shows a rudimentary structure of the VISA implemented in the developed framework.

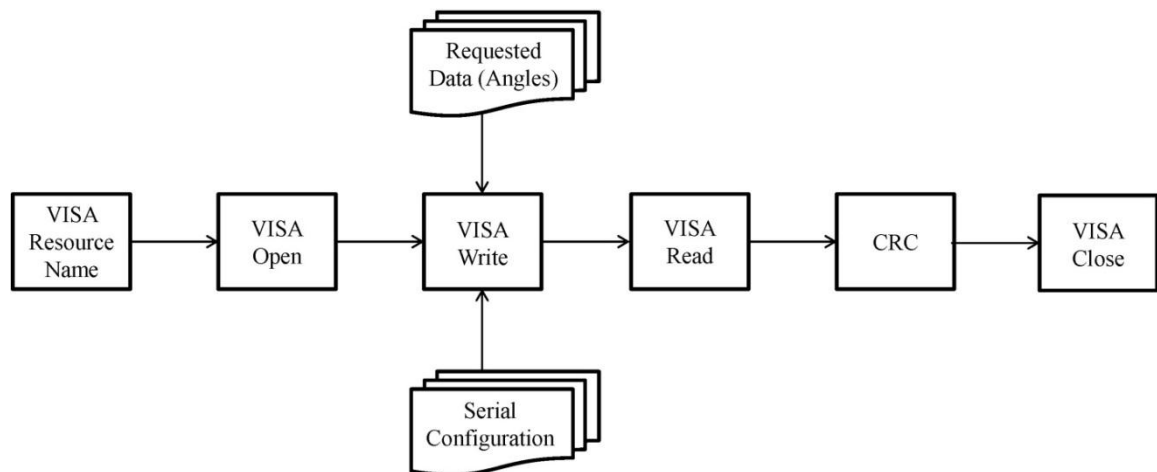


Figure 5.6: Developed VISA Interface Structure

As shown in this Figure, *VISA Resource Name* passes session information between instrument driver and SubVIs and is a unique identifier reference to the data collection device (e.g. COM1, COM2). *VISA Open* essentially opens a session to communicate with

the device specified by the *VISA Resource Name* and returns a session identifier that can be used to invoke operations on that device. This is equivalent to the function *PRIME::Initialize()* in the Figure 5.3. *Serial Configuration*, as stated before, sets the port configuration parameters specific to the device such as baud rate which is a measurement for communication speed equal to 38,400 HZ for the PNI TCM module. *VISA Write* has a performance similar to *PRIME::Control()* in the Figure 5.3. It extracts the Payload and requests needed angle measurements. Subsequently, *VISA Read* reads the requested data based on what was defined in *VISA Write*. The CRC will be performed to detect corrupted data packets by calculating the CRC-16 of the output string from the tracker and comparing it to the checksum at the end of the output string. In essence, this procedure is identical to what was described in Subection 5.2.1. Finally, similar to function *PRIME::Shutdown()*, *VISA Close* shuts down the port and terminates the software connection to the device.

The advantage of real time automated data collection is that it enables the simulation model to update itself in response to changes in the project environment. This can be achieved by continually collecting time-stamped data. However, before the raw data stream enters the simulation model or is used as input for visualization, it should be classified, analyzed, and converted to a format that defines the state and the context of the entity for which the data is collected. As such and as shown in Figure 5.1, the raw data collected using either the .NET or LabVIEW operational environments is passed onto the data classification and analysis module of the developed DDDAS framework. The following Subsection provides more details about this module.

5.3 Automated Data Classification and Analysis

One of the major challenges in collecting a large volume of heterogeneous information is that unnecessary data may also be inevitably collected. For example, in order to mathematically describe the motion of a loader's boom within the context of an earthmoving operation, a 3D orientation tracker mounted on the boom would capture three angular values namely yaw, pitch, and roll. However, given that the boom must be raised or lowered to load or unload a truck, the main piece of information needed to determine the start and end times of load or unload activities is the pitch angle. As such, potential trembles resulting in small changes in the roll angle and also possible motions such as sidewise movements and maneuvering of the loader leading to a change in the yaw value are to the most extent, redundant as far as detection the beginning and end of load and unload activities for the simulation model and having a smooth animation for visualization are concerned. Therefore, collected data must be carefully classified so that only relevant and useful information is passed onto the next steps.

Classified data also needs to be transformed into a proper format interpretable by the simulation model. One such format is a numerical representation of activity durations using probabilistic distributions. Since discrete events mark the beginning and end of each activity, identifying the duration of individual activities can be achieved by detecting time-stamped events corresponding to the beginning and end of that activity. Therefore, activity durations can be derived from the pool of classified collected raw data and suitable probability distributions will be then fit to the calculated duration values. In

the earthmoving example described above, the angle of the boom and the truck bed relative to the horizontal line can be used to identify the start and end points of load and unload activities and determine activity durations. For example, in the operation depicted in Figure 5.7, activity durations can be calculated by comparing the time stamps corresponding to when each event (i.e. raise boom, load truck, lower boom, haul, raise bed to dump, lower bed, return) occurs based on the orientation data (i.e. angles) received from sensors mounted on the equipment.

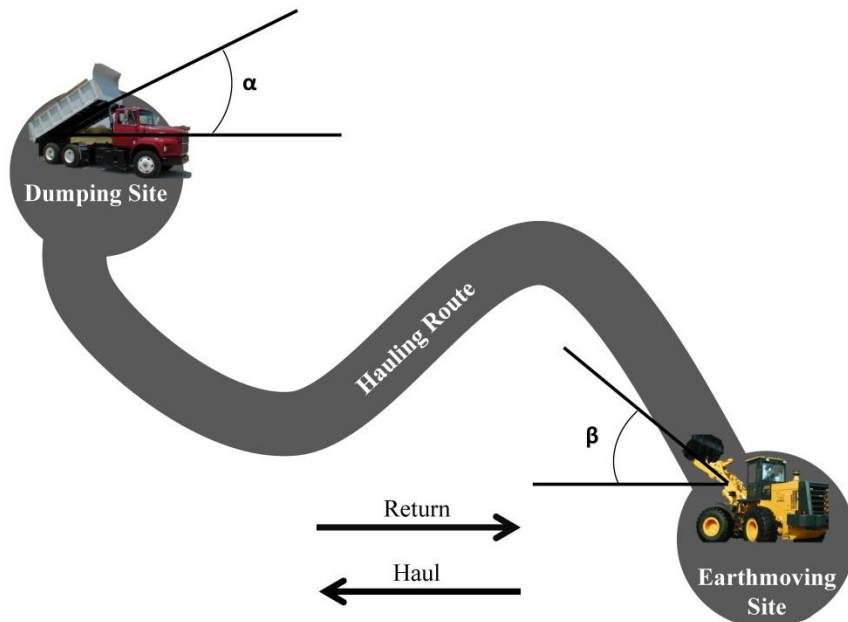


Figure 5.7: Simplified Layout of an Earthmoving Operation

An example of how a series of time-stamped data can be used to extract certain activities and their durations is illustrated in Figure 5.8.

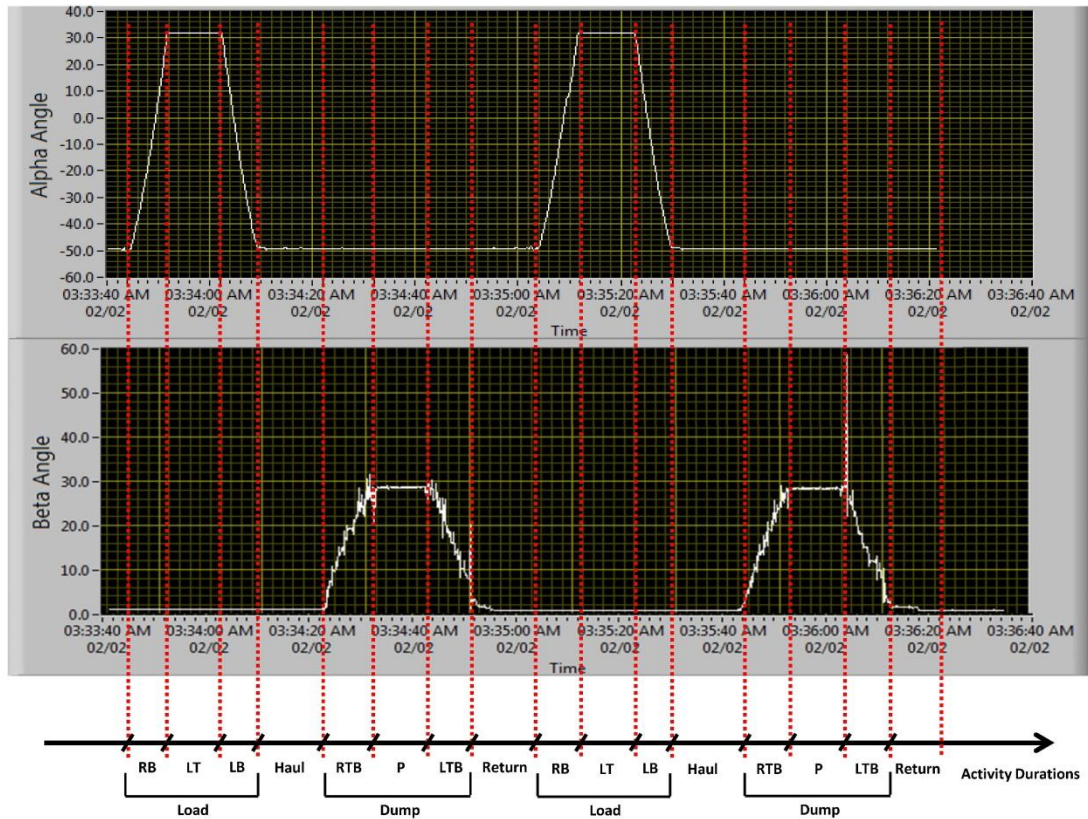


Figure 5.8: Activity Durations Based on the Variation of Equipment Body Orientation with Respect to Time
 (RB = Raise Bucket, LT = Load Truck, LB = Lower Bucket, RTB = Raise Truck Bed, P = Put, LTB = Lower Truck Bed)

In this Figure, the first diagram shows changes of angle α (loader boom angle relative to the horizontal line) and the second diagram shows angle β (truck bed angle relative to the horizontal line) over time. Considering angular variation histograms displayed in these two diagrams, a timeline representing the duration of each activity can be generated. For example, an increasing angle α and a constant angle β (close to zero) indicate that the loader is raising its boom while the truck is waiting to be loaded (RB in Figure 5.8). A near constant angle α (close to its peak value) and a constant angle β (close to zero) indicate that the loader is putting soil into the truck (LT in Figure 5.8). A decreasing

angle α and a constant angle β (close to zero) indicate that the loader is lowering its boom while the truck is preparing to move (LB in Figure 5.8). An instance of “Load” activity is completed when all three (RB, LT, and LB) processes are completed.

A similar analysis can be done to isolate instances of “Haul”, “Dump”, and “Return” activities. For instance, given that angle α is constant (at a value close to zero), if angle β is increasing from zero, the truck bed is being raised (RTB in Figure 5.8), if angle β is almost constant (close to its peak value), soil is being dumped (P in Figure 5.8), and if angle β is decreasing, the truck bed is being lowered (LTB in Figure 5.8). These three processes, put together, will constitute an instance of “Dump” activity. Since histogram data is time-stamped, duration values can be easily determined for all such instances. Mathematical models will then be applied to a well-populated pool of these calculated durations to determine a distribution function that best represents the duration of that activity. This distribution function is then used to describe the duration of that activity in the corresponding DES model [63].

It is worth mentioning that given the unavailability of GPS to obtain time-stamped positional data in an indoor environment (e.g. laboratory setting where the components of this framework was tested), a number of simplifying assumptions had to be made when developing the methodology for extracting the duration of activities. For example, it was assumed that the haul activity would not start until the loader lowers its boom and would not finish until the truck raises its bed. Likewise, return activity starts when truck’s β angle reaches zero and finishes at the beginning of the load activity, when the loader

starts raising its boom. It is clear that incorporating positional data into the proposed algorithms for calculating activity durations not only does eliminate the need to consider these and similar simplifying assumptions, but also enhances the accuracy of the algorithm. As such, future work in this research will include activities specifically targeting this need.

The classification and analysis module accepts input from data collection devices, outputs the classified data for pre-processed animation, and also analyzes data to be passed onto the simulation model. This guarantees that only relevant data is used and that the simulation model is not only the receiving end of the process but also can steer the data collection process by requesting additional field data to be collected, if necessary. The classification and analysis process also includes statistical analysis algorithms to categorize the activities based on the trend of the collected data and to remove the outliers and eliminate the non-relevant data. The next Subsection describes the developed system for implementing such algorithms.

5.3.1 VI Structure for Data Analysis Module

In order to create a standalone platform consisting of both data collection and data analysis modules, all mathematical and logical functions for data classification, extraction of activity durations, and statistical analysis were appended to the same VI. Figure 5.9 shows the VI structure.

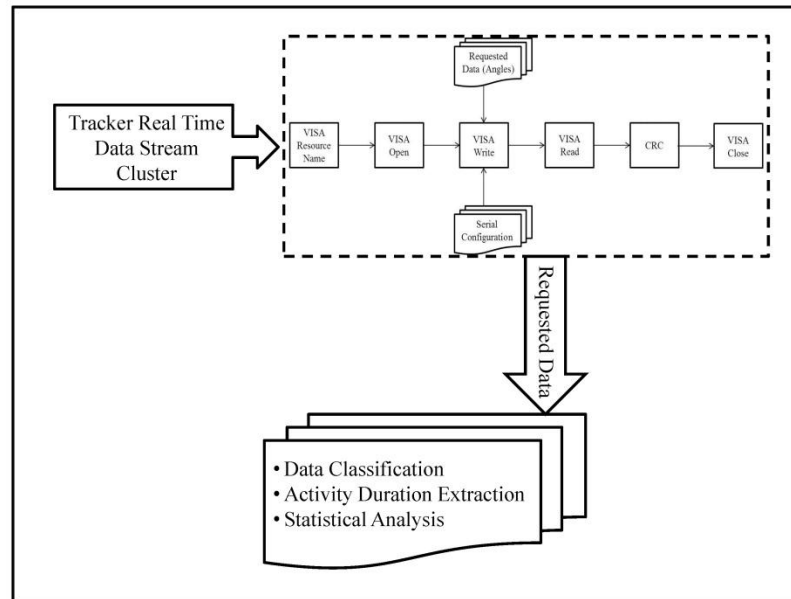


Figure 5.9: Single VI Containing Data Acquisition and Data Analysis Functions

A cluster of real time orientation data enters the VISA interface and undergoes steps depicted both in Figure 5.6 and by a dashed outline in Figure 5.9. This is followed by raw data being classified and time-stamped. The time-stamped data will then be used to calculate activity durations using several mathematical and logical commands built into the VI. Statistical analysis will be also performed on the well-populated pool of duration to calculate mean, standard deviation and other parameters required to describe activities in a data-driven simulation model.

5.4 Data-Driven Simulation

As far as the system architecture illustrated in Figure 5.1 is concerned, once the data is available after the classification and analysis step, input parameters for simulation model

are determined. Construction operations can be broken down into and modeled as a system of discrete activities which makes DES a viable method for simulating such operations. One of the most commonly used DES systems is STROBOSCOPE [47]. STROBOSCOPE, initially designed for construction operations, is an open-design programming language that enables users to make complex dynamic decisions and thus, control the simulation at run-time. The advantage of STROBOSCOPE over many other existing DES modeling platforms is that it considers the diversity of resources and their characterizations. In addition, it has been built upon the concept of traditional activity cycle diagram (ACD) which makes it suitable for modeling a large group of construction operations that are cyclic in nature. STROBOSCOPE models are based on a graphical network of interconnected modeling elements. A DES model of a sample earthmoving operation is illustrated in Figure 5.10.

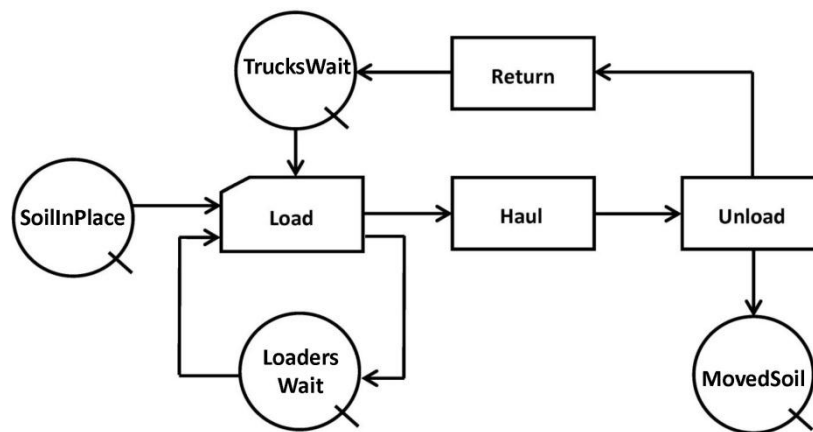


Figure 5.10: DES Model of a Typical Earthmoving Operation

In this Figure, *SoilInPlace*, *LoadersWait*, *TrucksWait*, and *MovedSoil* are queues where resources wait before being drawn to activities (if needed). Also, *Load* is called a combi

activity since it immediately follows a queue, and *Haul*, *Unload*, and *Return* are normal activities. In order for a STROBOSCOPE model to describe a real system, attributes such as activity durations, number of entities, and resource capacities must be known. In the absence of collected field data, assumptions and personal judgment is normally used to quantify such parameters. As previously stated, one of the main motivations behind this research was to investigate if further improvements can be made to the existing approach of assigning values to simulation parameters by designing a methodology that incorporates field data to obtain more realistic simulation parameters.

STROBOSCOPE models consist of a series of programming statements written in a script input file. All parameters pertinent to the characteristics of each model element should be defined through those programming statements. Therefore, once the appropriate simulation parameters are determined from the collected data, the simulation script is opened and updated based on the calculated parameters.

5.5 Pre- and post-processed Animations

Chapter 4 illustrated a detailed description about OSG, the visualization toolkit used in this study. Following the data flow illustrated in Figure 5.1, as soon as appropriate field data is collected and classified, a concurrent 3D dynamic animation of ongoing activities is created. This pre-processed (i.e. generated before data is fed into the simulation model) animation can assist in detecting potential conflicts and enhancing safety and monitoring of the project. The other benefit of this animation is that unlike many existing site monitoring systems which mainly rely on video streaming, finding the best spots to

install cameras such that every action can be monitored with a free line-of-sight is no longer an issue. This is due to the fact that once the animation is rendered on the screen, the user has complete control over the viewpoints and can change their locations and directions of look, if necessary. For example, the user can zoom in an out or navigate around the animated scene to gain a better visual perspective of certain parts of the operation since as stated in Chapter 4, OSG provides the opportunity to change the viewpoint to observe the scene from any desired angle.

In addition to the pre-processed animation, the results of the DES model can be used to create a post-processed animation. As demonstrated in literature, providing visualized output of a simulation model is preferred by many construction planners and analysts since very often, making decisions solely based on the textual output of conventional simulation systems is time consuming and prone to unwanted biases and mistakes [54, 72]. However, in addition to the benefits that general visualization of simulation models has, providing decision-makers with two identical animations, one based on the exact real movements occurring on the jobsite and the other based on the output of the updated simulation model with the latest data obtained from the field provides an extremely convenient way to evaluate and compare different scenarios with the concurrent field configuration and make more realistic decisions. For example, since each construction project is unique in terms of requirements and usage of its working space, having a real time data from the project and evaluating different scenarios based on the transformation, requirements, and limitations of the working space (e.g. maneuverability issues for loading and dumping activities in earthmoving operations, visibility problems for the

crane operator in steel girder erection, safety problems and detecting potential collision, overcrowding in particular work zones), prevent decision-makers from making general assumptions based on historical data or their expert eye on the work. Hence, displaying the pre-processed animation side-by-side the simulation-based post-processed animation enables decision-makers to see first-hand how current trends on the jobsite (reflected in the pre-processed animation) and the expected performance of resources (as displayed in the post-processed animation) are related, and hence effectively serves this purpose.

Finally, another major advantage of having pre- and post-processed animations is that comparing the two animations greatly facilitates the validation and verification of the simulation model. In this case, the modeler can intuitively make sure whether the model contains any modeling flaws or whether it performs as intended (i.e. verification). Also, it can be visually determined by people who are not construction experts whether the simulation model accurately represents the real world.

Creating realistic post-processed 3D animations of a simulated construction processes is a complicated task that has been previously studied by a number of researchers. A recent example of a post-processed 3D visualization platform is VITASCOPE. VITASCOPE is a general purpose 3D animation system for visualizing simulated processes modeled in simulation tools such as STROBOSCOPE, capable of writing formatted output during a simulation run. Based on the logged simulation model runtime data, VITASCOPE graphically illustrates modeled operations by processing sequential, time-ordered animation commands in an ASCII text file [10]. While VITASCOPE is a great tool for

creating post-processed (simulation-based) animations of construction activities, the existing visualization capabilities of the framework developed in this research enable the generation of a pre-processed animation using the same OSG-based environment and 3D CAD models of construction equipment. As described earlier, these two animation streams, when simultaneously displayed, can facilitate the process of validation and verification of the simulation model while providing a means to intuitively compare different scenarios tried in the simulation model.

It is worth mentioning that since the designed framework is intended to function in an automated manner, and since the OSG visualization toolkit is written and extended using the C++ programming environment, a middleware for linking LabVIEW (i.e. containing data collection and analysis functionalities) to Microsoft Visual C++ (i.e. containing OSG visualization platform) had to be designed. To this end, ActiveX Automation technology was used in this research. ActiveX has an interface that allows individual programs to be linked together to suit for specific computing needs [73].

5.6 What-If Analysis

Another building block of the framework as shown in Figure 5.1 is the “What-If Analysis” module. In order for a construction engineer to make necessary decisions regarding the complex processes, different scenarios need to be assessed and the cost and time associated with each scenario must be determined. For example, a decision regarding equipment fleet to be used in an earthmoving operation could be the one associated with the minimized expected cost [47]. Considering all possible configurations

in terms of crew sizes, number of equipment and their arrangements, operations logic, and construction methods, a decision-maker may end up having to choose from several combinations to perform a certain task. Using simple methods such as subjective mathematical comparisons or more complex optimization models, the engineer can then determine the best configuration that satisfies the predefined criteria (e.g. objective function, time, cost).

5.7 Decision-Making and Dynamic Feedback

The last component of the system architecture presented in Figure 5.1 is Decision-Making and Dynamic Feedback. The developed algorithms for data collection, classification and analysis, simulation and visualization, will be best used in the presence of a human decision-maker or a team of decision-makers who will be ultimately responsible for making the required modifications to the target construction process. As stated earlier, presented data to the user contains two juxtaposed animations; one identical to the actual process taking place in the jobsite (pre-processed), and the other, resulted from simulating alternative scenarios (post-processed). Also, the user is provided with the results of the simulation model and the output of the what-if analysis in order to decide which alternative solution is the most appropriate. Therefore, not only by intuitively watching side-by-side animations, but also through intelligently interpreting performance attributes (e.g. productivity rates) from the simulation output and various alternative scenarios, the decision-maker(s) will have the ability to further adjust future processes. Ultimately, and due to the dynamic nature of construction projects, the cycle

presented in Figure 5.1 will repeat to reflect any further changes occurring in the process. In other words, the next phase of data collection starts after expert modifications are applied to the construction resources, and activities and a new set of data will be classified, analyzed, simulated, and visualized. This guarantees that through continuous data collection from the equipment involved in a construction process, at any given time, the system will be functioning at its best performance level.

CHAPTER 6: LABORATORY SCALE EXPERIMENTS AND RESULTS

Chapter 5 outlined the individual components of the developed framework and their relationships in the context of the overall system. In this Chapter, results of preliminary proof-of-concepts experiments conducted in the Decision Support, Information Management, and Automation Laboratory (DESIMAL) at the University of Central Florida are provided to demonstrate the validity and applicability of the developed methodology and algorithms for data collection, data analysis, visualization, and data-driven simulation. In particular, the validation phase included a number of laboratory-scale equipment operations scenarios designed and implemented to test certain aspects of the developed framework. In addition, a comprehensive experiment was carried out in which the robustness, applicability, and overall functionality of the framework in terms of data collection and classification capabilities, ability to generate realistic pre-processed animations, and effectiveness to create data-driven simulation was validated. The following Subsections provide more insight about the details and outcomes of each of the validation experiments.

6.1 Preliminary Results

6.1.1 Experiment Tools and Peripheral Devices

Preliminary experiments were performed on a laboratory-scale Construction Equipment Automation Platform (CEAP) using remotely-controlled model construction equipment.

A NetCam XL IP-addressable camera and a Dell™ Precision T1500 desktop system were also deployed. The camera was used to demonstrate the correctness and precision of the pre-processed visualization and the desktop system was the main computing platform. Figure 6.1 illustrates the overall arrangement of the tools and peripheral devices namely the CEAP, model construction equipment, IP-addressable camera, and the computer system.

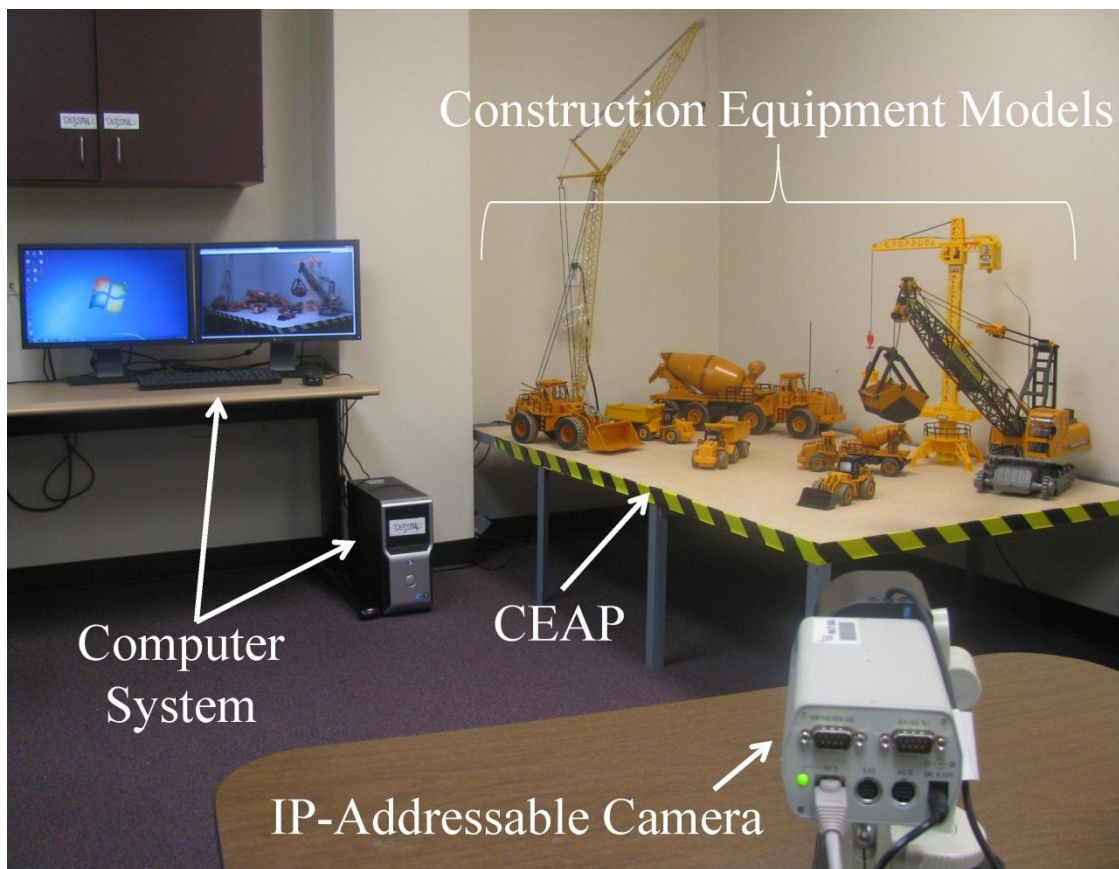


Figure 6.1: Overall Arrangement of Experiment Tools and Devices

In order to collect equipment motion data several PNI TCM 3D orientation trackers were used. These modules were mounted on model construction equipment to capture and

transmit three angular values namely yaw (heading), pitch (tilt), and roll. Figure 6.2 shows a PNI TCM 3D orientation tracker mounted on a model excavator with definitions of yaw, pitch, and roll angles. Also manufacturer’s specifications of this orientation tracker are listed in Table 6.1.

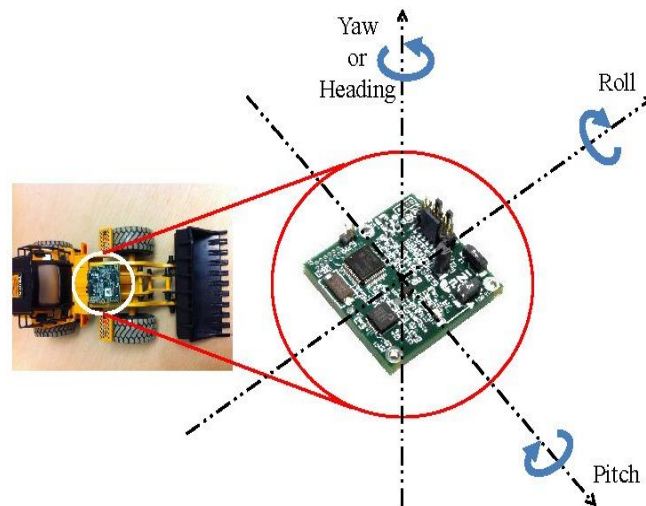


Figure 6.2: A Prime 3D Orientation Tracker Mounted on a Model Excavator with Definitions of Yaw, Pitch and Roll Angles

Table 6.1: Manufacturer’s Specifications of Prime 3D Orientation Tracker

Angle	Parameter	Value	
Heading	Range	360°	
	Accuracy (tilt ≤ 45°)	1° rms	
	Resolution	0.1°	
Tilt (Pitch/Roll)	Range	Pitch	±90°
		Roll	±180°
	Accuracy		1° rms
	Resolution		0.1°

6.1.2 Single Object Data Collection and Visualization

Initially, a series of small-scale validation tests were conducted using data collected from only one model construction equipment. Later, data collection, data analysis, and visualization algorithms were modified to enable data capturing and processing from multiple objects.

The first in a series of these experiments was conducted using an orientation tracker mounted on a model loader [74]. Figure 6.3 shows the loader on the CEAP and the orientation tracker attached to the boom of the loader.

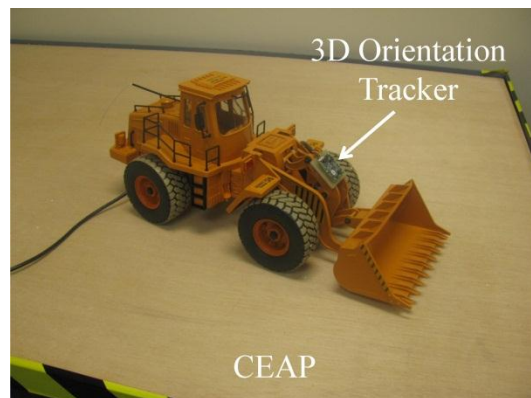


Figure 6.3: Orientation Tracker Mounted on a Loader's Boom

The first step in conducting each experiment was to collect and classify equipment motion data to provide necessary input for visualization and data-driven simulation. As stated in Chapter 5, data collection and analysis is performed using LabVIEW. Figure 6.4 shows the Front Panel (i.e. user interface) of the data collection system for the validation experiment using a single model loader. As shown in this Figure, the interface of the developed VI enables a user to specify a communication port to receive data from the

orientation tracker, start and stop the data collection task, and view the numerical values of the collected orientation angles.



Figure 6.4: Front Panel of Data Collection VI for a Single Loader

Once the VI is launched the data collection task begins. This is followed by a continuous stream of real time classified angular data displayed in the three indicators designed to show yaw, pitch, and roll values. These indicators are marked as YawLoader, PitchLoader, and RollLoader in Figure 6.4, respectively. As soon as the user switches off the data collection using the “Stop Data Collection” button the data stream stops. By clicking on “Start Animation”, an animation of the exact same movements of the loader’s boom will appear on the screen. Figure 6.5 shows snapshots from this test. In this Figure, several frames of the live video streams of the real system captured using the IP-

addressable camera are displayed next to the corresponding 3D animation frames generated by the developed framework in real time.

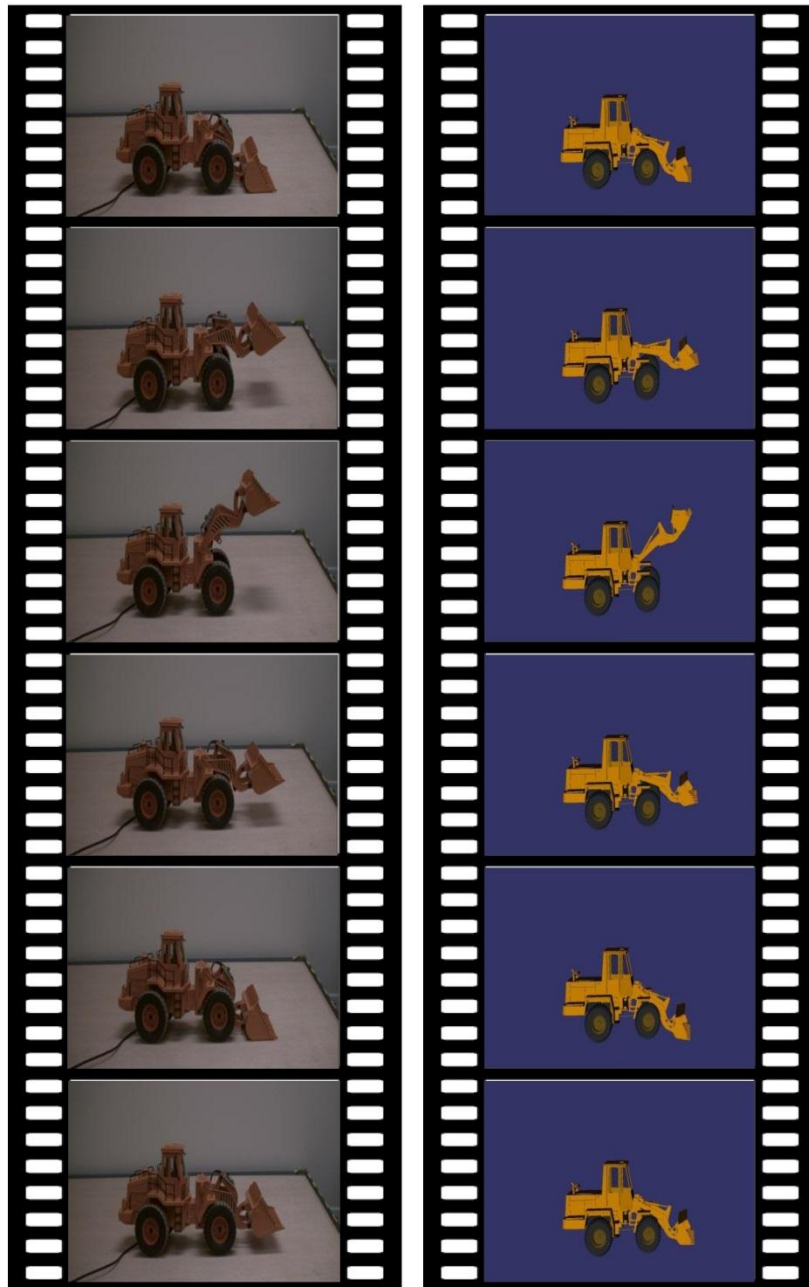


Figure 6.5: Real Time Display of Loader's Boom Movements and Corresponding 3D Animation Generated in Real Time

6.1.3 Double Object Data Collection and Visualization

In order to validate the generalizability of the overall framework and to demonstrate that the developed methods will properly function in situations where operational data from more than one piece of equipment has to be collected, the data collection algorithms in LabVIEW as well as the visualization methods (created in .NET using OpenSceneGraph) were slightly modified. In doing so, the major issue that was successfully addressed was to update the processes inside the data analysis module to be able to identify individual activities from a large pool of raw motion data collected from several pieces of equipment using multiple data collection devices (i.e. 3D orientation trackers), determine the logical relationships and interactions between a group of equipment, and consequently extract activity durations [63]. To validate the newly developed methods, a laboratory-scale experiment was set up where operational data was collected from two models, and the collected data was processed to generate a live 3D animation as well as the calculate the main input parameters needed by the data-driven simulation module to describe equipment activities in a meaningful format. In this experiment, two orientation trackers were mounted on a model loader and a model truck. Figure 6.6 shows these equipment placed on the CEAP while the orientation sensors are mounted on the loader's boom and the truck's bed.



Figure 6.6: Orientation Trackers Mounted on a Loader's Boom and a Truck's Bed

Similar to the validation experiment using only one object, a VI was created and implemented for data collection. However, this time not only the three angular data are shown, but also two diagrams containing a series of time-stamped data to extract activity durations are illustrated in the Front Panel. Each of these histograms shows the how incoming data collected from the orientation tracker changes over time. For example, the trend of data corresponding to the loader's boom indicates that the boom is first lowered from its initial state, raised and remained in a steady state for some time, lowered again and remained in a steady state for a while, and finally raised. By observing this data trend, one can conclude that the loader was involved in a cycle of digging soil (boom down position) followed by loading a truck (boom up position). Based on the collected data and using the developed algorithms for detecting individual activities from a series of angular data, mean and standard deviation of durations were calculated and displayed. Figure 6.7 shows the Front Panel used in this experiment.

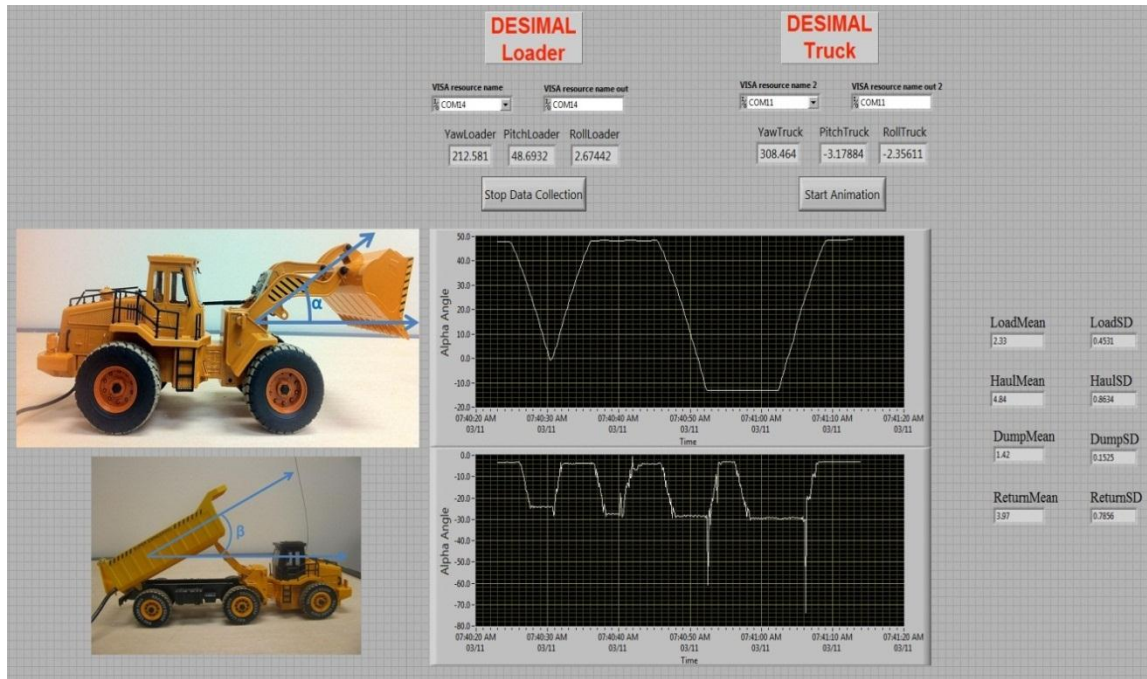


Figure 6.7: Front Panel of Data Collection VI for a Double Object Experiment

Similar to the scenario in which only one object was used, a real time stream of motion data is captured and displayed in the specific indicators on the Front Panel. Simultaneously, each activity is detected by the VI based on the existing data trends and activity duration is calculated using mathematical algorithms inside the corresponding Block Diagram. Figure 6.8 illustrates a portion of the extensive Block Diagram developed for data collection and analysis purposes.

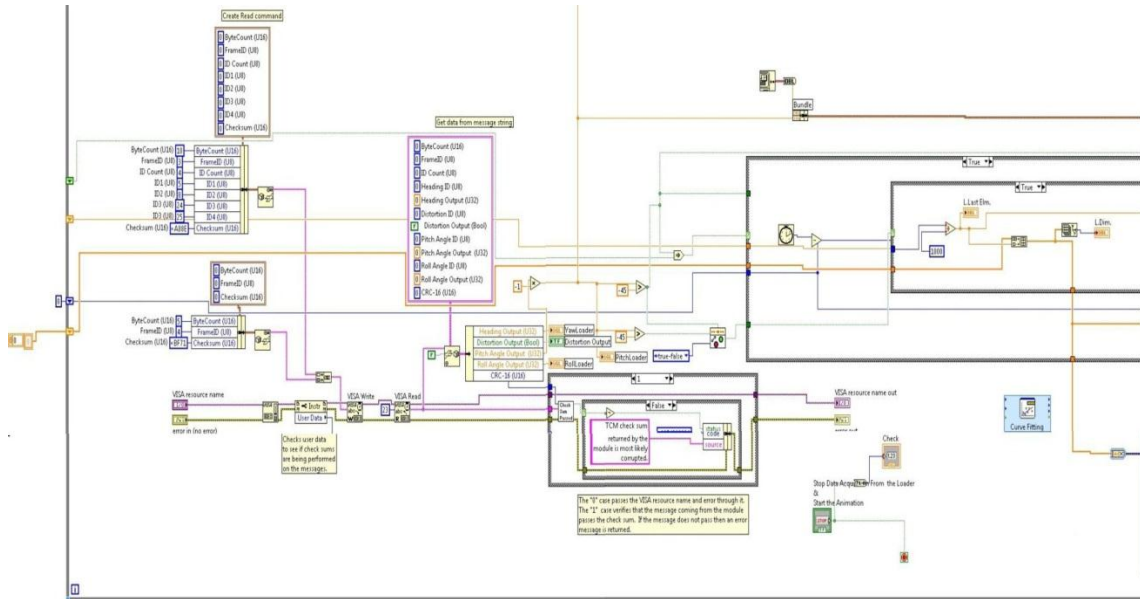


Figure 6.8: Partial View of the Extensive Block Diagram Developed in this Research

Individual activity durations are calculated as long as equipment motion data is streaming. The calculated values are used to populate numerical arrays. The content of each numerical array corresponding to a certain activity (e.g. load, dump) is evaluated in real time using statistical methods to determine the mean and standard deviation of the probabilistic normal duration that best fits all values.

Once the data collection is stopped by the user, and the start animation command is triggered, a 3D animation showing the exact same equipment movements appears on the screen. Figure 6.9 shows snapshots from the live video stream of the real system as well as the corresponding 3D animation created in real time.

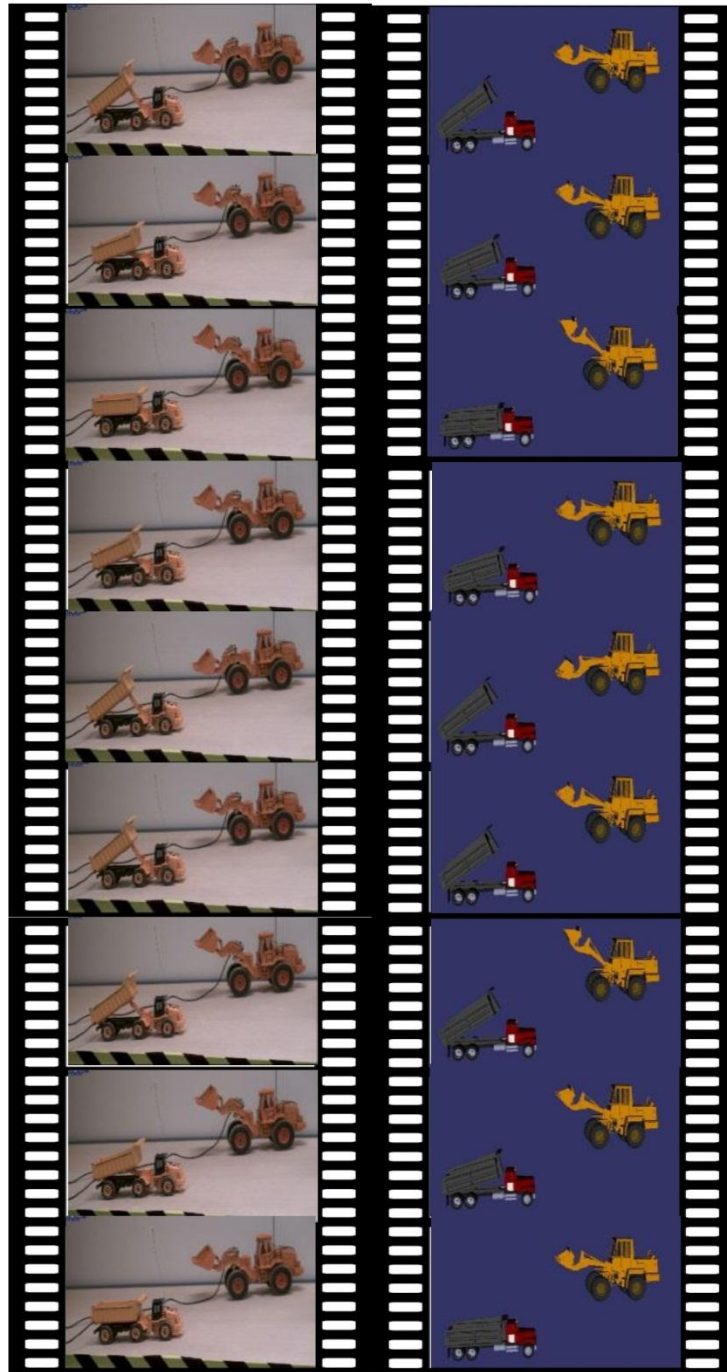


Figure 6.9: Real Time Display of Loader's Boom and Truck's Bed Movements and Corresponding Animations

6.2 Comprehensive Example: Data-Driven Simulation

In order to demonstrate the ability of the developed framework in supporting the prospect of data-driven simulation by collecting, processing, and integrating real time operational data with simulation modeling, a simplified yet comprehensive operational scenario was designed and carried out. In this experiment, the goal was to move 200 pieces of model rocks from a loading area (i.e. Area #1) to a dumping site (i.e. Area 2) for a dam construction project. A model loader was used to load a model truck. The truck would haul the rocks from the loading area to the dumping site. It was assumed that pieces of rock are so big and heavy that each truck can carry only one rock in each hauling cycle. In order to collect field data, two orientation trackers were mounted on the model equipment; one on the loader's boom, and the other on the truck's bed. Figure 6.10 shows the layout of the experiment conducted on the CEAP.

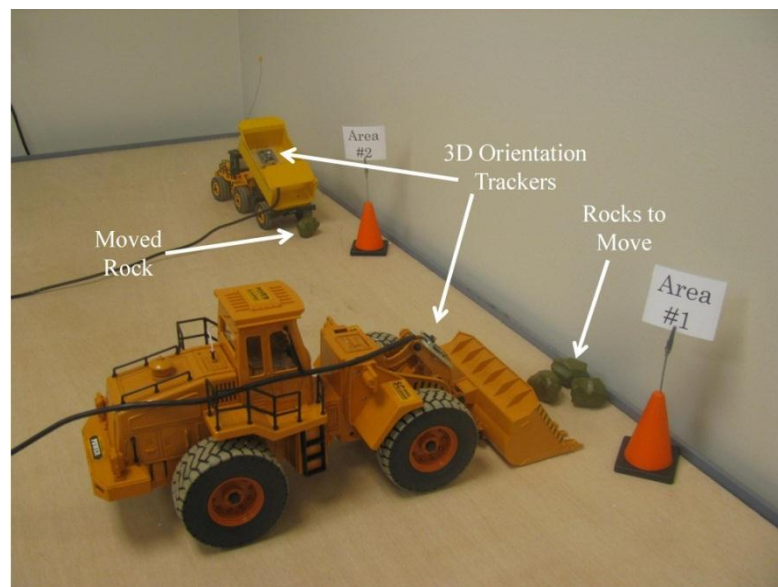


Figure 6.10: Experiment Layout of a Model Dam Construction Scenario

Figure 6.11 shows the DES network of this operation. In this Figure, *RocksToMove*, *LoadersWait*, *TrucksWait*, and *MovedRocks* are queues, *Load* is a combi activity (i.e. it immediately follows a queue), and *Haul*, *Dump*, and *Return* are normal activities. Also, all network elements (i.e. activities and queues) are connected by links. Each link has a specific name and can carry a certain type of resource (i.e. *Rock*, *Loader*, *Truck*) from one element to the other. For example, *RK2* is defined as a link connecting *Load* and *Haul* activities which carries the *Rock* resource.

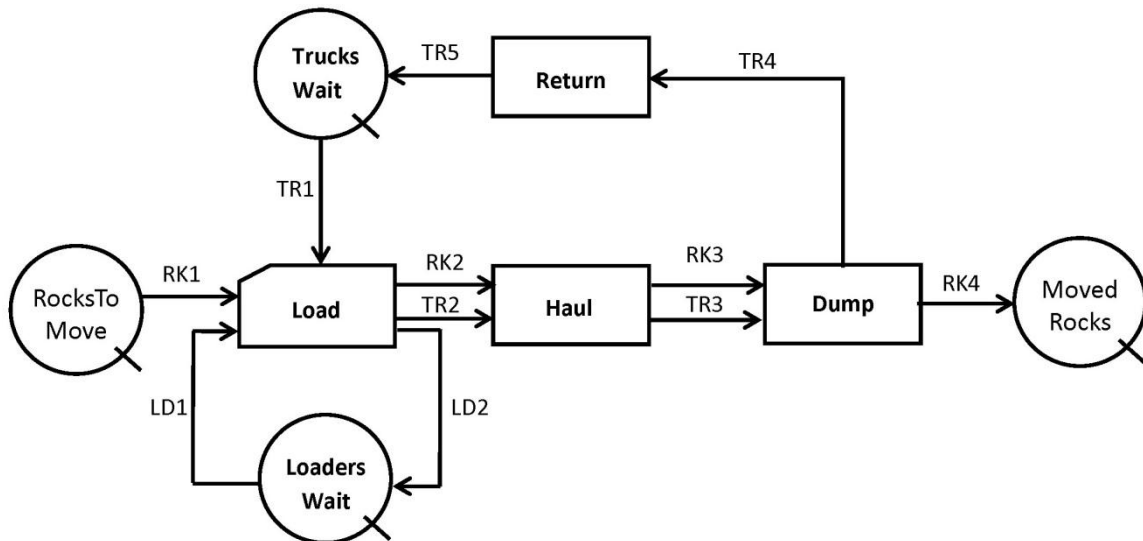


Figure 6.11: DES Model of Rock Hauling Activity

As stated before, during the planning stages of a project, simulation modelers generally rely on expert judgments or field reports from similar past projects to determine model parameters such as activity durations. Following the same logic and as shown in Figure 6.12, a DES script was initially created in STROBOSCOPE for the dam construction

scenario where activity durations were approximated based on the overall arrangement of resources and considering the motion speed of model equipment.

```

////////////////////////////////////
/Rock Hauling Operation Simulation
////////////////////////////////////

/* General Section for Problem Parameters
VARIABLE    NumLoaders    1;
VARIABLE    NumTrucks    1;

VARIABLE    NumberOfRocks 200;

/* Definition of Resource Types
GENTYPE    Loader;
GENTYPE    Truck;

GENTYPE    Rock;

/* Definition of Network Elements
QUEUE     LoadersWait Loader;
COMBI     Load;
QUEUE     TrucksWait  Truck;
NORMAL    Haul;
NORMAL    Dump;
NORMAL    Return;

QUEUE     RocksToMove Rock;
QUEUE     MovedRocks  Rock;

LINK      TR1     TrucksWait Load;
LINK      LD1     LoadersWait Load;
LINK      LD2     Load  LoadersWait;
LINK      TR2     Load  Haul  Truck;
LINK      TR3     Haul  Dump  Truck;
LINK      TR4     Dump  Return Truck;
LINK      TR5     Return TrucksWait;

LINK      RK1     RocksToMove Load;
LINK      RK2     Load  Haul  Rock;
LINK      RK3     Haul  Dump  Rock;
LINK      RK4     Dump  MovedRocks;

/* Activity Durations
DURATION  Load  'Normal[40,5]';
DURATION  Haul  'Normal[35,5]';
DURATION  Dump  'Normal[3,0.5]';
DURATION  Return 'Normal[35,5]';

/* Queue Initialization
INIT      TrucksWait NumTrucks;
INIT      LoadersWait NumLoaders;

INIT      RocksToMove NumberOfRocks;

/* Terminating Condition
SIMULATE;

/* Presenting Results
DISPLAY   "Total Working Time : " SimTime " Seconds";
REPORT;

```

Figure 6.12: STROBOSCOPE Simulation Input File

In this Figure, statements used to describe activity durations inside the simulation script are highlighted. In addition, necessary statements were added to assess and report the total completion time of the project. The output of this simulation model is shown in Figure 6.13. In this Figure, average waiting time of resources inside their corresponding queues is highlighted. Since the simulation parameter (i.e. activity durations) were approximated in the first place, the resulting waiting times may or may not represent the actual idle time of resources during the course of the real world project.

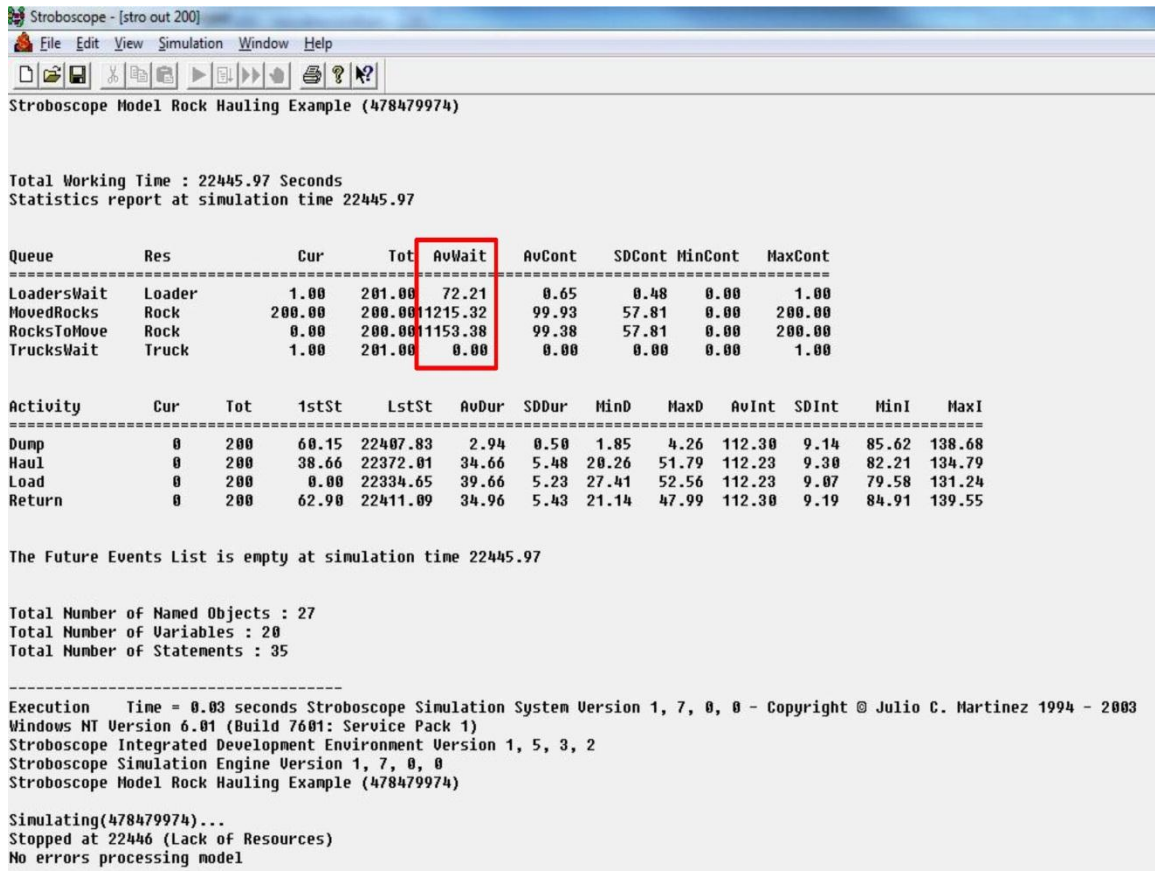


Figure 6.13: STROBOSCOPE Simulation Output File

Hence, it was decided to incorporate real time operational data collected from the model equipment into the DES model to create a more accurate and realistic output that better serve the decision-making process. To do so, data was collected from several complete operational cycles including *Load*, *Haul*, *Dump*, and *Return* activities. The collected data was further processed to determine and display the statistical mean and the standard deviation of each activity as identified in the corresponding VI by establishing rules relating equipment motions to the beginning and end events of individual activities. Figure 6.14 shows the VI and the results obtained for activity durations.

These statistical parameters were used to replace the approximate duration values by assigning more realistic Normal distributions to individual activity durations and update the DES model. The revised STROBOSCOPE simulation script is shown in Figure 6.15 where newly calculated activity durations are highlighted. The updated simulation model was then run and results were collected as illustrated in Figure 6.16.

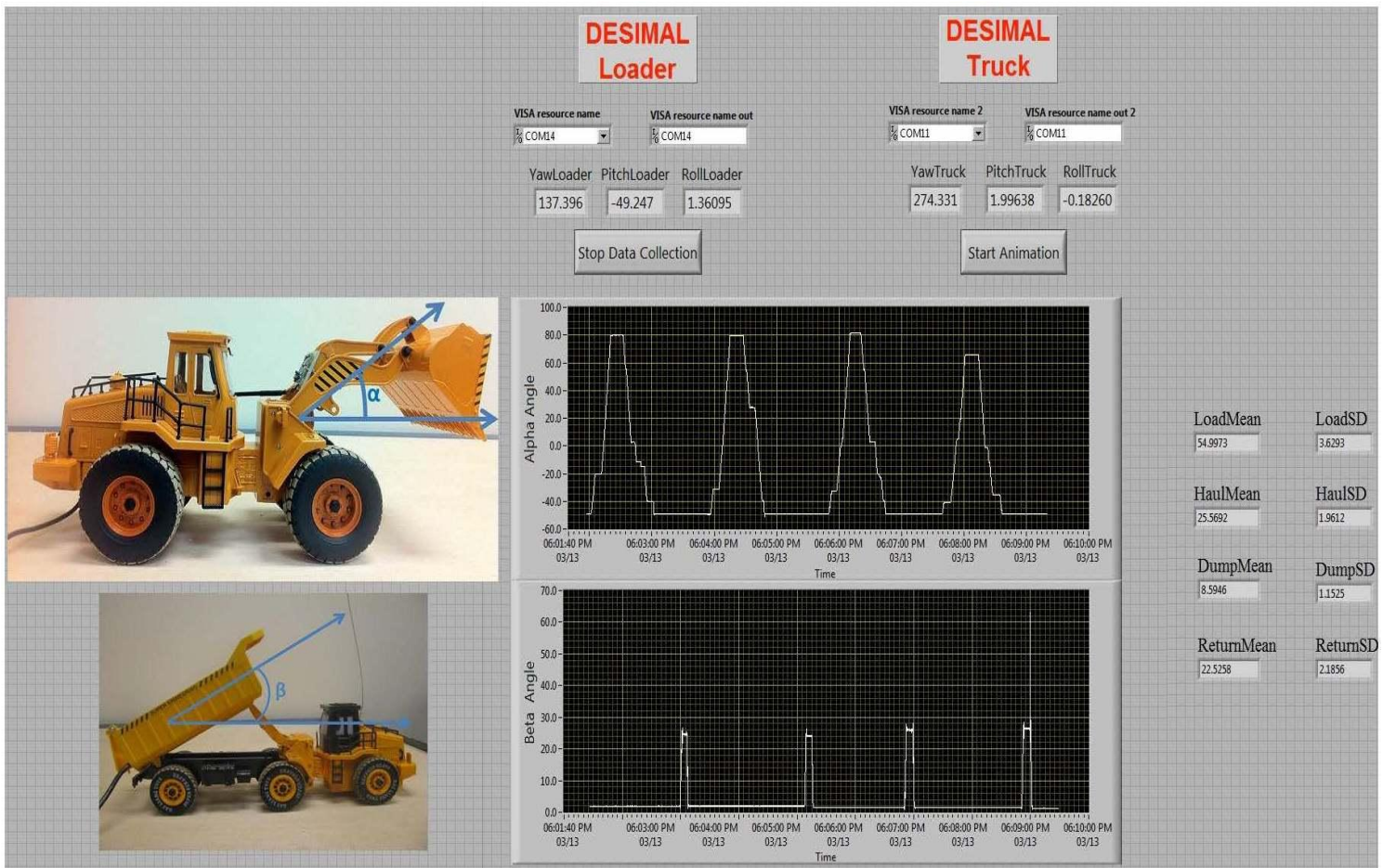


Figure 6.14: Developed VI for Data Collection and Analysis for Rock Hauling Example


```

File Edit View Simulation Window Help
[Icons]

////////////////////////////////////
/Rock Hauling Operation Simulation
////////////////////////////////////

/* General Section for Problem Parameters
VARIABLE NumLoaders 1;
VARIABLE NumTrucks 1;

VARIABLE NumberOfRocks 4;

/* Definition of Resource Types
GENTYPE Loader;
GENTYPE Truck;

GENTYPE Rock;

/* Definition of Network Elements
QUEUE LoadersWait Loader;
COMBI Load;
QUEUE TrucksWait Truck;
NORMAL Haul;
NORMAL Dump;
NORMAL Return;

QUEUE RocksToMove Rock;
QUEUE MovedRocks Rock;

LINK TR1 TrucksWait Load;
LINK LD1 LoadersWait Load;
LINK LD2 Load LoadersWait;
LINK TR2 Load Haul Truck;
LINK TR3 Haul Dump Truck;
LINK TR4 Dump Return Truck;
LINK TR5 Return TrucksWait;

LINK RK1 RocksToMove Load;
LINK RK2 Load Haul Rock;
LINK RK3 Haul Dump Rock;
LINK RK4 Dump MovedRocks;

/* Activity Durations
DURATION Load 'Normal[55,3.63]';
DURATION Haul 'Normal[25.57,1.96]';
DURATION Dump 'Normal[8.6,1.15]';
DURATION Return 'Normal[22.52,2.18]';

/* Queue Initialization
INIT TrucksWait NumTrucks;
INIT LoadersWait NumLoaders;

INIT RocksToMove NumberOfRocks;

/* Terminating Condition
SIMULATE;

/* Presenting Results
DISPLAY "Total Working Time : " SimTime " Seconds";

REPORT -

```

Updated Activity Durations



```

/* Activity Durations
DURATION Load 'Normal[55,3.63]';
DURATION Haul 'Normal[25.57,1.96]';
DURATION Dump 'Normal[8.6,1.15]';
DURATION Return 'Normal[22.52,2.18]';

```

Figure 6.15: STROBOSCOPE Simulation Input File Containing Updated Activity Durations

```

Stroboscope - [stro out 4]
File Edit View Simulation Window Help
Stroboscope Model Rock Hauling Example (372899430)

Total Working Time : 440.933 Seconds
Statistics report at simulation time 440.933

Queue      Res      Cur      Tot      AuWait    AuCont    SDCont    MinCont    MaxCont
-----
LoadersWait Loader    1.00    5.00    44.77     0.51     0.50     0.00     1.00
MovedRocks  Rock     4.00    4.00    185.71    1.68     1.17     0.00     4.00
RocksToMove Rock     0.00    4.00    167.05    1.52     1.11     0.00     4.00
TrucksWait  Truck    1.00    5.00     0.00     0.00     0.00     0.00     1.00

Activity    Cur      Tot      1stSt    LstSt    AvDur    SDDur    MinD     MaxD     AvInt    SDInt    MinI     MaxI
-----
Dump        0        4      81.50    413.88    8.59     0.77     7.61     9.41    110.80    6.75    104.98    118.20
Haul        0        4      55.91    391.82    25.31    2.59    22.06    28.38    111.97    5.76    105.32    115.53
Load        0        4       0.00    337.53    54.27    3.20    49.78    57.08    112.51    5.36    107.76    118.32
Return      0        4      89.10    422.79    22.06    2.75    18.14    24.43    111.23    5.38    106.79    117.21

The Future Events List is empty at simulation time 440.93

Total Number of Named Objects : 27
Total Number of Variables : 20
Total Number of Statements : 35

-----
Execution Time = 0.03 seconds Stroboscope Simulation System Version 1, 7, 0, 0 - Copyright © Julio C. Martinez 1994 - 2003
Windows NT Version 6.01 (Build 7601: Service Pack 1)
Stroboscope Integrated Development Environment Version 1, 5, 3, 2
Stroboscope Simulation Engine Version 1, 7, 0, 0
Stroboscope Model Rock Hauling Example (372899430)
Simulating(372899430)...
Stopped at 440.933 (Lack of Resources)
No errors processing model

```

Figure 6.16: STROBOSCOPE Simulation Output File Based on the Updated Durations

Comparing the output of the revised simulation model (Figure 6.16) with that of the original model (Figure 6.13), it is clearly seen that incorporating field data into the simulation modelling has resulted the average waiting time of the loader to significantly decrease from 72.21 seconds to 44.77 seconds. Also, the overall project completion time is noticeably improved as a result of using real equipment data to update activity durations. These improvements can potentially affect the outcome of the planning of projects tasks scheduled for the immediate future tasks as far as resource arrangements

and combinations are concerned. Table 6.2 summarizes the results of this comparative validation example.

Table 6.2:
Comparison between Estimated Durations and Actual Durations Based on Real Time data

Simulation Element	Approximated Duration (sec.)	Data-Driven Duration (sec.)
Load	N[40,5]	N[55,3.63]
Haul	N[35,5]	N[25.57,1.96]
Dump	N[3,0.5]	N[8.6,1.15]
Return	N[35,5]	N[22.52, 2.18]
Loader's Avg. Idle	72.21	44.77

CHAPTER 7: CONCLUSIONS AND FUTURE WORK

7.1 Conclusions

Operations level planning and control is one of the most critical components of managing ongoing activities in a construction site. Proper resource planning and control can guarantee that the best possible arrangement of resources are deployed which will in turn, result in substantial savings in project completion time and cost. To this end, simulation modeling as a powerful tool for analyzing complex construction operations has gained significant credibility during the past several years. Commonly, many simulation paradigms use static or historical data to create computer interpretable representations of real engineering systems. The suitability of this approach for modeling construction operations, however, has always been a challenge since most construction projects are unique in nature, and every project is different in design, specifications, methods, and standards. Therefore, there is a significant need for a methodology that not only does enable the modeling of main entities and logical relationships in a real system, but also allows that real time changes be incorporated into the simulation model.

The major requirement of a modeling platform capable of precisely representing the real world construction system is a data collection scheme capable of providing the simulation model with the latest information about the status of underlying processes and project entities. Given the dynamic nature and complexity of many construction processes, manually gathering the information necessary to create the corresponding simulation

model is a tedious if not an impossible task and thus, it is necessary to employ an automated system for collecting required data and convert them to a format understandable by and useful for the simulation model.

This Thesis document reported on a study conducted to investigate the requirements and applicability of a data-driven decision support system based on the relatively new simulation paradigm of dynamic data-driven application system (DDDAS). This paradigm was integrated with the traditional discrete-event simulation (DES) modeling to create a single decision-making framework for short-term scheduling and system control. The framework is capable of automatically collecting real time operational data from construction equipment and subsequently sorting, analyzing, and using them to create real time 3D animations of the concurrent construction processes, and also updating the simulation model describing the real operations based on the latest trends in the data stream collected from the construction jobsite.

The developed methodology was validated inside a .NET object-oriented environment along with a graphical programming and data collection platform, LabVIEW. To validate the functionality and robustness of the developed algorithms, 3D orientation trackers were used to collect motion data from moving parts of model construction equipment, and the collected data was analyzed and transformed into a format meaningful for the decision-making process. All preliminary experiments were performed in an indoor laboratory setting at the University of Central Florida.

The following summarizes the main milestones of this research that have been successfully achieved:

- A data collection platform was developed in LabVIEW for collecting angular data from 3D orientation trackers that transmitted data over an RS-232 serial port interface.
- Data classification and analysis algorithms were developed in LabVIEW for real time analysis of raw data and to convert them to proper format for use as input by the simulation model and visualization system.
- Dynamic concurrent animations of ongoing activities in real system were created inside the .NET environment using the OpenSceneGraph visualization toolkit.
- Necessary communication interfaces were created to facilitate data interoperability between the data collection and analysis module, the object-oriented programming platform used for visualization, and the discrete event simulation used to model ongoing construction activities.
- Laboratory-scale validation experiments were successfully conducted and results were documented to demonstrate the applicability and reliability of the developed data-driven decision support framework.

7.2 Future Work

The presented research is part of a much larger ongoing project which aims to facilitate the integration of real time operational data into the construction decision-making process. The next step in developing the current system will contain communication methods to capture Real Time Kinematics (RTK) GPS data for location tracking of construction equipment and also the deployment of more efficient orientation trackers that can adequately handle specific conditions of the jobsite in terms of communication range, accuracy, and ambient noise. In addition to spatio-temporal data (i.e. position, orientation), payload information is another potential source of data that can be collected and used to determine the state of equipment involved in operations such as earthmoving or steel erection where material is transported from one location to another. There are also other types of data that are not necessarily related to construction resources but can potentially affect the progress of field activities. Examples include weather-related (e.g. temperature, humidity) data and soil and topography data. To this end, future work in this research will include the design and implementation of robust algorithms to collect, process, and fuse such multiple-source heterogeneous data [75, 76].

Also, there is a need to examine the developed pre-processed visualization module in the presence of a post-processed (simulation-based) 3D visualization platform to highlight the advantages and identify potential shortcomings of the current framework. In addition, work needs to be done to improve the mathematical efficiency and statistical accuracy of the framework in order to more effectively handle, fuse, and process large volumes of

raw incoming data especially when multiple heterogeneous data collection devices are used.

Automating and optimizing equipment operations is another potential area for future work in this research. To achieve this, machine learning methods will be investigated to develop a self-learning system capable of observing activities that involve resource (i.e. equipment, material, personnel) interactions, extracting information by identifying data trends and cyclic motions, and subsequently generating knowledge-based action plans to streamline process flows on the jobsite.

APPENDIX A: C++ ALGORITHMS FLOWCHARTS

As stated in Chapter 5, the computing platform developed in this research takes advantage of a .NET object-oriented design as well as a graphical user interface (GUI) developed in LabVIEW. Platform interoperability features that facilitate the communication of information between the .NET environment and the LabVIEW interface are provided using the ActiveX automation interface.

In this Appendix, a detailed description of the .NET functionalities is presented by using flowcharts that describe how different programming modules communicate and what type of data is transferred between these modules. The illustrated flowcharts are only intended to supplement the discussion of the topics introduced in previous Chapters and to help interested readers gain a better understanding of the data flow in the developed platform.

There are four major C++ functions used inside the .NET environment. These functions include,

```
CreateAnimationPath()
```

This function plays the most critical role since it facilitates communication between C++ and LabVIEW to capture and store angular data as vectors and create and return an animation path.

```
CreateMovingModel()
```

This function imports 3D CAD files of articulated parts of model construction equipment and defines the parent-child hierarchical relationships between different nodes

constructing each object in the scene. This function creates and returns an intermediate or group node including all objects called `model`.

`CreateModel()`

This function defines the origin of the coordination system used to create the visualization scene. It also attaches the group node `model` to the `root` node and returns `root` as the highest point of the hierarchy.

`Main()`

This function initializes a LabVIEW interface, tilts the scene to arrive at the desired viewpoint, set the scene to render, and finally runs the animation.

Figures a.1 through A.4 illustrates detailed flowcharts of the above function.

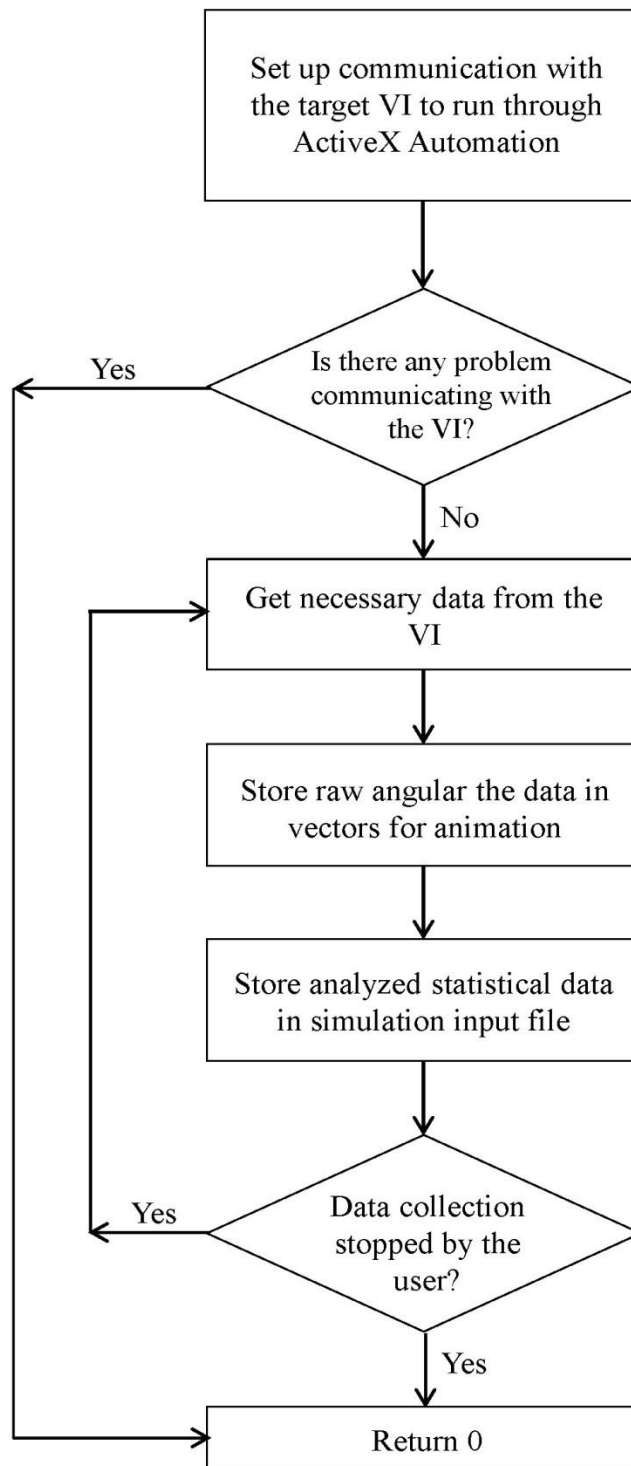


Figure A. 1: CreateAnimationPath () Function Flowchart

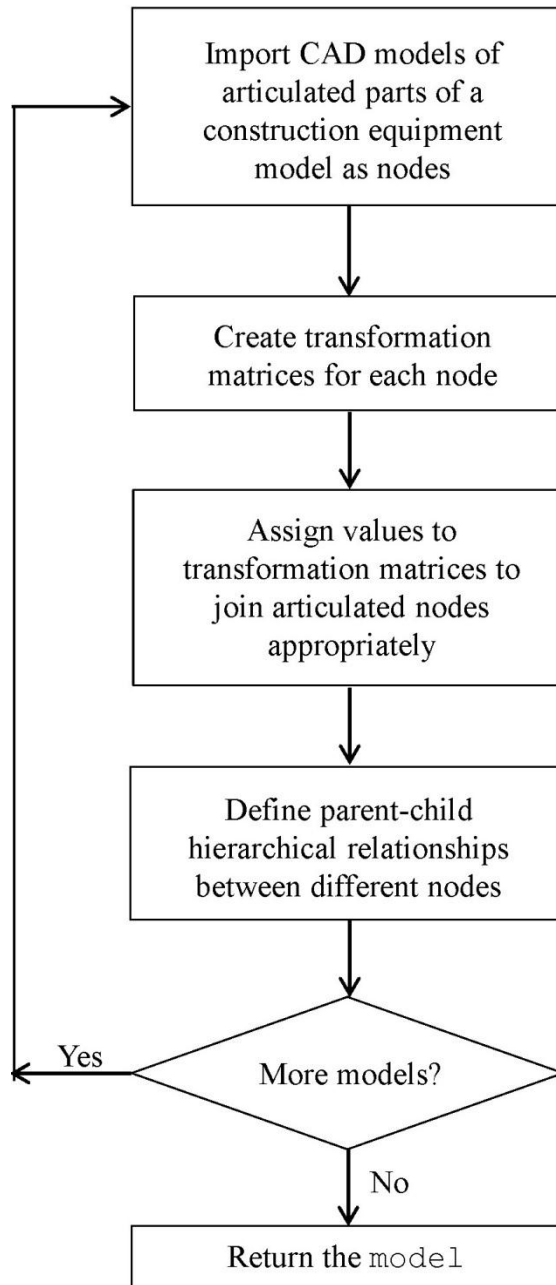


Figure A. 2: CreateMovingModel() Function Flowchart

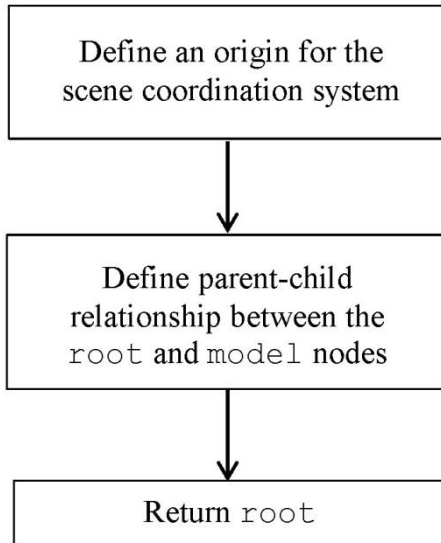


Figure A. 3: CreateModel() Function Flowchart

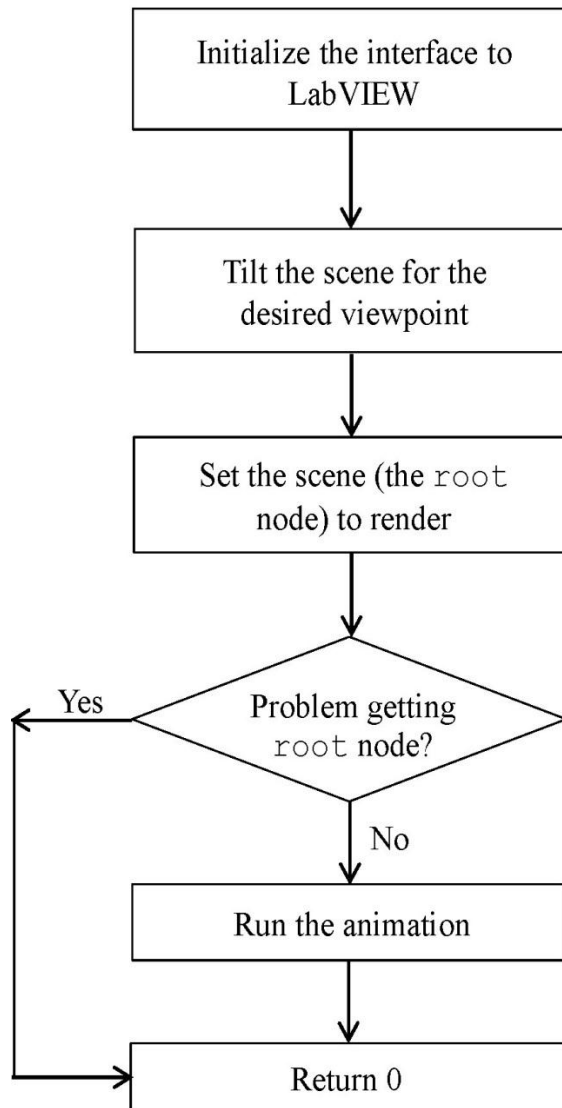


Figure A. 4: Main() Function Flowchart

APPENDIX B: LabVIEW GRAPHICAL PROGRAMMING AND ALGORITHMS

LabVIEW¹ (i.e. Laboratory Virtual Instrument Engineering Workbench) is a product of National Instrument (NI) and is a platform for designing engineering and scientific measurement and control systems. LabVIEW uses graphical programming (G) as a data flow language in which nodes, as operations or functions, operate on data received through “wires”. This approach provides an efficient way of handling and processing data especially when compared to most text-based programming languages which operate based on a sequential line by line manner. LabVIEW has built-in tools designed specifically for data collection, analysis, and presentation.

LabVIEW programs are called virtual instruments (VIs). Each VI has two windows: the user interface which is called the Front Panel, and the graphical code called the Block Diagram. The Front Panel provides users with interactive controls such as buttons, gages, graphs, and tables as well as tools to save data files or automatically generating reports. The Block Diagram, on the other hand, consists of icons and nodes that are connected together via wires. Figure B.1 shows a customized Front Panel and the corresponding Block Diagram.

¹ LabVIEW is a registered trademark of National Instruments (NI).

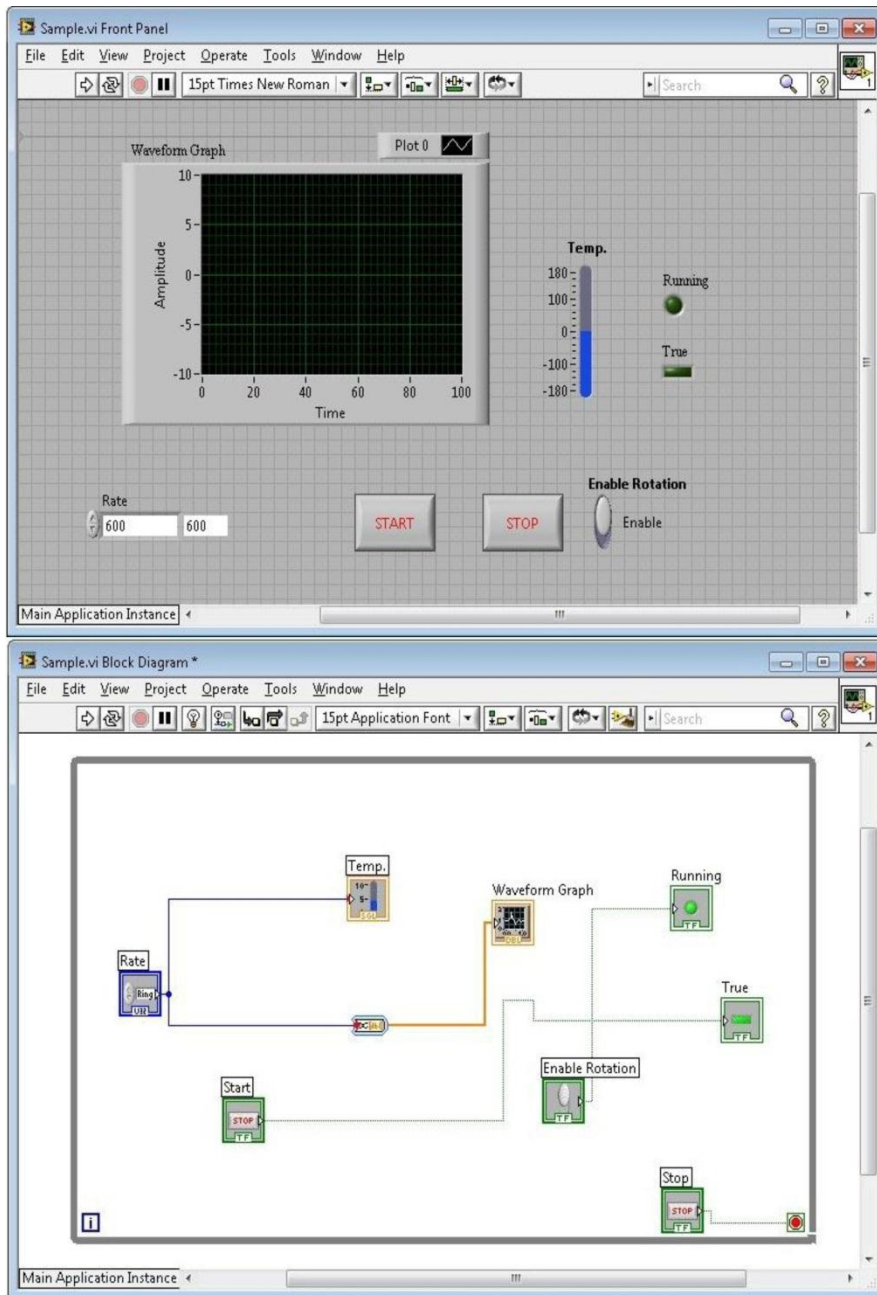


Figure B. 1: A Customized VI - The Upper Window is the Front Panel and the Bottom Window is the Block Diagram

Algorithms for data collection and data analysis components of the presented framework were developed using LabVIEW. To this end, a Plug and Play (P&P) instrument driver initially developed by the NI was modified, customized and appended to meet the required functionalities needed in this research. An instrument driver is a library of VIs that controls a programmable instrument. NI instrument drivers are provided as open-source well-documented libraries and can be customized by the end user to perform specific tasks. In this research, an instrument driver was used for communication with orientation trackers employed for data collection via RS-232 protocol for serial communication.

As stated in Chapter 5, Virtual Instrumentation Software Architecture (VISA), a standard I/O language and an application programming interface (API) for sensor programming was used in this research. VISA basically facilitates port communication by providing needed operations such as opening, writing to, reading from, and closing a port. Figures B.2 through B.5 show special nodes in LabVIEW for each of the indicated tasks.

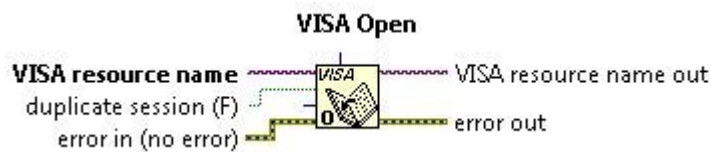


Figure B. 2: *VISA Open* Opens the Specified Port by the *VISA Resource Name*



Figure B. 3: *VISA Write* Writes Data to the Specified Port by the *VISA Resource Name*

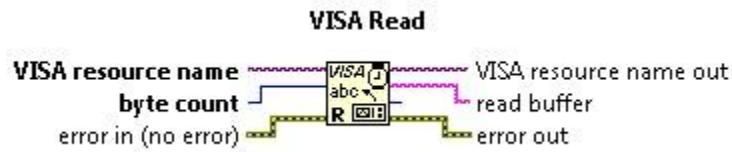


Figure B. 4: *VISA Read* Reads Data from the Specified Port by the *VISA Resource Name*

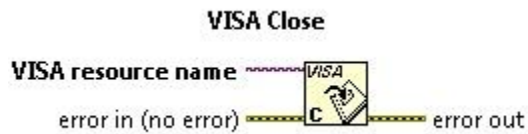


Figure B. 5: *VISA Close* Closes the Specified Port by the *VISA Resource Name*

Also, Figure B.6 illustrates a simplified layout of how these VISA functions are connected to each other via wires in the developed data collection system.

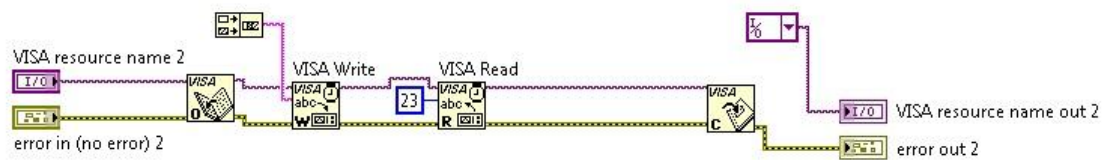


Figure B. 6: A Series of VISA Functions and Their Connections as Used in this Research

Data classification and analysis algorithms were also designed in the same VI. Figures B.7 through B.12 show built-in functions that were used to configure a relatively sophisticated graphical code capable of real time extraction of activity durations from angular raw data. Necessary information has been provided in each Figure caption.

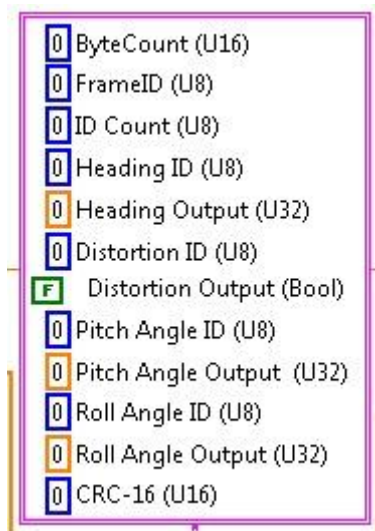


Figure B. 7: Requested Data Classified from Cluster of Real Time Orientation Data

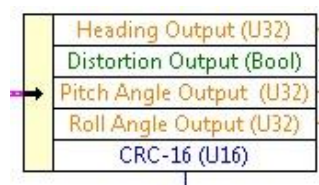


Figure B. 8: *Unbundled By Name* Function that Returns Cluster Elements Whose Names Have Been Specified

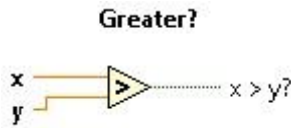


Figure B. 9: *Greater?* Function Returns True If x Is Greater than Y - This Function Was Used to Detect Data Exceeding a Specified Threshold

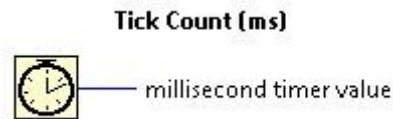


Figure B. 10: *Tick Count* Function That Returns the Value of a Timer – This Function Was Used to Measure the Duration of Each Activity

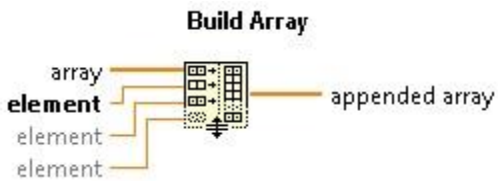


Figure B. 11: *Build Array* Function to Store Activity Durations in a Numerical Array



Figure B. 12: *Statistics* Tool Returns the Specified Statistical Characteristics of Input Arrays

In addition to the VI elements described above, there are a number of other functions, tools, and controls that were also developed and used for constructing the VI. For example *Case Structures*, *While Loops*, or *PointByPoint Analysis* functions that each one of which perform specific tasks under different conditions. The descriptions and technical details of these elements are, however, beyond the scope of this document. Interested readers are encouraged to contact the author or the DESIMAL research group at the University of Central Florida for more information.

REFERENCES

- [1] D.W. Halpin, L.S. Riggs, Planning and analysis of construction operations, Wiley, New York, 1992.
- [2] R.F. Cox, R.R.A. Issa, D. Aherns, Management's perception of key performance indicators for construction, *J. Constr. Eng. Manage.* 129 (2) (2003) 142-151.
- [3] W. Ibbs, Quantitative impacts of project change: size issues, *J. Constr. Eng. Manage.* 123 (3) (1997) 08-11.
- [4] NRC, National Research Council (NRC), Federal Facilities Council Technical Report No. 149 Reducing Construction Costs: Uses of Best Dispute Resolution Practices by Project Owners, N.A. Press, Washington, DC, 2007.
- [5] D.W. Halpin, CYCLONE - A Method for Modeling Job Site Processes, *ASCE Journal of Construction Division* 103 (3) (1997) 489-499.
- [6] B.C. Paulson, W.T. Chan, C.C. Koo, Construction operation simulation by microcomputer, *J. Constr. Eng. Manage.* 113 (302) (1987) 302-314.
- [7] Y.L. Liu, P.G. Ioannou, Graphical Object-Oriented Discrete-Event Simulation System, in: *Proceeding of the 1992 Winter Simulation Conference (WSC)*, Association for Computing Machinery (ACM), Arlington, VA, 1992, pp. 1285-1291.
- [8] J. Martinez, P.G. Ioannou, General purpose simulation with stroboscope, in: *Proceeding of the 1994 Winter Simulation Conference (WSC)*, Association for Computing Machinery (ACM), Lake Buena Vista, FL, 1994, pp. 1159-1166.
- [9] J. Shi, Activity-Based construction ABC modeling and simulation method, *J. Constr. Eng. Manage.* 12 (354) (1999) 354-360.
- [10] V.R. Kamat, J.C. Martinez, General-purpose 3D animation with VITASCOPE, in: *Proceeding of the 2004 Winter Simulation Conference*, Washington, DC, 2004.

- [11] A.H. Behzadan, V.R. Kamat, Automated generation of operations level construction animations in outdoor augmented reality, *J. Comput. Civil Eng.* 23 (6) (2009) 405-417.
- [12] L. Song, A framework for real-time simulation of heavy construction operations, in: *Proceeding of 2008 Winter Simulation Conference (WSC)*, Miami, FL, 2008, pp. 2387-2395.
- [13] G. Adkins, U.W. Pooch, Computer Simulation: a Tutorial, *Computer* 10 (4) (1977) 12-17.
- [14] J. Banks, *Handbook of simulation : principles, methodology, advances, applications, and practice*, Wiley, New York, 1998.
- [15] W.S. Jang, M.J. Skibniewski, Wireless sensor technologies for automated tracking and monitoring of construction materials utilizing Zigbee networks, in: *Proceeding of ASCE Construction Research Congress*, Grand Bahamas Island, 2007.
- [16] A.H. Behzadan, Z. Aziz, C.J. Anumba, V.R. Kamat, Ubiquitous location tracking for context-specific information delivery on construction sites, *Autom. Constr.* 17 (6) (2008) 737-748.
- [17] D. Grau, C. Caldas, P. Goodrum, C. Haas, Assessing the impact of materials tracking technologies on construction craft productivity, *Autom. Constr.* 18 (7) (2009) 903-911.
- [18] J. Messner, M. Horman, D. Riley, An Interactive Visualization Environment for Construction Engineering Education, in: *Proceeding of Construction Research Congress 2005: Broadening Perspectives* San Diego, CA, 2005, pp. 1279-1288.
- [19] S. Han, F. Peña-Mora, M. Golparvar-Fard, S.Roh, Application of a visualization technique for safety management, in: *Proceeding of the 2009 ASCE International Workshop on Computing in Civil Engineering*, Austin, Texas, 2009, pp. 543-551.
- [20] A.H. Behzadan, V.R. Kamat, Interactive augmented reality visualization for improved damage prevention and maintenance of underground infrastructure, in:

Proceeding of the 2009 Construction Research Congress, Seattle, WA, 2009, pp. 1214-1222.

[21] R. Akhavian, A.H. Behzadan, Dynamic Data Driven Simulation of Ongoing Construction Processes, in: Proceeding of 3rd International and 9th Construction Specialty Conference, Ottawa, Canada, 2011, pp. CN-154 (151-110).

[22] I. Davidson, M. Skibniewski, Simulation of automated data collection in buildings, *J. Comput. Civil Eng.* 9 (1) (1995) 9-20.

[23] B. Mccullouch, Automating field data collection in construction organizations, in: Proceeding of the 1997 ASCE Construction Congress Minneapolis, Minnesota, 1997, pp. 957-963.

[24] G.S. Cheok, R.R. Lipman, C. Witzgall, J. Bernal, W.C. Stone, NIST Construction Automation Program Rep. No: 4, Non-Intrusive Scanning Technology for Construction Status Determination, Gaithersburg, Md, 2000.

[25] R. Navon, Automated project performance control of construction projects, *Autom. Constr.* 14 (4) (2005) 467-476.

[26] S. Taneja, A. Akcamete, B. Akinci, J.H. Garrett, E.W. East, L. Soibelman, Analysis of Three Indoor Localization Technologies to Support Facility Management Field Activities, in: Proceeding of International Conference on Computing in Civil and Building Engineering, Nottingham, UK, 2010.

[27] K.S. Saidi, A.M. Lytle, W.C. Stone, Report of the NIST Workshop on Data Exchange Standards at the Construction Job Site, in: Proceeding of 20th International Symposium on Automation and Robotics in Construction (ISARC), Eindhoven, The Netherlands, 2003, pp. 617-622.

[28] J. Song, C.T. Haas, C. Caldas, E. Ergen, B. Akinci, Automating the task of tracking the delivery and receipt of fabricated pipe spools in industrial projects, *Autom. Constr.* 15 (2) (2006) 166-177.

[29] J. Song, C.T. Haas, C. Caldas, Tracking the location of materials on construction job sites, *J. Constr. Eng. Manage.* 132 (9) (2006) 911-918.

- [30] J. Teizer, M. Venugopal, A. Walia, Ultrawideband for Automated Real-Time Three-Dimensional Location Sensing for Workforce, Equipment, and Material Positioning and Tracking, *Journal of Transportation Research Record* 2081 (6) (2008) 56-64.
- [31] D.T. Grau, C.H. Caldas, Methodology for automating the identification and localization of construction components on industrial projects, *J. Comput. Civil Eng.* 23 (1) (2009) 3-13.
- [32] H.M. Khoury, V.R. Kamat, Indoor User Localization for Context-Aware Information Retrieval in Construction Projects, *Autom. Constr.* 18 (4) (2009) 444-457.
- [33] E.J. Jaselskis, M.R. Anderson, C.T. Jahren, Y. Rodrigues, S. Njos, Radio Frequency Identification Applications in Construction Industry, *J. Constr. Eng. Manage.* 121 (2) (1995) 189-196.
- [34] G.M. Jog, I.K. Brilakis, D.C. Angelides, Testing in harsh conditions: Tracking resources on construction sites with machine vision, *Autom. Constr.* 20 (4) (2011) 328-337.
- [35] A.A. Oloufa, M. Ikeda, H. Oda, Situational awareness of construction equipment using GPS, wireless and web technologies, *Autom. Constr.* 12 (6) (2003) 737-748.
- [36] C.H. Caldas, D.T. Grau, C.T. Haas, Using global positioning system to improve materials-locating processes on industrial projects, *J. Constr. Eng. Manage.* 132 (7) (2006) 741-749.
- [37] U.S. Coast Guard Navigation Center, <http://www.navcen.uscg.gov/?pageName=GPS>, (accessed Sep. 27, 2011).
- [38] E. Ergen, B. Akinci, B. East, J. Kirby, Tracking Components and Maintenance History within a Facility Utilizing Radio Frequency Identification Technology, *J. Comput. Civil Eng.* 21 (1) (2007) 11-20.
- [39] J. Teizer, D. Lao, M. Sofer, Rapid Automated Monitoring of Construction Site Activities using Ultra-Wideband, in: *Proceeding of 24th International Symposium on Automation and Robotics in Construction*, Cochin, Kerala, India, 2007, pp. 23-28.

- [40] S. Razavi, O. Moselhi, Indoor Construction Location Sensing using Low Cost Passive RFID Tags, in: Proceeding of 3rd International and 9th Construction Specialty Conference, Ottawa, Canada, 2011, pp. CN1-10.
- [41] A.H. Behzadan, V. Kamat, Visualization of construction graphics in outdoor augmented reality, in: Proceeding of the 2005 Winter Simulation Conference, 2005.
- [42] A.H. Behzadan, V. Kamat, General-purpose modular hardware and software framework for mobile outdoor augmented reality applications in engineering, *Adv Eng Inform* 22 (1) (2008) 90-105.
- [43] I. Brilakis, M.W. Park, G. Jog, Automated vision tracking of project related entities, *Adv Eng Inform* 25 (4) (2011) 713-724.
- [44] J.C. Martinez, P.G. Ioannou, General-purpose systems for effective construction simulation, *J. Constr. Eng. Manage.* 125 (4) (1999) 265-276.
- [45] D.W. Halpin, An Investigation of the Use of Simulation Networks for Modeling Construction Operations, Ph.D Dissertation, University of Illinois at Urbana-Champaign, Illinois, 1973.
- [46] A.A. Oloufa, Modeling operational activities in object-oriented simulation, *J. Comput. Civil Eng.* 7 (1) (1993) 94-106.
- [47] J.C. Martinez, *Stroboscope : state and resource based simulation of construction processes*, 1996.
- [48] M. Lu, Simplified discrete-event simulation approach for construction simulation, *J. Constr. Eng. Manage.* 129 (5) (2003) 537-546.
- [49] M.P. Hunter, R.M. Fujimoto, W. Suh, An investigation of real-time dynamic data driven transportation simulation, in: Proceeding of the 2006 Winter Simulation Conference, Monterey, CA, 2006, pp. 1414-1421.

- [50] S. Tavakoli, A. Mousavi, A. Komashie, A generic framework for real-time discrete event simulation (DES) modelling, in: Proceeding of 2008 Winter Simulation Conference (WSC), Miami, FL, 2008, pp. 1931-1938.
- [51] X. Guo, S.Y. Huang, W.J. Hsu, M.Y.H. Low, Yard crane dispatching based on real time data driven Simulation for container Terminals, in: Proceeding of the 2008 Winter Simulation Conference (WSC), Miami, FL, 2008, pp. 2648-2655.
- [52] T.H. Chung, Y. Mohamed, S.M. AbouRizk, Bayesian updating application into simulation in the North Edmonton Sanitary Trunk tunnel project, J. Constr. Eng. Manage. 132 (8) (2006) 882-894.
- [53] S. AbouRizk, A. Hizaji, D.W. halpin, Effect of input modeling on simulation output, in: Proceeding of Computing in Civil Engineering (New York), 1989.
- [54] V.R. Kamat, Visualizing simulated construction operations in 3D, J. Comput. Civil Eng. 15 (4) (2001) 329-327.
- [55] Y. Kui, The ASDMCon project: the challenge of detecting defects on construction sites, in: Proceeding of the Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'06), 2006.
- [56] J.J. Shi, Iconic animation of construction simulation, in: Proceeding of the 1999 Winter Conference on Simulation, 1999, pp. 992-997.
- [57] J.O. Henriksen, General-purpose concurrent and post-processed animation with Proof, in: Proceeding of the 1999 Winter Conference on Simulation, 1999, pp. 176-181.
- [58] F. Darema, Dynamic data driven applications systems: new capabilities for application simulations and measurements, in: Proceeding of Computational Science -- ICCS 2005, Atlanta, USA, 2005, pp. 610-615.
- [59] National Science Foundation (NSF), Dynamic Data Driven Application Systems (DDDAS), 2000 NSF Sponsored Workshop Report, http://www.nsf.gov/cise/cns/dddas/dd_das_work_shop_rprt.pdf, (accessed Sep. 29, 2011).

- [60] J. Mandel, Towards a dynamic data driven application system for wildfire simulation, in: Proceeding of Computational Science - ICCS 2005. 5th International Conference. , 2005.
- [61] C.C. Douglas, Y. Efendiev, A dynamic data-driven application simulation framework for contaminant transport problems, Journal of Computers & mathematics with applications 51 (11) (2006) 1633-1646.
- [62] M. Gaynor, A dynamic, data-driven, decision support system for emergency medical services, in: Proceeding of Computational Science - ICCS 2005. 5th International Conference., 2005.
- [63] R. Akhavian, A.H. Behzadan, An Integrated Data Collection and Analysis Framework for Remote Monitoring and Planning of Construction Operations, Adv Eng Inform (26) (2012) (in press).
- [64] D.R. Nadeau, Visualizing stars and emission nebulas, Computer graphics forum 20 (1) (2001) 27-33.
- [65] P. Martz, OpenSceneGraph Quick Start Guide, Pmartz, 2007.
- [66] V.R. Kamat, J.C. Martinez, Scene graph and frame update algorithms for smooth and scalable 3D visualization of simulated construction operations, Journal of Computer-aided civil and infrastructure engineering 17 (4) (2002) 228-245.
- [67] Woolford, D. (2003), Understanding and Using Scene Graphs, http://www.itee.uq.edu.au/~comp4201/scene_graphs_dsaw.pdf, (accessed Mar. 13, 2012).
- [68] OpenSceneGraph (OSG) Website, <http://www.openscenegraph.org/>, (accessed Mar. 13, 2012).
- [69] J.G. Proakis, M. Salehi, Digital communications, 5th ed., McGraw-Hill, Boston, 2008.
- [70] PNI, User Manual, CompassPoint Prime, 3-Axis Electronic Compass Module, in: PNI (Ed.), PNI Sensor Corporation, Santa Barbara, CA, 2011.

[71] National Instruments (NI) Website, <http://www.ni.com/>, (accessed Mar. 13, 2012).

[72] P. Ioannou, Animation of complex construction simulation models, in: Proceeding of Computing in Civil Engineering (New York), 1996.

[73] C. Microsoft, Automation programmer's reference : using ActiveX technology to create programmable applications, Microsoft® Press, Redmond, Wash, 1997.

[74] R. Akhavian, A.H. Behzadan, Dynamic Simulation of Construction Activities Using Real Time Field Data Collection, in: Proceeding of 18th Workshop of Intelligent Computing in Engineering and Architecture (EG-ICE), Enschede, Netherlands, 2011, pp. 1-10.

[75] A. R. Pradhan, B. Akinci, An Approach for Fusing Data from Multiple Sources to Support Construction Productivity Monitoring, in: Proceeding of International Conference on Computing in Civil and Building Engineering, Nottingham, UK, 2010, pp.

[76] J. Dong, D. Zhuang, Y. Huang, J. Fu, Advances in multi-sensor data fusion: algorithms and applications, Sensors 9 (10) (2009) 7771-7784.