Enhanced Path Planning Method for Improving Safety and Productivity of Excavation Operations

Seied Mohammad Langari

A Thesis
in
The Department
of
Building, Civil and Environmental Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of Master of Applied Science (Building Engineering) at
Concordia University
Montreal, Quebec, Canada

December 2015

CONCORDIA UNIVERSITY
SCHOOL OF GRADUATE STUDIES

This is to certify that the thesis prepared

By: Seied Mohammad Langari

Entitled: Enhanced Path Planning Method for Improving Safety and Productivity of Excavation Operations

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science (Building Engineering)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

Dr. Zhenhua Zhu — Chair

Dr. Chun-Yi Su — External Examiner

__________ External to Program

Dr. Tarek Zayed — Examiner

Dr. Zhenhua Zhu — Examiner

Dr. Amin Hammad — Thesis Supervisor

Approved by

____
Chair of Department or Graduate Program Director

__________ __________
Dean of Faculty

ABSTRACT

Enhanced Path Planning Method for Improving Safety and Productivity of Excavation Operations

Seied Mohammad Langari

Improving safety and productivity of earthwork operations is of paramount importance, especially in congested sites where collisions are more probable. Real-time Location Systems and Automated Machine Guidance and Control technologies are expected to improve both safety and productivity of earthwork operations by providing excavator operators a higher level of support regarding the path planning of excavators based on site conditions. However, in spite of the large number of studies related to automated path planning of excavators using well established algorithms from robotics, such as Rapidly-exploring Random Trees (RRT) and Probabilistic Roadmaps (PRM), these studies do not fully consider the engineering constrains of the equipment and do not result in smooth and optimal paths that can be applied in practice. This research aims to improve the path planning of excavators in a congested site where the speed of the algorithm and the quality of the path significantly influence the overall performance of the earthwork operations. The proposed method is implemented and tested in Unity 3D game engine environment for visualization and verification purposes. The efficiency of the proposed method in generating a collision-free path, which can ensure improved productivity, is verified both quantitatively and visually. The comparative results with other recent and modified versions of the RRT algorithm show that the proposed algorithm is able to find a higher quality path in a shorter time.

ACKNOWLEDGEMENTS

I would like to use this opportunity to express my gratitude to everyone who supported me throughout the course of this Master degree. My deepest gratitude to my supervisor, Dr. Amin Hammad, for his expertise, patience and understanding. Without his methodical and moral support, I could not have made such progress in this research. He taught me how to question thoughts and express ideas. His constant enthusiasm, motivation and guidance helped me in all the time of research and writing of this thesis.

Besides, I am writing to extend my appreciation to all professors who provided me with their valuable knowledge during my study at Concordia University. My sincere thanks also goes to Mr. Homam Albahnassi for helping me in the implementation of the existing research in the first case study; Mr. Henrique Machado and Bruno Paes in the implementation of the Multi-Agent System; besides my friend Faridaddin Vahdatikhaki who involved me in his research regarding proactive and reactive methods in construction equipment safety.

I appreciate my friends and fellow lab-mates in Concordia University: Mohammad Mostafa Soltani, Mohammad Mowlana, Khaled Al-Emmari, Alhussain Taher, and Seyed Amirhossein Sharif for all the exciting discussions and all the joys we had.

Last but not least, I would like to thank my family for the support they provided me through my entire life and in particular, I would like to express my deepest appreciation to my wife, without whose love and encouragement I would not have finished this thesis.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

2D	Two-dimensional
3D	Three-dimensional
4D	Four-dimensional
A^*	A-star
ABS	Agent-Based Simulation
AI	Artificial Intelligence
AWCBC	Association of Worker's Compensation Board of Canada
CAD	Computer-Aided Design
CPM	Critical Path Method
CPU	Central Processing Unit
C-space	Configuration Space
D^*	Dynamic A^*
DES	Discrete Event Simulation
DH-notation	Denavit-Hartenberg notation
DoF	Degree of Freedom
DRRT	Dynamic Rapidly-exploring Random Trees
DEW	Dynamic Equipment Workspaces
FK	Forward Kinematics
GPS	Global Positioning System
IES	Intelligent Excavation System
IK	Inverse Kinematics
LAEW	Look-Ahead Equipment Workspaces
MAS	Multi-Agent System
NIOSH	National Institute for Occupational Safety and Health
NRTS	Near Real-Time Simulation
PDF	Probability Density Function
PERT	Program Evaluation Review Technique
PRM	Probabilistic RoadMaps
RRT	Rapidly-exploring Random Trees
RRTBiasedLimCon	RRT Biased Limited Connect

1. INTRODUCTION

1.1 General Background

Heavy equipment are extensively used in the construction industry due to their significant effect on the productivity of construction operations. However deployment of these heavy equipment have resulted in a high increase in the number of injuries and fatalities. According to Association of Worker's Compensation Board of Canada (AWCBC, 2014) construction is known as the most hazardous industry in Canada in terms of the number of fatalities. Struck by incidents (involving vehicles and equipment) are the second cause of death in the construction industry. Cranes (17 %), excavators (15), tractors (15), loaders (9%), and pavers (7%) are the most frequently involved construction equipment in fatality incidents (NIOSH, 2010).

The recent advances in computer visualization and simulation have provided a substantial platform for researchers in the construction domain to simulate and analyze construction processes at the macro and micro levels (Hammad et al., 2014a). A large amount of research has been done in the area of construction simulation in order to improve the productivity and safety of construction operations. Simulation techniques are used extensively in order to enable site practitioners and researches to investigate possible scenarios before the actual operations in order to mitigate any unforeseen risks. Path planning algorithms have the potential to play a significant role in simulation tools for planning and coordinating construction equipment. However these algorithms are designed to solve pure mathematical problems. Therefore, in order to adopt them to construction applications, they must be customized to consider the engineering constraints of construction equipment so that they can generate realistic motion of the equipment.

Many research works have focused on simulation and visualization for path planning of large construction equipment, such as tower cranes and hydraulic cranes. In these types of simulation, the safety and productivity of one or a few equipment participating in a specific task have been analyzed and tested (Kim et al., 2003; Kang and Miranda, 2006; Albahnassi and Hammad, 2011; Chang et al., 2012; Zhang and Hammad, 2012; Marzouk and Ali, 2013; Lin et al., 2014a; Lin et al., 2014b). Among the recent sample-based path planning algorithms, Rapidly-exploring Random Tree (RRT) proposed by La-Valle (1998) is one of the quickest path planners and is widely used at the micro-level simulation research. RRT is an effective path planner for construction equipment, which in most cases has a large number of Degrees of Freedom (DoFs). However, the generic RRT path planner explores the search space to find a collision-free path without any considerations of the specific properties of the construction equipment. This problem can be solved by embedding heuristic rules in the RRT path planner considering the operational and engineering constraints of the construction equipment. For example, Albahnassi and Hammad (2011) proposed a framework for near real-time path planning of cranes considering their engineering constraints. Another limitation of RRT is that it might not be fast enough for the dynamic construction projects where there are many pieces of equipment working in a congested area. Besides, RRT and other sample-based path planners are probabilistically complete but they are not optimal. Completeness is the ability of the path planner to find the path when there is one (Albahnassi and Hammad, 2011; Zhang and Hammad, 2012). The jaggedness and non-optimality of the path clearly affect the cycle-time of the equipment and will ultimately decrease its productivity. In a research for improving erection processes using robotic cranes, the initial generated path was not applicable to the crane due to the low quality of the path and the operational aspects of the equipment (Kang and Miranda, 2006). Several post-processing methods were applied to improve the quality of the path and to fit it to the crane movement.

1.2 Research Objectives

Focusing on excavators, this research has the following objectives:

(1) Improving the path planning of excavators in a congested site where the speed of the algorithm and the quality of the path significantly influence the safety and productivity of the earthwork operations;

(2) Developing a new method to adjust the motion plan to consider the hydraulic power constraints of the excavator, which will affect productivity; and

(3) Proposing a basic approach to integrate micro and macro path planning using a Multi-Agent System.

1.3 Thesis Structure

The literature review is covered in Chapter 2, in which the most recent and significant related topics including, robotics kinematics, agent-based simulation, and path planning algorithms are discussed and compared. Chapter 3 is divided into three parts. In the first part, a new simulation approach for excavator cyclic operation is proposed along with a new advanced and modified path planning algorithm, which accommodates the engineering constraints of excavators. A new method is proposed in the second part, which adjusts the path generated by the planner according to rotational speed constraints of the joints. In the third part, a simulation model is proposed for earthwork operations in which the proposed micro path planning method is integrated with macro planning. Chapter 4 provides two case studies in which the results of each part correspond to the respective part of Chapter 3. Unity game engine is used as the simulation platform. A summary, the research contributions and future work are represented in Chapter 5.

2. LITERATURE REVIEW

2.1 Introduction

This chapter reviews and analyses the previous research, methods, frameworks and algorithms with regards to construction simulation, automation of construction equipment, path/motion planning and robotic applications. Reviewing the available trends in research and industry is done to identify the research gaps and the opportunities for the application of advanced path planning methods to increase the safety and productivity of the construction site. In-depth review is done regarding the research related to earthwork operations and excavators, which are extensively used in construction operations and play a major role in the productivity and safety of construction site.

2.2 Simulation of Construction Operations

Due to the recent vast advance in computer visualization and simulation, these techniques are being used as substantial platforms for researcher in the construction domain to analyze construction processes at macro and the micro level (Hammad et al., 2014a). At the macro level, the focus is on high-level managerial concerns, such as the selection of the best combination of construction equipment and their deployment to ensure productivity and safety. In the macro-level simulation, in most of the cases, it is enough to find the path for the relocation of equipment in a 2D space using simple algorithms. The path planning of construction equipment is a new type of micro-level simulation of construction operations that has benefitted from algorithms from robotics and computer science (Kim et al., 2003; Soltani et al., 2003; Zhang and Hammad, 2012; Marzouk and Ali, 2013; Lin et al., 2014a; Lin et al., 2014b).

2.2.1 Macro-level planning and simulation

At the macro level the challenge mostly goes to high level managerial issues, such as the best combination of construction equipment to ensure high productivity (Zayed and Halpin, 2004). Discrete Event Simulation (DES) is a well-known and conventional computer simulation technique for construction processes at the macro level. In this approach, events follow the assigned probability density function, which can be extracted from on-site observations. DES is based on data collected from actual construction projects, such as observed time for each task and the construction equipment specifications (Zayed and Halpin, 2001). It has limitations accommodating spatial interaction between construction equipment and the environment; in other words, it may not be able to fully address complexity of construction operations.

Agent-based simulation (ABS) approach is a solution for this limitation. To consider safety and space-limitation issues on the job-site, Marzouk and Ali (2013) simulated piling operation using agent-based simulation. In contrast to conventional DES technique, ABS is a bottom-up approach where the combination of the behavior of each independent agent makes the final simulation outcome. Researchers utilized path planning algorithms to create this behaviors. Marzouk and Ali (2013) considered many other aspects in their model, such as traffic congestion, space, breakdown of equipment and soil behavior as an engineering constraints. To incorporate the effect of equipment traffic congestion in the job-site on productivity, they support the agents with A* path planning algorithm. However, the construction site was assumed as a 2-D discrete grid representation where agents utilized the A* algorithm to find their safe paths from initial positions to the goal positions, such as breakdown location, next piling location, etc.

A* is a path planning algorithm suitable for 2-D discrete space (Zhang and Hammad, 2012) and it is applicable for the relocations of equipment at the macro level. (Hammad et al., 2014a). However, its efficiency drop off severely by increasing problem dimensionality. Soltani et al. (2003) compared the performance of Dijkstra, A* and Genetic Algorithm (GA) in finding equipment path in a construction job-site. The desirable criteria of a path were defined as shortness of path, safety and visibility. Then they modeled the construction site by a rectangular grid, and different matrices were attached to the site layout including visibility, safety and distance data. The result showed that A* is capable of finding a near-optimal solution more efficiently than Dijkstra. A* benefits from a heuristic function that guides it toward the goal. Compared to Dijkstra, A* sacrifices optimality for a faster result. The level of this tradeoff is severely affected by the heuristic functions. But both suffer from dimensionality, which makes them suitable for problems involving objects with a low number of Degrees of Freedom (DoFs) (Soltani et al., 2003).

2.2.2 Micro-level simulation

Planning of construction operations has widely benefitted from robotics and computer science, especially at the micro level. A micro level problem considers the construction equipment as a robot with multiple DoFs. For example, a typical hydraulic excavator's upper-structure has four DoFs (Swing, Boom up/down, Stick in/out and Bucket curling) and a crawler crane has seven DoFs (Lin et al., 2014). There are many path planning algorithms to handle a robot with such high DoFs. RRT, Probabilistic Road Map (PRM) and their variations are capable of generating obstacle-free and safe path for construction equipment in job-sites (Kavraki et al., 1996). RRT was firstly proposed by La-Valle (1998) and proved to be efficient and reliable path planner in construction sites with dynamic and static obstacles (Zhang and Hammad, 2012).

2.2.2.1 Path planning in static environments

Lin et al. (2014a) utilized RRT to simulate trajectory of articulated construction equipment to help the site planner in site layout planning and replanning. The results showed that the proposed algorithm is capable of handling the problem in near real-time to help the planner in decision making and visualization. In this research, they consider three DoFs for these construction machineries, which is X,Y and θ, however in this type of car-like motion problem is constrained with control variables ($\boldsymbol{v}$, the speed and phi, the steering angle), which is called non-holonomic problems (Lin, 2014a). RRT is capable of solving non-holonomic problem, which is an advantageous of this path planning algorithm. In another research, Lin et al. (2014b) applied a modified version of bidirectional RRT to simulate motion of Crawler crane with seven DoFs. The crawler crane is able to move with load, and this motion is consist of both holonomic and non-holonomic DoFs. Due to random nature of sampling-based algorithm such as RRT, the generated path is jagged and zigzag in Configuration Space (C-space). This jagged path in C-space results in an unnatural movement of construction robot, which means longer cycle-time and lower productivity. This is considered as a limitation of these types of algorithms, because although the path is obstacle-free, it is not suitable from practical and operational point of view. Zhang and Hammad (2012) improved the quality of path for mobile cranes by considering a cost function for smoothness and time required to execute the path. The proposed algorithm is called RRT-Connect-Connect-Mod, which is a modified version of bidirectional RRT. The results showed 11.51% improvement in quality of path compared to RRT-Connect-Connect. Using a modified sampling and tree extension strategy, Lin et al. (2014b) improved the quality of path for crawler crane, which resulted in a more practical path. To generate a more practical and smoother path using RRT, some researchers applied a post-processing technique (i.e. after generating the path by the path planner) (Lin et al., 2014a; Kang and Miranda, 2006).

For tower crane robot path planning, Kang and Miranda (2006) proposed three fast algorithms named, *QuickLink*, *RandomGuess* and *QuickGuess*. The idea behind the proposed algorithm are very close to RRT, they start searching the C-space by randomly sampling collision-free node to find the obstacle-free path. Then they applied path refining method in consecutive steps for improving it to fit the operational characteristic of tower crane. PRM is another sampling based algorithm for path planning purposes. Chang et al. (2012) applied this algorithm to generate a safe path for lift operations with single and dual mobile crane. The result is satisfactory for near real-time scenario applications while maintaining safety requirements. In the mentioned research, all obstacles are assumed to be static, but this may not be able to address the dynamic challenge in a real construction job-site.

2.2.2.2 Path planning in dynamic environments

Real-time field data acquisition tools and replanning algorithms, are two main assets have to be attached to Robot construction machinery to be able to make the safe path in a dynamic construction environment (Zhang et al., 2009). Dynamic RRT (DRRT) proposed by Ferguson (2006) is a modified version of RRT that is capable of generating path in changing environment.

2.3 Robotics Kinematics

2.3.1 Forward kinematics

The concern in robotics is the location of objects in three-dimensional space (Craig, 2005). Robotics Kinematics express robots members in a mathematical form and the cause of those configuration and motions is outside the scope. Generally speaking, a robot consists of a series of members and those members are connected via joints. The relationship between two consecutive joint can be expressed in terms of relative position and orientation, as shown in Fig. 2.1. In order to compute these measures, a frame has to be considered fixed to each joint. Homogeneous transformation of consecutive frames are the concept behind the study of Robotics (Craig, 2005). As in Equation 2.1, the homogeneous transformation represents the mathematical relationship between two consecutive frames (Craig, 2005):

$$\begin{bmatrix} {}^{A}P \\ 1 \end{bmatrix} = \left(\begin{array}{c|c} {}^{A}_{B}R & {}^{A}P_{BORG} \\ \hline 0\ \ 0\ \ 0 & 1 \end{array} \right) \begin{bmatrix} {}^{B}P \\ 1 \end{bmatrix} \tag{2.1}$$

Where ${}^{A}P$ and ${}^{B}P$ are 3×1 matrices that represent the position of end effector in the frame A and B respectively. ${}^{A}P_{BORG}$ represents the origin of frame B with respect to frame A. ${}^{A}_{B}R$ is a 3×3 matrix that represents the rotation. Euler method is a well-known method to express this matrix. The relationship between two points are expressed by a 4×4 matrix, which is called Homogeneous transformation (Craig, 2005).

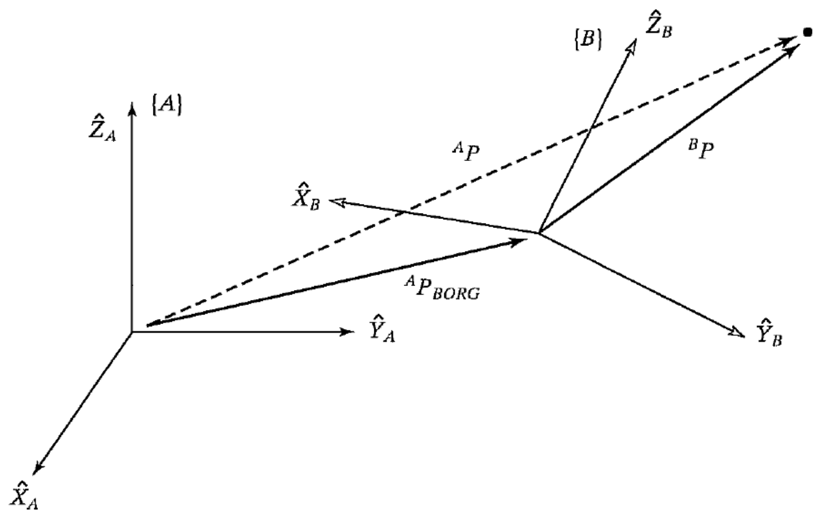


Figure 2.1 Rotation and translation between two consecutive frames (Craig, 2005)

Most of robots have revolute joints or sliding joints, called prismatic joints. There are rare case that a robot is built with a joint with multiple DoFs. Even a robot joint with n degrees of freedom can be formulated as n joints with one degree of freedom, which are connected together without any distance. Therefore, any types of robot can be modeled by homogeneous transformation of *Denavit-Hartenberg notation* (Turner et al., 1984). Fig. 2.2 represents the parameters of Denavit-Hartenberg method.

The Equation 2.2 represents the homogeneous transformation matrix, proposed by Denavit-Hartenberg:

$$ {}^{i-1}_{i}T = \begin{bmatrix} s\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -s\alpha_{i-1} d_i \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & c\alpha_{i-1} d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.2} $$

Where ${}^{i-1}_{i}T$ represents the transformation matrix between two consecutive joints of $i-1$ and i.

Most of the pieces of construction equipment have a series of simple 1-DoF joints of either revolute or prismatic. Then a construction equipment can be represented by a chain of joints and the configuration of final end effector, such as bucket of an excavator, can be modeled by a series of Denavit-Hartenberg transformation matrices. Figure 2.3 represents an example of 3 degrees of freedom robot, which only has revolute joints.

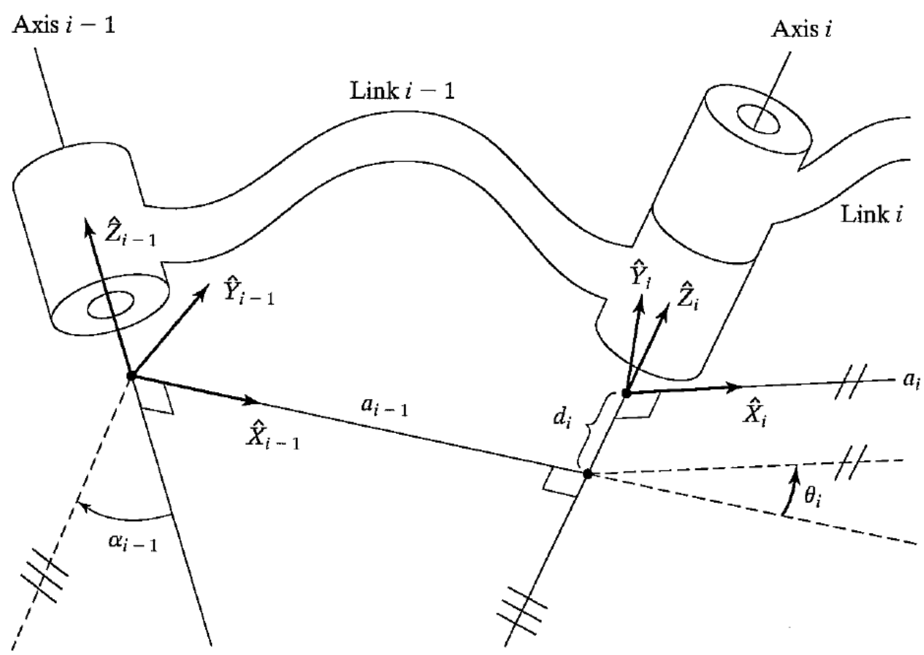


Figure 2.2 Parameters of Denavit-Hartenberg notation (Craig, 2005)

For example, the upper-structure of an excavator has four DoFs with only revolute joints (as will be explained in Chapter 3). The pose of the end-effector of the excavator (tip of the bucket) can be easily calculated using the DH transformation matrix for any given values of joints' angles.

2.3.1 Configuration space

Most of the path planning algorithm benefits from the concept of Configuration Space (C-space) to generate a collision-free path. The idea of C-space is proposed by Lozano-Pérez and Wesley (1979). The C-space of a robot is the space of all possible configurations of that robot, which can be an abstract space when the path planning problem has more than three degrees of freedom (Albahnassi, 2010).

To understand the concept of C-space it is easier to start with a 2-dimensional path planning problem, where a circle-shape agent, with radius r, has to move from a Start position to Target, as shown in Fig. 2.4 To avoid the collision between the agent and obstacle, it has to keep the distance of at least r. This buffer around the obstacles, create a new boundary around them, which is called C-space obstacle (Kim et al, 2012; Klingensmith, 2013). Fig. 2.4 represents an example of 2-dimensional problem where a planner benefits from C-space concept to generate a collision-free path. Now consider another path planning problem where the agent has more than two degrees of freedom.

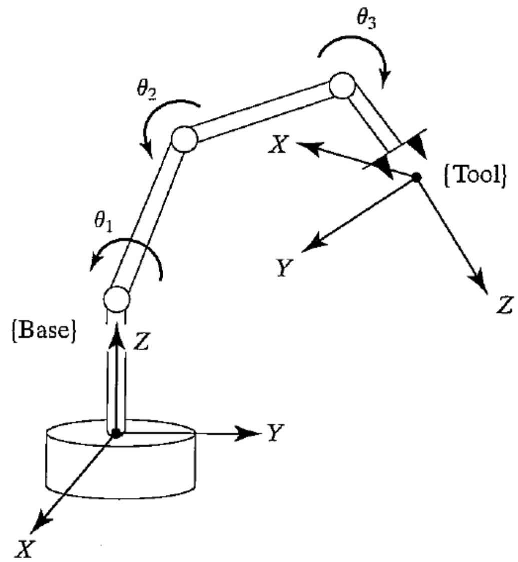


Figure 2.3 An example of 3-DoFs robot with revolute joints (Craig, 2005)

Fig. 2.5 represents another more complex path planning problem in which the agent has three degrees of freedom, two for its position (X, Y) and one for the orientation, θ, with regards to the horizontal line. As shown in Fig. 2.5, the agent has to move from the Start configuration $(X_{Start}, Y_{Start}, \theta_{Start})$ to Goal configuration$(X_{Goal}, Y_{Goal}, \theta_{Goal})$. In this scenario, the agent is not a circle and applying of a circle may result in a conservative solution and consequently a non-optimal path. In order to find a collision-free path, all possible configurations should be checked. Fig. 2.6(a) shows that collision detection is done for all (X, Y) of the agent in a $\theta = 90$, and $\theta = 0$ respectively. The same procedure can be done in every θ between 0 and 90. As shown in Fig. 2.6(b) at $\theta = 90$, those configurations in red represent a collision with the obstacle, which can shape a C-space obstacle. Therefore, the θ can be assumed as another dimension then C-space obstacle can be generated based on three degrees of freedom of the agent, shown in Figs. 2.6(d) and (e). The abstract generated space is called, Configuration Space (C-space).

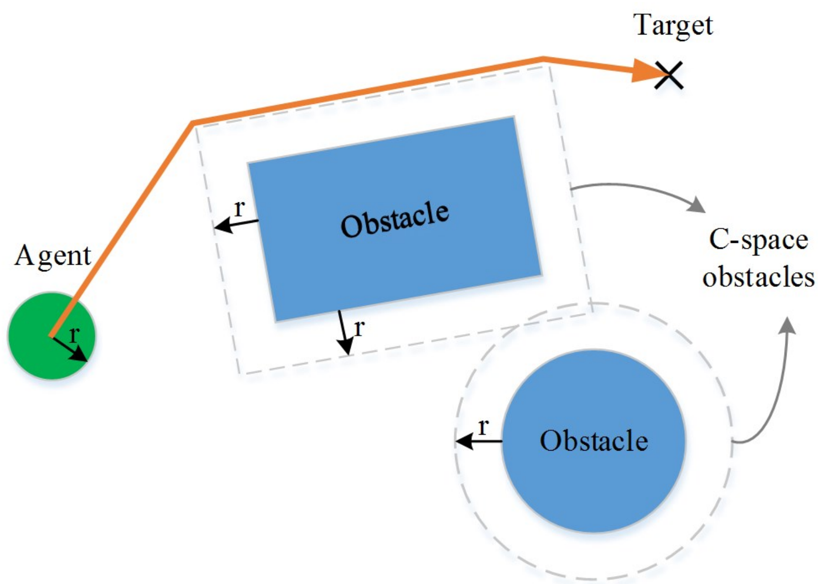


Figure 2.4 A 2D example of C-space

As the result, the real-world three degrees of freedom problem of Fig. 5(a) can be translated to the abstract three dimensional problem shown in Fig. 5(d). This configuration space could be much more complex and bizarre when the path planning problem has more degrees of freedom.

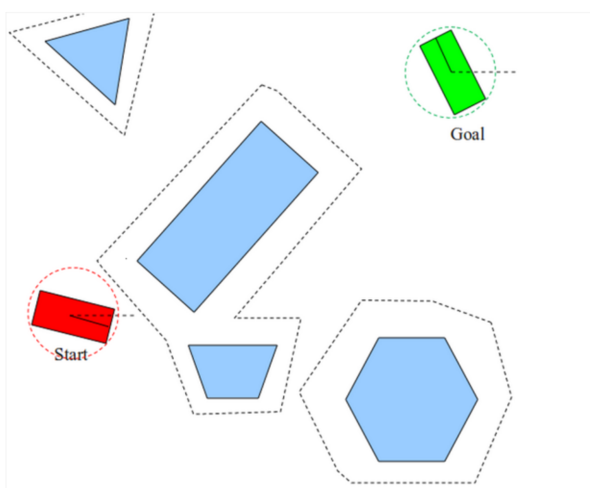
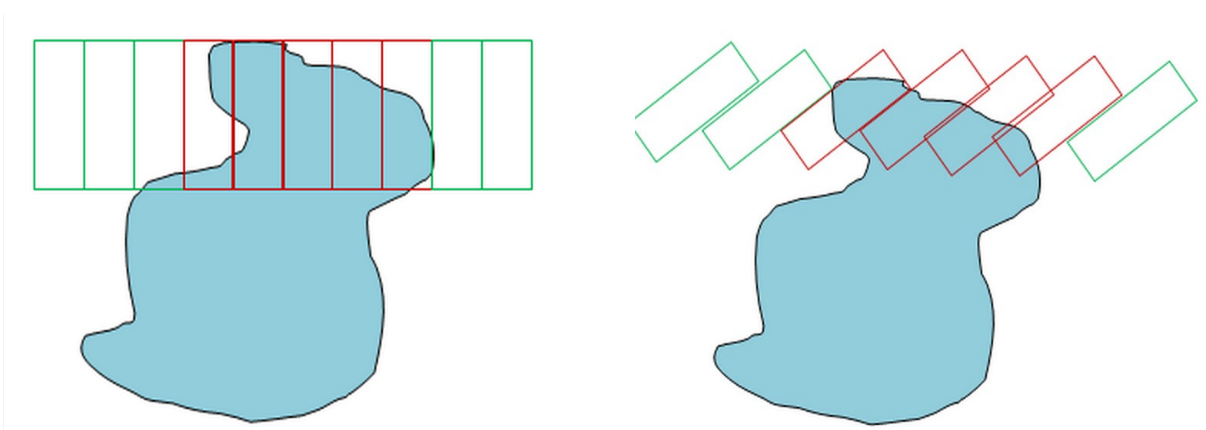


Figure 2.5 A 3D example of path planning problem (Klingensmith, 2013)

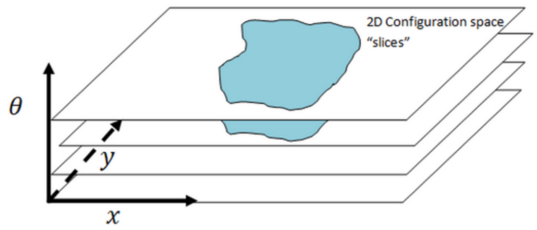
2.4 Earthwork Simulation Models and Path Planning of Excavators

2.4.1 Earthwork simulation models

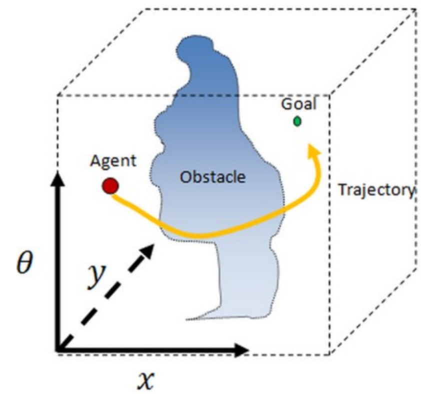
Singh and Cannon (1998) proposed a methodology for earthwork planning is proposed for excavators and the loaders. The method benefits from the knowledge of expert operators and encode it into the planner in two steps, course planning and refined planning. With a vision to semiautonomous or autonomous construction operation, Kim and Russel (2003a) proposed a system architecture for Intelligent Earthwork System in which Artificial Intelligence (AI) and some emerging technologies such as GPS, sensors and wireless communication are applied. Kim and Russel (2003b) presented an extension of their previous work in which an automated task identification/scheduling and a resource allocation method are introduced. Seo et al. (2011) proposed an automated excavation systems in which the earthwork volume is divided into smaller pieces in order to be done by excavators. The research is a part of bigger research project, initiated in Korea, to form a fully Intelligent Excavation System (IES), which is composed of three parts of Excavation Task Planner, Robotic Controller and Actuators/Software Manager.



(a) Collision detection in X Dof   (b) Collision detection for θ Dof



(c) Creation of layers, which is the outcome of collision detection process



(d) Creation of obstacle in the C-space

Figure 2.6 Step-by step generation of example of a 3D C-space (Klingensmith, 2013)

The proposed model benefits from sensing technologies such as 3D laser scanner to capture the existing site conditions, then it recognize the earthwork volume by comparing the existing condition with the design data. The proposed method is able to divide the excavation into smaller pieces, called Working Area. The work areas are later divided into smaller pieces considering the operational aspects of excavators. In the research of Kim and Seo (2011) another part of the IES is presented, excavation task planner for autonomous excavator, which is developed to take the planning consideration into account.

An Automated Landfill System (ALS) is proposed by Tsergn et al. (2000), which could increase the productivity of the whole operation. The system applies several algorithms for multi-equipment landfill operations. The Global Navigating System (GPS) allows the 3D site model to interact with the real information from equipment, then the system creates a collision-free path for each compactor or truck. The equipment benefits from the model to do the compaction in an optimal way.

2.4.2 Simulation of excavator operations

Kamat and Martinez (2005) attempted to make the connection between DES and animation of earthwork operations with a focus on articulated construction equipment. The tie points between DES and animation are only the instances in time in which the state of equipment changes. In some cases, it is needed to divide the states or processes into smaller pieces in order to enrich this connection by adding more time instances. The excavator operation is then animated using iterative IK.

In the research of Kim et al. (2003), macro path planning of construction equipment in unknown environment is investigated. The path planning algorithm is able to generate the continuous collision-free path from an initial position to a target position. In order to implement the proposed model, a range sensor, a displacement sensor, a GPS device, a communicator device have to be mounted on the construction equipment. The proposed path planner, called Sensbug, was proved to be more efficient compared to previous Bug-based path planning methods. In the research of Alzraiee et al. (2012a), System Dynamics is used to include the project uncertainty into the planning of earthmoving project. These uncertainties such as change in scope of work, or level of operator skill, etc. are not included into the conventional planning methods (CPM and PERT). The proposed dynamic planning model showed improved project modeling.

To increase the safety of construction site, Lee et al. (2011b) created a prototype simulator to train the excavator operator. In this research, the human can operate and control the virtual excavator in with a joystick, pedals. Considering the Excavator's upper-structure 4 DoFs, Lee et al. (2011a) developed a model to control the excavator's end-effector. Given a path from a skilled excavator operator, the controller applies the same path on the end-effector such as digging the ground using inverse kinematics. Liu et al. (2010) focused on the trajectory control of the upper-structure and the boom of a remote excavator. With a vision of autonomous excavation, Hoan et al. (2011) presented a model for the excavator manipulator. The torque applied to joints is computed and tested using simulation. To help the construction site manager in a better site layout plan, Lin et al. (2013) proposed a decision support system, which assesses the accessibility of tractor trailer from operational and safety point of view. In order to test the proposed model, they utilized Microsoft XNA as the game engine combined with Nvidia PhysX game engine.

2.4.3 Parameterized script for excavators

Many researcher conducted research about different construction equipment such as tower crane, mobile crane, roller, etc. however few of them investigated the path planning of excavator at micro level. Generally speaking, operation of an excavator consists of sequential processes of digging, swinging to dump, dumping and returning to dig face again. In a mining or a big earthwork project, this operation might happen thousands of times for only one excavator in a very similar way. That is the reason it may be different from operations of other construction operations. Simulation of excavator operation requires considerations particular to this equipment to ensure maximum productivity and safety. In an attempt to a fully automation excavation, Stentz et al. (1999) developed a semi-autonomous excavator. The excavator follows a predefined motion, which is called, script based motion planning. The robot excavator is equipped with two scanning sensors, which makes the excavator aware of its environment, and the software automatically finds the position of truck and consequently the exact dump position.

In a simple word, parameterized scripts is a set of state and commands that imitate the operation of the excavator. In an investigation to minimize the cycle-time of excavator repetitive operation, Rowe (1999) proposed adaptive motion planning for autonomous mass excavation. He considered requirement of excavator to generate a feasible path and finally optimized it using memory based learning. The optimization is done using two objective functions following: 1) to minimize the time and, 2) constraints error. Finally the results showed that performance of the proposed method is even better than an expert operator.

2.4.4 Automated machine guidance and control

To improve the quality and productivity of earthwork operations, Automated Machine Guidance and Control (AMG/C) is introduced to the industry (Vahdatikhaki, 2015). The available systems are able to help the operator with supportive information received from high precision tracking and sensing technologies (e.g., GPS, robotic total station, laser augmentation) (Kiongoli, 2010). This support can be provided in two levels: guidance and control. At the guidance level, the relevant information regarding the existing situation is compared with the design data and the guidance signals are given to the operator. In the case of a grader for example, the real-time information regarding the design level and the existing level are shown on the screen in the cabin. At the control level, the grader blade is automatically controlled according the information received from the GPS and/or laser augmentation systems, as shown in Fig. 2.7.

Figure 2.7 Grader with Dual GPS (Kiongoli, 2010)

2.5 Path Planning Algorithms

2.5.1 Bug algorithm

Bug algorithm is considered as a reactive planner and is one of the simplest path planners for 2D configuration space. Assume a simple case where an agent is moving in a 2D environment. It has to move from the initial configuration to the goal configuration and there are couple of obstacles on the way. Lumelsky et al. (1987) proposed two varieties of a method to resolve such problems, called Bug 1 and Bug 2.

(a) Bug1: Fig. 2.8 represents the behavior of agent that follows Bug 1 algorithm. The aim is to generate a feasible path from the Start to the Target. The agent leaves the Start toward the Target in a straight line, then the first leave point is called L_0 . The agent continues moving toward Target, until the first obstacle is reached at hit point H_1. Then it follows the boundary of the obstacle to the hit point again. As the agent follows the boundary it measures the distance to the target continuously and the minimum distance is stored as Q_m and the location is stored as L_1. In the next round, the agent leaves the obstacle at L_1 toward the Target again. The procedure will be repeated until the Target is reached. The following steps represent the Bug 1 algorithm (Lumelsky et al., 1987).

Step 1. From the point L_{i-1}, go toward the Target along the straight line until one of the following occurs: (a) The Target is reached. (b) An obstacle i is reached, then H_i is stored. Go to step 2.

Step 2. Follow the obstacle boundary. Stop if the Target is reached. Otherwise continue to get back to H_i, then find the leave point L_i based on the Q_m, the minimum distance from the object boundary to the Target. Go to step 3.

Step 3. If the Target is reachable, leave the object at L_i, set $i = i + 1$ and go to step 1.

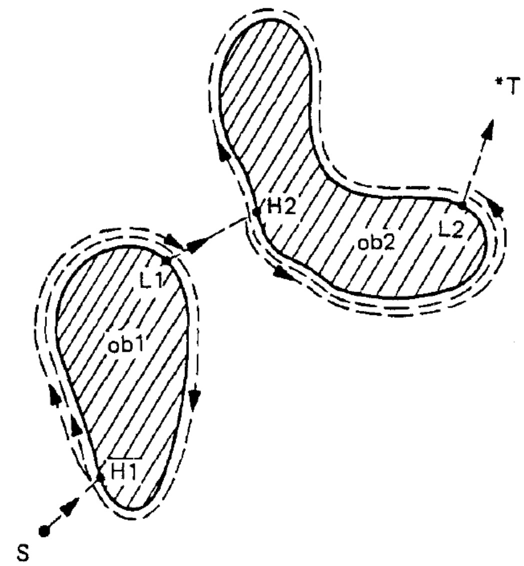


Figure 2.8 Example of Bug 1 algorithm (Lumelsky et al., 1987)

(b) Bug 2: Same as previous, the aim is to generate a collision-free path for an agent from Start to Target. The following explains the Bug 2 algorithm (Lumelsky et al., 1987), which can be verified by an example of the algorithm shown in Fig. 2.9:

Step1. From the point L^{j-1}, move along the straight line toward Target until one of the following occurs:

The Target is reached. The procedure stops.

An obstacle is encountered and hit point, H^j, is defined. Go to step 2.

Step 2. Follow the obstacle boundary until one of the following occurs:

The Target is reached. The procedure stops.

The line Start-to-Target crosses the boundary of the obstacle at another point Q, as $distance(Q, Target) < distance(H^j, Target)$, Define that point as the leave point, L^j. Set $j = j + 1$ and go to step 1.

The agent returns to H^j and thus completes a closed curve (boundary of the obstacle) without finding the next hit point H^{j+1}. The Target is not reachable, the procedure stops.

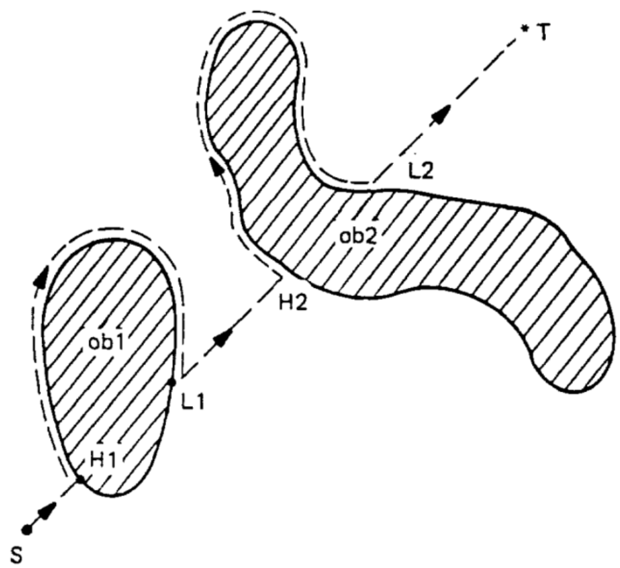


Figure 2.9 Example of Bug 2 algorithm (Lumelsky et al., 1987)

2.5.2 A* algorithm

A* algorithm proposed by Hart et al. (1968) is an extension of Dijkstra algorithm (Dijkstra, 1959). Benefitting from heuristics, A* shows a better performance comparing with Dijkstra in finding the shortest collision-free path in a C-space (Soltani et al., 2003). From a given initial node to a given goal node, A* moves forward in the graph along the lowest expected total distance or cost. The cost of a node (n) is determined based on the following equation:

$$f(n) = g(n) + h(n) \tag{2.3}$$

Where $g(n)$ is the operating cost function, which is the actual operating cost from the initial node to the current node (n). Heuristic function $h(n)$ estimate the cost of movement from the current node to the goal node on the graph. The heuristic is added to the original Dijkstra algorithm, $h(n)$, which is the admissible heuristic that means it must not overestimate the distance to the goal (Choset, 2015).

Starting from initial node, it keeps aside a list of nodes with priority called, open list. The node with the lower $f(n)$ has the higher priority. In each step the node with the lowest priority, called best node, is removed from the open list and the open list would be updated according to the new set of nodes in neighborhood of the best node that are not in the close list. Close list consists of those nodes that are already explored. The algorithm keeps moving in the graph until the goal is reached. The shortest distance between the initial and goal node is $f\ (Goal - Node)$. However, to generate the path between these nodes, the algorithm requires to follow from the goal node to its predecessors until the initial node is reached. The Fig. 2.10 explains the A* algorithm.

$C(n1, n2)$ estimates the cost of movement between two neighbor nodes of $n1$ and $n2$. The Manhattan method and the diagonal shortcut are two common methods for this measure. The Manhattan method estimates the distance to the next node considering vertical and horizontal movements. This method is a bit faster than the diagonal shortcut however it is not admissible. On the other hand, the diagonal shortcut is slower but it is admissible (Laster, 2005).

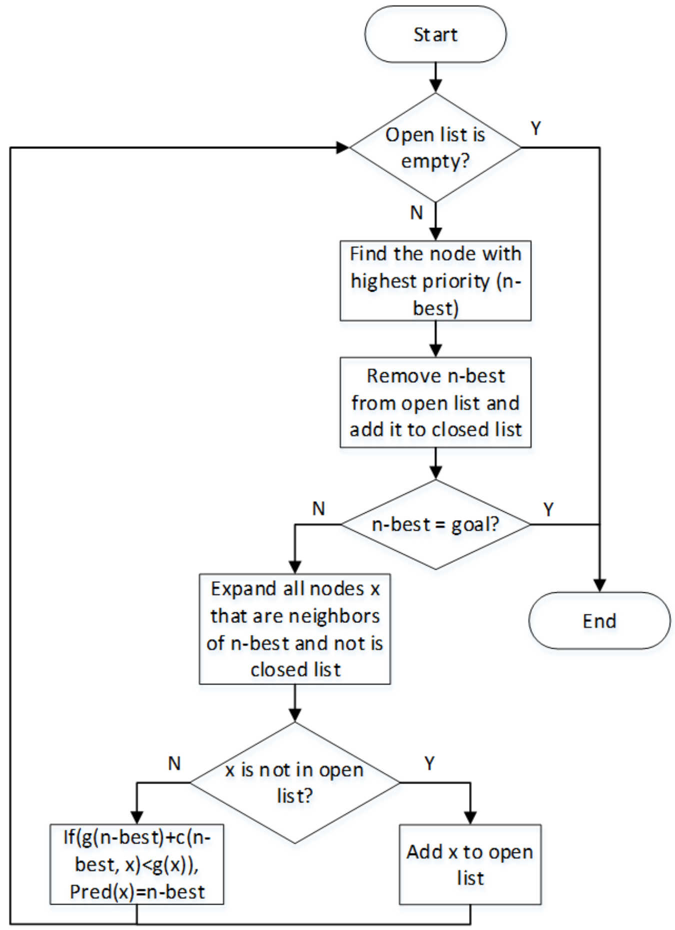


Figure 2.10 A* algorithm (Adapted from Choset, 2015)

2.5.3 D* algorithm

A* is able to generate a collision-free path between initial node to the goal node. It is able to find an optimal solution in an unknown environment with fixed obstacles. A* can be used for replanning, however it is not efficient in terms of computation effort. The Fig. 2.11 explains A*-replanning algorithm.

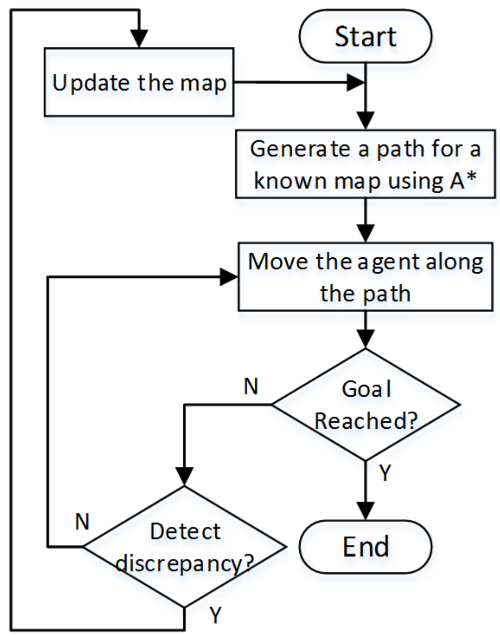


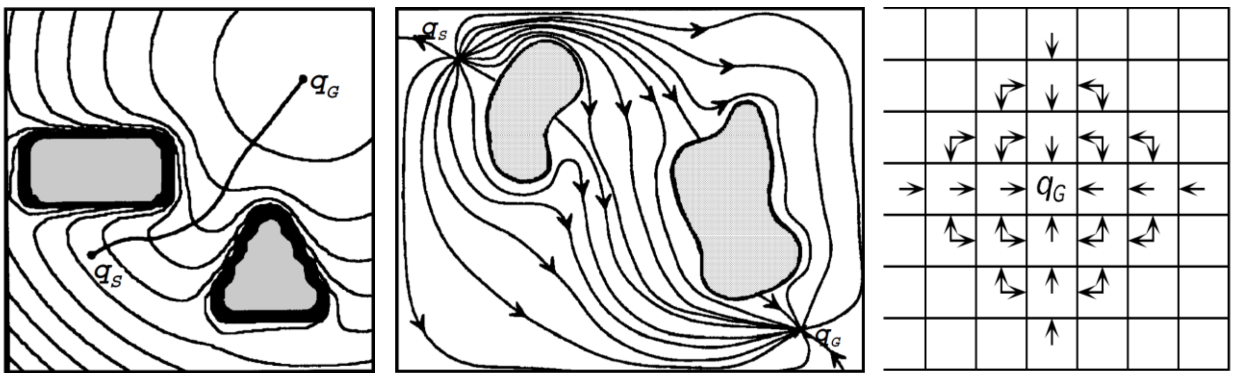
Figure 2.11 A* re-planning algorithm (Choset, 2015)

However, it cannot deal with dynamic obstacles efficiently. D* is proposed by Stentz (1995) to overcome this limitation. D* stands for "Dynamic A* search" in which the cost parameters can change during the process (Choset, 2015).

2.5.4 Potential field

Potential Field technique is different from other methods, as this path planning technique guide the agent in the C-space based on a vector that is defined on the agent and obstacles positions. In this technique there is no trajectory (Klingensmith, 2013).

This method benefits from a scalar function called, Potential. The goal configuration has the lowest potential value and obstacles have a high value of potential. The agent has to move from the initial configuration toward the lower potential value, which is along the negative gradient of the potential. The obstacles are assumed as positive high potential values, since the agent is expected to avoid them. One of the deficiency of this method is that the agent can be trapped in the local minimum. The agent may be trapped in a local minimum instead of reaching the goal configuration. Fig. 2.12 represents three examples of different types of potential field maps (Henrich, 1997).



(a) Minimum distance to goal

(b) Simulated current flow

(c) Propagated move directions

Figure 2.12 Different types of global potential field maps (Henrich, 1997)

The result of this planner is a realistic and smooth path. From a computation point of view, this algorithm can be considered as a fast path planner, which is an advantage.

2.5.5 Probabilistic RoadMaps (PRM)

PRM is one of the general solutions to path planning problems with high number of DoFs. The method can be compared with Visibility Graph, but in high dimensional C-space. The method benefits from probabilistic random node generations to reduce the computation effort. Based on randomly added nodes to the configuration space, a graph would be generated and the algorithm searches to find the path within the graph. The generation of random nodes decreases the number of collisions; however the optimality of the solution is not guarranteed.

The PRM is probabilistically complete, which means it is able to find a feasible path if there is no time constraints. The optimality of the method also depends on the number of generated nodes. The higher number of nodes would result in a more optimal solution (Chang et al., 2012).

2.5.6 Rapidly-exploring Random Trees (RRT)

As shown in Fig. 2.13, starting from the initial configuration as the root of the tree, the algorithm randomly and incrementally adds leaves to the tree. This process will continue until the predefined goal is reached. This agility and efficiency of RRT is mainly due to this randomness behavior; hence increasing the chance of success by reducing the search space into a set of randomly added nodes. In the case of excavators, the search space is a multi-dimensional space and each dimension represents one angle of a rotational joint as a DoF. The initial and goal configurations can be translated into two sets of angles in case of an excavator. Then, random set of angles, representing a new configuration (i.e. position) in the C-Space, would be generated. Based on this new random configuration, the tree would be extended from the nearest leave of the tree toward that node. This process will continue until the goal is reached. The following pseudo code of Build-RRT adapted from (La Valle and Kuffner, 2000) shows how the tree grows.

Build_RRT ()

T.init (x_{init});

For $k = 1$ to K do

    x_{rand} = Random_Configuration ();

x_{near} = Nearest_Neighbor (x_{rand}, T);

x_{new} = Extend (x_{near}, x_{rand});

If (x_{new} =! null)

T.Add (x_{new});

End

Return T;

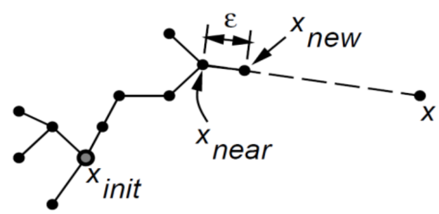


Figure 2.13 The tree in the C-Space (La Valle and Kuffner, 2000)

Due to the random nature of sampling-based algorithms such as RRT, the generated path is jagged and zigzag. This jaggedness results in an unnatural movement of the construction equipment, which means longer cycle-time and lower productivity. This is considered as a limitation of these types of algorithms because although the path is obstacle-free, it is still unsuitable from practical and operational points of view.

2.5.6.1 RRTBiasedLimCon Algorithm (Albahnassi and Hammad, 2011)

To improve efficiency of the RRTBiasedLimCon algorithm, the following modifications are applied to the basic RRT to improve the efficiency of the algorithm while considering engineering constraints. The proposed method mainly benefits from three features, biasness, Connect (extension) function and rules of action. With a user-defined probability of P, the Goal-Node is chosen as the target for the tree. This probability of P determines the biasness of the algorithm with respect to the Goal-Node. As shown in Fig. 2.14, instead of growing the tree with a step of ε for each generated Rand-Node (as shown in Fig. 2.13), the tree keeps extending several times until an obstacle or the Goal-Node is reached (Albahnassi, 2010). This heuristic rule increases the greediness of the search, hence making the algorithm faster and more efficient. Rules of action are another modification that is shown in Fig. 2.14. In the basic RRT, a regular extension of the tree happens along the Rand-Node from the nearest node, which means movement in multiple DoFs. However this movement may not be suitable for cranes because it would result in the movement of multiple joints. In addition, a constraint has been imposed on the algorithm to avoid extending the tree away from the goal configuration. As shown in Fig. 2.14, the tree stops at q_{new} instead of continuing in the dotted line.

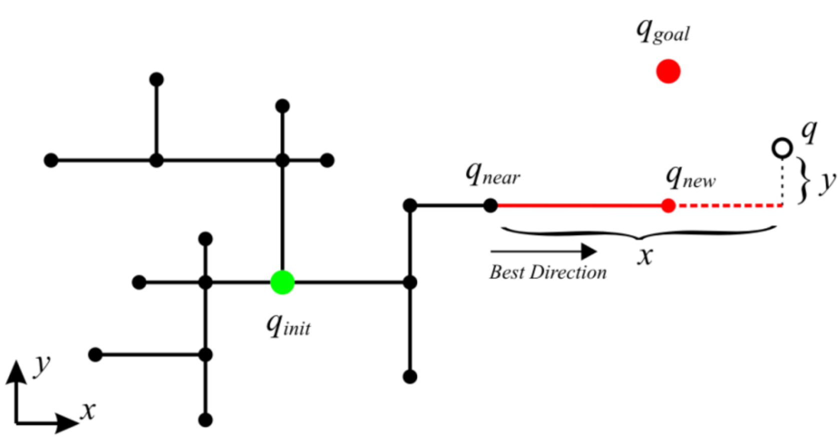


Figure 2.14 Limited extension of the tree along best DoF (Adapted from Albahnassi, 2010)

2.5.6.2 Problem Definition

Comparing with DRRT, the RRTbiasedLimCon algorithm is proven to be an efficient path planner in construction projects. Besides, the results of the RRTbiasedLimCon showed a significant improvement in terms of the performance (Albahnassi, 2011). Speed of the algorithm is an important criterion that has been addressed for near-real time simulation. A rapid and fast path planning is required to re-act to the challenging and dynamic construction environment with many pieces of equipment and humans. However, there are other concerns such as engineering constraints, which may not be fully addressed in the previous research, such as quality of the generated path that would consequently affect the productivity.

In order to take into account the applicability of the generated path for cranes, RRTbiasedLimCon has considered rules of action that would lead to movement of only one DoF at a time. Although these rules may be acceptable for cranes, where safety has a much higher weight than productivity, applying them to excavators would result in a robot-like movement and a lower productivity. An efficient and fast movement of excavators requires applying movements in multiple DoFs. For example, an excavator operator may move two or more joints of the equipment while loading the truck to reduce the cycle-time (Stentz et al., 1999).

2.6 Comparison of Different Types of Path Planning Algorithms

Table 2-1 shows a comparison between different path planning algorithms considering optimality, completeness, query type, DoFs, type of obstacles and computation speed (Klingensmith, 2013; and Bruce 2004). Different types of path planning can be used for different scenarios with regards to their capabilities and characteristics. Usually A* or Bug algorithm can be used for macro path planning as the problem has low DoFs (Kim et al., 2003, Kim et al., 2012, Marzouk and Ali, 2013). In this research, A* is used for the macro path planning because it is complete and provides an optimum path, which can dramatically decrease the cycle-time of the equipment. However, it is applicable only for a small number of DoFs when the concern is to solve only the macro path planning of the equipment. On the other hand, in order to consider the micro path planning of the equipment, other algorithms may be more suitable, such as the RRT algorithm, which is a reliable technique in dealing with a large number of DoFs; although it does not provide an optimal path (AlBahnassi & Hammad, 2011).

Table 2-1 Comparison of path planning algorithms (Adapted from Bruce, 2004; Klingensmith, 2013)

Path planning algorithm	Optimality	Completeness	Query type	DoFs	Obstacle	Speed	Non-Holonomic
Bug	No	Yes	Reactive	2	Any	Excellent	No
Visibility graph	Yes	Yes	Multi	2-3	Polygon	Medium	No
A*	Yes	Resolution	Single	2-3	Grid	Slow	No
Potential field	No	No	Reactive	2-3	Any	Excellent	No
RRT	No	Probabilistic	Multi	2-100	Any	Good	Yes

As shown in Table 2-1 Bug algorithm is able to generate the path, if available. This algorithm is effective when the knowledge about the environment is low, the agent move forward and follow simple rules to get to the target. However, it does not have any notion of optimality then it greatly influences on the productivity of the construction operations.

a) Optimality

Optimality of the generated path is one of the desirable metrics of a path planner. Generally speaking, sampling-based algorithms such as RRT do not guarantee optimality of the path. A*, on the other hand, is an optimal path planner however it is not efficient enough for problems with high number of DoFs (Soltani et al., 2003).

b) Completeness

Although Bug planner is a very simple algorithm, it is complete in most of the cases; meaning that it is capable of finding the path in a reasonable time if there is any or it can report there is none, if it is not feasible (Albahnassi, 2010). Kim et al. (2003) proposed a modified version of the Bug, called SensBug, to improve the capability of this planner for path planning of construction equipment. In the discussion of Lee (2008), it is shown that it may fail to reach the goal in some problems. Therefore, completeness is a fuzzy criteria. In sampling based planners, this criteria is affected by the algorithm parameters. Or for the A*, it can be complete or not complete, which depends on the size of the grid.

d) DoFs

Bug is a simple planner in a 2D space where a robot needs to find the way from the starting position to the goal position. Visibility graph can be applied for a 3D C-space. Currently, it is a well-known planner for game developers. Unity 3D game engine uses A* for 3D space on the Nav-mesh, which is a type of graphs. However, these algorithms cannot generate the path for more than 3D C-space efficiently if applicable. That is mainly because of increased search space will result in a huge number of possible configurations and consequently will decrease the efficiency. Sampling based algorithms are proposed to address difficulties of such problems. Instead of searching throughout the C-space, RRT only searches for random configurations in the C-space and check the collision for those specific configuration instead. That is the reason sampling based algorithms such as RRT and PRM are much more efficient in high dimensional problems.

f) Non-holonomic

Very basic path planners such as the Bugs consider the robot or the agent as a point. In such case, the planner attempts to find a path from the start to the goal configuration. However, in many cases, it is not practical since any robot or agent has dimensions. Some research overcame this limitation by simply considering a safety buffer to the obstacle; this buffer could be radius of the surrounding circle of the robot. This conservative approach guarantees that there would be no collision if the planner considers the robot as a point. This is enough for a problem in 2D space however in a more complicated problem with more DoFs a more complicated path planner is required. Sampling based planners are proven to be efficient in a more complicated problem such as a robot with four or five DoFs. They are capable to rapidly find a series of obstacle-free configurations in the C-space and the robot may find a collision-free path from the initial to the goal configuration by following them. For example, the basic RRT may find a series of collision-free configuration to make the path based on them, however it is not always feasible for the robot to follow them since the basic RRT generates random configuration blindly and without any notion of possible constraints of the movement. A car or a robot wheeled loader is not able to rotate 10 degrees in the same place. These type of motions, called non-holonomic, are constraints by controllable variables instead of DoFs. RRT is capable of handling such problems (Lin et al., 2014b).

2.7 Limitation of Previous Research

According to the literature review, RRT has many advantages when the problem is to deal with multiple number of DoFs. RRT is introduced as an effective algorithm for the path planning of construction equipment with three to seven DoFs. However, it has the following limitations:

1- Smoothness: The path generated by existing RRT methods is not smooth, and the agent equipment has to follow a fluctuated path, especially when it comes to reality, it seems to be almost impossible to apply a path generated by RRT to a real construction equipment.

2- Optimality: Another problem of existing RRT methods is that they do not have any sense of optimality, hence the final path will consist of many unnecessary movements, which will result in a less productivity.

3- Natural movement: In case of excavator agent, RRT does not have notion of the work, so it has to benefit from parametric script to make a full free motion planning in a way that a real operator works to ensure the safety.

For the case of agent excavator, the final path is preferred to encompass the requirement of the excavator to minimize the energy consumption, considering the applicability of the path in the real world. Previous path planners did not fully consider the obstacles in the work site. An agent for the excavator must be able to find the practical path to achieve its assigned duty, and at the same time, it must be able to recognize and avoid the unforeseen possible obstacles.

2.8 Summary

Path planning is an essential and integral part of the agent-based simulation in the construction domain. Many researchers have benefitted from these algorithms to analyze the construction operations and processes. For example, Marzouk and Ali (2013) investigated the agent-based simulation of the piling operation in order to address the limitations of DES. These limitations are mostly related to spatial aspects of the simulation. That paper concentrates on the safety/productivity aspects of the project and can mostly be used in the design stage. However, in another research of Lin et al. (2014b), a more detailed part of construction operations is investigated, hence a more advanced path planner is used to address the specific kinematics of crawler cranes. Parametric path planning (Rowe, 1999) is proposed to address the challenge of autonomous excavation. However it cannot handle the collision avoidance in the case of obstacles such as underground pipes or workers.

Many research has been done in the area of path planning of heavy construction equipment, particularly mobile and tower cranes. To the best knowledge of the author, there are few investigations in the area of excavator's path planning considering its specific engineering constraints and operational logics.

3. METHODOLOGY

3.1 Introduction

As explained in Chapter 2, many research works have been done regarding the simulation of construction operations at different levels of detail, which depends on whether the concern is managerial or it is a local safety or productivity of a single construction equipment. A relatively rich investigation is done regarding the application of path planning in construction equipment, however there are still areas that need further research.

This chapter can be divided into three parts. In the first part, path planning is used to enrich the planning agents to deal with ongoing problems that may happen during the course of operations such as underground pipes, other construction equipment, etc. As the excavators play a significant role in the productivity and safety of earthwork operations, many modifications are applied to the basic RRT to fit the algorithm to the engineering constraints of the equipment. In the second part of this chapter, we move one step beyond the path planning algorithm and introduce time into it. It will allow us to quantify the productivity measures. In the third part, an agent-based excavation operation simulation model is presented, which considers the interaction between the excavator and the truck. The model is designed in a way to increase the overall performance of the excavation operations.

3.2 Proposed Micro Path Planning

In this research, as the main focus is on improving the safety and productivity of the excavation operations where the excavator's operator is supervising the execution of work based on the automated path planning results. A synthetic approach of several path planning methods is used to balance the needs of safety and practicality as will be explained in Section 3.5.

3.2.1 Micro Path Planning of Excavators

The routine motion of the boom-stick-bucket of the excavator between the dig point and the dumping point on top of the truck includes the swing of the upper-structure, boom up or down, stick in or out, and the curling of the bucket. A skillful worker can apply a combination of these four DoFs in order to shorten the cycle-time of one loading pass. Planning this path based on the trajectory of the bucket is not practical because of the complexity of modeling the actions of the hydraulic cylinders (Rowe 1999). Therefore, a parametric rule-based approach is adapted from (Rowe 1999). The assumption in this path planning is that the movements between one pass and another are similar and that the parameters of the rules can be adjusted to consider the exact locations of the dig point and the dumping point. Fig. 3.1 shows several snapshots of three excavators loading trucks with different settings with respect to the relative heights and locations of the excavator, the dig point, and the truck. In Fig. 3.1(a), the digging is at a very low level, forcing the boom and the stick to extend almost to their maximum angles. In Fig. 3.1(b), the excavator is in the middle between the digging area and the truck, resulting in about 180 degree swinging of the boom. In Fig. 3.1(c), the truck is at a lower level than the excavator, resulting in a slightly shorter cycle-time because the boom does not have to be raised above the bed of the truck before and after dumping. Another assumption is that, except for the truck that is considered as a known obstacle in the parametric rule-based approach, the repetitive work of the excavation between the dig and dumping points is obstacle-free.

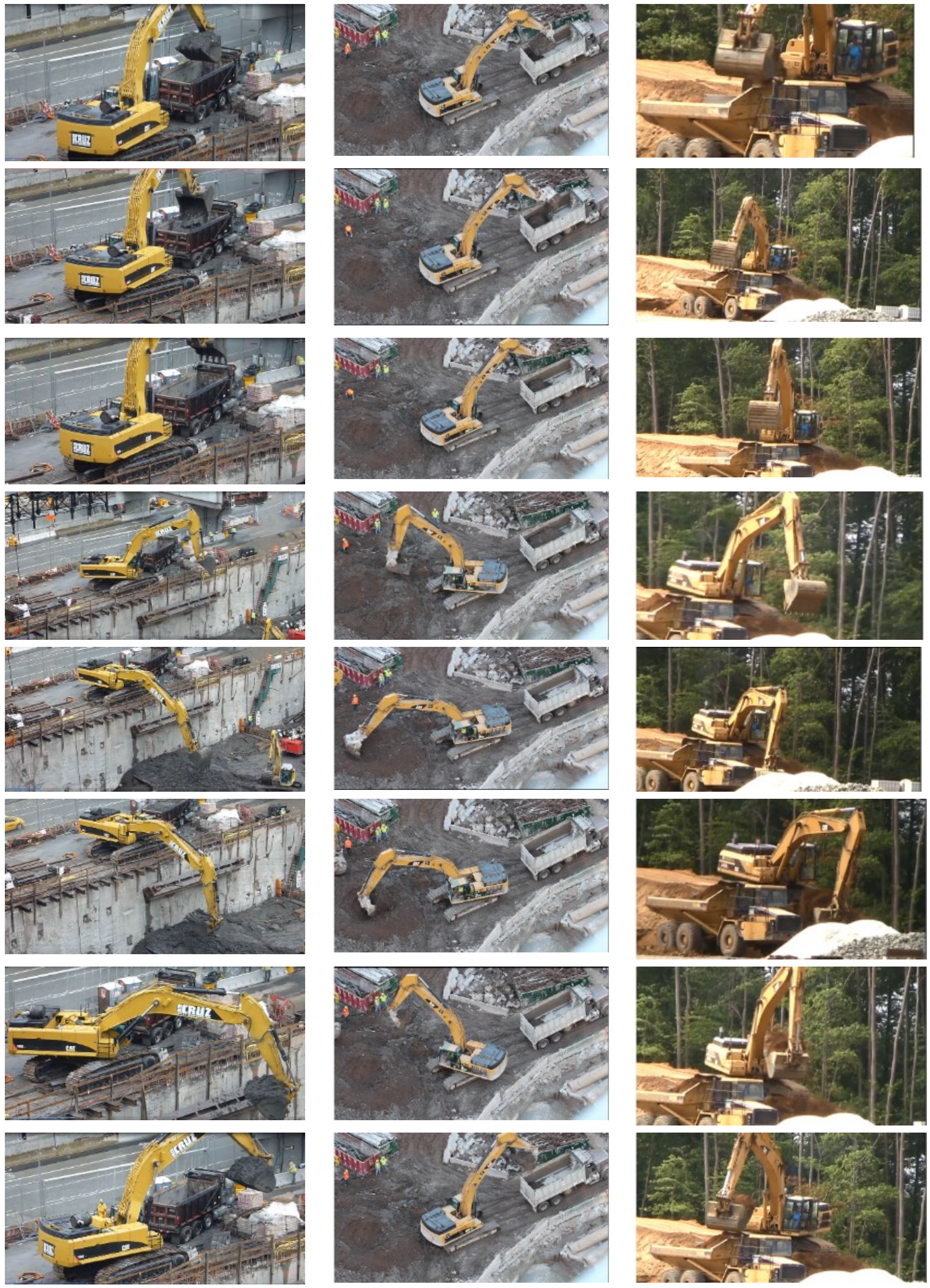


Figure 3.1 Snapshots of excavators loading trucks: (a) The digging is at a very low level forcing the boom and the stick to extend almost to their maximum angles, (b) The excavator is in the middle between the digging area and the truck, (c) The truck is at a level lower than the excavator. (Hammad et al., 2014b)

This assumption does not hold in many cases where the site is congested with fixed and/or dynamic obstacles. For example, Fig. 3.2 shows an excavation working near underground pipes, which represent fixed obstacle that have been detected while excavating and which will obstruct the motion of the stick and bucket based only on the parametric rule-based approach.

Figure 3.2 Unsafe situation because of underground pipes (Hammad et al., 2014b)

Fig. 3.3 shows two examples of two excavators working in a tight area and should consider each other. In these conditions, it is necessary to use a more detailed path planning algorithm such as RRT (AlBahnassi and Hammad 2011). This issue will be elaborated in depth in this chapter.

(a) Two backhoes loading trucks and hammering

(b) Two backhoes working in a tight area

Figure 3.3 Unsafe situation because of congestion (Hammad et al., 2014b)

3.3 Embedding Heuristic Rules in RRT Path Planning

3.3.1 Representing excavator cyclic motion

The cyclic operation of an excavator can be divided into the following subtasks: *digging*, *moving to dump*, *dumping*, and *returning to the next dig*. An excavator's free motion planning is defined as the above-mentioned sequential subtasks except *digging*. The present research focuses on the free motion and excludes the digging subtask because it requires considering the interaction between the bucket and the soil. Fig. 3.4 shows the four DoFs of the upper structure of a typical excavator where θ_i is the angle of a joint. θ_1 is the swing angle, which is the angle of rotation of the upper structure with respect to the lower body. A typical excavator has the ability to swing 360°. θ_2 is the angle between the boom and a horizontal plane. θ_3 is the angle between the boom and the stick. The angle between the bucket and the stick is called θ_4. The ranges of rotation for the last three DoFs can be extracted from the specifications of the equipment.

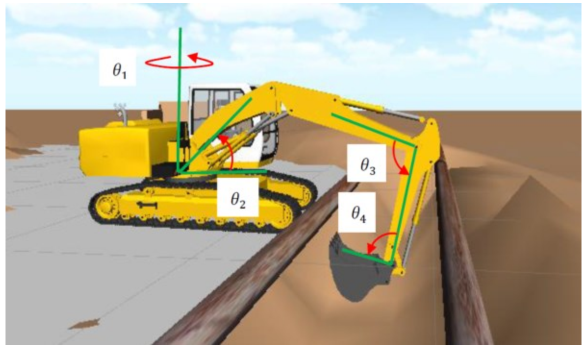


Figure 3.4 Four DoFs of upper structure of a typical excavator

Table 3-1 shows that a free motion planning of an excavator can be achieved in three steps. In the *moving to dump* subtask, the soil in the bucket should be maintained while moving toward the truck by constraining the tilt angle of the bucket with respect to the horizontal plane as shown in Fig. 3.5. To consider the engineering constraint of the bucket when loaded, the following equation has to be respected.

$$\sum_{i=2}^{4} \theta_i^j = \sum_{i=2}^{4} \theta_i^0 = constant. \tag{3.1}$$

where in θ_i^j, i is the joint index and j represents different values for that DoF. θ_i^0 equals to θ_i , which is represented in Fig. 3.4. The remaining three DoFs can be biased in this subtask toward the Goal-Node. When the excavator reaches the dump configuration, it has to unload the soil by curling out the bucket and applying a small inward rotation of the stick. This subtask can be done either by an advanced path planner or by the parametric path planning (Stentz et al., 1999) because of its simplicity. The last subtask is to move to the next dig point and all four DoFs of the excavator are free for path planning.

Table 3-1 Biasing and constraining the DoFs for the subtasks within the free motion of excavator

	Subtask	Initial Configuration	Goal Configuration	Remarks	Number of DoFs
Free motion planning	Moving to dump	$\begin{bmatrix}\theta_1^0\\ \theta_2^0\\ \theta_3^0\\ \theta_4^0\end{bmatrix}$	$\begin{bmatrix}\theta_1^1\\ \theta_2^1\\ \theta_3^1\\ \theta_4^0\end{bmatrix}$	• Biasing the first three DoFs toward the Goal-Node, which results in moving the boom up while swinging. • Bucket is constrained to maintain soil.	3
	Dumping	$\begin{bmatrix}\theta_1^1\\ \theta_2^1\\ \theta_3^1\\ \theta_4^0\end{bmatrix}$	$\begin{bmatrix}\theta_1^1\\ \theta_2^1\\ \theta_3^2\\ \theta_4^1\end{bmatrix}$	• As it is a simple subtask, it can be done using either parametric path planning (Stentz et al., 1999) or the new proposed method.	2
	Returning to next dig	$\begin{bmatrix}\theta_1^1\\ \theta_2^1\\ \theta_3^2\\ \theta_4^1\end{bmatrix}$	$\begin{bmatrix}\theta_1^2\\ \theta_2^2\\ \theta_3^3\\ \theta_4^2\end{bmatrix}$	• Biasing the four DoFs toward the Goal-Node.	4

(a) Near digging configuration (b) Boom-up

Figure 3.5 Constraining the loaded bucket to maintain the soil when moving from (a) to (b)

To generate the path using a path planning algorithm, the initial and goal configurations must be defined for each subtask using inverse kinematics (IK) as explained below. This would result in a set of sequential intermediate configurations to link the path segments of the free motion cycle.

In the research of Kamat and Martinez (2005) a new iterative inverse kinematics technique is used to find the configuration of the excavator given the position of the end effector. However, the analytical inverse kinematics of Rowe (1999) is preferred in this research as it is faster. There are three engineering constraints that have to be respected: (1) While loaded, the bucket has to remain parallel to the horizontal plane (refer to Equation 3.1), (2) When loaded, the bucket has to be considered fixed and the other three DoFs of the excavator have to be determined based on the final location of the truck bed.

$$(X,Y)_{end\ effector} = (X,Y)_{Middle\ of\ truck\ bed} \tag{3.2}$$

(3) In dumping, the final pose of the end effector should be close enough to the truck bed to unload the soil without hitting the bed. Depending on the bucket size and the truck-bed size, the pose of the excavator end-effector changes while the truck is being loaded by the excavator. This change is in Z direction and can be calculated based on the following equations:

$$Z_{end-effector} = Z_{truck\ bed} + B + S + (i-1) \times K \tag{3.3}$$

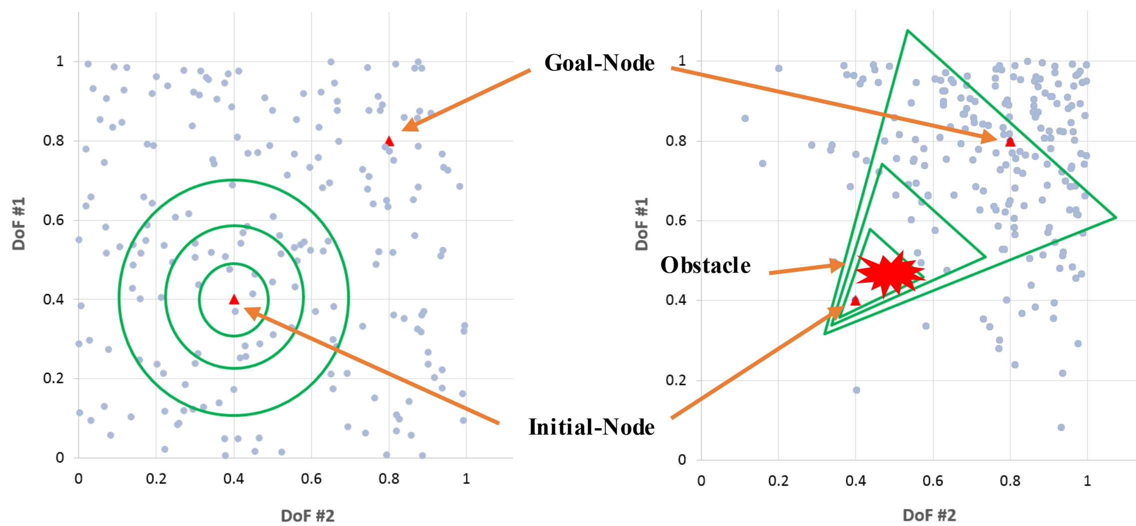
$$K = \frac{bucket\ capacity}{truck\ capacity} \times depth\ of\ truck\ bed \tag{3.4}$$

where i is the number of the bucket load that is being dumped over the truck bed, B is the bucket length, S is the safe distance to avoid hitting the truck bed. The length of the bucket plus a safe distance has to be respected between the bucket and the truck bed to avoid any damage to the bed.

3.3.2 NonuniformRRT

As explained in Chapter 2, RRT is expected to be a fast and effective path planner for equipment that has a high number of DoFs, such as excavators. Fig. 3.6 shows two simple 2D examples of the C-Spaces that will be used to illustrate the concept of the proposed algorithm. The Initial-Node and Goal-Node are indicated and the other dots are randomly generated configurations called

Rand-Nodes. Fig. 3.6(a) shows a basic RRT where the random configurations are uniformly distributed making the tree grows uniformly. This uniform growth is schematically shown by circles. Fig. 3.6(b) represents the concept proposed in this paper for a NonuniformRRT where the random nodes are generated using a biased random number generator; making the tree grows biased toward the Goal-Node instead of searching the whole C-Space.



(a) Basic RRT

(b) NonuniformRRT (Adapted from Langari and Hammad, 2015)

Figure 3.6 Schematic 2D representation of tree growth

The NonuniformRRT is expected to scarify completeness for a faster and more optimal path. Probabilistically speaking, instead of growing in all directions, the tree extends in a certain direction toward the Goal-Node. The comparison of the basic RRT and NonuniformRRT is similar to the comparison of Dijkstra and A* searching algorithms (Soltani et al., 2003). Dijkstra searches for the path considering all possible solutions; however A* benefits from a heuristic function to narrow down the search space. The advantages of the NonuniformRRT are: (1) Maintaining the agility, flexibility and randomness behavior in overcoming obstacles, NonuniformRRT explores the C-Space focusing on the area leading to the Goal-Node, resulting in a more efficient search;

and (2) The path is expected to be smoother since the tree grows toward the Goal-Node from the Initial-Node. This would result in an effective and multiple movements of joints toward the goal as will be demonstrated in Chapter 4. On the other hand, due to its biasness, the NonuniformRRT may have a lower possibility of success compared with the basic RRT, especially when there are too many obstacles between the initial and the goal nodes. The limitations of NonuniformRRT are addressed in the Section 3.3.3. The shape parameters of NonuniformRRT are fully explained in Section 3.2.2.1.

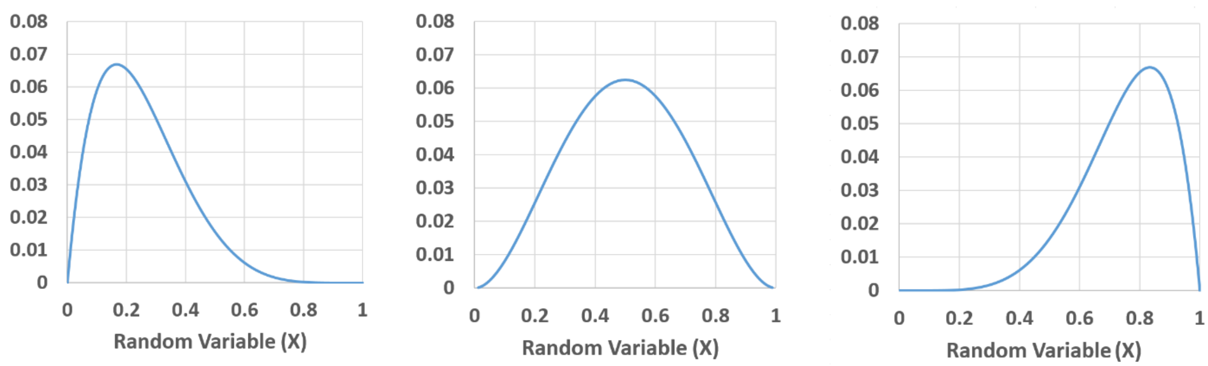
3.3.2.1 Shape Parameters of NonuniformRRT

In order to create a biased tree for the NonuniformRRT as shown in Fig. 3.6(b), a new random number generator is required to generate biased random nodes around the Goal-Node. There are many probability density functions (PDF) available for different engineering purposes. However, to generate biased random nodes in the C-Space, the Beta PDF is selected in this research since it can be defined in a finite range, which is compatible with the finite range of each DoF, and the random numbers can be generated with the desirable biasness according to its parameters. Equations 3.5 and 3.6 represent the standard Beta PDF in the range of [0, 1] and the associated expected value, respectively (Montgomery and Runger, 2014):

$$PDF(x) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha,\beta)} \tag{3.5}$$

$$E(x) = \frac{\alpha}{\alpha+\beta} \tag{3.6}$$

where α and β are positive values of the shape parameters and x is the random variable. The PDF shape parameters determine the shape of the Beta PDF and its biasness. For example, $\alpha = \beta$ would result in a PDF with an expected value of 0.5 as shown in Fig. 3.7(b). The application of this type of PDF results in a non-uniform generation of random numbers concentrated in the middle. However, when $\alpha < \beta$ as shown in Fig. 3.7(a), the PDF leans toward the lower limit of the range [0, 1], and it leans toward the upper limit of that range when $\alpha > \beta$, as in Fig. 3.7(c). The idea is to generate random numbers with the expected value of the Goal-Node.



(a) $\alpha = 2, \beta = 6$ (b) $\alpha = 3, \beta = 3$ (c) $\alpha = 6, \beta = 2$

Figure 3.7 Effect of shape parameters on the Beta probability density function

Fig. 3.8 represents the Beta PDF is used for each DoF in a 2D C-space example in order to create random nodes (shown in small blue dots). The most desirable Beta PDF is the one with the expected value (Mean of the PDF) resulting in the generation of the coordinate values which fit to the Goal-Node, as it assures the guidance of the tree toward the Goal-Node. Therefore, a set of appropriate shape parameters, α and β, have to be determined for the i^{th} DoF using the following equation:

$$E(x) = \frac{\alpha}{\alpha + \beta} = \frac{\theta_i^{Goal} - \theta_i^L}{\theta_i^U - \theta_i^L} \tag{3.7}$$

where θ_i^{Goal}, θ_i^U, and θ_i^L are the goal, upper, and lower limits of the i^{th} DoF, respectively.

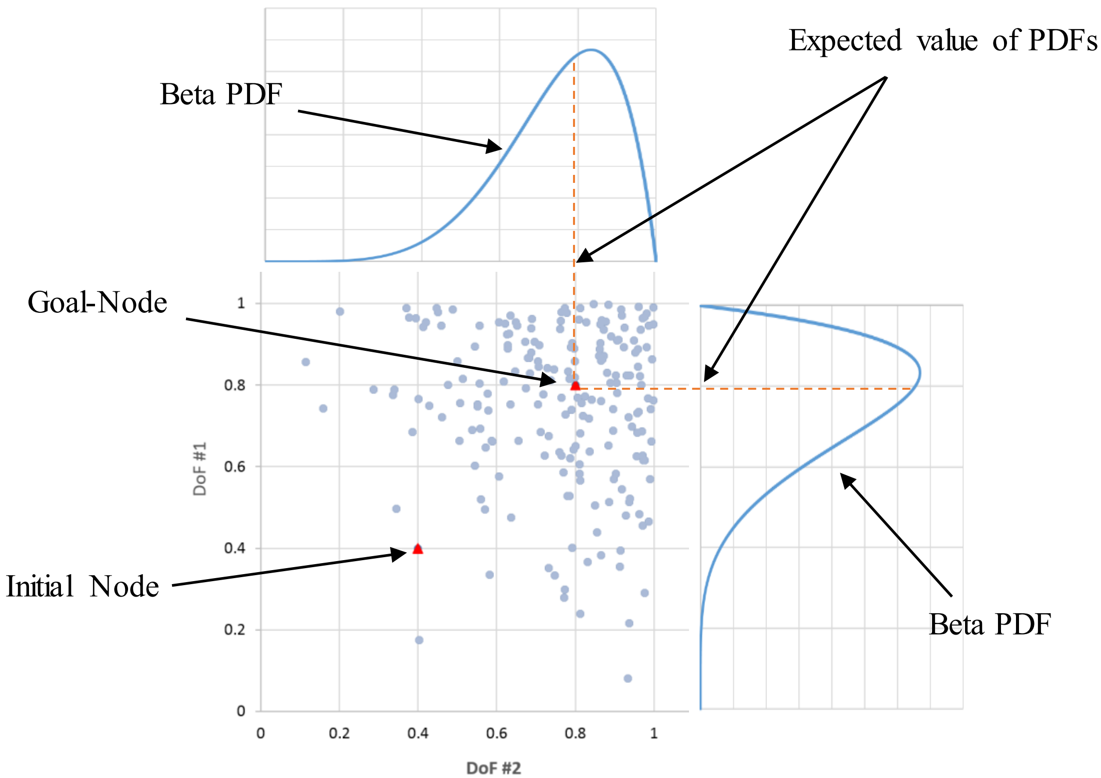


Figure 3.8 The expected value of Beta PDF fit to the goal value

Depending on whether the coordinate value of the Goal-Node is bigger or smaller than the middle of the range of values for a specific DoF, a pair of α and β can be determined to result in a PDF with an expected value matching the Goal-Node. Therefore, α and β can be determined for each DoF in each subtask to match the biasness of the Beta PDF with the goal of the subtask as in the following pseudo code:

If $(\theta_i^{Goal} < \theta_i^{middle})$

$$\beta_i = \frac{\theta_i^U - \theta_i^G}{\theta_i^G - \theta_i^L} \text{ and } \alpha_i = 1 \; (\alpha_i < \beta_i)$$

Else

$$\alpha_i = \frac{\theta_i^G - \theta_i^L}{\theta_i^U - \theta_i^G} \text{ and } \beta_i = 1 \; (\alpha_i \geq \beta_i)$$

End

where θ_i^{middle} is the average value of the i^{th} DoF ($\theta_i^{middle} = \frac{\theta_i^U - \theta_i^L}{2}$). In the above pseudo code, the equation of θ_i^{Goal} assures that the random number generator creates numbers with an expected value matching the Goal-Node. However, since there are two parameters, the results are going to be an infinite set of α and β pairs. For the sake of simplicity, the smaller parameter is assumed equal to one and the larger parameter can be calculated based on that. However, the higher values of α and β will result in lower variance of the randomly generated numbers, which directly affects the dispersion of the nodes, and consequently the shape of the tree. More disperse nodes means more flexibility for the tree to grow, especially when the environment is full of obstacles, and vice versa.

3.3.3 NonuniformRRT+

Fig. 3.9 represents the flowchart of the new improved version of NonuniformRRT (Called NonuniformRRT+), which benefits from other algorithms including UniformRRT, ExtendTowardGoal, Greediness and Perpendicular extension. At the initial stage of the search, UniformRRT, which uses a uniform random number generator, starts extending the tree. As discussed in Section 3.3.2, this will improve the deficiency of NonuniformRRT at the initial stage in avoiding obstacles. After a certain user-defined limit, the algorithm switches to NonuniformRRT mode. ExtendTowardGoal algorithm, explained in Section 3.3.3.2, is always active during the search. At any stage of the search, the tree extends toward the Goal-Node when there is an obstacle-free straight line between the current node and the Goal-Node. Perpendicular extension of the tree is used to avoid collisions when needed and improve the performance as explained in Section 3.3.3.3. As explained in Section 3.3.3.4 and 3.3.3.5, the greediness and a path smoothing method are adapted to increase the quality of the generated path. NonuniformRRT (Langari and Hammad, 2015) fails to generate a fast result in problem such as scenario I (explained in Section 4.2.2 (a)) where the pipes are in the way to the goal configuration and located very close to initial configuration. However, the new NonuniformRRT+ is able to improve that limitation.

3.3.3.1 Overcoming the Limitation of Initial Stage Search

As mentioned in the Section 3.3.2, the NonuniformRRT has limitations in dealing with specific types of problems where the obstacles are too close to the Initial-Node, such as the case shown in Fig. 3.6(b). In such cases, the number of collision checks increases greatly, which results in a relatively inefficient computation effort.

The triangles in Fig. 3.6(b) schematically represent the growth of the tree during the search. The abovementioned limitation can be visualized by adding an obstacle near the head of the triangle and very close to the Initial-Node. Then it would be very hard for the path planner to find a feasible path. To improve this limitation, the new NonuniformRRT+ is benefitting from a heuristic. In the initial stage of the search, the tree utilizes the regular random number generator (UniformRRT). However, when the path planning passes its initial stage, it uses the non-uniform random number generator for the rest of the search. This initial stage of the search can be translated to a number of user-defined limit of S, as shown in Fig. 3.9. This rule increases the capability of the algorithm to overcome the obstacles that are too close to the Initial-Node.

3.3.3.2 Extend Toward Goal Node

A direct movement of all DoFs toward the Goal-Node is desirable, as it is the shortest and smoothest path in the C-Space. At any stage of search process, NonuniformRRT+ tries to choose the Goal-Node with a probability less than a certain value (PROB). This value can be defined by the user and will add another type of biasness to the algorithm (Albahnassi and Hammad, 2011). The tree continues extending itself until reaching Goal-Node or an obstacle is found, as shown in Fig. 3.9 In the case of an obstacle, the algorithm tries to find another way to avoid this obstacle using UniformRRT or NonuniformRRT. In other words, at any stage of the search, the tree would be connected to the Goal-Node when there is no obstacle in between. This is expected to result in a faster and more optimal solution.

3.3.3.3 Perpendicular Extension

As the NonuniformRRT algorithm makes the whole tree biased toward the goal, it could be harder to move around obstacles. When the extension of the tree is repeatedly rejected due to collisions more than *T* times specified by user, it means that the tree is trapped behind the obstacles in the C-space. In order to increase the flexibility of the tree when it faces obstacles, a new modification is applied to the algorithm (called Perpendicular Extension), which is also shown in Fig. 3.9.

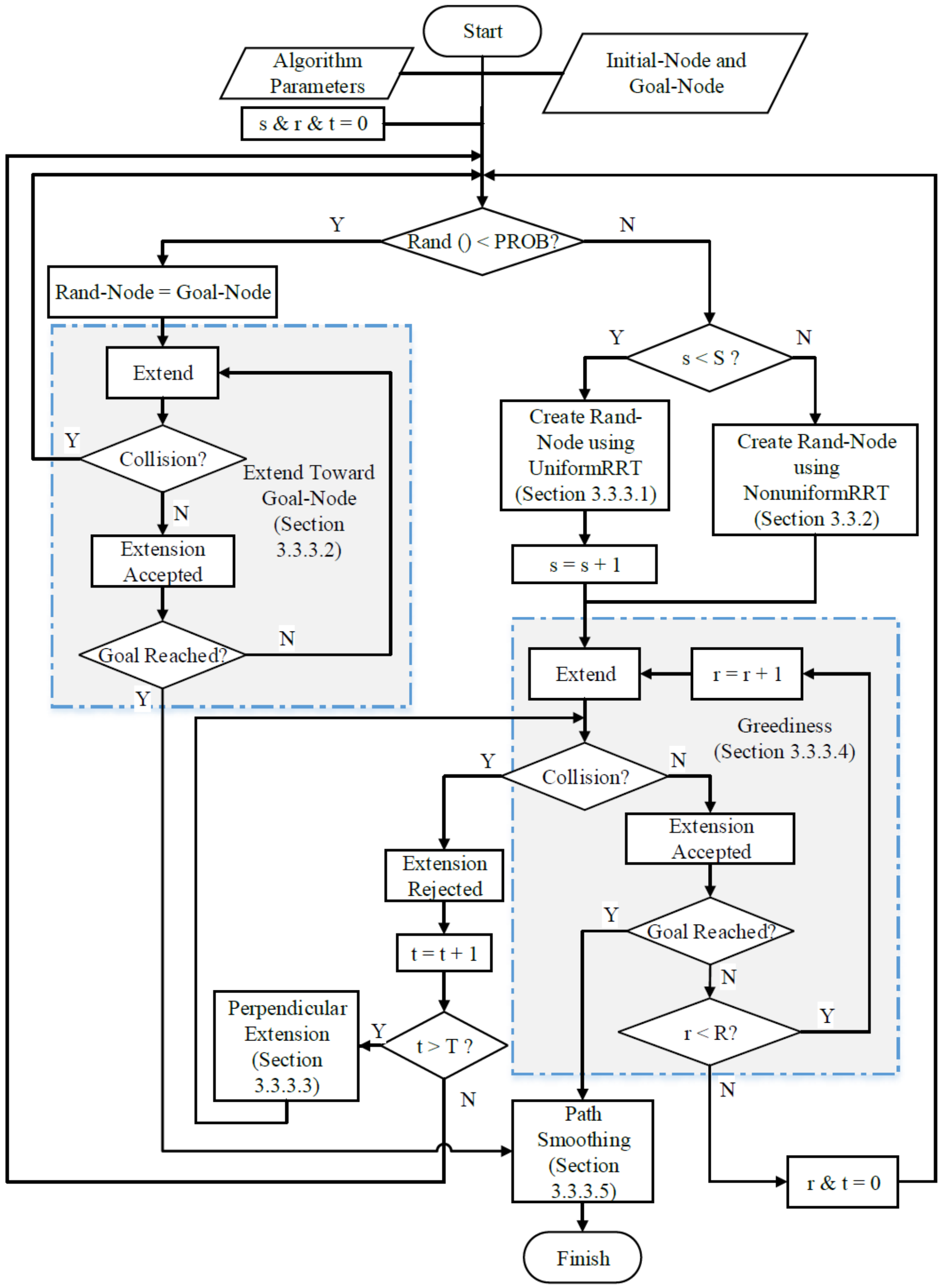


Figure 3.9 Flowchart of NonuniformRRT+

In general, regular extension is used to expand the tree during the search, however when this method fails to grow the tree, the Perpendicular Extension is used to increase the chance of overcoming the obstacles and reducing the computation time by adding the new node to the tree on a line perpendicular to the line passing through the Chosen-Node and the Rand-Node as shown in Fig. 3.10. A tree in the C-space is represented in this figure, which is biased toward the Goal-Node due to the non-uniform generation of random numbers. The Initial-Node and Goal-Node are represented by stars, and the obstacle by an irregular shape. The random node is shown using a grey dot, and the closest node on the tree is shown using a black dot. As shown in the Fig. 3.10 the tree is blocked by the obstacle, however it can be extended using Perpendicular Extension.

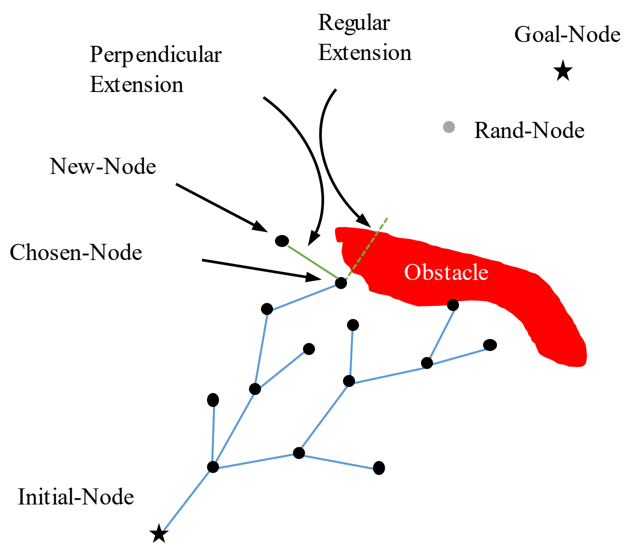


Figure 3.10 2D example of C-space

3.3.3.4 Greediness

In order to improve both the quality of the path and computation time, greediness is used

(Albahnassi, 2010; Lin et al., 2014b). Instead of one step extension, the tree may grow as up to user-defined parameter for greediness (R) as shown in Fig. 3.9.

3.3.3.5 Path Smoothing

When the path is generated, still there may be some jaggedness on the path. Post-processing is required to smoothen this jaggedness, which is shown in Fig. 3.9. The following pseudo code represents the path smoothing:

```
PathSmoothing (Path)
NewPath.Node.Add(Node(1));
   For (s=1 to (Path.Node.Count-1); s++)
        For (k=Path.Node.Count to s+1; k--)
                If (ε < Distance (Path.Node (s), Path.Node (k)) < C × ϵ)
                        NewPath.Node.Add (Node (k)); ExitLoop(k);
                Else if (k==s+1)
                        NewPath.Node.Add (Node (s+1)); ExitLoop(k);
                End
        End
   End
Return NewPath;
```

As shown in the pseudo code, for any chosen node on the path, the algorithm looks for any heretofore node that is within a distance range of ε to $C \times \epsilon$. The coefficient C has to be given by the user, which ranges from 1 to 2. In this research, we used $C = 1.5$ based on trying different values. This post processing adjustment to the path may result in a shortcut and a shorter path at the end. The collision detection is not needed for these shortcuts as the distance between nodes is very small. This technique is represented in Fig. 3.11, which is expected to improve the quality of the path both in terms of smoothness and length. The shortcut dot-line constitutes a new path, which avoids the unnecessary movement of the excavator.

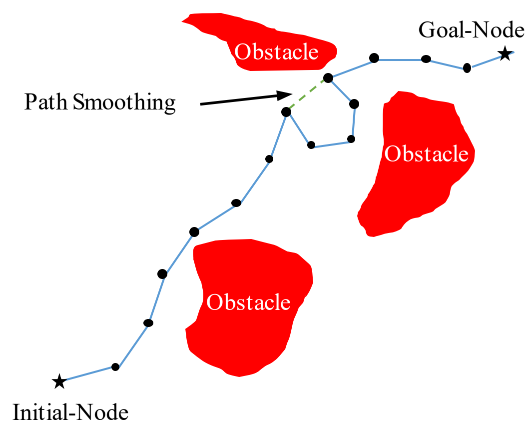


Figure 3.11 Path smoothing

3.3.4 Path evaluation metrics

Metrics are necessary to evaluate the performance of the proposed method. According to previous research, metrics can be classified into two main classes, *computation efforts* (effort metrics), and *quality of the path* (quality metrics) (Zhang and Hammad, 2012; Lin et al., 2014b). Quality metrics evaluate the algorithm based on the roughness and the length of the generated path. A smooth path is also expected to be energy efficient. Besides, a desirable algorithm has to be computationally efficient.

3.3.4.1 Quality Metrics

(a) *Path Roughness*: As discussed before, RRT-based path planners usually generate a jagged path in the C-Space, which is the result of its probabilistic nature. This would lead to an unnecessary, inefficient and even impractical movement of the excavator; and consequently, it will result in a lower productivity. Zhang and Hammad, (2012) have improved the quality of the path. However, their metric does not represent the jaggedness of the path. In this research, a new metric is proposed to measure this specific quality of the path. The assumption is that a smooth movement of the equipment is the result of a linear relationship between any three consecutive nodes on the path. Fig. 3.12 represents a 2D C-Space with three consecutive nodes in three different hypothetical paths. The path with $q"(i)$ as the middle node is the most undesirable path as it results in a unsmooth movement of DoFs. The path with $q'(i)$ as the middle node represents a better path compared with the previous one. The path with $q(i)$ in the middle is the ideal path that would result in a smooth movement of the equipment. In the case of the last path, the variations are linear resulting in the following equation:

$$\frac{\theta_i^j - \theta_i^{j-1}}{\sqrt{\sum_{k=1}^{m}(\theta_k^{j-1} - \theta_k^j)^2}} = \frac{\theta_i^{j+1} - \theta_i^j}{\sqrt{\sum_{k=1}^{m}(\theta_k^{j+1} - \theta_k^j)^2}} \tag{3.8}$$

where m is the number of the DoFs of the equipment. θ_i^{j-1}, θ_i^j and θ_i^{j+1} are the coordinates of the i^{th} DoF of three consecutive nodes $j-1$, j and $j+1$. Any variation from the linear composition of nodes in the C-space will result in an undesirable motion, which is called in this research, *path roughness*. Given *n* as the number of nodes, the path roughness is represented by the following equation:

$$R = \sum_{i=1}^{m}\sum_{j=2}^{n-1} \left| \frac{\theta_i^j - \theta_i^{j-1}}{\sqrt{\sum_{k=1}^{m}(\theta_k^{j-1} - \theta_k^j)^2}} - \frac{\theta_i^{j+1} - \theta_i^j}{\sqrt{\sum_{k=1}^{m}(\theta_k^{j+1} - \theta_k^j)^2}} \right| \tag{3.9}$$

(b) *Length of the path*: Fig. 3.13 represents a 2D C-Space and two paths of the same length. However one path is jagged and rough. Therefore it is important to differentiate between the roughness and the length of the path. In a simple 3D movement of an object, the length of path can easily be calculated as the sum of all relocations along the path. This length can be defined both in 3D-space and C-Space.

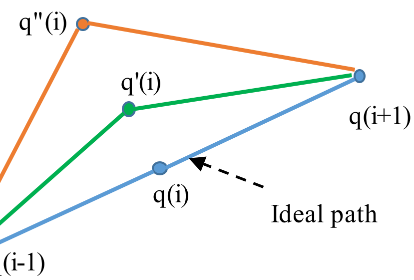


Figure 3.12 Segments of three different paths with different path roughness values

Lin et al. (2014b) considered the movement of the load for a crawler crane as a metric for measuring the path length. Their formulation is designed to measure the length of the path for a crawler crane, which is the movement of the crawler crane's end-effector. In this research, the same concept of end-effector's movement is used to propose the following formulation for the excavator:

$$L = \sum_{i=1}^{m} \sum_{j=1}^{n-1} r_i |\theta_i^{j+1} - \theta_i^j| \tag{3.10}$$

where r_i is the effective arm for that rotational joint. It should be noted that r_i is a function of the overall configuration of the excavator. Fig. 3.14 represents the effective arm in a specific configuration of the excavator. L_i is the length of the excavator's part, which can be determined from the equipment specifications. Given the length of equipment parts, the effective arms r_i can then be calculated using a simple equation for specific values of θ_i (Appendix A).

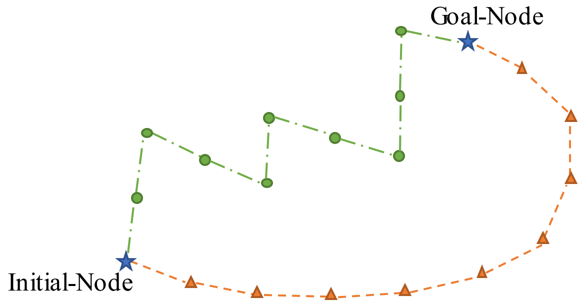


Figure 3.13 Two paths with the same length but different path roughness values

3.3.4.2 Effort Metrics

(a) $Time$: The time it takes for the algorithm to find the collision-free path is the primary metric for computation efforts. This is a desirable metric when comparing two searching algorithms. However, there is another metric that can capture the computation efforts independent from the central processing unit (CPU) and other system specifications, which is the number of nodes in the tree.

(b) The *number of rejected nodes:* This is the number of nodes that could not be added to the tree due to collisions. This is another metric that shows computation efforts since collision detection requires considerable computation efforts (Lin et al., 2014b). The Number of rejected nodes can help to reveal the capabilities and limitations of the algorithm in dealing with obstacles.

Figure 3.14 Effective arms for length of the path

3.4 Proposed Motion Planning Method

3.4.1 Motion planning

Assume a sample-based path planner such as RRT generates a collision-free path for an agent in a 2D environment. The path consists of a series of consecutive poses, which guarantees a safe relocation of the agent from the Initial-pose to Goal-pose in the 2D space. The maximum distance between these poses is ε, which is usually specified by the user at the initial setup of the planner. In the context of this research, the relocation of the agent from on pose to the next pose is called STEP. In other words, the path is a series of those consecutive STEPs.

Fig. 3.15 represents the results of a 2D example of motion planning problem if the nodes are spaced equally in time. The horizontal axis is the time and the vertical axis represents the DoFs. The first DoF is shown by lines with circle dots and the second one is shown by lines with square dots. If the agent follows the proposed consecutive poses (called Intermediate-Nodes in the C-space), it would reach to the Goal-Node safely. However, it ignores the time it can take to reach to the goal pose, which depends on the kinematic properties (such as speed) of the object in the problem. In the case of construction equipment, usually there are speed constraints, for example the excavator cannot exceed a certain value of rotational speed in swinging.

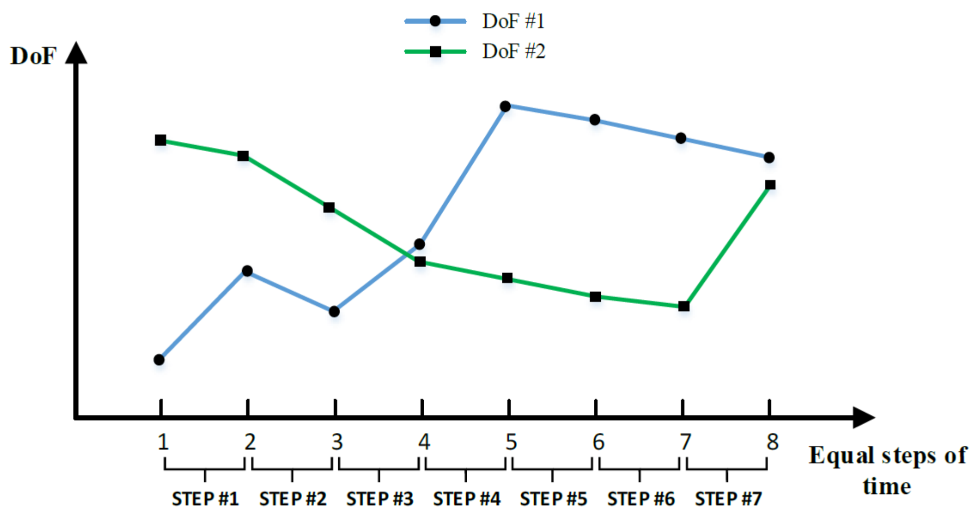


Figure 3.15 The path generated by the path planner for a 2D C-space example

Going back to the 2D example of Fig. 3.15, for the sake of simplicity, one value is considered as the maximum speed for both DoF #1 and DoF #2. This maximum speed has to be reflected in the motion planning of the object. The example of Fig. 3.15 represents these STEPs with the same rate in time, for example in the STEP #1 (represents the transition from pose #1 to pose #2) the agent has a critical movement in DoF #1 direction, as the movement is bigger than the movement in DoF #2 direction. Then in the motion planning phase, the maximum velocity must be taken into account for DoF #1. This will result in a time adjustment in STEP 1 for which the DoF #2 has to be adjusted accordingly. The same principle has to be applied for the subsequent STEPS. Sometimes the critical DoF is DoF #2, such as in the last STEP.

Fig. 3.16 represents the adjusted motion plan according to the maximum speed constraints of the

DoFs. The horizontal axis schematically represents the time, and the spacing between nodes 1 and 2 has increased to satisfy the maximum speed constraint of DoF #1. On the other hand, the spacing between poses 5 to 7 is compressed to minimize the total time of the path. In order to maintain the same path during the adjustment process, the DoF #2 is adjusted according to the adjustment of DoF #1, as explained in Section 3.4.3. In order to assure that the maximum velocity of DoF #2 is respected, the spacing between pose 7 to 8 is extended. It should be noted that the total adjusted time in the Fig. 3.16 is expected to be shorter than the total time in Fig. 3.15.

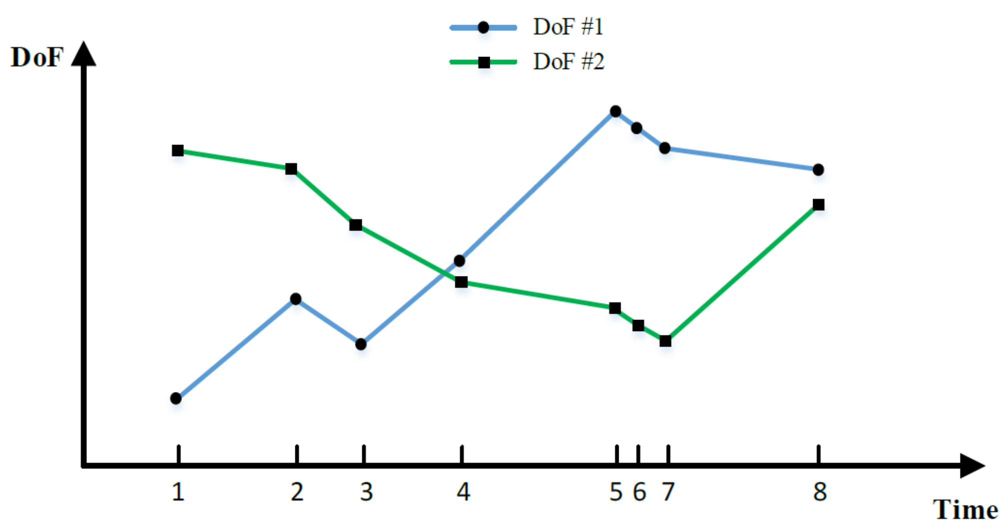


Figure 3.16 Adjusted path according to the time

3.4.2 Excavator cycle-time

In the context of agent-based simulation, the execution time must be added to the computation time to reflect the realistic time that a task is consuming.

$$T_{Cycle} = T_{Execution} + T_{Computation} \tag{3.11}$$

where $T_{Computation}$ is the time required for the path planner to generate a path, and it depends on the efficiency of the planner and the computational power of the computer. Because agents work independently in the simulation process, one cannot wait for the planning phase of another agent.

A full cycle-time of an excavator can be determined by summing up the time required to accomplish each of sub-tasks of *digging*, *moving to dump*, *dumping* and *returning to next dig*.

$$T_{Execution} = T_{digging} + T_{moving\ to\ dump} + T_{dumping} + T_{returning\ to\ next\ dig} \tag{3.12}$$

A subtask execution time is the summation of times required to follow the nodes in the path of that subtask, as expressed in Equation 3.13.

$$T_{subtask} = \sum_{k=0}^{K-1} \max(t_1^k, t_2^k, \dots, t_N^k) \tag{3.13}$$

where K and N represent the number of nodes in the generated path and the total number of DoFs, respectively. t_i^k is the execution time between node k to $k+1$ for the i^{th} DoF, which depends on the maximum rotational velocity of that joint of the excavator and the angular distance between the nodes for each DoF. Then Equation 3.11 can be re-written in the following form.

$$T_{subtask} = \sum_{k=0}^{K-1} \max\left(\frac{|\theta_1^{k-1} - \theta_1^k|}{\omega_1}, \frac{|\theta_2^{k-1} - \theta_2^k|}{\omega_2}, \dots, \frac{|\theta_N^{k-1} - \theta_N^k|}{\omega_N}\right) \tag{3.14}$$

where θ_i^j represents the value of i^{th} DoF in the j^{th} node of the path, and ω_i represents the rotational velocity of the i^{th} DoF.

3.4.3 Linear adjustment of the path after motion planning

A path generated by RRT is composed of multiple consecutive nodes, which are connected with a line in the n-dimensional C-space. This line represents a pose transition of the equipment, called STEP, as explained in Section 3.4.1. This value is usually the same as the ε value, which is a parameter of RRT provided by the user. This means that between any of those nodes, the motion is assumed to be linear. However, in order to assure safety, it is required to guarantee that the path will be the same after the motion planning process.

Assume t_{Max}^{k} is the maximum values of the time required for the STEP k between all DoFs. It means that this STEP cannot be done faster than this.

$$t_{Max}^{k} = \max\left(t_1^k, t_2^k, \ldots, t_N^k\right) = \max\left(\frac{\left|\theta_1^{k-1} - \theta_1^k\right|}{\omega_1}, \frac{\left|\theta_2^{k-1} - \theta_2^k\right|}{\omega_2}, \ldots, \frac{\left|\theta_N^{k-1} - \theta_N^k\right|}{\omega_N}\right) \qquad (3.15)$$

Then the rotation velocities of the rest of DoFs have to be corrected accordingly, as expressed in Equation 3.16.

$$^{adj}\omega_i^k = \frac{\theta_i^{k-1} - \theta_i^k}{t_{Max}^k} \qquad for\ (i \neq Max) \qquad (3.16)$$

where $^{adj}\omega_i^k$ is the adjusted rotational velocity of i^{th} DoFs between nodes $k-1$ and k, which assures the path would remain the same after motion planning process. As shown in Figs. 3.14 and 3.15, the same adjustment is happening in those figures, which does not change the path but affects the motion by extension in some parts and compression in other parts of the path. Still there are other engineering constraints that affect the motion planning of the excavator, which will be explained in Section 3.4.4.

3.4.4 Coupled motion effect

As explained before, the upper-structure of a typical hydraulic excavator has four DoFs, which correspond to upper-structure swing, boom, stick and bucket rotation. These rotations are supported by four independent actuators, however there are only two hydraulic pumps that support these four actuators. The swing and the stick movements are supported by one hydraulic pump and the other pump empowers the boom and bucket movements (Rowe, 1999). In some hydraulic excavators three hydraulic pumps are devised to increase the productivity but they are limited to very large pieces of equipment (Bennink, 2011).

Rowe (1999) conducted some experiments on the motion of the hydraulic excavator. The results show that the independent motion of joints can be done in a shorter time than when the coupled motion is applied. For example, a 100° swing of an excavator takes 6 seconds, however it takes 11 seconds if this happens simultaneously with a 90° of stick rotation as they share the same hydraulic pump. The effect of coupled motion is worst for the case of the simultaneous swing and boom rotation. The hydraulic pressure moves along the least resistance direction, and as the result, it takes 4.5 seconds for boom sweep of 10° in the coupled motion, which is three times longer than the time it takes for independent boom rotation of 10°(almost 1.5 seconds). So there are two rotational velocities for each DoF of a hydraulic excavator, ω_{ic}, which represents the maximum rotational velocity of i^{th} DoF in the coupled motion and ω_i, which is the maximum rotational velocity of i^{th} DoF, then:

$$\omega_{ic} \leq \omega_i \tag{3.17}$$

In order to calculate an accurate execution time, it is needed to use ω_{ic} instead of ω_i when the motion is coupled. The effect of coupled motion is outside the scope of this research.

3.5 Embedding the proposed path planning into the macro planning

Our previous research introduced the concept of agents that can support the excavator and truck operators (Hammad et al., 2013). Fig. 3.18 shows the flowchart of the excavator. It is assumed that the excavation site is divided into work areas (WAs) where the excavator is supposed to dig the earth. It is not possible to excavate the whole amount of earth in a WA from one position of the excavator and the WA should be excavated based on several strips. When digging is done in the first WP, the excavator has to move backward to excavate the next strip (See Fig. 3.17).

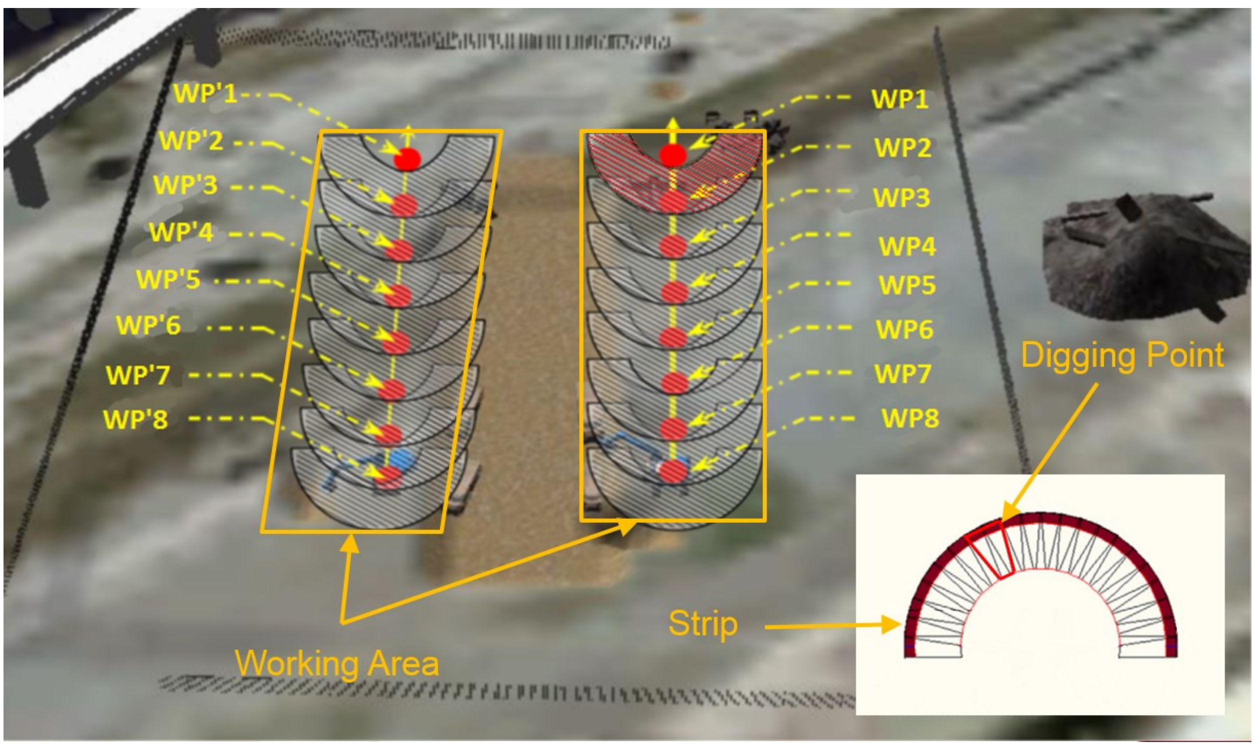


Figure 3.17 Terms used in Multi-Agent System (Adapted from Vahdatikhaki, 2015)

To move backward, a path planning algorithm (e.g., A*) is used. When excavation is done in one WA, the excavator moves to the next area using the A* algorithm to ensure avoiding dynamic and static obstacles. At each WP, the excavator regularly utilizes a parametric script for digging the earth and dumping it into a truck. However, in case of any potential collisions with other equipment or obstacles, the parametric script is not reliable. Therefore, the path planning is done using a more sophisticated algorithm such as RRT (LaValle, 1998).

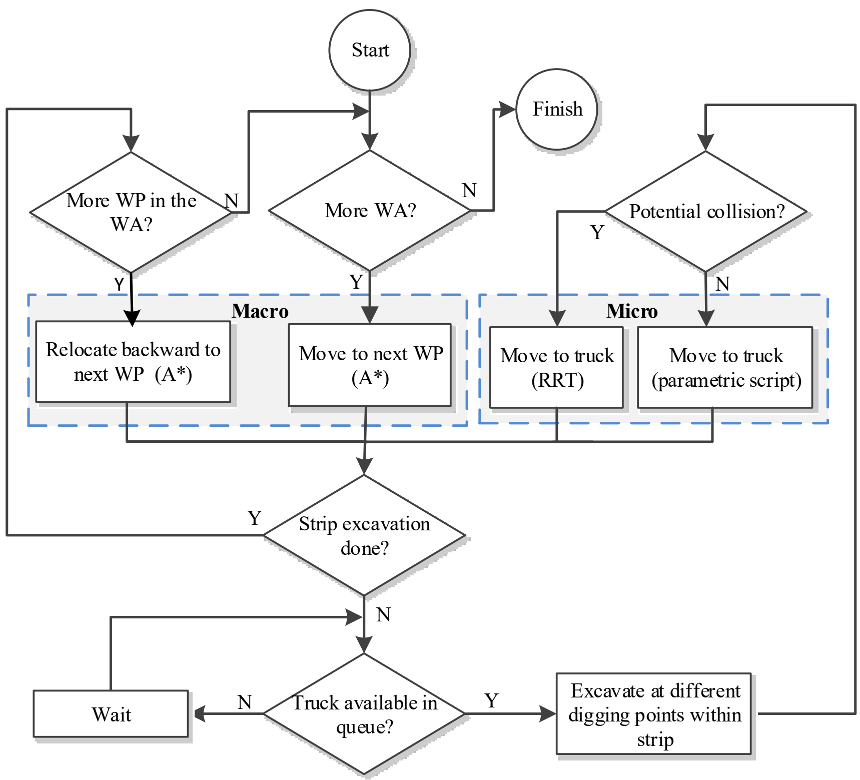


Figure 3.18 Excavator agent flowchart

To accomplish the earthmoving operation, the collaboration of haulers with the excavators is essential. Usually the back cycle-time of the truck is longer than the excavator's service time. Therefore, to achieve the equipment fleet's optimal productivity, several trucks are served by each excavator and the truck may need to wait in a queue before being served. Fig. 3.19 illustrates how a truck works to achieve its duty in the fleet. When there is more than one fleet of equipment, a truck has to select the excavator that has the shortest queue. The truck can also uses A* to avoid its obstacles when hauling to dump and returning. Then truck will wait until it is completely filled with material. Then it will find its optimal way to the dumping area using A*. After each dumping, the truck will find the shortest queue among the active excavators if available.

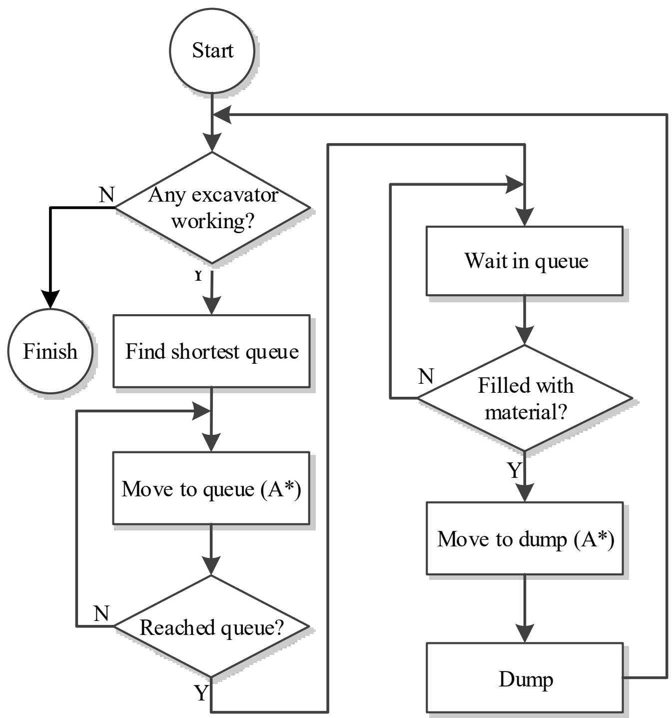


Figure 3.19 Truck agent flowchart

3.6 Summary and Conclusions

In the first part of this chapter, an advanced RRT-based path planner is proposed for cyclic operation of excavators including moving to dump, dumping and moving to next dig point. NonuniformRRT is the core of the presented algorithm, which benefits from the non-uniform random number generator instead of the uniform random number generator. NonuniformRRT improves the performance of the planner by limiting the search space and making the tree biased toward the Goal-Node. Besides, the generated path resulted from NonuniformRRT is probabilistically more inclined toward Goal-Node, which is an improved path in terms of quality. However, preliminary results show that it has limitations dealing with the cases in which the obstacles are between the Initial-Node and the Goal-Node and very close to the Initial-Node. This is expected as in the initial stages of search, the tree is trapped behind the obstacles and NonuniformRRT keeps pushing it toward the Goal-Node.

With many modifications, NonuniformRRT+ is introduced to address the limitations of NonuniformRRT, which increase the agility of the planner and the quality of the path. NonUniformRRT+ benefits from six heuristics in order to improve the quality of the path and to satisfy the engineering constraints (Langari and Hammad, 2015): (1) instead of the uniform random number generator, it uses a non-uniform random number generator. These non-uniform random number generator generates random numbers with the expected value of the Goal-Node for each DoF. The beta probability density function is used for this purpose. (2) At the initial stage of the path finding process, it benefits from the conventional RRT to avoid being trapped nearby the obstacles. (3) It benefits from ExtendTowardGoalNode heuristic, which makes it capable to connect to the goal node if there is no obstacle between the current node and the Goal-Node. This reduces the computation time and improve the quality of the path. (4) For cases when the regular extension of tree cannot overcome obstacles, a new extension method is proposed, called Perpendicular Extension, which allows the algorithm to avoid the obstacles and decreases the number of collision checks. (5) Another type of greediness is added to the algorithm to choose the Goal-Node as the random node with a user-defined chance. (6) At the end, the NonuniformRRT+ uses a path smoothing algorithm in order to improve the quality of the generated path.

In the second part of this chapter, a new concept is proposed in order to adjust the path generated by the planner according the time. The method is expected to maximize the productivity of the hydraulic excavator. This chapter finishes with a simulation model for excavation projects considering excavator and truck agents, which are extensively used in construction projects. Benefitting from the proposed path planning algorithm, the proposed model overcomes the limitations of previous simulation models at macro and micro path planning levels. A* is used for the macro path planning where the agents decides to relocate from one working area to another one, for example, the excavator moves back from one digging point to the next digging point. A sophisticated RRT-based path planner is proposed to handle the excavator operations as it is able to avoid the obstacles such as workers, etc.

4. IMPLEMENTATION AND CASE STUDIES

4.1 Introduction

This Chapter is arranged in the following form: Section 4.2 introduces the evolution of the proposed path planning method. The proposed algorithm, NonuniformRRT+, is implemented in Unity Game Engine (2015), and the results are shown. The proposed algorithm of Albahnassi (2010), RRTBiasedLimCon, is adapted for excavators in Unity Game Engine. A performance comparison between the adapted RRTBiasedLimCon and NonuniformRRT+ is then presented in Section 4.2.3. In Section 4.2.5, the results of time-adjustment are shown. In Section 4.3, the implementation of Multi-Agent System is presented where four trucks are working with two excavators to accomplish the earthwork. Finally, the application of the path planning method is investigated in other cases in Section 4.3.

4.2 First Case Study: NonuniformRRT and NonuniformRRT+

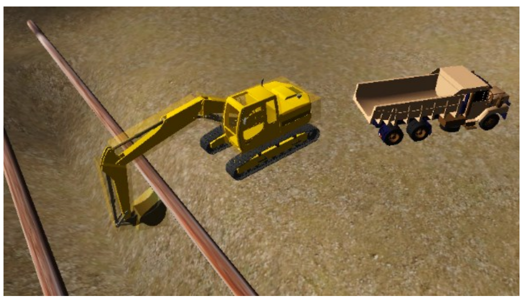
In order to test the proposed method, a real case study is simulated in unity game simulation environment. As shown in Fig. 4.1, the operator has to carefully navigate the bucket between two pipes in order to dig the soil. After digging, the soil should be carried to the dump position. The site is congested with other pieces of equipment, workers, etc. Therefore, the operator has to pay special attention to avoid collisions. For example, when a worker was in the excavator's workspace, the operator brought the stick toward the body of the excavator to avoid collisions.

Figure 4.1 Excavator digging between two pipes in a congested site

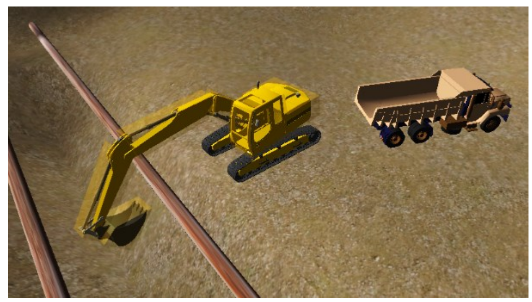
4.2.1 Simulation environment in game engine

As shown in Fig. 4.2 the scenario of Fig. 4.1 is implemented in Unity game engine (Unity game engine, 2015) to test the efficiency of the proposed method and to compare it with RRTBiasedLimCon. The earthwork operation in this simulation is induced from a real-world operation, where the excavator-operator is digging the earth between two pipes. The operation is taking place very carefully to avoid collision between pipes and the excavator. The algorithm considers the pipes and the ground as obstacles. In addition to those, the truck has been added to the scenario as another obstacle to make the situation more challenging, which could reveal the limitations or advantageous of the proposed algorithm.

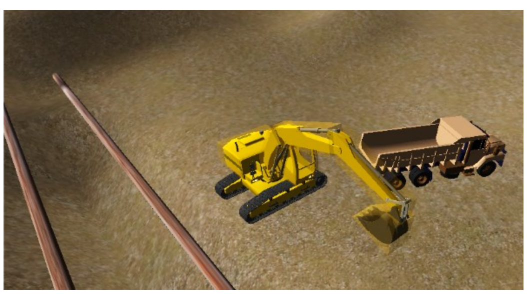
The proposed algorithm is implemented in C# scripts, which are attached to the scene game objects to find the collision-free path for the excavator. In this simulation, the excavator has two models: a visual model (realistic model) and a simplified model. The simplified model is composed of a few basic convex volumes representing the parts of the excavator and is used for path planning calculations. When the path is found, the visual model is used to visualize the motion of the excavator. Using the simplified model to find the path has two main benefits. First, collision detection for this model is easier; hence it greatly reduces the computation efforts. Second, a buffer can be added around the simplified model to conservatively consider safety (Albahnassi and Hammad, 2011). The proposed algorithm is implemented in a way that can be modified to fit to any similar holonomic construction equipment with multiple DoFs (e.g. front shovel). The upper and lower limits of each joint movement have to be defined. Each joint can be selected as a prismatic or rotational type.



(a) Initial configuration



(b) Passing between two pipes



(c) Moving toward the truck

(d) Goal configuration

Figure 4.2 Simulated excavation environment with two pipes and a truck

In addition, the user has to define the maximum number of nodes and ϵ for the path planner. The extension of the tree occurs with ϵ distance step, which is the maximum distance between two consecutive nodes in the C-Space. The value of ϵ depends on the dimensions of the equipment and obstacle size in the scene. As a general rule, a smaller ϵ would result in a safer path, but as it increases the computation effort increases dramatically. The maximum number of nodes is a limit that the algorithm should use to search for a solution.

Given the positions of the excavator and the truck and the digging points, the initial and goal configurations can be determined using inverse kinematics. After digging, when the bucket is full, it has to remain constrained in order to maintain the soil. Considering these issues, the proposed method has been tested in the game environment and it showed satisfactory results in finding a collision-free path. The quantitative and qualitative comparisons between the proposed method and the RRTBiassedLimCon are given in the following sections.

4.2.2 Implementation of NonuniformRRT

In order to get reliable comparison of the evaluation metrics explained in Chapter 3, both the Nonuniform and basic RRT path planners are executed 50 times on a computer with a CPU of 3.40 GHz and memory of 6 GB. Table 4-1 represents the average results of each metric. The results show a significant improvement in computational efforts and the quality of the path.

Table 4-1 Comparison of evaluation metrics of NonuniformRRT and basic RRT

Path planner	Average No. of Nodes	Average run duration (ms)	Average length (m)	Average Smoothness
Basic RRT	716.2	186	1057.5	268.9
Non-uniform RRT	40.0	12	453.3	41.6

More investigation on the performance of NonuniformRRT revealed the limitation of this path planning method. In some cases the performance of the algorithm drops dramatically where the obstacles are between Initial-Node and Goal-Node and very close to Initial-Node. It takes a lot of time to deals these type of problem and the average number of collision detection increases greatly as well. This is expected as the algorithm continuously attempts to extend the tree toward the Goal-Node while it is trapped behind the obstacles. Then none of the extension will be accepted due to the collision.

4.2.3 Comparing performance of NonUniformRRT+ with RRTbiasedLimCon

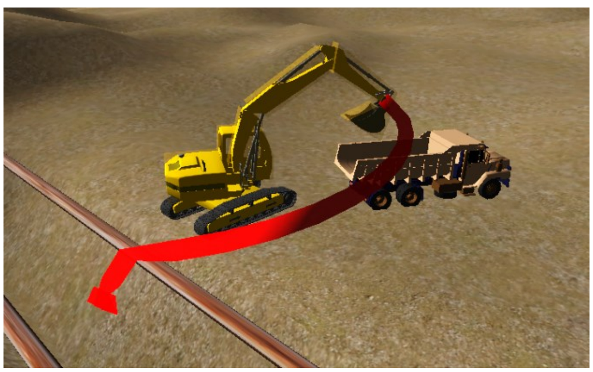
In order to get a reliable comparison of the metrics explained in Section 3.3.4, both NonUniformRRT+ and RRTBiasedLimCon path planners are executed 50 times on a computer with a CPU of 3.40 GHz and memory of 6 GB.

(a) Scenario I (three DoFs test): As shown in Fig. 4.2 the excavator has to dump the earth to the truck's bed when digging is done. As explained in Section 3.3.1 this subtask has three DoFs. Table 4-2 represents the average results of each metric.

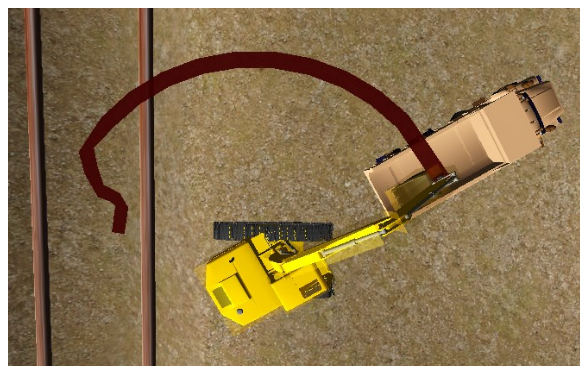
Table 4-2 Comparison of evaluation metrics – Scenario I with three DoFs

Path planner	Ave. path roughness	Ave. length of path (m)	Ave. Time (s)	Ave. No. of Collision Checks
NonUniformRRT+	9.05	31.06	0.68	2273
RRTBiasedLimCon	35.20	36.81	0.81	547
Improvement ratio (%)	74	16	16	N.A.

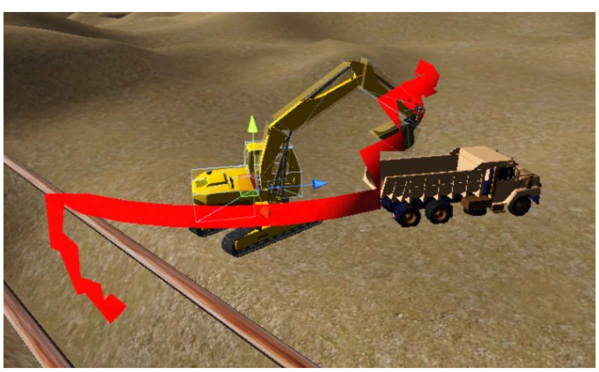
NonUniformRRT+ shows a significant improvement in terms of quality of the path compared with RRTBiasedLimCon. On average, the NonUniformRRT+ results in a shorter path by 16%. It means an increase in productivity and consequently a reduction of fuel consumption. Visual analysis of the results show very close correlation between path generated by the proposed algorithm and the path generated by a human operator. Fig. 4.3 represents the bucket trajectory of the excavator. Figs. 4.3(a) and (b) shows a path generated by NonUniformRRT+ from the side and top views, respectively. For the same path planning problem, the bucket trajectory is shown in Figs. 4.3(c) and (d) for RRTBiasedLimCon.



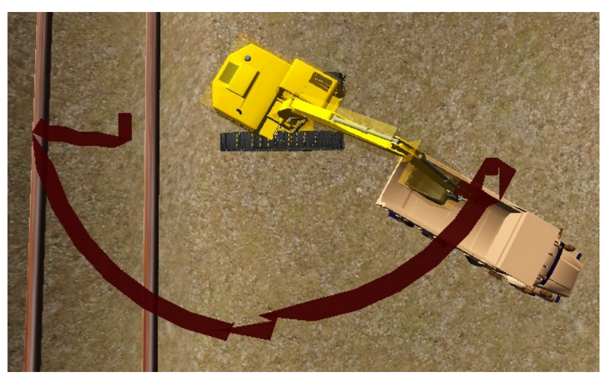
(a) Side-view of a path generated by NonUniformRRT+



(b) Top-view of the same path by NonUniformRRT+



(c) Side-view of a path generated by RRTBiasedLimCon

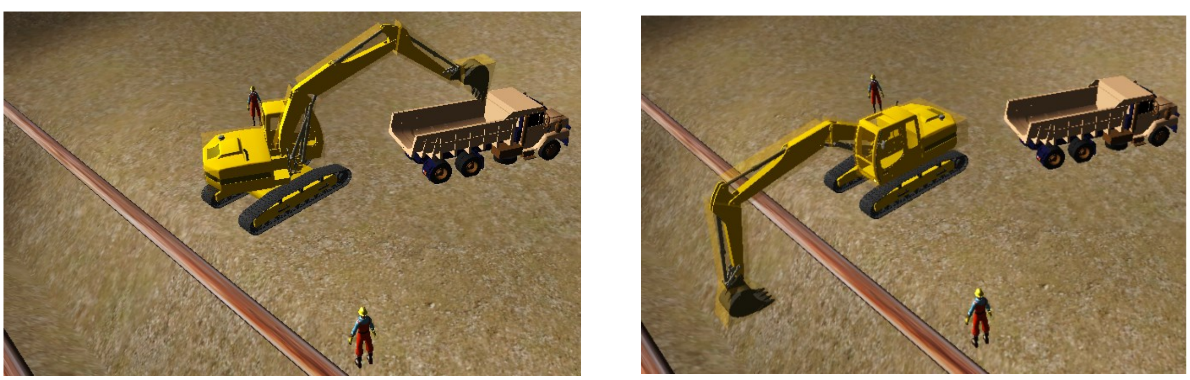


(d) Top-view of the same path by RRTBiasedLimCon

Figure 4.3 Trajectory of the bucket movement

The visual assessment of the results shows that NonUniformRRT+ is capable of generating paths very similar those generated by a human-operator. It avoids obstacles when needed, otherwise a smooth and jagged-free path is followed. The motion of the excavator is jagged-less and smooth. The excavator's joints are moving toward the goal configuration simultaneously, which will result in a less cycle-time and a higher productivity. The proposed approach generates a path 74% smoother than that of RRTBiasedLimCon. Compared with RRTBiasedLimCon, it takes almost 16% less time for NonUniformRRT+ to generate the path, which makes it suitable for near-real time simulation.

(b) Scenario II (four DoFs test): As shown in Fig. 4.4, both algorithms are tested in another scenario where the excavator has already dumped the earth in the truck's bucket and has to go back to the next digging point.



(a) Initial configuration (b) Goal configuration

Figure 4.4 Scenario II: where the excavator is moving to next dig point

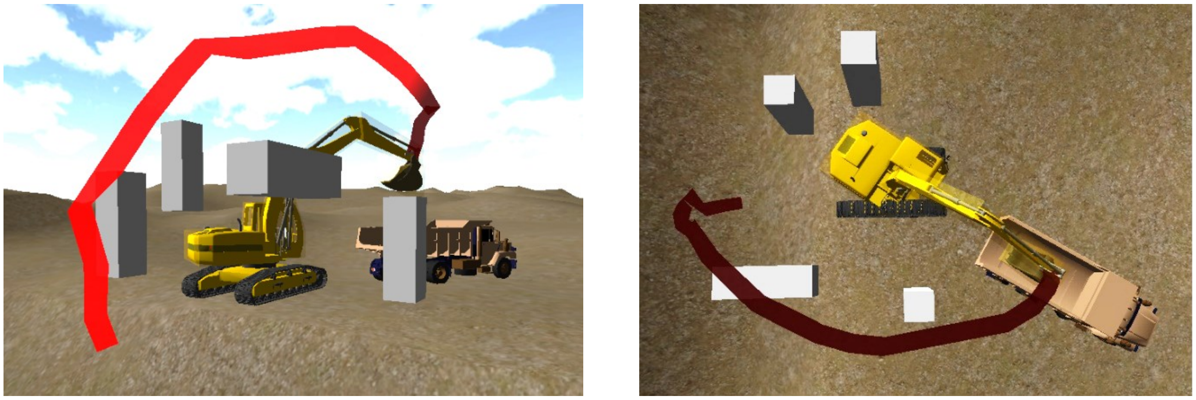
To create a congested construction site, two workers are added as another obstacle to this scenario to make the path planning problem more challenging. The initial and goal configurations are assumed according to the digging plan. As explained in Section 3.3.1, four DoFs of the excavator are engaged in this subtask. The results are represented in the Table 4-3. The proposed method shows improved performance in both computation effort and quality of the path. Compared with RRTBiasedLimCon, the path generated by the proposed method shows 65% and 9% improvement in terms of roughness and length of the path, respectively. In addition, the NonUniformRRT+ takes almost 50% of the time of RRTBiasedLimCon.

4.2.4 Performance of NonuniformRRT+ with more obstacles

NonuniformRRT+ is developed for hydraulic excavator, which repeatedly has to dig the earth and dump it into the truck bed. In Section 4.2, the proposed algorithm is tested in the most complex situation that can happen in construction site by considering workers, pipes and truck as obstacles. However to test the capability of the algorithm to generate a collision-free path, it is also tested in a more complex situation where four relatively big obstacles are irregularly arranged around it. As shown in Fig. 4.5, it is able to generate the collision-free path.

Table 4-3 Comparison of evaluation metrics – Scenario II with four DoFs

Path planner	Ave. path roughness	Ave. length of path (m)	Ave. Time (s)	Ave. No. of Collision Checks
NonuniformRRT+	24.82	41.29	2.12	3145
RRTBiasedLimCon	70.32	45.28	4.18	1063
Improvement ratio (%)	65	9	49	N.A.



(a) Side view (b) Top view

Figure 4.5 Path generated by NonuniformRRT+ in a more complex situation

4.2.5 Motion planning

Referring to the case study developed in Section 4.2 (Scenario I), the proposed motion planning is tested and analyzed. Fig. 4.6 represents the results of a path generated by NonuniformRRT+ for a problem of excavator with three DoFs. The path planning problem is the subtask of "Moving to dump" where the swing, boom and stick movements are active. Before motion planning, the excavator follows that in an equal-time spaced manner; in which the time space equals to the time for the most critical STEP. The time required for all the STEPs is assumed equal to the time required by the joint with the maximum value of $\frac{\Delta\theta_i}{\omega_i}$ (see Equation 3.15). Without any motion planning, this subtask takes 30.20 seconds to be accomplished assuming that the steps are equally spaced in time.

Using the hydraulic excavator's kinematic information in Table 4-4, the motion planning is applied on the same path generated by NonuniformRRT+, which is represented in Fig. 4.7. The subtask takes 14.3 seconds to be accomplished. As a result, the path is stretched in some parts and compressed in other parts to reach the maximum overall speed of the excavator, and this consequently minimizes the time to accomplish the subtask. Based on the assumptions made in this section, the motion planning improves the productivity using the maximum speed of the equipment. The motion planning in Fig. 4.7 reduces the time by 53% compared with the presumed motion plan in Fig. 4.6 (in which STEPs are spaced equally according to the time based on the most critical one).

Table 4-4 Excavator joint speeds (Adapted from Rowe, 1999)

	Independent motion
Swing speed (Deg/Sec)	15
Boom rotation speed (Deg/Sec)	5.5
Stick rotation speed (Deg/Sec)	22.5
Bucket rotation speed (Deg/Sec)	31.71

Table 4-5 represents the cycle-time of the hydraulic excavator with details regarding subtasks, explained in Section 3.3.1, according to the bucket capacity (Peurifoy and Schexnayder, 2011). These numbers are determined based on average conditions such as 30 to 60 degrees for swing to dump subtask, and the excavator and the truck are considered at the same level.

Considering the average case when the bucket size is 3 cubic yard (cy), it takes 5 seconds for the excavator to swing about 45 degrees; based on that it can be presumed that the excavator requires about 15 seconds to do the swing to dump subtask with 135 degrees. This is comparable with the 14.3 second which is the outcome of the motion planning with swing to dump subtask of 150 degrees.

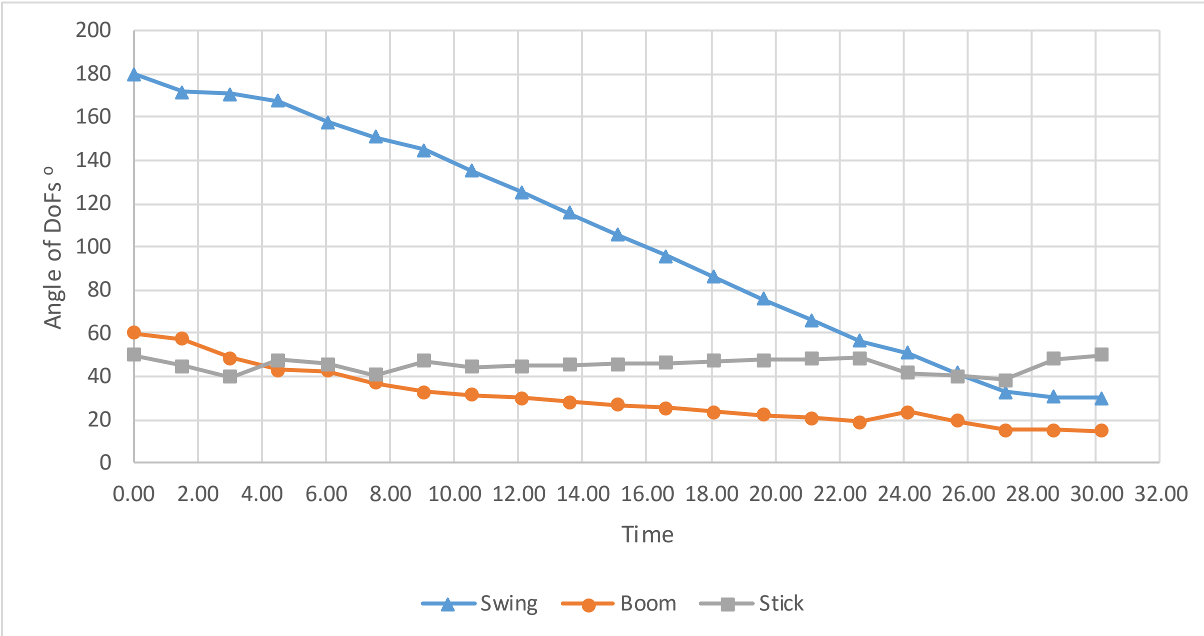


Figure 4.6 The equal-timely spaced path generated by NonuniformRRT+

4.2.5.1 A typical motion vs. motion planning of excavators

Fig. 4.8 represents a schematic and typical motion of an excavator, which is inspired from parameterized path planning (Stentz et al., 1999). The only obstacle is the truck for this path planner and it may fail to generate a collision-free path in other cases when there are other objects around the excavator. As shown in Fig. 4.8, the motion includes loading, swing to truck and dumping subtasks. θ_1 ,θ_2 ,θ_3 and θ_4 represent bucket, stick, boom movements and swing, respectively. The black line in Fig. 4.8 shows that the bucket remains fixed in the second subtask, Swing to Truck. The blue line represents a constant rotation of excavator upper-structure in the same subtask.

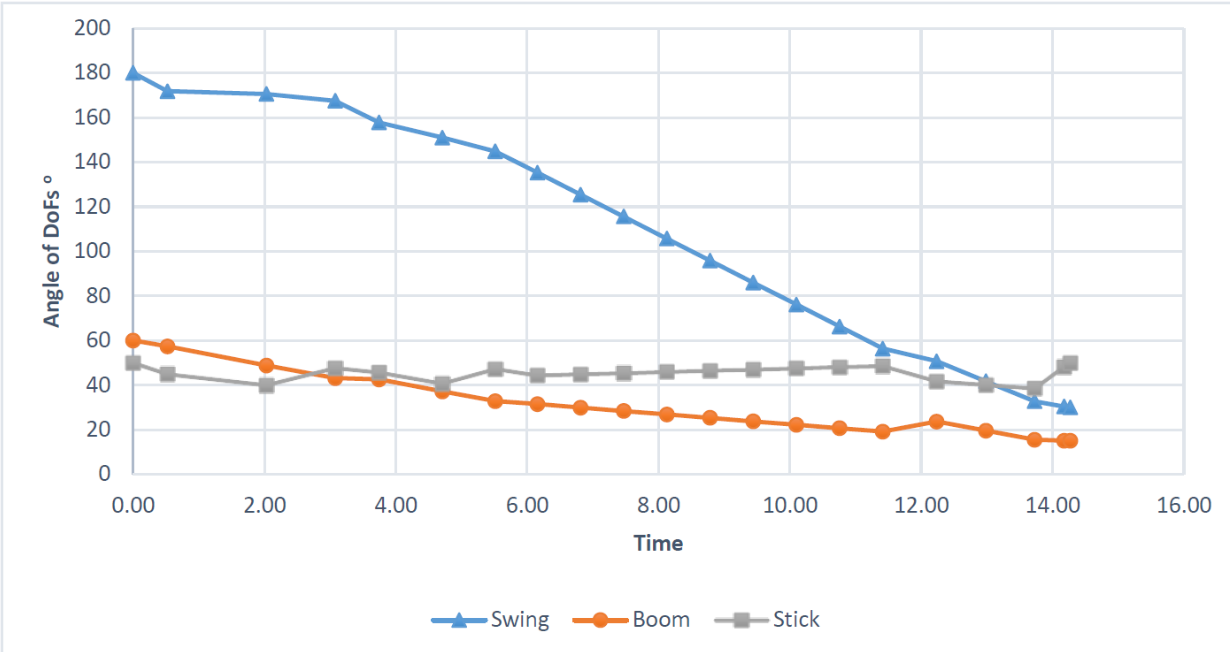


Figure 4.7 The motion planning of a path generated by NonuniformRRT+

In order to compare, motion planning is applied on a path generated by NonuniformRRT+ for the same subtasks, as shown in Fig. 4.9. In the first view, it looks different from Fig. 4.18, but it mainly roots from the conventions. In contrast with Fig. 4.8, the bucket movement in Fig. 4.8 is not constant as it should maintain the soil while moving to truck (Langari and Hammad, 2015). The blue line represents the swing, which is constant in loading and dumping subtasks but almost constantly changes in Swing to Truck. Detailed comparison reveals small deviations between two figures, which is coming from two facts: (1) Fig. 4.8 represents a typical motion but Fig. 4.9 shows a motion planning in a scenario with more obstacles, (2) The motion shown in Fig. 4.9 is based on a path generated by NonuniformRRT+.

Table 4-5 Excavation cycle time of hydraulic excavator under average condition* (Peurifoy and Schexnayder, 2011)

Bucket size (cy)	Load bucket (sec)	Swing loaded (sec)	Dump bucket (sec)	Swing empty (sec)	Total cycle (sec)
< 1	5	4	2	3	14
$1 - 1\frac{1}{2}$	6	4	2	3	15
$2 - 2\frac{1}{2}$	6	4	3	4	17
3	7	5	4	4	20
$3\frac{1}{2}$	7	6	4	5	22
4	7	6	4	5	22
5	7	7	4	6	24

*Depth of cut 40 to 60% of maximum digging depth, the swing angle 30 to 60 degrees, loading haul unit on the same level as the excavator

4.2.6 Discussion

The proposed approach is designed for movements in multiple DoFs like a skilled excavator operator who moves multiple DoFs of the equipment to achieve the best performance and consequently optimal productivity. Regarding the obtained results, there are couple of issues that are explained in the following:

- Regarding the computation effort of both scenarios, the results show a great capability of NonUniformRRT+ in dealing with a very challenging situation on the construction site. It is able to generate a safe path in a minimal time. Considering the limitations of on-site processors, this efficacy of the planner would become very essential. NonUniformRRT (Langari and Hammad, 2015) fails to generate a fast result in a problem such as scenario I where the pipes are in the way to the goal configuration and located very close to initial configuration. However, the new NonUniformRRT+ is able to improve that limitation.

- Generally speaking, a path planning problem becomes harder by increasing the number of dimensions of the C-Space (number of DoFs) and it requires more computation effort. In other words, it is expected to take more time for a planner to generate a path when the problem has multiple DoFs, as the multiple number of dimensions of the C-space specifies a bigger and more complex search space. However, it is not always the case since the computation effort is affected by other factors such as the initial configuration, settings of the obstacles in the C-Space or even the planner paramters.

- Considering the length of the path for a full free cycle, RRTBiasedLimCon generates a path with length of 82.09 meters, and NonUniformRRT+ generates a path of 72.35 meters. These values are calculated by summing the length of the pathes in scenario I and II. Then in the whole cycle, there is a decrease of 12% in the length of the path, and also there is a great improvement in the quality of the path, which will increase the cycle-time of the equipment dramatically.

- In the context of agent-based simulation, agents are supposed to be independent. Hence, the computation time affects the whole scenario and it should be carefully considered. The idea is that other agents would not wait for a specific agent to make the decision. Then the obstacle setting may change a lot while a planner is generating a path for one agent. Then the computation time should be considered in the whole accomplish time, as shown in Equation 3.11.

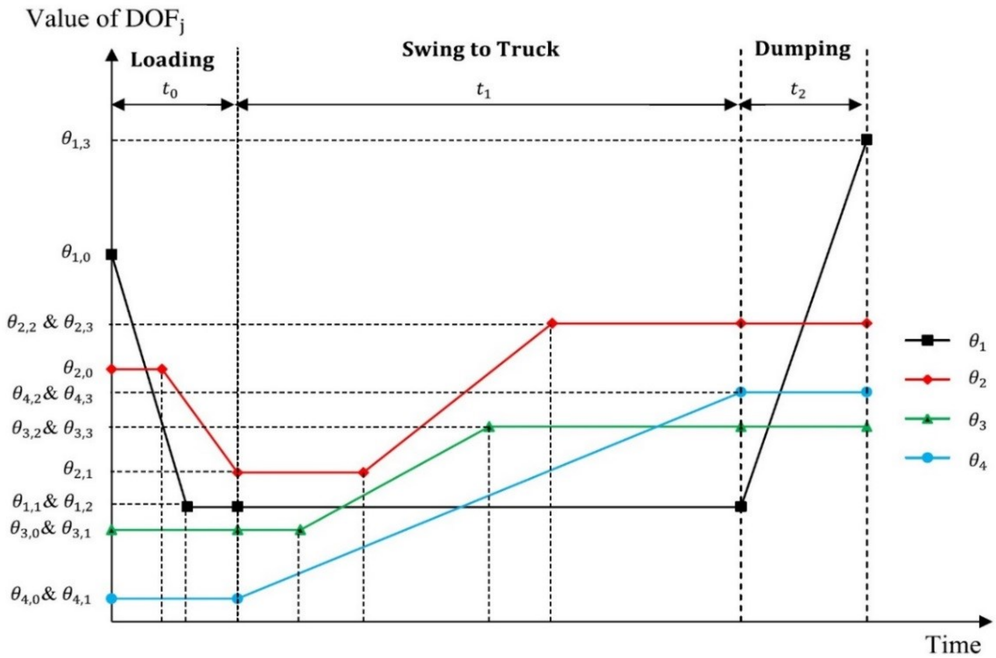


Figure 4.8 A typical motion planning of an excavator (Vahdatikhaki, 2015)

Although in this research the proposed approach is applied on the excavator, the authors believes that it is also suitable for similar construction equipment with capability of movement in multiple DoFs at the same time.

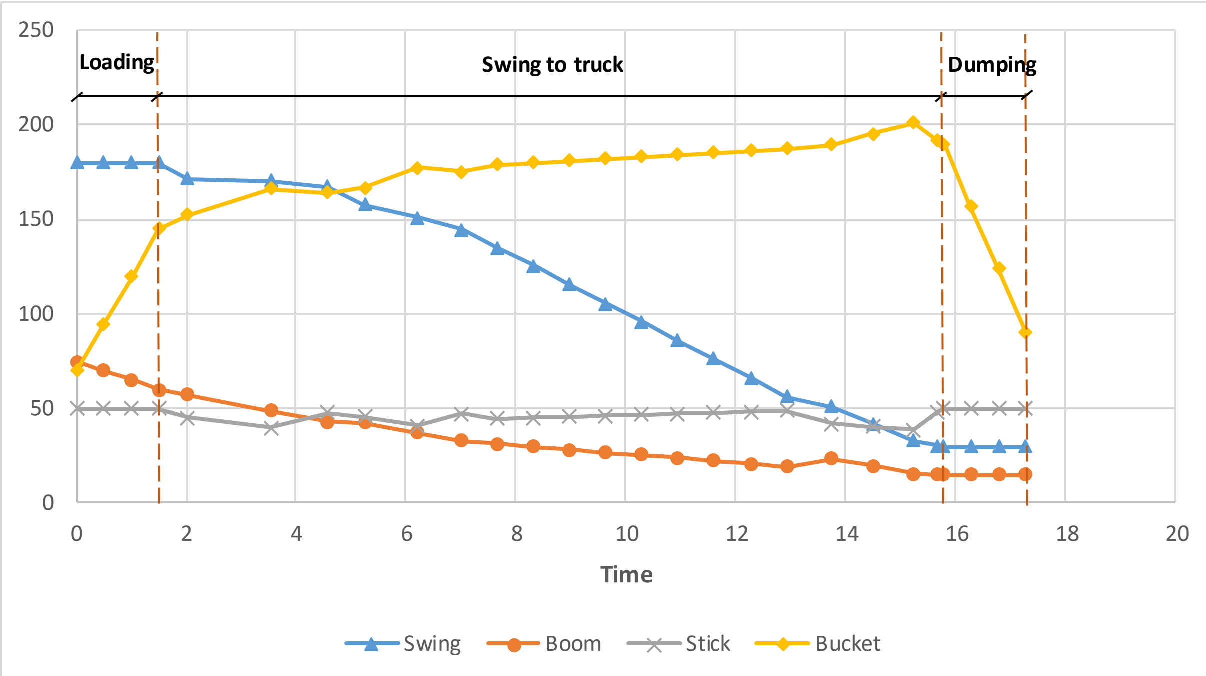


Figure 4.9 Motion planning of a path generated by NonuniformRRT+

4.3 Second Case Study: Implementation of Multi-Agent System

As shown in Fig. 4.10, a simulated earthmoving operation is designed in Unity where two excavators and four trucks are working together as a team. The case study assumes that the operation information (such as operation schedule, operation logic, 3D design of the model, etc.) has already been transferred to the agents of this scenario. Two excavators and four trucks are considered as the initial set of agents. The operation schedule is given in the form of the type of tasks to be completed by the agents, and the start and the end position of the operation.

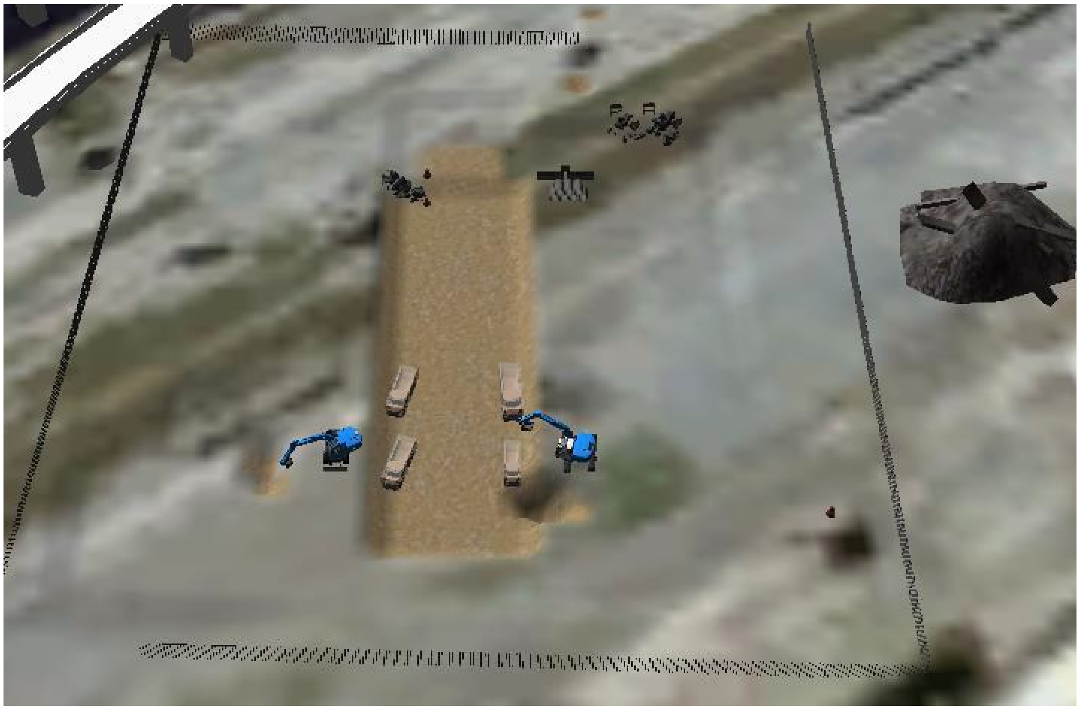


Figure 4.10 Layout of the simulated case study

In this case study, the macro plans for the excavator are given through manager-defined plans as shown in Fig. 4.11, following the pattern presented by Seo et al. (2011). As shown in this figure, each excavator has a designated start and end points and a route connecting the two points, which are defined through a graphical user interface. Points DS_1 to DS_8 and DS'_1 to DS'_8 represent the macro plans for excavators one and two, respectively. Once the simulation runs, the agents follow their plans based on their respective operation logics.

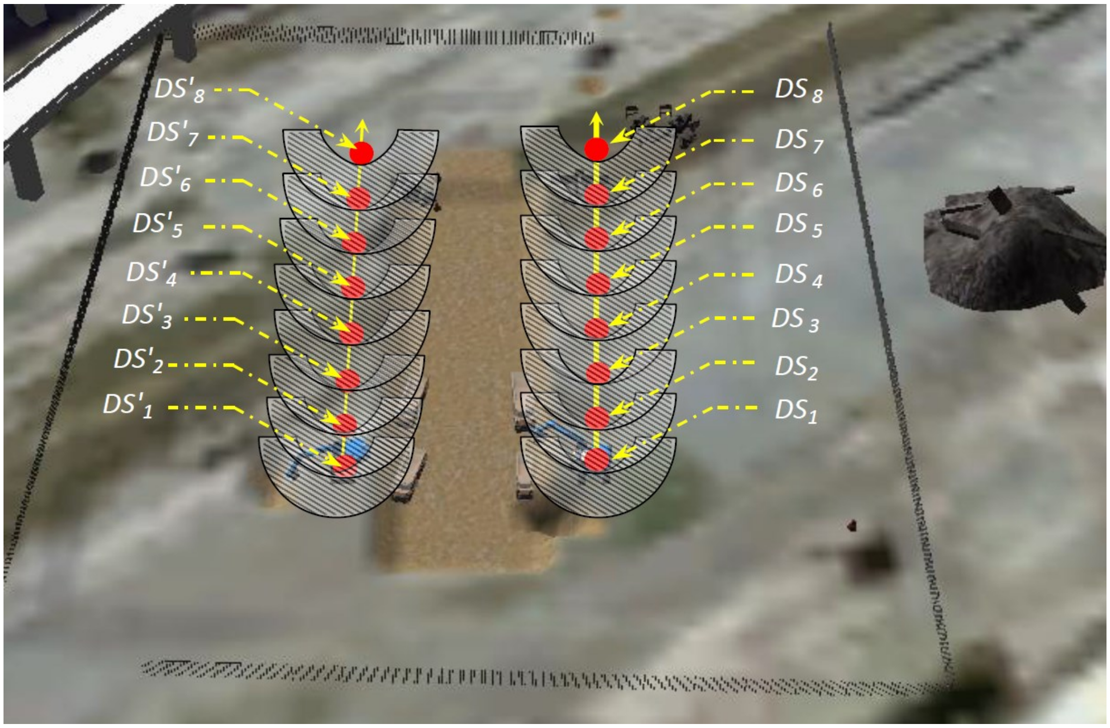


Figure 4.11 Schematic representation of the digging plans for excavators

Look Ahead Equipment Workspace (LAEW) is a proactive method, which forecasts the movement of equipment in order create the equipment workspace for the next couple of seconds (Vahdatikhaki, 2015). The feasibility of applying LAEWs for collision-free path planning is investigated. In this scenario, which is inspired from a real excavation site shown in Fig. 4.12(a), two excavators are digging the ground between two pipes. Excavator 1, which is assumed to have a higher priority than Excavator 2, is expected to swing to Truck 1, and Excavator 2 is expected to swing away from Truck 2. The excavators are working in close proximity and should avoid the pipes and the terrain as static obstacles. As shown in Fig. 4.12(b), the initial paths of the excavators are planned using parametric scripting (Stentz et al., 1999). Fig. 4.12(c) shows the risk map of Excavator 1 with a threshold of 0.8 for the next 2.3 seconds. Since the initial path of Excavator 2 collides with the generated LAEW, Excavator 2 uses RRT path planner to generate a new path that will avoid the potential collision. The bucket trajectory of Excavator 2 and the final configurations of both excavators are shown in Fig. 4.12(d). The results illustrate that the Multi-Agent System (MAS) (Vahdatikhaki, 2015) is able to effectively use LAEWs to predict the potential future collisions and generate new paths to avoid such collisions.

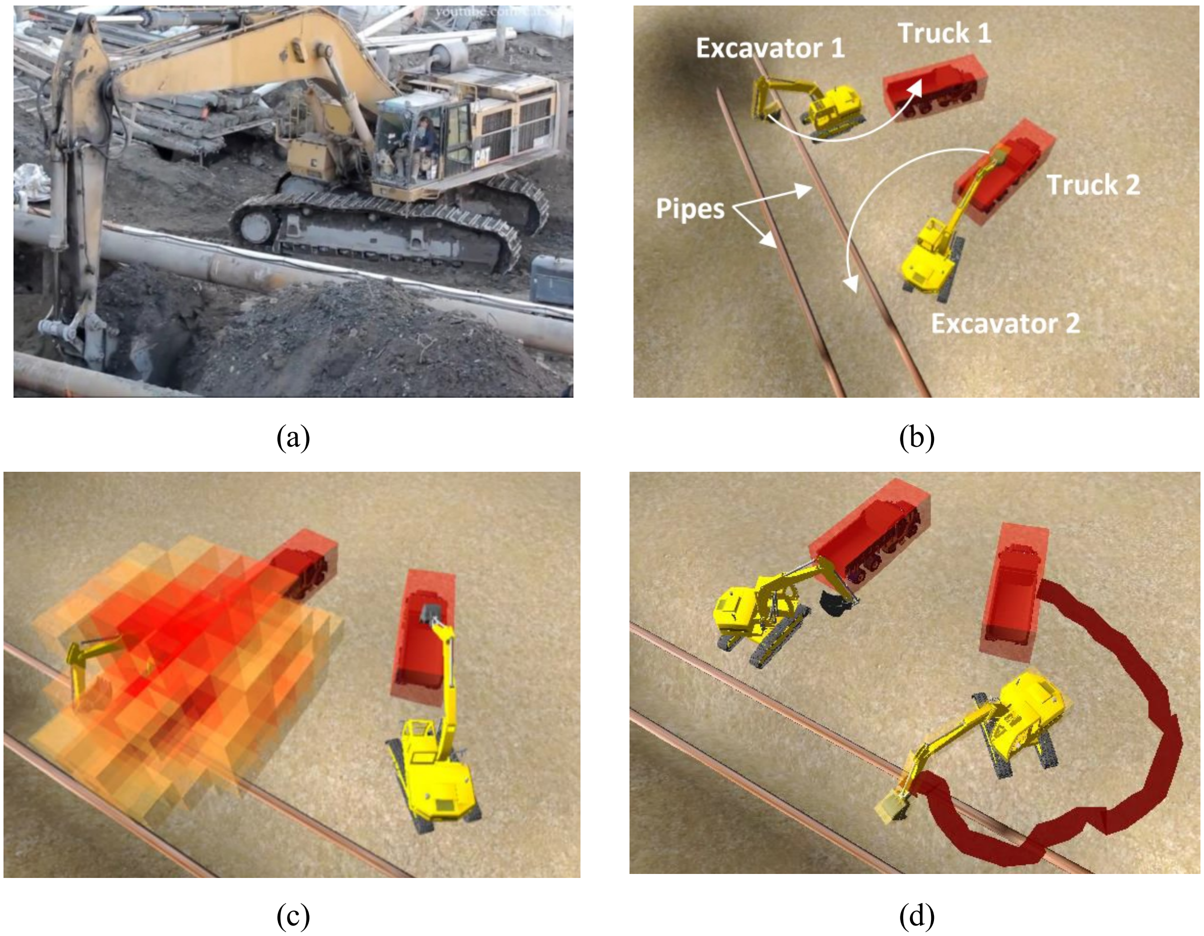


Figure 4.12 Path planner avoids the possible collision (b) Current poses and initial paths of excavators, (c) LAEW of excavator 2, and (d) Final path of excavator 2 (Vahdatikhaki, 2015)

4.4 Summary and Conclusions

Case studies, developments of the proposed models, limitations and experiments are explained and discussed in this chapter. The chapter starts with the case studies that we adapted from a real world project. The site is congested with many obstacles while the excavator is working. We created the same environment in the Unity Game Engine (2015) using C# programming language. The NonuniformRRT is first implemented on the excavator in which the advantages and limitations are revealed. Then implementation of the NonuniformRRT+ is explained and the results of performance comparison with RRTBiasedLimCon are presented. The results show that NonuniformRRT+ can generate a safe path faster than RRTBiasedLimCon. In terms of quality, the path generated by NonuniformRRT+ is significantly improved as well.

Besides, motion adjustment is applied to the path generated by NonuniformRRT+ and it is then compared with previous research. The results of the motion adjustment show that the performance of the NonuniformRRT+ is comparable to an average skilled excavator operator. In addition, the implemented screenshot of the macro planning is shown and discussed. At the end, application of the proposed path planning method combined with the LAEW concept is investigated (Vahdatikhaki, 2015).

5. SUMMARY, CONCLUSIONS AND FUTURE WORK

5.1 Summary

Simulation of excavator path planning is developed, at different levels of details in this thesis. First, many relevant research backgrounds in the areas of agent-based simulation, path planning algorithms, and robotics are studied in the literature review. Due to its significant effect on earthwork's productivity, the excavator operation is selected to be investigated in order to improve the safety and performance of the construction site. The excavator is treated as a semi-robot with rigid parts and mathematical expressions governing the kinematics of the equipment.

As the upper-structure of a typical excavator has four DoFs, an advanced path planning algorithm, RRT, is used as a base to address the requirements of the construction equipment. Previous path planners were developed to solve the problems in a pure mathematical setting and they were not be able to consider the requirements of construction equipment. Many modifications are applied and tested on the planner in the simulation environment to ensure the safety, productivity and constraints are respected. A new NonUniformRRT+ algorithm is proposed to solve the path-planning problem of the excavators by embedding heuristic rules for (1) biasing the extension of the tree in RRT, (2) extending the tree directly toward the goal when there is no obstacle, and (3) applying heuristic rules for obstacles when the tree is trapped. New metrics are proposed to measure and evaluate NonuniformRRT+ by comparing with the latest available planners.

A new metric is proposed to measure the quality of the path. The results show a significant improvement in terms of the quality of the path in both scenarios. Besides, the proposed method has shown that it requires less computation time than RRTBiasedLimCon. From the qualitative point of view, the movement of the excavator is quite smooth with minimal jaggedness. The new approach is able to address the limitation of previous path planners when applied to hydraulic excavators. It is able to consider different engineering constraints, embedded in the motion of hydraulic excavator, in order to increase the productivity of earthwork projects. The modified planner is expected to be applicable to other types of construction equipment with the same nature.

The results show that the performance of the proposed algorithm is good and can address the fast-paced requirements of construction site. A scenario is extracted from the real world to evaluate the proposed algorithm. The excavator has to pass the loaded bucket between two pipes and move it above the truck to dump. The simulation showed that the algorithm is capable of efficiently finding the collision-free path from a given initial configuration to a goal configuration. In the context of safety planning, the proposed algorithm is developed and tested where two excavators are working very close to each other. The excavators can perform their duty without any safety problems using the proposed planner.

Unity game engine 3D is used as the simulation platform where agents can be defined in the scene environment and the user can add behavior to them using C# script or Java script code. We have generated an earthwork operation scenario in the scene with excavators, trucks and workers. We have tested the proposed algorithm in case of cyclic operation of excavators. The user can define the number and type of joints, whether prismatic or revolute, inside the program with the constraints for each. The user also can specify the amount of many parameters that the planner requires. In order to implement the collision detection, a buffered volume is considered around each equipment pieces, which has two advantages; first, it decreases the computation effort while the safety is not compromised, second, it can be used as a virtual entity for collision detection without affecting the visual excavator.

The productivity aspect is mostly neglected in previous research as there is a gap between the path planning and the productivity. The time is the linkage between these two. NonuniformRRT+ generates collision-free path for excavators, which will increase the productivity of operations by producing a high quality path within a minimal period of time. In Chapter 3 of this thesis, a motion planning technique is proposed for a path generated by NonuniformRRT+ in order to calculate the productivity of the hydraulic excavator within the simulation environment. To generate a motion planning of hydraulic excavator there are engineering constraints to be respected such as joint limits, joint speed limit, and coupled motion effect. Besides, the motion planning of each STEP has to be done based on the critical one, and the rest of joints must be adjusted accordingly to assure the safety is guaranteed.

5.2 Contributions and Conclusions

The followings represent the contribution made in this research:

(1) A new NonuniformRRT+ algorithm is proposed for path-planning of excavators by embedding heuristic rules for engineering constraints of excavators to improve the quality of the path and performance of the planner.

(2) A new definition of the path roughness metric is proposed to measure the quality of the path in addition to the other metrics (path length and computation time).

(3) In order to determine the effect of path planning on productivity, a new method is proposed in which the time is added to the path to generate the motion plan for excavators while respecting the maximum rotational speeds of the joints.

(4) A synthetic macro and micro path planning approach is proposed in the Multi-Agent System

for earthwork operations with excavators and trucks.

And the followings represent the conclusions of this research:

(1) RRTBiasedLimCon, proposed by Albahnassi (2010), is adapted for excavators and implemented within the Unity Game Engine for purpose of comparison. The comparison between NonuniformRRT+ and RRTBiasedLimCon is done for two scenarios, 3DoFs and 4DoFs. In both cases, the results show a significant improvement in terms of all metrics.

(2) The proposed path planning is able to capture the cyclic operation and constraints of the excavator.

(3) The visual comparison shows that the path generated by NonuniformRRT+ is smoother than the path generated by RRTBiasedLimCon and similar to the path generated by a skilled excavator operator.

(4) The motion planning is applied for the path generated by NonuniformRRT+, and the results show that the performance of the motion planner is comparable with the cycle-time of an average skilled operator.

(5) The application of micro and macro path planning within the Multi-Agent System is investigated and an basic prototype of Multi-Agent System is developed in which many parameters such as productivity, cycle-time of each equipment, idle times etc. can be measured.

5.3 Limitations and Future Work

This research aimed to fill the gap in the available research background. However, there are many other areas of research, which can be considered as future work:

(1) The interaction between the worker and construction equipment is one area of research, which

can greatly increase the safety in construction sites. Usually the collision detection is used in order to increase the safety between the construction equipment and other objects. However, the interaction of worker workspace and the equipment workspace is required to be further investigated.

(2) Equipment pose/state detection is another area of research, which can increase the safety of construction sites. In order to make sure all safety aspects are respected, construction equipment planning must be supported by these tracking methods.

(3) A full agent-based simulation of construction project requires the clear definition of agents along with their specifications and behaviors, and more importantly the interaction between agents and the environment. In order to bring the construction operation logic into consideration, a fleet management method is required.

(4) A full real-world implementation of the proposed methods can be a challenge, which is outside of the scope of this research; however, it reveals more engineering and practical aspects that may be neglected in the simulation environment.

(5) Creation of soil model is another area of research in the simulation of earthwork operations. Simulation of small soil particles requires very high computation effort, which makes it difficult for conventional software and hardware.

References

AlBahnassi, H., Real-Time Motion Planning and Simulation of Cranes in Construction. *Master Thesis, Concordia University*, Canada, 2010.

AlBahnassi, H. and Hammad, A., Near Real-Time Motion Planning and Simulation of Cranes in Construction: Framework and System Architecture. *Journal of Computing in Civil Engineering,* 26(1): 54-63, 2011.

Alzraiee, H., Moselhi, O., and Zayed, T., Dynamic Planning of Earthmoving Projects Using System Dynamics. *Gerontechnology*, *11*(2), 316, 2012a.

Alzraiee, H., Moselhi, O., and Zayed, T., A Hybrid Framework for Modeling Construction Operations Using Discrete Event Simulation and System Dynamics. *Construction Research Congress*, 1063-1073, 2012b.

Association of Workers' Compensation Boards of Canada (AWCBC), 2014, Number of Fatalities, By Industry and Jurisdiction. *On-line:* <*http://awcbc.org/?page_id=14#fatalities*> Accessed June 9th, 2015.

Bennink, C., 2011, Dig Into Excavator Productivity. January 5th. *On-line :*< *http://www.forconstructionpros.com/article/10212146/dig-into-excavator-productivity*> Accessed October 10th, 2015.

Bruce, J., 2004. Robot Motion Planning. Available at <*http://www4.cs.umanitoba.ca/~jacky/Robotics/Papers/Bruce-ERRTMotionPlanningSlides.pdf* > accessed June 1, 2015.

Chang, Y.C., Hung, W.H. and Kang, S.C., A Fast Path Planning Method for Single and Dual Crane Erections. *Journal of Automation in Construction,* 22, pp. 468-480, 2012.

Choset, H., 2015, Robotic Motion Planning: A* and D* Search. Available at <*http://www.cs.cmu.edu/~motionplanning/lecture/lecture.html*> accessed May 29, 2015.

Craig, J.J., Introduction to Robotics: Mechanics and Control. *Pearson education international,* 3rd edition, 400 p, 2005.

Dijkstra, E.W., A Note on Two Problems in Connexion with Graphs. *Numerische Mathematik*, 1, 269-271, 1959.

Ferguson, D., Kalra, N., and Stentz, A., Replanning with RRTs. *Proceeding of the IEEE International Conference on Robotics and Automation*, pp. 1243-1248, 2006.

Hart, P.E., Nilsson, N.J., Raphael, B., A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transaction on Systems Science and Cybernetics,* 4(2): 100-107, 1968.

Hammad, A., Vahdatikhaki, F., and Zhang, C., A Novel Integrated Approach To Project-Level Automated Machine Control/Guidance Systems in Construction Projects. *Journal of Information Technology in Construction, ITcon*, 18:162-181, 2013.

Hammad, A., El Ammari, K., Langari, M., Vahdatikhaki, F., Soltani, M., AlBahnassi, H., Paes, B., Simulating Macro and Micro Path Planning of Excavation Operations Using Game Engine. *In Proceedings of Winter Simulation Conference*, pp. 4111-4112, Savannah, USA, 2014a.

Hammad, A., El Ammari, K., Langari, S.M., Vahdatikhaki, F., Soltani, M., Albahnassi, H., Paes, B., Simulating Macro and Micro Path Planning of Excavation Operations Using Game Engine, *Internal report*, 2014b.

Henrich, D., Fast Motion Planning By Parallel Processing–A Review. *Journal of Intelligent and Robotic Systems,* 20.1 pp. 45-69, 1997.

Hoan, L.Q., Jeong, C.S., Kim, H.S., Yang, H.L., and Yang, S.Y., Study on Modeling and Control of Excavator. *In proceedings of 28th International Symposium of on Automation and Robotics in Construction,* 969-974, Seol, Korea, 2011.

Kamat, V.R., and Martinez, J.C., Dynamic 3D Visualization of Articulated Construction Equipment. *Journal of Computing in Civil Engineering*, 19(4), 356-368, 2005.

Kang, S., and Miranda, E., Planning and Visualization for Automated Robotic Crane Erection Process in Construction. *Journal of Automation in Construction,* 15: 398-414, 2006.

Kavraki, L.E., Svestka, P., Latombe, J.C., Overmars, M.H., Probabilistic Roadmaps for Path Planning In High-Dimensional Configuration Spaces. *IEEE Transactions on Robotics and Automation,* 12(4): 566-580, 1996.

Kim, S., Russell, J.S., and Koo, K., Construction Robot Path-Planning for Earthwork Operations. *Journal of Computing in Civil Engineering*, 17(2):97-104, 2003.

Kim, S.K., and Russell, J.S., Framework for an Intelligent Earthwork System: Part I. System Architecture. *Journal of Automation in Construction*, 12(1), 1-13, 2003a.

Kim, S.K., and Russell, J.S., Framework for an Intelligent Earthwork System: Part II. Task Identification/Scheduling and Resource Allocation Methodology. *Journal of Automation in construction*, 12(1), 15-27, 2003b.

Kim, J., Seo, J., Task Planner for Autonomous Excavator Considering Work Environment. *In proceedings of the 28th International Symposium on Automation and Robotics in Construction*, 770-771, Seol, Korea, 2011.

Kim, S., Seo, J., and Russell, J.S., Intelligent Navigation Strategies for an Automated Earthwork System. *Journal of Automation in Construction*, 21:132-147, 2012.

Kiongoli, S., Testing the Accuracy of Machine Guidance in Road Construction. *Undergraduate dissertation, University of Southern Queensland*, 2010.

Klingensmith, M., 2013, Overview of Motion Planning. *On-line: <http://www.gamasutra.com/blogs/MattKlingensmith/20130907/199787/Overview_of_Motion_Planning.php>* accessed May 29, 2015.

Laster, P., 2005, A* Path Finding for Beginner. *On-line: < http:// www.policyalmanac.org/games/aStarTutorial.htm>* accessed May 28, 2015.

La Valle, S.M., 1998, Rapidly-exploring Random Trees: a New Tool for Path Planning. Computer science department, Iowa state University. *On-line:* <*msl.cs.uiuc.edu/~lavalle/papers/Lav98c.pdf*> accessed May 22, 2015.

La Valle, S.M., and Kuffner, J., Rapidly-exploring Random Tree: Progress and Prospect. 2000.

Langari, S.M., Hammad, A., Embedding Heuristic Rules in RRT Path Planning of Excavators. *In Proceedings of 32nd International Symposium on Automation and Robotics in Construction and Mining*, Oulu, Finland, 15-18 June 2015.

Lee, J., and Bernold, L.E., Ubiquitous Agent-based Communication in Construction. *Journal of Computing in Civil Engineering*, *ASCE*, 22(1):31-39, 2008.

Lee, C.S., Bae, J.H., and Hong, D.H., A Study on Working Plan of Intelligent Excavator. *In Proceedings of the 28th International Symposium on Automation and Robotics in Construction*, 993-995, Seol, Korea, 2011a.

Lee, G.W., Im, S., and Kim, E.K., Development of an Excavator Simulator for an Intelligent Excavating System. *In proceedings of the 28th International Symposium on Automation and Robotics in Construction*, 1445-1448, Seol, Korea, 2011b.

Lin, J.J.C., Yang, C.E., Hung, W.H., and Kang, S.C., Accessibility Evaluation System for Site Layout Planning–a Tractor Trailer Example. *Journal of Visualization in Engineering*, 1(1), 1-11, 2013.

Lin, J.J., Hung, W., and Kang, S., Motion Planning and Coordination for Mobile Construction Machinery. *Journal of Computing in Civil Engineering,* 10.1061/ (ASCE) CP.1943-5487.0000408 , 04014082, 2014a.

Lin, Y., Want, X., Wu, D., Wang, X., Gao, S., Lift Path Planning for a Nonholonomic Crawler Crane. *Journal of Automation in Construction*, 44: 12-24, 2014b.

Liu, Y., Hasan, M.S., and Yu, H.N., Modelling and Remote Control of an Excavator. *International Journal of Automation and Computing*, 7(3), 349-358, 2010.

Lozano-Pérez, T., and Wesley M.A., An Algorithm for Planning Collision-free Paths Among Polyhedral Obstacles. *Communications of the ACM* 22.10: 560-570, 1979.

Lumelsky, V. J., & Stepanov, A. A., Path-Planning Strategies for a Point Mobile Automaton Moving Amidst Unknown Obstacles of Arbitrary Shape. *Algorithmica*, 2(1-4), 403-430, 1987.

Marzouk, M., and Ali, H., Modeling Safety Consideration and Space Limitations in Piling Operations Using Agent-Based Simulation. *Journal of Expert System with Application*, 40, 4848-4857, 2013.

Montgomery, D.C., and Runger, G.C., Applied Statistics and Probability for Engineering. *John Wiley & Sons, United States of America*, 4(12): pp. 148-150, 2014.

The National Institute for Occupational Safety and Health (NIOSH), *Online* <*http://www.cdc.gov/NIOSH/* > Accessed December, 2nd, 2015.

Peurifoy, R.L., and Schexnayder, C.J., Construction Planning, Equipment, and Methods. *McGraw-Hill Education, 8th edition*, 2011.

Rowe, P.S., Adaptive Motion Planning for Autonomous Mass Excavation. *Ph.D. dissertation, Carnegie Mellon University*, 1999.

Rowe, P., and Stentz, A., Parameterized Scripts for Motion Planning. *Intelligent Robots and Systems*, 2:1119-1124, 1997.

Seo, J., Lee, S., Kim, J., and Kim, S. K., Task Planner Design for an Automated Excavation System. *Jounal of Automation in Construction*, 20(7), 954-966, 2011.

Singh, S., and Cannon, H., Multi-resolution Planning for Earthmoving. *In Proceedings of the IEEE International Conference on Robotics and Automation, IEEE*, 121-126, 1998.

Soltani, A.R., Tawfik, H., Goulermas, J.Y., and Fernando T., Path Planning in Construction Sites: Performance Evaluation of the Dijkstra, A*, and GA Search Algorithms. *Journal of Advanced Engineering Informatics*, 16: 291-303, 2003.

Stentz, A., The Focused D* Algorithm for Real-time Replanning. *In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 1995.

Stentz, A., Bares, J., Singh, S., & Rowe, P., A Robotic Excavator for Autonomous Truck Loading. *Autonomous Robots*, 7(2): 175-186, 1999.

The National Institute for Occupational Safety and Health (NIOSH), 2010, Goal 1: Reduce the Major Risks Associated with Traumatic Injuries and Fatalities in Construction. *Online:* <*http://www.cdc.gov/niosh/nas/construction/*> Accessed 10/11/2015.

Tserng, H. P., Ran, B., and Russell, J. S., Interactive Path Planning for Multi-equipment Landfill Operations. *Journal of Automation in construction*, 10(1), 155-168, 2000.

Turner, T., Craig, J., and Gruver, W., A Microprocessor Architecture for Advanced Robot Control. *14th ISIR,* Stockholm, Sweden, October 1984.

Unity Game Engine. *On-line:* <*http://unity3d.com*> Accessed January 28, 2015.

Vahdatikhaki, F., Hammad, A., Risk-Based Look-Ahead Workspace Generation for Earthwork Equipment Using Near Real-Time Simulation. *Journal of Automation in Construction*, 58: 207-220, 2015.

Vahdatikhaki, F., Towards Smart Earthwork Sites Using Location-Based Guidance and Multi-Agent Systems. *Ph.D. dissertation, Concordia University*, 2015.

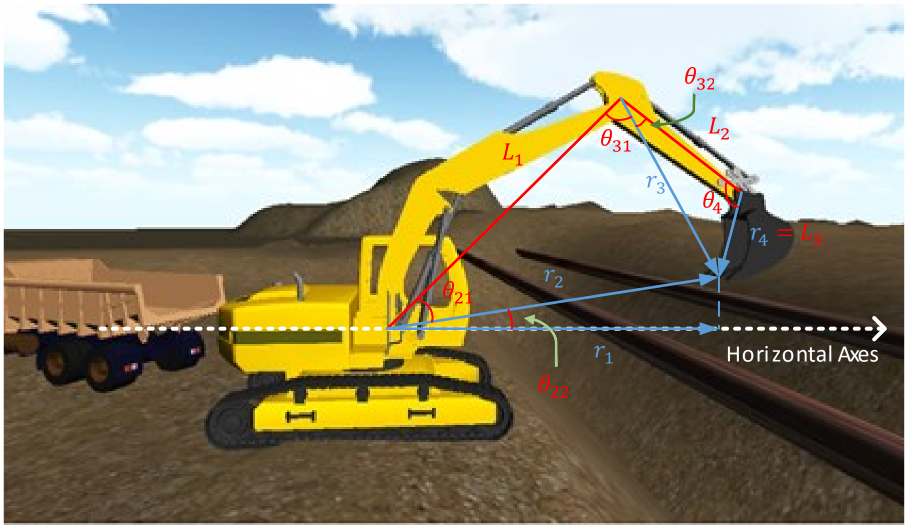
Zayed, T.M., and Halpin, D., Simulation of Concrete Batch Plant Production. *Journal of Construction Engineering and Management*, 127(2), 132-141, 2001.

Zayed, T., Halpin, D. W., Simulation as a Tool for Pile Productivity Assessment. *Journal of Construction Engineering and Management*, 130:3(394), 2004.

Zhang, C., Hammad, A., Bahnassi, H., Collaborative Multi-Agent Systems for Construction Equipment Based on Real-time Field Data Capturing. *Journal of Information Technology in Construction*, 14: 204-228, 2009.

Zhang, C., and Hammad, A., Improving Lifting Motion Planning and Re-planning of Cranes with Consideration for Safety and Efficiency. *Journal of Advanced Engineering Informatics*, 26: 396-410, 2012.

Appendix A – Excavator effective arms



Target of calculation:

To find the effective arms of a typical excavator, called r_1, r_2, r_3 and r_4, with respect to the DoFs of the upper structure, as mentioned in Section 3.3.4.1.

Assumptions:

$\theta_3 = \theta_{31} + \theta_{32}$

$\theta_2 = \theta_{21} + \theta_{22}$

- $\boldsymbol{r_4 = L_3}$

θ_2, θ_3 *and* θ_4 are given and also L_1, L_2 *and* L_3 are obtained from the excavator specifications.

- $\boldsymbol{r_3 = \sqrt{r_4^2 + L_2^2 - 2r_4L_2\cos\theta_4}}$

- r_2:

$r_4{}^2 = r_3{}^2 + L_2{}^2 - 2r_3 L_2 \cos\theta_{32}$ then,

$$\theta_{32} = \cos^{-1}\left(\frac{r_3{}^2 + L_2{}^2 - r_4{}^2}{2r_3 L_2}\right)$$

$$\theta_{31} = \theta_3 - \theta_{32}$$

$$\boldsymbol{r_2 = \sqrt{L_1^2 + r_3^2 - 2r_3 L_1 \cos\theta_{31}}}$$

- r_1:

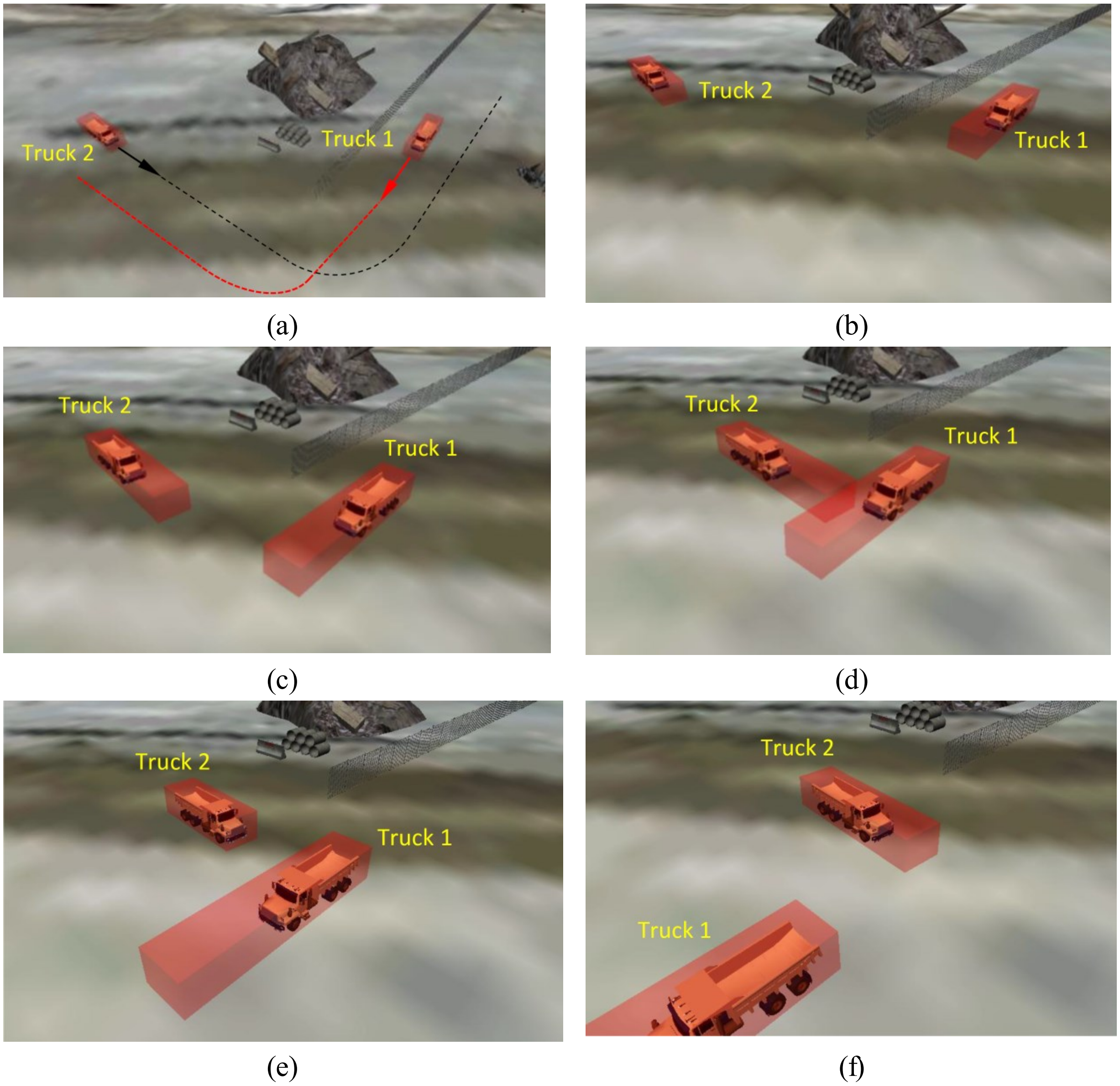
$r_3{}^2 = r_2{}^2 + L_1{}^2 - 2r_2 L_1 \cos\theta_{21}$ then,

$$\theta_{21} = \cos^{-1}\left(\frac{r_2{}^2 + L_1{}^2 - r_3{}^2}{2r_2 L_1}\right)$$

$$\theta_{22} = \theta_2 - \theta_{21}$$

$$\boldsymbol{r_1 = r_2 \cos\theta_{22}}$$

Appendix B – Truck dynamic workspace



Dynamic Work-space (a) The layout of the second scenario, (b) and (c) Dynamic sizes of DEWs, (d) Collision detection between DEWs, (e) Stoppage of Truck 2, and (f) Both equipment continue their paths

Dynamic Equipment Workspace is another safety concept proposed by Vahdatikhaki (2015) to avoid collision; this concept is used as the last line of defense. The above figure shows an instance of how DEW-based safety management is capable of avoiding immediate collisions between equipment (Vahdatikhaki, 2015). As shown in Fig. (a), while Truck 1 is hauling the material to the dumping point, Truck 2 is returning from the dumping point to the corresponding DS. It is assumed that although their planned paths have been collision-free, Truck 1 fell behind its planned path, which could lead to a potential collision. In this case, DEWs can be used as the last line of defense to avoid the collision by requiring one of the equipment to stop. A higher priority is given to Truck 1 since it is loaded. Figs. (b) and (c) show how the size of DEWs is changing based on the speed characteristics of the equipment. Fig. (d) depicts the collision between the two DEWs. Given its lower priority, Truck B stops and waits until Truck A passes, as shown in Figs. (e) and (f).

Appendix C - Publications

(a) Articles in refereed journals

Langari, S.M., Hammad, A., Improving the Performance of RRT Path Planning of Excavators by Embedding Heuristic Rules. *To be submitted to Journal of Automation in Construction*.

Vahdatikhaki, F., Langari, S.M., Hammad, A., Taher, A., Framework for Fleet-level Automated Equipment Guidance in Earthwork Projects Using Multi-Agent System. *Journal of Automation in Construction*, 2015 (Under review).

(b) Articles in refereed conferences

Bolourian, N., Langari, S.M., Soltani, M.M., Park, J., and Hammad, H., Framework for Bridge Inspection Using Unmanned Aerial Vehicle and LiDAR Scanning. *Abstract submitted to 33rd International Symposium on Automation and Robotics in Construction,* 2016.

Vahdatikhaki, F., Hammad, A., Langari, S.M., Multi-Agent System for Improved Safety and Productivity of Earthwork Equipment using Real-time Location Systems. *In proceedings of 11th Construction Specialty Conference*, Vancouver, BC, Canada, 2015.

Langari, S. M., Hammad, A., Embedding Heuristic Rules in RRT Path Planning of Excavators. *In Proceeding of 32nd International Symposium on Automation and Robotics in Construction and Mining*, Finland, 1-8, 2015.

Hammad, A., El Ammari, K., Langari, M., Vahdatikhaki, F., Soltani, M., AlBahnassi, H., Paes, B., Simulating Macro and Micro Path Planning of Excavation Operations Using Game Engine. *In proceedings of Winter Simulation Conference*, Savannah, USA, 4111-4112, 2014.