

**Model-Based Reliable Mission Planning of Off-Road Autonomous Ground Vehicles
Under Uncertain Environments**

by

Yixuan Liu

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Industrial and Systems Engineering)
in the University of Michigan-Dearborn
2022

Doctoral Committee:

Assistant Professor Zhen Hu, Chair

Associate Professor Shan Bao

Associate Professor Jian Hu

Assistant Professor Youngki Kim

Assistant Professor Xiaoge Zhang, Hong Kong Polytechnic University

Yixuan Liu

yixuanli@umich.edu

ORCID iD: 0000-0002-5657-4412

© Yixuan Liu 2022

Acknowledgments

First, I would like to express my greatest appreciation to my academic advisor, Professor Zhen Hu, for his continuous support and guidance. His foresight during the research helps me to find proper research topics and put my effort in the right place. He is always willing to share the precious knowledge and experience in both academia and life which not only guides me through my Ph.D. career but will also inspire me in the future. To be honest, sometimes I doubted if I could finish my Ph.D., but Prof. Hu is always able to help me out of my struggling and help me push forward. He has really changed my life. I am also thankful to Prof. Yubao Chen, who introduce Prof. Hu to me when I decided to start my Ph.D. journey.

I am grateful to the committee members, Prof. Youngki Kim, Prof. Shan Bao, Prof. Jian Hu and Prof. Xiaoge Zhang. Their insights help me find the defects in my research so I can continuously improve the dissertation.

I would also like to thank my peers and friends for their inspiration. Special thanks to Dr. Chen Jiang for the guidance and encouragement when I first start my research; thanks to Dr. Jice Zeng and Dr. Jianhua Yin for the support and advice during my research.

This work was supported in part by the Automotive Research Center (ARC) in accordance with Cooperative Agreement W56HZV-04-2-0001 U.S. Army CCDC Ground Vehicle Systems Center (GVSC), Warren, MI. The support is gratefully acknowledged. Also, many thanks to Prof. Zissimos P. Mourelatos, who have provided me a lot of opportunities and advice.

Last but not least, I would like to thank my parents for their selfless support and faith in me. Without their support, I would not be able to complete this journey.

Table of Contents

Acknowledgments.....	ii
Table of Contents	iii
List of Tables.....	vii
List of Figures	viii
Abstract	xii
1. Introduction.....	1
1.1. Backgrounds	1
1.2. Vehicle mobility model	4
1.3. Vehicle mobility reliability	6
1.4. Path planning	7
1.5. Organization and Contributions	8
2. Mobility Prediction of Off-Road Ground Vehicles Using a Dynamic Ensemble of NARX Models	11
2.1. Background	11
2.1.1. Off-road AGV dynamic Model	11
2.1.2. Nonlinear Auto Regressive eXogenous (NARX) model	15
2.1.3. Long Short-Term Memory (LSTM).....	17
2.2. Dynamic Ensemble of NARX Models with Different Numbers of Lags for Vehicle Mobility Prediction	18
2.2.1. GMM for partition of dynamic domains.....	19
2.2.2. Dynamic ensemble of NARX models (DENA).....	23
2.2.3. Dynamic ensemble of NARX models with different numbers of lags	

(DENA-DIFF).....	25
2.3. Case Study.....	28
2.3.1. Generation of synthetic mobility data.....	29
2.3.2. Comparison of mobility prediction using different surrogate models ..	32
2.4. Summary	37
3. Simulation-Based Mission Mobility Reliability Analysis of Off-Road Autonomous Vehicles.....	39
3.1. Background	39
3.1.1. Off-Road Vehicle Mobility Model.....	39
3.1.2. Mission Mobility Reliability.....	41
3.2. Simulation-Based Off-Road Ground Vehicle MMR Analysis	44
3.2.1. Overview.....	44
3.2.2. Simulation-Based MMR Analysis in Mission-Planning Phase	46
3.2.3. Dynamic Updating of MMR in the During-Mission Phase.....	52
3.3. Case Study.....	58
3.3.1. Description of the MMR Analysis Problem.....	58
3.3.2. Mission-Planning Phase: Surrogate-Modeling Based MMR Analysis .	61
3.3.3. During-Mission Phase: Dynamic Updating of MMR.....	62
3.4. Summary	66
4. Reliability-Based Multi-Vehicle Path Planning Under Uncertainty Using a Physarum-Based Approach	67
4.1. Background	68
4.1.1. Off-Road Vehicle Mobility Analysis Model	68
4.1.2. Deterministic Mission Planning.....	71
4.1.3. Off-Road AGV Mobility Reliability.....	72
4.2. Reliability-Based Mission Planning Using a Bio-Inspired Method.....	76
4.2.1. Adaptive Surrogate Modeling for Model-Based Mobility Reliability	

Analysis.....	77
4.2.2. Physarum solver-based mission planning.....	83
4.2.3. Implementation Procedure.....	90
4.3. Numerical Example.....	90
4.3.1. Maps of Interest.....	91
4.3.2. Generation of Mobility Reliability Map.....	92
4.3.3. Case Study 1: Path Planning for a Single Vehicle.....	95
4.3.4. Case Study 2: Mission Assignment and Planning for Multiple Vehicles.....	97
4.4. Summary.....	98
5. Reliability-Based Path Planning Under Uncertainty with Multiple Constrains Using RRT*-Based Approach.....	100
5.1. Background.....	100
5.1.1. Off-Road Vehicle Mobility Model.....	100
5.1.2. Physics-based M&S.....	102
5.1.3. Path Planning by RRT*.....	103
5.1.4. Challenges in Path Planning of Off-Road AGVs.....	104
5.2. Reliability-Based Path Planning of Off-Road AGVs Subject to Multiple Failure Modes.....	106
5.2.1. Overview.....	106
5.2.2. Reliability Analysis for the Maximum Attainable Speed.....	107
5.2.3. Dynamic Reliability Analysis with Respect to Vertical Acceleration..	111
5.2.4. Reliability-Based Path Planning Subject to Multiple Failure Modes..	117
5.3. Case Study.....	121
5.3.1. Map Information and Soil Properties.....	121
5.3.2. Construction of Mobility Model.....	122
5.3.3. Results of Path Planning.....	124

5.4. Summary	128
6. Conclusion and Future Works.....	130
7. Bibliography	132

List of Tables

Table 2.1 Optimal number of lags for DENA-DIFF (Scenario 1).....	33
Table 2.2 Comparison of mean RMSE and absolute errors of different approaches for ten testing paths using 40 and 80 training paths	37
Table 3.1 Distribution and correlation parameters of slope and soil properties	60
Table 4.1 Input parameters of the SCMDeformableTerrain module	70
Table 4.2 Input parameters of the SCMDeformableTerrain module	78
Table 4.3 Statistical information of different slope/soil parameters.....	91
Table 4.4 Comparison between different approaches for single vehicle path planning...	97
Table 4.5 Comparison between different approaches for multi-vehicle path planning....	98
Table 5.1 Statistical information of different slope/soil parameters.....	122
Table 5.2 Comparison of different methods	128

List of Figures

Figure 1.1 Three steps of autonomous vehicle design	2
Figure 1.2 Overall Structure of The Thesis.....	4
Figure 1.3 Different slope representation of the same location	6
Figure 1.4 Organization of The Thesis.....	8
Figure 2.1 Simulation of an off-road autonomous ground vehicle	12
Figure 2.2 Simulation of AGV mobility in PyChrono	14
Figure 2.3 Partition of the training data and training of NARX models.....	23
Figure 2.4 Proposed prediction approach using dynamic ensemble of NARX models...	25
Figure 2.5 Proposed prediction approach using dynamic ensemble of NARX models with different numbers of lags	28
Figure 2.6 Simulation of HMMWV for the case study.....	29
Figure 2.7 Ten different terrain height maps generated using KL expansion	30
Figure 2.8 A realization of terrain map described as random field	30
Figure 2.9 Five paths of each terrain map for data collection in Scenario 1.....	31
Figure 2.10 (a) Path elevation and (b) vertical acceleration for Path 1 of Figure 2.9.....	32
Figure 2.11 Dirichlet process to estimate the optimal number of clusters: (a) Dirichlet iterative process; (b) posterior distribution of number of clusters.....	33
Figure 2.12 Clustering of the training dataset (Scenario 1)	33
Figure 2.13 Clustering of regions for a test path 1.....	34
Figure 2.14 Comparison of GP-NARX, LSTM, and NARX-Net for a testing path.....	34
Figure 2.15 Comparison of different DENA methods for a testing path	35
Figure 2.16 Comparison of different DENA-DIFF methods for a testing path	35
Figure 2.17 Absolute prediction error comparison of GP-NARX, LSTM, and NARX-Net	

for a testing path.....	35
Figure 2.18 Absolute prediction error comparison of DENA methods for a testing path	36
Figure 2.19 Absolute prediction error comparison of DENA-DIFF methods for a testing path.....	36
Figure 2.20 Prediction error distribution of different methods for test path 5	37
Figure 3.1 Uncertainty propagation of vehicle mobility	41
Figure 3.2 A mission path over a terrain map	43
Figure 3.3 Overview of the proposed framework for simulation-based off-road ground vehicle MMR analysis	45
Figure 3.4 Simulation of a space-dependent variables with different slope ID or soil types	47
Figure 3.5 Relationship between coordinates, soil type, slope parameters, and soil properties.....	49
Figure 3.6 Illustration of a vehicle proceeds on a mission path.....	53
Figure 3.7 Flowchart of the dynamic updating of mission mobility reliability	54
Figure 3.8 Flowchart of generating samples for the unfinished part of the mission path	56
Figure 3.9 Original map from the satellite [79]	59
Figure 3.10 The slope and soil IDs of the selected area.....	60
Figure 3.11 Route chosen from the map (Soil)	61
Figure 3.12 Slope and Soil ID of each coordinate on the path	61
Figure 3.13 Mission mobility failure probability w.r.t. the added number of function evaluations	62
Figure 3.14 Plot of vehicle speed vs. coordinates.....	63
Figure 3.15 Distribution of the soil property parameters at the first point	64
Figure 3.16 Comparison of the estimated soil property parameters and true values	65
Figure 3.17 Updating history of $p_f(\mathcal{Q}) \mathbf{v}_{1:c}$ as the vehicle proceeds on the mission path	65
Figure 3.18 Updating history of the remaining mobilable distance (RMD) on a path	66

Figure 4.1 Process of mission planning for off-road vehicles	67
Figure 4.2 Two simulation examples of PyChrono: Vehicle: (a) gator UTV under deformable terrain, (b) MAN truck under rigid terrain.....	69
Figure 4.3 Simulation flowchart of PyChrono::Vehicle.....	70
Figure 4.4 A simulation model of off-road AGV in PyChrono	71
Figure 4.5 Uncertainty sources in terrain-related parameters	72
Figure 4.6 Uncertainty sources in terrain-related parameters	74
Figure 4.7 Vehicle mobility simulation model to predict the maximum attainable speed	78
Figure 4.8 Flowchart of adaptive surrogate modeling-based reliability analysis	80
Figure 4.9 Flowchart of mobility reliability-based mission planning.....	82
Figure 4.10 Comparison of uniform mesh and navigation mesh.....	84
Figure 4.11 Graphical representation of the maze using a network.....	85
Figure 4.12 Flowchart of the Physarum solver algorithm.....	87
Figure 4.13 Flowchart of the proposed approach.....	90
Figure 4.14 (a) Slope map and (b) Soil map of interest.....	91
Figure 4.15 HAMMV vehicle used in this numerical example	92
Figure 4.16 (a) Convergence history of U_{min} in adaptive surrogate modeling; (b) Convergence history comparison of adaptive surrogate modeling and the commonly used variance minimization-based surrogate modeling method	93
Figure 4.17 (a) Mean speed map and (b) SMR map	94
Figure 4.18 GO/NO-GO map (a) $R_t = 0.8$ (b) $R_t = 0.9$ and (c) $R_t = 0.95$	94
Figure 4.19 Navigation mesh of the map	95
Figure 4.20 (a) Original map and (b) Dynamic obstacle in the map.....	96
Figure 4.21 (a) Shortest path, (b) Dynamic path planning from the middle, and (c) Dynamic path planning from the starting point.....	96
Figure 4.22 (a) Original map and (b) Dynamic obstacle of the map	97
Figure 4.23 (a) original mission assignment and (b) dynamic mission assignment	98
Figure 5.1 Illustration of path planning for Off-road AGV	102

Figure 5.2 Illustration of tree-rewriting in RRT*	104
Figure 5.3 Uncertainty sources and its impact on AGV mobility	105
Figure 5.4 Overview of the proposed framework	107
Figure 5.5 An example of simulating movement of an AGV in off-road environment in PyChrono	109
Figure 5.6 Effects of soil parameters on the vertical acceleration	112
Figure 5.7 Flowchart of Dynamic Ensemble of NARX model (DENA)	116
Figure 5.8 (a) Height map and (b) Soil map of interest	122
Figure 5.9 The results of adaptive GP and comparison with other methods.....	123
Figure 5.10 The height contour and generated eight paths	124
Figure 5.11 Comparison between DENA and GP-NARX	124
Figure 5.12 Path planning by different methods; (a) RRT*; (b) R-RRT*; (c) MC-RRT*; and (d) MC-RRT*-2.....	125
Figure 5.13 Illustration of a high-risk area of mobility loss	126
Figure 5.14 Illustration of bumpy terrain condition.....	127
Figure 5.15 The MCS results of RRT* with 5,000 samples	128

Abstract

Off-road autonomous ground vehicles (AGVs) and other types of robotics are drawing increased attention in recent years as they are able to replace human in dangerous or boring working environments such as in the presence of wildfires or earthquakes, battlefield, and agricultural field. Mission planning of off-road AGVs plays a vital role in ensuring the successful and autonomous operation of AGV in the off-road environment. The unstructured off-road environment and various sources of uncertainty pose challenge to mission planning of off-road AGVs.

This research focuses on mission planning of off-road AGVs with mobility reliability considerations. The goal is to identify a path that is not only shortest in terms of travel distance, but also reliable to ensure the success of a mission. It is a systematic work that contains three main steps: (i) vehicle mobility modeling, (ii) mobility reliability analysis, and (iii) reliability-based path planning. This dissertation aims to provide novel approaches to properly perform the three steps. We firstly construct a dynamic ensemble of Nonlinear Autoregressive Network with Exogenous inputs (NARX) models over time to accurately prediction mobility of off-road AGVs. Secondly, a simulation-based mission mobility reliability (MMR) analysis framework is developed to account for uncertainty in mobility prediction of off-road AGVs in mission planning phase and a dynamic updating scheme is proposed to update the MMR estimation using online mobility data. Then the adaptive surrogate modeling is used to calculate the state mobility reliability and incorporate the soil information into mobility reliability map. A reliability-based path planning method is developed for both single and multiple vehicle using a Physarum-based algorithm and the navigation map. Finally, a rapidly-exploring random trees star (RRT*) algorithm is studied

to account for multiple the mission mobility reliability (MMR) constraints in mission planning path. Multiple case studies are used to demonstrate the effectiveness of the proposed approaches. The results show that the proposed approaches can effectively identify optimal paths while satisfying certain mission mobility reliability requirements.

Chapter 1

Introduction

1.1. Backgrounds

In recent years, Autonomous vehicle (AV) is considered as one of the game-changing technologies that will change the transportation styles of current society. Research shows that autonomous vehicle has promising potential in improving road safety by eliminating human errors and easing traffic congestions through optimal control. As one specific application of the autonomous vehicle, the off-road autonomous ground vehicles (AGVs) are drawing increased attention as they can work in dangerous situations such as wildfires or earthquakes and replace humans in fields such as agricultural and battlefield.

For an autonomous vehicle, the main purpose is to find the optimal path from a beginning location to a target location (path/mission planning). However, the area for path/mission planning is not always stable. There are usually uncertainties in the area. As a result, quantifying the uncertainty and analyzing the reliability of path/mission planning for autonomous vehicles is vital to ensure the safety of autonomous vehicles. The design of reliable autonomous engineering systems is a challenging issue since it requires the consideration of autonomous operations into design to guarantee the reliability of the system during operation. To better perform reliability analysis, a proper mobility model of the vehicle is also important. Therefore, the development of autonomous vehicles can be divided into three steps including mobility modeling, reliability analysis and path planning algorithm.

The relation between three steps is shown in Figure 1.1. We use the mobility model to calculate the mobility reliability and also use the reliability result to update the mobility

model. Also, we use the reliability model to guide our path planning and make sure the planned path maintains the expected reliability. The three steps combine with each other during the design process.

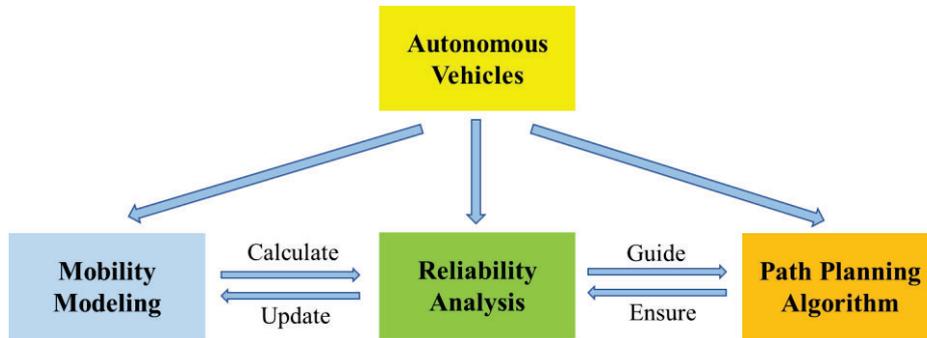


Figure 1.1 Three steps of autonomous vehicle design

To modify the problem of the autonomous vehicles, vehicle mobility analysis and its modeling and simulation (M&S) are important steps in developing and operating off-road vehicles. M&S tools to predict the capability of a vehicle to go through a certain area can provide important guidelines in designing off-road ground vehicles and planning operations. A vehicle mobility model is an analytical or simulation model that is used to predict the vehicle mobility characteristics for given environment conditions and vehicle conditions. To obtain the mobility model, a series of testing and simulation scenarios or standards are needed.

The format of the mobility model can be different in different applications. For off-road vehicles, the mobility characteristics can be the vehicle speed or acceleration. The environment input can be the terrain condition that the vehicle is running on. And the vehicle input can be the vehicle design and control parameters such as vehicle weight, length, throttle and brake. In order to make good use of off-road ground vehicle modeling, the U.S. Army Tank Automotive Research, Development, and Engineering Center (TARDEC) which is now called Ground Vehicle Systems Center (GVSC), developed the NATO Reference Mobility Model (NRMM) in the 1960s. Other than the developed NRMM and its upgrades, a Next-Generation NRMM (NG-NRMM) using multibody

dynamic modeling and new simulation techniques is being developed to further improve the prediction capability and accuracy of the mobility models.

Before the mass production and the deployment of autonomous vehicles in real environment, the safety and reliability of AVs are recognized as one of the most important factors that need to be assured. There are different uncertainty sources in the mobility model. For on-road vehicles, they can be the uncertainty in the traffic such as the behavior of other drivers and the uncertainty in the road condition or weather condition. For off-road vehicles, they can be the uncertainty in vehicle design and the uncertainty in the terrain.

The mobility reliability is defined as the probability that the vehicle mobility characteristics is within a certain range so that the vehicle can safely operate during the mission. It can be solved by combining the mobility model and uncertainty sources.

After obtaining a mobility model and clarify the mobility reliability, the next step is to perform a path planning for the vehicles. For a given vehicle design, a properly planned path using model-based approach will help the autonomous vehicle to better accomplish the mission in the post-design stage with higher speed, lower energy consumption and higher reliability.

In conclusion, the structure of the thesis is shown in Figure 1.2. For the mobility model, it includes simulation model using different software or package and surrogate model such as adaptive sampling and piecewise Gaussian process. For reliability analysis, it can be separated by number of constrains or by the definition of reliability function. And the constrain can also be different depends on different purpose or focus of the autonomous vehicles. For the path planning, it includes different types of algorithms and need to be combined with mobility modeling and reliability analysis.

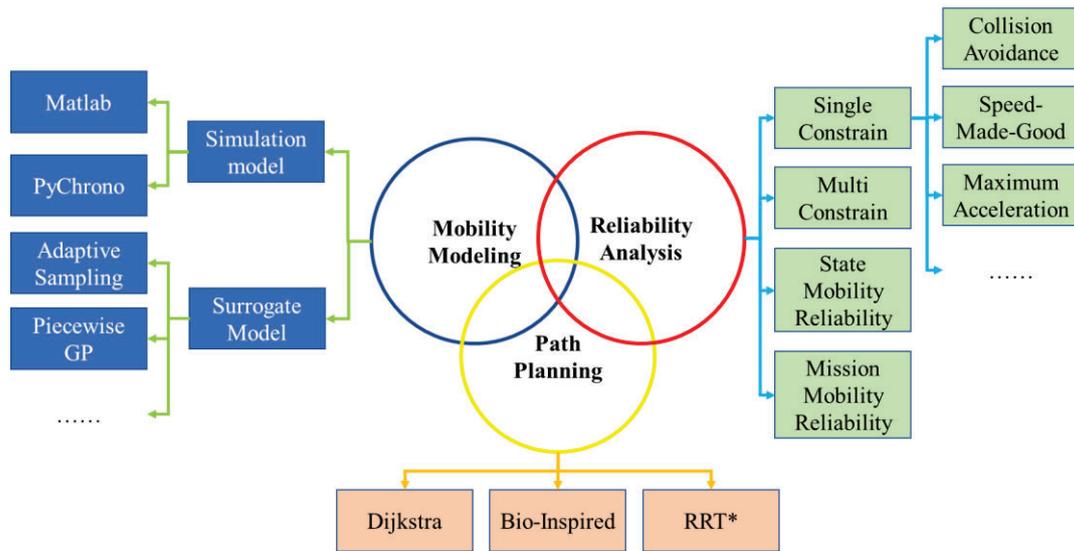


Figure 1.2 Overall Structure of The Thesis

1.2. Vehicle mobility model

In order to make good use of off-road ground vehicle M&S, the U.S. Army Tank Automotive Research, Development, and Engineering Center (TARDEC) which is now called Ground Vehicle Systems Center (GVSC), developed the NATO Reference Mobility Model (NRMM) in the 1960s [1-5]. Other than the developed NRMM and its upgrades, a Next-Generation NRMM (NG-NRMM) using multibody dynamic modeling and new simulation techniques is being developed to further improve the prediction capability and accuracy of the mobility models [3, 6, 7].

In addition to the NRMM and NG-NRMM, many other mobility analysis models have also been developed in the past decades to predict ground vehicle performance under different off-road conditions. For instance, using Bekker's equations, an analytical tool called Bekker's Derived Terramechanics Model (BDTM) has been developed to evaluate vehicle off-road mobility [8]. Recuero et al. [9] developed a mobility simulation model using nonlinear finite element tires on granular material. Serban et al. [10] proposed a co-simulation framework for vehicle-terrain interaction simulation. By developing a computer program called WinMaku, Korlath et al. [11] performed a simulation of mobility performance for off-road vehicles. Also, Hetherington et al. [12] suggested a model using

mean maximum pressure (MMP) to specify off-road performance of vehicles.

Similarly, for NG-NRMM, the stochastic mobility modeling is still one of the most important topics in mobility prediction [13]. Several approaches have been proposed to meet the need of performing stochastic mobility prediction in NG-NRMM. For example, using a hardware-in-the-loop real-time simulator, Iagnemma and Gonzalez [6] proposed a Monte Carlo simulation method (MCS) to predict the vehicle performance under uncertainty; Gonzalez et al. used the Kriging surrogate modeling approach to model terrain elevation in predicting off-road mobility over large spatial regions [14]; they also used Kriging surrogate model to develop a stochastic mobility map [15]; a dynamic Kriging-based method is developed in Ref. [13] to improve the effectiveness of Kriging surrogate modeling in generating stochastic off-road mobility map; Detweiler et al. [16] developed interpolation approaches for terrain maps to get a high-fidelity 3D terrain surface.

For mobility modeling, the first challenge is how to obtain the input and output of the mobility model. The direct way is set a series of test using real vehicles. But this is too costly. So a simulation model is the best alternative. However, running a high-fidelity simulation still cost too much time and computational effort. So how to effectively reduce the number of function evaluations to the computer simulation model or the number of vehicle tests becomes important. The second challenge is about surrogate modelling. Even though we have different type of surrogate models, and some can handle large number of dataset, but different types of data represent different mobility situations. Directly feeding all the data to a surrogate model at once will lower the accuracy of the model and increase the computational effort. So we need to properly separate the dataset to train a better surrogate model. The third challenge is about the reality of the mobility model input. In many mobility predictive models, the slope is one important factor that will greatly affect the vehicle performance. However, when we try to generate the vehicle mobility of a certain location or among the mission area, we will find that the slope at any location depends on

the vehicle as shown in Figure 1.3. As a result, the slope of a location can only be observed after the vehicle moving direction at that location has been provided in the first place. For the same reason, there won't be any slope map in real applications. The information we can obtain before path planning is the height map. So, it's important to transfer the height information into slope in order to perform the mobility analysis and reliability-based path planning.

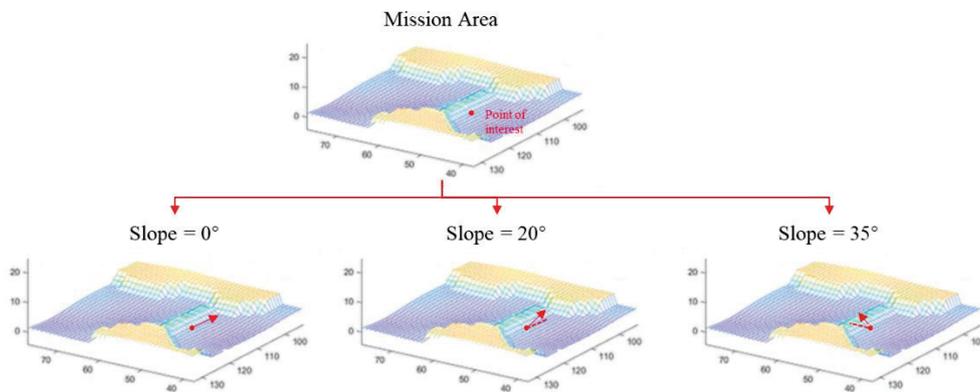


Figure 1.3 Different slope representation of the same location

1.3. Vehicle mobility reliability

In reality, there are many sources of uncertainty such as the uncertainty in vehicle parameters, uncertainty in soil property, and uncertainty in the soil map [17], which lead to uncertainty in the predicted off-road vehicle mobility. In order to account for uncertainty in mobility prediction, many methods have been developed. Lessem et al. [18, 19] and Priddy et al. [20] converted deterministic NRMM into a stochastic model so that variability of vehicle mobility can be included in the prediction. Haug et al. [21] described the domain of mobility when an object is moving in an area with obstacles. Choi et al. [22] proposed a framework to construct a reliability-based stochastic mobility map. Jayakumar et al. [23] built a method to generate a mobility map using machine learning algorithms such as support vector machines and neural networks. Hu et al. [17] established a testing design optimization approach to reduce the uncertainty when generating the mobility map.

For mobility reliability analysis, the first challenge is how to quantify the uncertainty. We need to consider the correlation of the uncertainty over the mission area. The second challenge is how to use the reliability refine the mobility model and further more reduce the number of simulation or test. The third challenge is how to update the reliability if new observation is given.

1.4. Path planning

For reliability-base path planning Yang [24] proposed a Nonperiodic B-Spline Curves-based path planning approach for autonomous robots or vehicles to avoid obstacles in the mission area. Rapidly-exploring random trees (RRT) has been used for path planning considering uncertainty in the terrain [25]. Liu et al. [26] proposed an A-star based approach to find a robust path in complex off-road environment. Sun et al. [27] introduced a stochastic extended linear quadratic regulator (SELQR) approach by minimizing the expected value of a user-defined cost function and modeling the uncertainty with a Gaussian distribution. A linear-quadratic Gaussian motion planning (LQG-MP) with RRT has been developed by Berg [28] to generate a proper path under Gaussian uncertainty. Zhang et al. [29] proposed a geometric reinforcement learning algorithm for path planning by considering risk in the reward matrix. Chao et al. [30] introduced a grid-based RRT* approach that provides a path with minimum risk in the dangerous area. Moreover, in order to consider the dependence of the uncertain terrain-related parameters over the target map, Jiang et al. [31] proposed a R2-RRT* method by combining a state mobility reliability (space independent) and mission mobility reliability (space dependent)) to identify an optimal path satisfying a required mission reliability level.

For the path planning, the first challenge is how to use the mobility reliability model and uncertainty sources to transfer the mission area into a map that is ready to apply the path planning algorithms. The second challenge is how to make sure the planned path is

overall reliable. The third challenge is how to perform a multi-vehicle path planning with self-updating function that can handle dynamic problems.

1.5. Organization and Contributions

This dissertation aims to overcome the previously mentioned challenges and provide a complete process of autonomous off-road dissertation vehicle under uncertainty. The organization of the thesis is summarized in Figure 1.4.

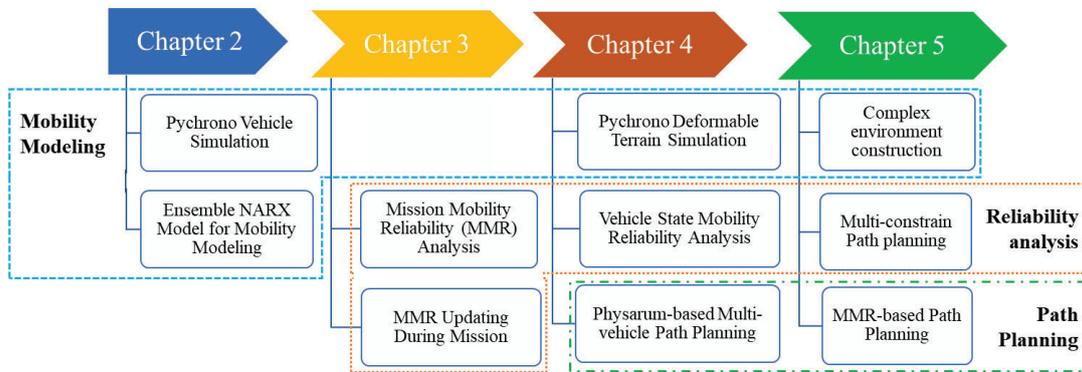


Figure 1.4 Organization of The Thesis

Chapter 2 uses a dynamic ensemble of Nonlinear Autoregressive Network with Exogenous inputs (NARX) models over time. The data is partitioned into different segments using a variational Gaussian mixture model to represent different vehicle dynamic behaviors. Based on the partitioned data, multiple surrogate models are constructed under the NARX framework with different numbers of lags.

Chapter 3 develops a simulation-based mission mobility reliability (MMR) analysis framework to account for uncertainty in mobility prediction of off-road ground vehicles in mission planning. A single-loop Kriging surrogate modeling method is employed for MMR analysis and a dynamic updating scheme is proposed to update the MMR estimation using online mobility data during a specific mission and for a particular vehicle.

Chapter 4 builds a novel bio-inspired approach is provided for model-based multi-vehicle mission planning under uncertainty for off-road AGVs subjected to mobility

reliability constraints in dynamic environments. Based on physics-based simulations, the vehicle state mobility reliability in operation is then analyzed using an adaptive surrogate modeling method. Subsequently, a bio-inspired approach called the Physarum-based algorithm is used in conjunction with a navigation mesh to identify an optimal path satisfying a specific mobility reliability requirement.

Chapter 5 use Pychrono as the simulation model for both mobility data extraction and path planning execution. Multiple constrains (e.g. vehicle speed, acceleration or fuel consumption) are used to generate the mobility reliability of the vehicle in the area. Then a rapidly-exploring random trees star (RRT*) algorithm is performed based on the mission mobility reliability (MMR) to find the optimal path. The algorithm is furtherly simplified using the GO/NO-GO map generated by the state mobility reliability (SMR).

The main contributions of the thesis are summarized as follows:

For mobility modeling

- A high-fidelity and open-source simulation model is selected to simply generate off-road mobility data and test the path planning approaches. A dynamic ensemble of NARX models is used for the off-road mobility modeling to improve the accuracy of the surrogate model.

For reliability analysis

- SMR is proposed to represent the mobility reliability of a certain location so that a GO/NO-GO can be generated for future path planning.
- MMR is proposed to properly represent the overall mobility reliability of the whole path so the results are more trustable.
- A single-loop Kriging surrogate modeling method and an adaptive surrogate method are employed to reduce the calculation effort. The MMR can be easily updated with the new observation during the mission phase.

For reliability-based path planning

- Different path planning algorithms have been applied to provide the optimal path.
- The planned path can automatically be redirected to the new optimal path if obstacles appear.
- More than one reliability constraints are considered during the path planning process to guarantee the MMR of the planned path.

Chapter 2

Mobility Prediction of Off-Road Ground Vehicles Using a Dynamic Ensemble of NARX Models

As the first step of path planning for off-road AGVs, the mobility modeling is firstly addressed in this chapter. This chapter aims to overcome the challenges in surrogate modeling of vehicle mobility with non-stationary dynamic behaviors by developing a dynamic ensemble framework of Nonlinear Autoregressive network with Exogenous inputs (NARX) models using a variable number of lags. Multiple NARX models are trained for different dynamic behaviors based on clustering of the training data. The NARX models are then assembled together dynamically over time to predict the dynamic prediction of AGV mobility over a long time period. The accuracy of the proposed framework is investigated using multiple machine learning models including Gaussian process regression and neural networks. A case study demonstrates the advantages of the proposed surrogate modeling method using a dynamic ensemble of NARX models over several classical surrogate modeling methods. This part has been published at:

- [32] Liu, Y., Barthlow, D., Mourelatos, Z. P., Zeng, J., Gorsich, D., Singh, A., and Hu, Z., 2022, "Mobility Prediction of Off-Road Ground Vehicles Using a Dynamic Ensemble of NARX Models," *Journal of Mechanical Design*, 144(9), p. 091709.

2.1. Background

2.1.1. Off-road AGV dynamic Model

2.1.1.1. The need of a data-driven dynamic model

The importance of understanding and being able to determine the response of complex dynamic systems has long been recognized [33]. Analysis traditionally begins with mathematical relations and physical laws that work as descriptions and representations of

the various behaviors of interacting components within systems. However, in modeling the dynamic behavior of systems, the governing physical laws are not always available, and explicit analytical solutions are not always tractable. This has motivated an increasing amount of research towards data-driven methodologies using of data-driven analysis and modeling [34].

The off-road AGV of Figure 2.1 is an example of a complicated dynamic problem, involving complex geometries, nonlinear interactions within components, and often stochastic interactions with the environment. While advancements in software and computation have promoted the use of techniques such as finite-element modeling and other numerical simulations, problems in this field are commonly broken into parts and addressed in a limited sense using simplified math models. This is often done by allowing key system characteristics to be represented with simple continuously deformable bodies, concentrated masses, and idealized springs and dampers, such that the often nonlinear dynamics are modeled and analyzed [33]. Many vehicle dynamics simulation codes are constructed and operated in this manner.

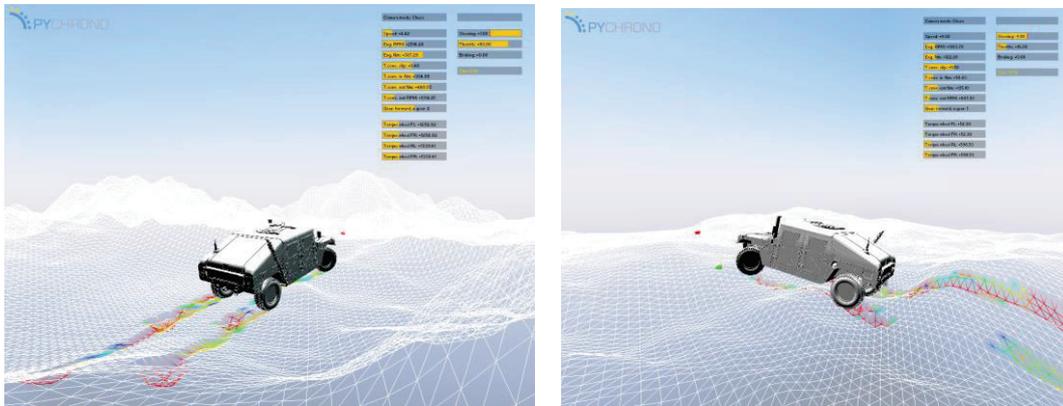


Figure 2.1 Simulation of an off-road autonomous ground vehicle

This chapter utilizes one such software, a Python version of Chrono: An Open-Source Multi-physics Dynamics Engine, to simulate vehicular dynamics in an off-road environment [35]. Using a collection of information and data pertaining to the terrain morphology, the vehicle motion is provided as output of the software. We use the latter to build a data-driven approach to meta-model the complex dynamics of a high mobility multipurpose wheeled vehicle traversing over an off-road terrain. The data-driven model drastically reduces the required computational effort for AGV mobility prediction, and thus supports reliability analysis and AGV mission planning under uncertainty [36].

2.1.1.2. Off-road AGV mobility simulation using PyChrono

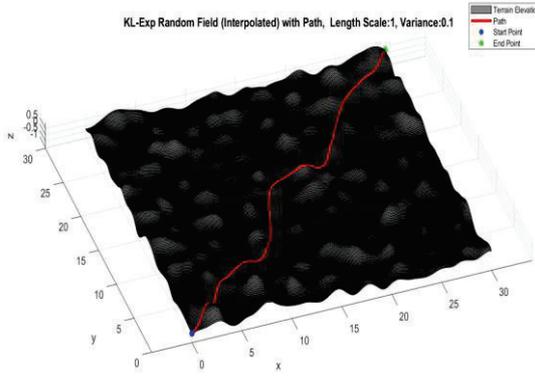
PyChrono (a Python version of Chrono) is an open-source multi-physics dynamics engine. The software specializes in handling complex and large dynamic systems with many rigid bodies that interact through various contact methods [35, 37]. Chrono is capable of solving problems involving multiple physics regimes simultaneously. For example, it can compute the dynamics of interconnected rigid bodies (such as moving parts of a mechanism) whilst computing the dynamics of deformable bodies (such as terrain-mechanics). For this reason, it has been widely employed for vehicle simulation research. Multiple “demo vehicles” and a “demo terrain” are currently included within its installation files. This allows users to easily begin simulating vehicles in a stochastic environment and obtain synthetic mobility data.

A user begins a vehicle simulation by specifying several inputs: a vehicle model type, vehicle parameters (suspension, driveline type, etc.), a terrain height map, a terrain type (rigid or deformable), a contact method for the vehicle-terrain interaction, initial location and orientation of the vehicle on the terrain, and several additional inputs on how the vehicle is powered and controlled throughout the simulation. The latter input can include for example a fixed or variable throttle, manual controls, or a speed and/or steering controller to follow a predefined path on the terrain. In addition, parameters to control the numerical computations, such as mesh/grid size of the deformable terrain, and simulation time step size are input by the user.

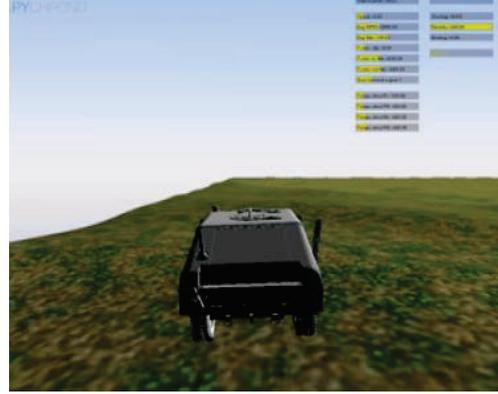
A large selection of output data can be extracted using the PyChrono vehicle simulation. Essential kinematic information can be recorded for multiple parts of the vehicle such as the global position of the vehicle’s center of mass, position of the driver, vehicle chassis velocity, Center of Mass (CoM) velocity, vehicle driver’s acceleration and CoM acceleration, wheel velocity, and vehicle rotation. Additional information can be captured involving the vehicle mechanisms such as steering, driveshaft speed, axle torque, wheel force, powertrain output, etc. For our research, the vehicle’s vertical acceleration is obtained and used as an output. The reason for selecting vertical acceleration as the variable of interest is that for off-road autonomous vehicles, it represents the bumping of the vehicle. When the vehicle is transporting people or cargo during the mission, the vertical acceleration can’t be too high or the people inside the vehicle will get injured or the cargo

will get damaged. So, the accurate prediction of the vertical acceleration of off-road autonomous vehicle will help the decision-maker to find a proper mission path.

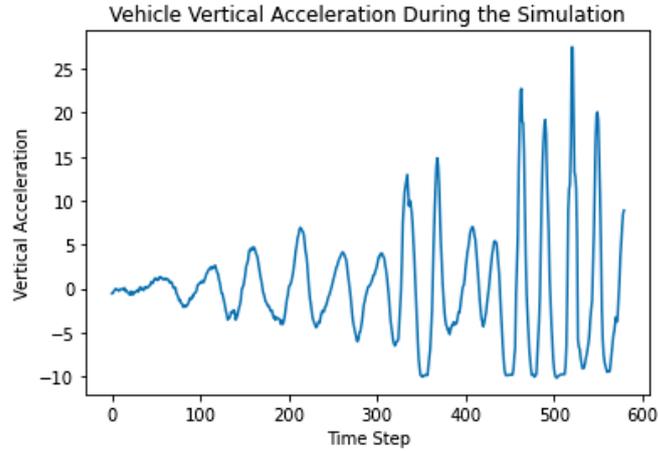
As indicated in Figure 2.2, PyChrono uses a terrain map (Figure 2.2(a)) and a vehicle model (Figure 2.2 (b)) to predict various vehicle mobility quantities, such as vertical acceleration (Figure 2.2(c)) for a given path.



(a) Terrain map and a path



(b) PyChrono simulation



(c) Predicted AGV vertical acceleration over a path

Figure 2.2 Simulation of AGV mobility in PyChrono
Mathematically, the PyChrono simulation model can be represented as

$$\mathbf{a} = g(\mathbf{u}) + \boldsymbol{\varepsilon}, \quad (2.1)$$

where $g(\cdot)$ stands for the PyChrono simulation model, $\mathbf{u} = [u_1, u_2, \dots, u_N]$ is a vector of N discretized elevation values over a path, $\boldsymbol{\varepsilon}$ is a vector of simulation noise values due to numerical discretization and model assumptions, and $\mathbf{a} = [a_1, a_2, \dots, a_N]$ is the vector of the predicted vehicle vertical accelerations. Note that the length of \mathbf{u} is different for

different paths and therefore N varies with the path.

In this chapter, the random terrain elevation is modeled as a random field using the Karhunen–Loève (KL) expansion method. For a given coordinate \mathbf{d} on the map (Figure 2.2(a)), the terrain elevation is represented by

$$Y_u(\mathbf{d}) = \mu_u(\mathbf{d}) + \sigma_u(\mathbf{d}) \sum_{l=1}^{N_{KL}} \sqrt{\lambda_l} \xi_l \eta_l(\mathbf{d}), \quad (2.2)$$

where $\mu_u(\mathbf{d})$ and $\sigma_u(\mathbf{d})$ denote the mean and standard deviation of the terrain elevation at \mathbf{d} respectively, ξ_l , $l = 1, 2, \dots, N_{KL}$ are independent standard normal random variables, N_{KL} is the number of expansion terms, and λ_l and η_l are the eigenvalues and eigenvectors of a correlation matrix whose elements are given by

$$\rho(\mathbf{d}_i, \mathbf{d}_j) = \exp\left[-\beta_1(d_{1i} - d_{1j})^2 - \beta_2(d_{2i} - d_{2j})^2\right], \quad \forall i, j = 1, 2, \dots, N_d, \quad (2.3)$$

where $\mathbf{d}_i = [d_{1i}, d_{2i}]$ are the spatial coordinates at location \mathbf{d}_i , and N_d is the total number of discrete spatial points in the map used for KL expansion.

Even though the PyChrono simulation can effectively predict various vehicle mobility quantities, the computation is expensive if a very fine mesh is used. The high computational cost hinders its application in reliability analysis and mission planning under uncertainty. A data-driven model must be constructed to replace the high-fidelity simulation model.

2.1.2. Nonlinear Auto Regressive eXogenous (NARX) model

Many dynamic systems evolve over space and time. To describe and analyze such systems, appropriate methods must be employed. Most dynamic systems are modeled using a state variable representation providing the evolution of state variables through time. This implies utilizing a collection of various time-series representing the system input and output.

Time series prediction and forecasting is often done using autoregressive modeling, which accounts for potential for relationship (correlation) within the sequence of values. An autoregressive model uses a basis where the predictands are previous occurrences of the predictor. If the basis is a linear combination of previous outputs, it is called an AR

model and is given by

$$y_i = c_1 y_{i-1} + c_2 y_{i-2} + \dots + c_q y_{i-q}, \quad (2.4)$$

or

$$y_i = \sum_{j=1}^q c_j y_{i-j}, \quad (2.5)$$

where y_i is the output (state variable) of interest, i is an time index up to q steps or “lags”, and $c_j, j = 1, \dots, q$ are the coefficient terms to fit the model. A simple example is the rule to generate the Fibonacci sequence where any term is computed as the sum of the two previous terms.

If the model parameters are adequately fit (typically done by minimizing an error metric using a data training set), the model can be used for prediction. The value at a future time step can be then computed using the values of previous steps. A computer code can iterate this process allowing the model to march forward in time for forecasting. It should be mentioned, that the prediction phase may need to assume prior values (up to q lags) to initiate the simulation and continue with forward propagation.

Such models can also incorporate external terms as input if they affect the state variable of interest. These are called *exogenous* terms and can be similarly included by appending a similarly structured sequence of exogenous lags as additional predictands as

$$y_i = \sum_{j=1}^q c_j y_{i-j} + \sum_{k=1}^p b_k u_{i-k}, \quad (2.6)$$

in which $u_{i-k}, k = 1, \dots, p$ is the exogenous input, and $b_k, k = 1, \dots, p$ are input coefficients to fit the model. Note that if the present time exogenous input (i.e. $k = 0$) is known, it can also be included as a predictor.

Lastly, if a non-linear basis of terms is used for prediction, or if u and y are used as arguments of a nonlinear function, the method (and naturally the name) becomes *Nonlinear Auto Regressive eXogenous* (NARX) modeling as

$$y_i = F(u_{i-1}, \dots, u_{i-p}; y_{i-1}, \dots, y_{i-q}), \quad (2.7)$$

where the function $F(\cdot)$ is a nonlinear mapping from lags of previous outputs and inputs to the current output. This nonlinear function can be for example, a polynomial, a Gaussian process regressor (GP-NARX) [38], or a feed forward artificial neural network (NARX-Net) [39].

Modeling of this kind is akin to finite difference approximations of differentiation, and numerical methods in solving differential equations. For this reason, NARX models are particularly preferred if there are underlying or otherwise undiscovered differential relations governing the behavior of the system, which is often the case for physics-based dynamic problems. Other related techniques have also been developed and have shown success in time series modeling of dynamic systems. Examples are the Dynamic Mode Decomposition (DMD), the popular SINDy algorithm, system identification methods, and a host of many different recurrent neural network architectures (LSTM, GRU, etc.) [34, 40]. All of these techniques share the common presumption that there exists a tractable relationship between a variable's future value in time and its previous values in time. Some other methods are mathematically similar [41, 42]. Great success in modeling even complex and highly nonlinear dynamical systems has been achieved using techniques such as in [34, 40, 43]. However, extrapolating to forecast significantly far into the future of a system's evolution, and still retaining good prediction accuracy, is still an active research topic.

In this chapter, both the GP-NARX and NARX-Net frameworks will be investigated in modeling and forecasting the vertical acceleration of an AGV as it traverses over an off-road terrain. In addition to GP-NARX and NARX-net, the proposed method is also be compared with LSTM which has gained much attention in recent years for modeling of various dynamic systems.

2.1.3. Long Short-Term Memory (LSTM)

LSTM is a variant of a recurrent neural network. It has attracted much interest in dynamic system modeling, since it allows a model to sequentially learn and determine whether to forget previous states and/or update the states. A LSTM module first determines what previous information to forget using a “forget gate” [44, 45] and then determines what new information \tilde{C}_t to store using an “input gate” and a tanh layer. Based on this, an

LSTM cell updates the new cell state C_t by combining the information inherited from the previous cell state C_{t-1} with the new candidate information \tilde{C}_t . Finally, the output of an LSTM cell is obtained.

The four-step operation of an LSTM cell is given below [44, 45]

$$\begin{aligned}
\text{Step 1: } f_t &= \sigma(W_f[h_{t-1}, x_t] + b_f); \\
\text{Step 2: } i_t &= \sigma(W_i[h_{t-1}, x_t] + b_i) \\
\tilde{C}_t &= \tanh(W_C[h_{t-1}, x_t] + b_C); \\
\text{Step 3: } C_t &= f_t \times C_{t-1} + i_t \times \tilde{C}_t; \\
\text{Step 4: } o_t &= \sigma(W_o[h_{t-1}, x_t] + b_o) \\
h_t &= o_t \times \tanh(C_t),
\end{aligned} \tag{2.8}$$

where W_f , W_i , W_C and W_o denote the weight matrices of each unit, and b_f , b_i , b_C and b_o are the bias terms associated with each unit.

2.2. Dynamic Ensemble of NARX Models with Different Numbers of Lags for Vehicle Mobility Prediction

As shown in Figure 2.2(c), the vertical acceleration of an AGV in off-road environment exhibits a non-stationary behavior. This poses challenges for surrogate modeling of vehicle mobility, resulting in a poor performance for GP-NARX, NARX-net, and LSTM as indicated in the case study of Section 2.3.

Different approaches have been developed in the past decade and recently to deal with non-stationary response in surrogate modeling, such as a non-stationary Gaussian process regression model [46], hierarchical multiscale surrogate models [47, 48], and neural network-based methods [49]. The current methods, however, are not generalizable and are difficult to implement in practice. For instance, the approach of non-stationary Gaussian process cannot be applied to neural network-based models. Moreover, most of the current methods focus on static, quasi-static, or univariate time series modeling. In this section, we propose an alternative approach to the existing methods. The proposed dynamic ensemble

of NARX models is easy-to-implement and generalizable to different types of models including both GP-based and neural network-based methods since it can capture different patterns of the dynamic behavior of the system.

The basic idea of the proposed method is to first partition the prediction domain into different segments. This is achieved through clustering techniques using a Gaussian Mixture Model (GMM). Using the partition of the prediction domain, different NARX models are constructed and are then ensembled together dynamically over time to predict vehicle mobility. Two strategies are proposed under this dynamic ensemble framework: (1) a Dynamic Ensemble of NARX models (DENA) with a fixed number of lags, and (2) a Dynamic Ensemble of NARX models with a DIFFerent number of lags (DENA-DIFF). In what follows, we explain the proposed framework in detail.

2.2.1. GMM for partition of dynamic domains

If N_p paths are used to generate synthetic mobility data for the off-road AGV, we have the terrain elevation along the N_p training paths as $u_{i,j}, i=1, 2, \dots, N_p; j=1, 2, \dots, N_i$, where N_i is the length of the i -th training path. The mobility simulation model of Section 2.1.1 is run for each of the N_p training paths to obtain the corresponding vertical acceleration of the AGV as $a_{i,j}, i=1, 2, \dots, N_p; j=1, 2, \dots, N_i$. According to the NARX scheme of Section 2.1.2, the synthetic mobility data of the N_p paths can be converted into a training data matrix as

$$\mathbf{x}_T = \begin{bmatrix} \mathbf{u}_1 & \mathbf{a}_1 \\ \mathbf{u}_2 & \mathbf{a}_2 \\ \vdots & \vdots \\ \mathbf{u}_{N_T} & \mathbf{a}_{N_T} \end{bmatrix}, \mathbf{a}_T = \begin{bmatrix} a_{T1} \\ a_{T2} \\ \vdots \\ a_{TN_T} \end{bmatrix}, \quad (2.9)$$

in which \mathbf{x}_T is the input training data matrix including previous lags of both terrain elevation and vehicle accelerations, \mathbf{a}_T is the output training data matrix (i.e. vertical acceleration of AGV). The vectors $\mathbf{u}_1 = [u_{11}, u_{12}, \dots, u_{1p}] \in \mathbb{R}^{1 \times p}$ and $\mathbf{a}_1 = [a_{11}, a_{12}, \dots, a_{1q}] \in \mathbb{R}^{1 \times q}$ are respectively the first p and q time steps of the first training

path, and the corresponding output is $a_{T1} \triangleq a_{1(q+1)}$. The remaining rows of the training data in Eq. (2.9) can be explained similarly. Finally, N_T is the total number of training data collected from the N_p paths.

Since the vehicle has a different dynamic behavior in different segments of a path, we partition the training data of Eq. (2.9) into different segments using a Gaussian Mixture Model (GMM). A GMM is a unsupervised machine learning method, which approximates the joint probability density function (PDF) of multi-variate data as a weighted sum of multi-variate Gaussian components as [50]

$$f(\mathbf{x}_T) = \sum_{i=1}^Q w_i \phi(\mathbf{x}_T | \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i), \quad (2.10)$$

where \mathbf{x}_T is the vector of input variables (terrain elevation and vehicle acceleration) with a dimension of $p+q$, Q is the number of Gaussian components, w_i is the weight of the i -th Gaussian component, $\sum_{i=1}^Q w_i = 1$, and $\phi(\mathbf{x}_T | \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$, $i = 1, 2, \dots, Q$ is the PDF of the i -th multi-variate Gaussian distribution

$$\phi(\mathbf{x}_T | \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) = \frac{1}{(2\pi)^{N/2} |\boldsymbol{\Sigma}_i|^{1/2}} \exp \left\{ -\frac{1}{2} (\mathbf{x}_T - \boldsymbol{\mu}_i)' \boldsymbol{\Sigma}_i^{-1} (\mathbf{x}_T - \boldsymbol{\mu}_i) \right\}, \quad (2.11)$$

in which $\boldsymbol{\mu}_i$ is the mean vector and $\boldsymbol{\Sigma}_i$ is the covariance matrix.

The application of GMM entails the estimation of (1) the GMM parameters, and (2) the optimal number of components/clusters. A commonly used and matured technique for the GMM parameter estimation is the Estimation-Maximization (EM) algorithm [51]. However, the EM algorithm is sensitive to the starting point and may converge to a local optimum. In this chapter, we use the variational inference (VI) method to overcome this limitation. In VI, the variational posterior distribution of GMM parameters is expressed as a multiplication of a Gaussian and a Wishart distribution as [52]:

$$f(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) = N(\boldsymbol{\mu}_i | \mathbf{m}_i, (\beta_i \boldsymbol{\Sigma}_i)^{-1}) W(\boldsymbol{\Sigma}_i | \mathbf{O}_i, \nu_i), i = 1, \dots, Q, \quad (2.12)$$

where $N(\cdot)$ and $W(\cdot)$ are respectively the PDF of a Gaussian and a Wishart distribution,

\mathbf{m}_i is the posterior mean vector of $\boldsymbol{\mu}_i$, β_i is the confidence in the estimation of \mathbf{m}_i , and v_i represents the extent of effect on the estimated \mathbf{O}_i , which is a scaled parameter of the covariance matrix $\boldsymbol{\Sigma}_i$.

The i -th posterior mean and covariance matrix for the i -th cluster, $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$, can be estimated based on the conjugacy properties of the normal inverse Wishart prior distribution [52]. As a result, \mathbf{m}_i can be interchanged with $\boldsymbol{\mu}_i$. For a given number of Gaussian components/clusters, the GMM parameters can be estimated using the VI method.

In order to address the second issue related to the optimal number of clusters, the Dirichlet Process (DP) is employed in this chapter to identify the cluster number in an iterative way based on a DP prior distribution. The probability of assigning data samples to the i -th cluster can be estimated by a famous Chinese restaurant process as [53]

$$p(i | \alpha) = \begin{cases} \frac{N_{ci} + \alpha}{N_T + \alpha - 1}, & \text{if } i < Q \\ \frac{\alpha}{N_T + \alpha - 1}, & \text{otherwise} \end{cases}, \quad (2.13)$$

where N_{ci} is the number of samples in the i -th cluster, α is a parameter that governs the cluster behavior and reflects the growth rate of clusters in observed data samples. It implies that the total number of clusters increases with α .

To select an optimal α , a prior PDF is usually used and then α is calibrated with Gibbs sampler based on data samples. The PDF of α is derived by Escobar and West [54] as

$$p(\alpha | \eta, i) = R_\eta G(\alpha + i, b - \log(\eta)) + (1 - R_\eta) \Gamma(\alpha + i - 1, b - \log(\eta)), \quad (2.14)$$

where R_η is a weighting factor defined as $R_\eta / (1 - R_\eta) = (\alpha + i - 1) / (N_T (b - \log(\eta)))$, η is a continuous quantity between 0 and 1 following a beta distribution $p(\eta | \alpha, i) = B(\alpha + 1, \eta)$, and $\Gamma(\cdot, \cdot)$ stands for a Gamma distribution. Using Eqs. (2.13) and (2.14), the optimal number of clusters Q and optimal α can be estimated through Gibbs sampling method [55].

If $\boldsymbol{\theta} \triangleq \{w_i, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i, i = 1, 2, \dots, Q\}$ includes the GMM parameters and $P(\mathbf{x}_{T,j}, i | \boldsymbol{\theta})$ is

the probability that the j -th data sample belongs to the i -th cluster which is predicted using the learned GMM, we have $\sum_{i=1}^Q P(\mathbf{x}_{T,j}, i | \boldsymbol{\theta}) = 1$. In order to partition the data matrix of Eq. (2.9) into different segments while maintaining continuity of the surrogate model prediction, we also impose the following condition during partition

$$I(\mathbf{x}_{T,j}, i) = \begin{cases} 1, & \text{if } P(\mathbf{x}_{T,j}, i | \boldsymbol{\theta}) \geq \frac{0.05}{Q-1}, \\ 0, & \text{otherwise,} \end{cases} \quad (2.15)$$

where $I(\mathbf{x}_{T,j}, i) = 1$ means $\mathbf{x}_{T,j}$ is classified with the i -th cluster. A user-specified value of $\frac{0.05}{Q-1}$ ensures continuity of the response. The case study of Section 2.3 shows that this value works well.

Based on Eq. (2.15), the input training data of the i -th cluster is

$$\mathbf{x}_{c,i} = \{\mathbf{x} | I(\mathbf{x}, i) = 1, \forall \mathbf{x} \in \mathbf{x}_T\}, i = 1, \dots, Q. \quad (2.16)$$

According to Eqs. (2.15) and (2.16), it is quite possible that a sample may belong to multiple clusters indicating that such samples will be used as training data for the training of multiple NARX models. If $\mathbf{a}_{c,i}$ is the corresponding output training data of the i -th cluster, we have

$$\begin{aligned} \mathbf{x}_T &= \{\mathbf{x}_{c,1} \cup \dots \cup \mathbf{x}_{c,Q}\}, \\ \mathbf{a}_T &= \{\mathbf{a}_{c,1} \cup \dots \cup \mathbf{a}_{c,Q}\}. \end{aligned} \quad (2.17)$$

Moreover, if the number of samples in a certain cluster is too low, the NARX model prediction will be of very low quality. If the low quality NARX model is then ensembled with the other models, it may ruin the overall prediction. To avoid this situation, another condition during the partition of the data is set as

$$L(\mathbf{x}_{c,i}) > \frac{N_T}{Q}, \quad (2.18)$$

where $L(\mathbf{x}_{c,i})$ represents the number of samples in the i -th cluster.

If the above condition is violated during the data partition, samples from the nearby

clusters will be added into the corresponding cluster to ensure the condition of Eq. (2.18) is satisfied. After the partition of the data matrix of Eq. (2.9), the data is grouped into different clusters to represent different dynamic behaviors of the AGV in the off-road environment. Finally, we train Q NARX models as

$$a = \hat{g}_{a,i}(\mathbf{u}_p, \mathbf{a}_q), i = 1, 2, \dots, Q, \quad (2.19)$$

where $\mathbf{u}_p = [u_1, u_2, \dots, u_p]$ and $\mathbf{a}_q = [a_1, a_2, \dots, a_q]$ represent the p and q lags of the terrain elevation and vehicle acceleration, and $\hat{g}_{a,i}(\cdot)$ is the i -th NARX model trained using data $\mathbf{x}_{c,1}$ and $\mathbf{a}_{c,1}$.

The NARX models in Eq. (2.19) can be of GP-NARX or NARX-Net type. Other types of models can also be used such as mixed GP-NARX and NARX-net models. Figure 2.3 summarizes the overall procedure for the partition of dynamic domains using GMM and subsequent training of NARX models.

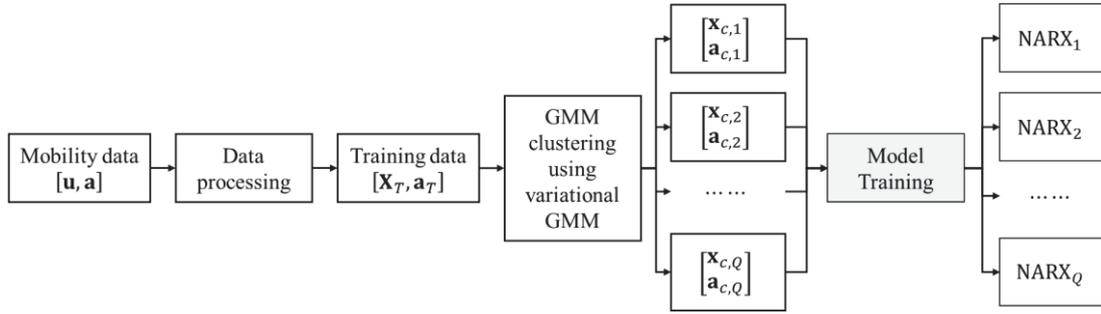


Figure 2.3 Partition of the training data and training of NARX models

2.2.2. Dynamic ensemble of NARX models (DENA)

After the Q NARX models are trained, they must be dynamically assembled together over time to predict the vehicle vertical acceleration over any given new path. If the terrain elevation of a new path be $\mathbf{u}_{new} = [u_1, u_2, \dots, u_{N_{new}}]$, where N_{new} is the number of steps on the new path, the task is to use the NARX models to predict the vehicle mobility without running the high-fidelity mobility simulation.

Taking the prediction at step t_k as an example, the inputs $\mathbf{x}_{T,k} = [\mathbf{u}_k, \mathbf{a}_k]$ to the NARX models are

$$\mathbf{u}_k = [u_{k-p-1}, u_{k-p+1}, \dots, u_k], \mathbf{a}_k = [a_{k-q}, a_{k-q+1}, \dots, a_{k-1}], \quad (2.20)$$

where \mathbf{u}_k are the previous p lags of terrain elevation of t_k and \mathbf{a}_k represents the previous q lags of vehicle vertical acceleration.

Using the inputs of Eq. (2.20), the probability that the inputs/dynamic behavior $\mathbf{x}_{T,k}$ at t_k belongs to the i -th cluster is predicted using the trained GMM of Section 2.2.1 as

$$p_{ki} = P(\mathbf{x}_{T,k}, i | \boldsymbol{\theta}), i = 1, 2, \dots, Q, \quad (2.21)$$

in which $\sum_{i=1}^Q p_{ki} = 1$.

Using the predicted probabilities, the NARX predictions can be then ensembled together to obtain the predicted vehicle acceleration at t_k as

$$\hat{a}_k = \sum_{i=1}^Q w(\mathbf{x}_{T,k}, i) g_{a,i}(\mathbf{u}_k, \mathbf{a}_k), \quad (2.22)$$

where $w(\mathbf{x}_{T,k}, i) = \hat{p}_{ki} / \sum_{i=1}^Q \hat{p}_{ki}$ with \hat{p}_{ki} given by

$$\hat{p}_{ki} = \begin{cases} p_{ki}, & \text{if } p_{ki} \geq \frac{0.05}{Q-1}. \\ 0, & \text{otherwise} \end{cases} \quad (2.23)$$

Eq. (2.23) indicates that if the probability the input/dynamic behavior belongs to a certain cluster is very low, the sample may be at the boundary of the cluster. For such cases, the NARX model usually has a relatively poor performance. Removing these NARX models from the ensemble avoids contamination of the prediction by NARX models with inaccurate predictions. As shown in Eq. (2.22), the ensemble weights $w(\mathbf{x}_{T,k}, i), i = 1, \dots, Q$ depend on the input $\mathbf{x}_{T,k}$. This implies that the ensemble weights will adjust dynamically over time according to the behavior of the response enabling us to dynamically ensemble the NARX models over time. In addition, as mentioned above, the NARX models can be of any type (e.g. GP-NARX or NARX-Net). If GP-NARX models are used, the prediction mean and standard deviation at step t_k are

$$\mu_{a,i} = \sum_{i=1}^Q w(\mathbf{x}_{T,k}, i) \mu_{a,i}(\mathbf{u}_k, \mathbf{a}_k), \quad (2.24)$$

$$\sigma_{a,i} = \sqrt{\sum_{i=1}^Q w^2(\mathbf{x}_{T,k}, i) \sigma_{a,i}^2(\mathbf{u}_k, \mathbf{a}_k)}, \quad (2.25)$$

where $\mu_{a,i}(\mathbf{u}_k, \mathbf{a}_k)$ and $\sigma_{a,i}^2(\mathbf{u}_k, \mathbf{a}_k)$ are the mean and variance of the prediction from the i -th GP-NARX model at t_k .

After we obtain the prediction of vehicle acceleration at t_k using the ensemble of the NARX models, the prediction is used as input to predict the vehicle mobility at the next step t_{k+1} as

$$\mathbf{u}_{k+1} = [u_{k+1-p}, u_{k+2-p}, \dots, u_k], \quad \tilde{\mathbf{a}}_{k+1} = [a_{k+1-q}, \underset{\text{predicted}}{a_{k+2-q}}, \dots, \hat{a}_k]. \quad (2.26)$$

The updated inputs are then used to determine the ensemble weights for t_{k+1} using Eqs. (2.22) and (2.23). The above process is repeated recursively over time allowing us to predict the vehicle mobility over a long time period. This is crucial for mobility prediction of off-road AGVs since the length of paths may be different. Figure 2.4 shows the flowchart of the proposed dynamic ensemble of NARX models based on the partition of dynamic domains using GMM.

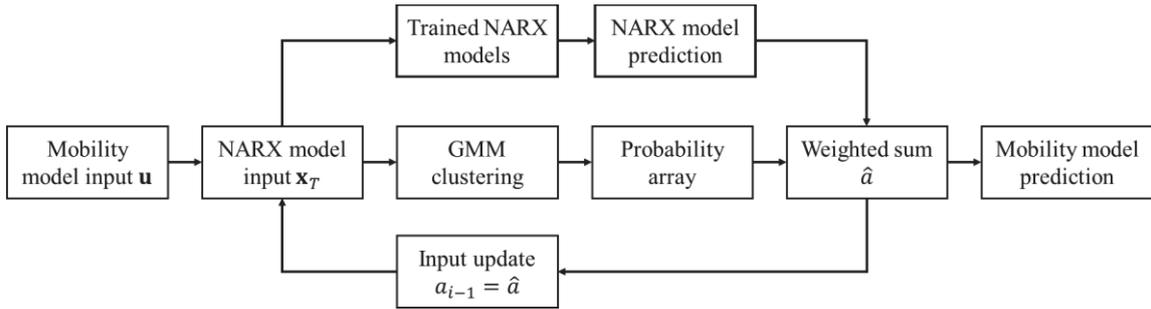


Figure 2.4 Proposed prediction approach using dynamic ensemble of NARX models

2.2.3. Dynamic ensemble of NARX models with different numbers of lags (DENA-DIFF)

In the DENA method presented in Sections 2.2.1 and 2.2.2, the number of lags is the same across different NARX models. Considering that the response over time may vary for different dynamic behaviors, a constant number of lags for each NARX model may not

be optimal. In the ideal scenario, the number of lags should be adaptable to the nonlinear dynamic response the NARX model should predict. For instance, the number of lags can be large for high-frequency response exhibiting many peaks in a short time, and small for low-frequency response. Motivated by this observation, this subsection develops a DENA with different number of lags method (DENA-DIFF) based on the dynamic ensemble method of Section 2.2.2. The basic idea is to use different number of lags for different clusters, where each cluster represents a different dynamic behavior and has its own NARX model.

To explain the proposed DENA-DIFF method, we represent the training data of the i -th cluster from Section 2.2.2 as

$$\mathbf{x}_{c,i} = \begin{bmatrix} \mathbf{u}_{i1} & \mathbf{a}_{i1} \\ \mathbf{u}_{i2} & \mathbf{a}_{i2} \\ \vdots & \vdots \\ \mathbf{u}_{iN_T} & \mathbf{a}_{iN_T} \end{bmatrix}, i = 1, \dots, Q. \quad (2.27)$$

where $\mathbf{u}_{i1} = [u_{k-p}, u_{k-p+1}, \dots, u_{k-1}] \in \mathbb{R}^{1 \times p}$, $\mathbf{a}_{i1} = [a_{k-q}, a_{k-q+1}, \dots, a_{k-1}] \in \mathbb{R}^{1 \times q}$. We then generate V different training datasets by adding more lags to the data matrix of Eq. (2.27) as

$$\mathbf{x}_{c,i}^{(r)} = \begin{bmatrix} \tilde{\mathbf{u}}_{i1}^{(r)} & \tilde{\mathbf{a}}_{i1}^{(r)} \\ \tilde{\mathbf{u}}_{i2}^{(r)} & \tilde{\mathbf{a}}_{i2}^{(r)} \\ \vdots & \vdots \\ \tilde{\mathbf{u}}_{iN_T}^{(r)} & \tilde{\mathbf{a}}_{iN_T}^{(r)} \end{bmatrix}, i = 1, \dots, Q; r = 1, \dots, V, \quad (2.28)$$

where $\mathbf{x}_{c,i}^{(r)}$ is the r -th training dataset of the i -th cluster. The $\tilde{\mathbf{u}}_{i1}^{(r)}$ and $\tilde{\mathbf{a}}_{i1}^{(r)}$ vectors of its first row are respectively the updated lags for terrain elevations and vertical accelerations by adding r additional lags to the original training dataset of $\mathbf{x}_{c,i}$. They are given by

$$\begin{aligned} \tilde{\mathbf{u}}_{i1}^{(r)} &= [\underbrace{u_{k-p-r}, \dots, u_{k-p-1}}_{\text{Added } r \text{ elements}}, \mathbf{u}_{i1}] \in \mathbb{R}^{1 \times (p+r)}, \\ \tilde{\mathbf{a}}_{i1}^{(r)} &= [\underbrace{a_{k-q-r}, \dots, a_{k-q-1}}_{\text{Added } r \text{ elements}}, \mathbf{a}_{i1}] \in \mathbb{R}^{1 \times (q+r)}. \end{aligned} \quad (2.29)$$

Similar expressions hold for the other rows of $\mathbf{x}_{c,i}^{(r)}$.

For each of the V different training datasets generated from $\mathbf{x}_{c,i}$, we divide the input training data and corresponding output data into a training set and a validation set as

$$\begin{aligned}\mathbf{x}_{c,i}^{(r)} &= \{\mathbf{x}_{T,i}^{(r)} \cup \mathbf{x}_{Va,i}^{(r)}\}, \\ \mathbf{a}_{c,i} &= \{\mathbf{a}_{T,i} \cup \mathbf{a}_{Va,i}\},\end{aligned}\quad \forall r = 1, \dots, V, \quad (2.30)$$

where $\mathbf{x}_{T,i}^{(r)}$ and $\mathbf{x}_{Va,i}^{(r)}$ are respectively the training and validation sets of the r -th new dataset generated from the i -th cluster.

Subsequently, we train V NARX models for the i -th cluster using $\mathbf{x}_{T,i}^{(r)}$, $r = 1, 2, \dots, V$ and $\mathbf{a}_{T,i}$ as

$$a = \hat{g}_{i,r}(\mathbf{u}_{p+r}, \mathbf{a}_{q+r}), \quad \forall i = 1, 2, \dots, Q; r = 1, \dots, V, \quad (2.31)$$

where $\hat{g}_{i,r}(\mathbf{u}_{p+r}, \mathbf{a}_{q+r})$, $r = 1, 2, \dots, V$ represents the NARX model of the i -th cluster trained using $\mathbf{x}_{T,i}^{(r)}$ and $\mathbf{a}_{T,i}$.

In order to determine the optimal number of lags for the i -th cluster, we compute the following prediction error of the r -th NARX model for the validation dataset

$$\mathcal{E}_{V,i}^{(r)} = \|\mathbf{a}_{V,i} - \hat{\mathbf{a}}_{V,i}^{(r)}\|_2, \quad r = 1, \dots, V \quad (2.32)$$

where $\|\cdot\|_2$ is the l_2 norm and $\hat{\mathbf{a}}_{V,i}^{(r)}$ is the prediction obtained from the r -th NARX model as

$$\hat{\mathbf{a}}_{V,i}^{(r)} = \hat{g}_{a,i}(\mathbf{x}_{Va,i}^{(r)}, r), \quad (2.33)$$

where $\mathbf{x}_{Va,i}^{(r)} = [\tilde{\mathbf{u}}_{Va,i}^{(r)}, \tilde{\mathbf{a}}_{Va,i}^{(r)}]$, with $\tilde{\mathbf{u}}_{Va,i}^{(r)}$ and $\tilde{\mathbf{a}}_{Va,i}^{(r)}$ being the $\tilde{\mathbf{u}}_{i1}^{(r)}$ and $\tilde{\mathbf{a}}_{i1}^{(r)}$ in Eq. (2.29) for the validation set of the i -th cluster after adding r more lags to the original cluster.

The optimal number of lags for the i -cluster is then determined by

$$r_i^* = \arg \min_{r=1, \dots, V} \{\mathcal{E}_{V,i}^{(r)}\}, \quad (2.34)$$

in which $\mathcal{E}_{V,i}^{(r)}$ is the error given in Eq. (2.32).

The above procedure is implemented for all clusters. Eqs. (2.28)-(2.34) allows us to use

different number of lags for different clusters each representing a different dynamic behavior. After the optimal number of lags is identified for all Q clusters, the NARX models, each with a different number of lags, are ensembled for prediction at t_k as

$$\hat{y}_k^* = \sum_{i=1}^Q w(\mathbf{x}_{T,k}, i) g_{i,r_i^*}(\mathbf{u}_{p+r_i^*}, \mathbf{a}_{q+r_i^*}), \quad (2.35)$$

where $w(\mathbf{x}_{T,k}, i)$ are the weights determined similarly as in the DENA approach of Section 2.2.2. The remaining recursive implementation procedure of DENA-DIFF is the same with that of the DENA method. Figure 2.5 summarizes the overall procedure of the DENA-DIFF method.

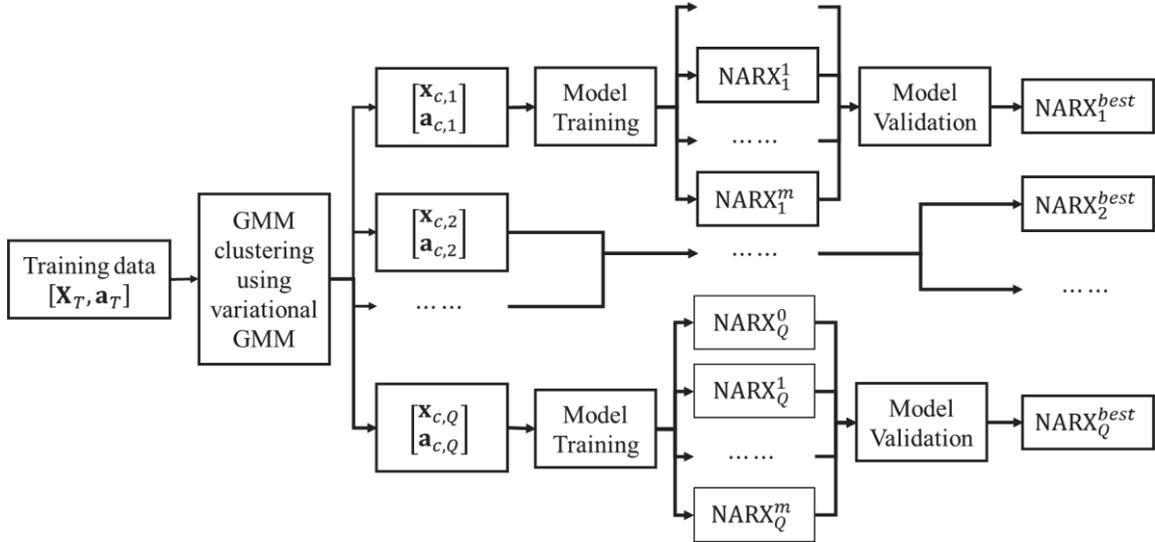


Figure 2.5 Proposed prediction approach using dynamic ensemble of NARX models with different numbers of lags

2.3. Case Study

In this section, we use an off-road AGV example to demonstrate the proposed DENA and DENA-DIFF methods. Both methods will be implemented using GP-NARX and NARX-Net. The DENA method is referred as “DENA-GP” and “DENA-Net” respectively in this section to indicate its implementation with GP-NARX and NARX-Net. Similarly, the implementations of DENA-DIFF with GP-NARX and NARX-net are referred as “DENA-DIFF-Net” and “DENA-DIFF-GP,” respectively. The DENA-GP, DENA-Net, DENA-DIFF-GP, and DENA-DIFF-Net methods will be compared with the classical GP-

NARX and NARX-Net methods without ensemble of NARX models and the LSTM model with extensive tuning. The LSTM model's performance will be optimized by tuning the number of layers, the number of neurons, the dropout rate, and the learning rate. Next, we first present how the synthetic AGV mobility data is collected for the training and testing of different methods. Then, the performance of different methods is compared.

2.3.1. Generation of synthetic mobility data

The off-road AGV mobility simulation model discussed in Section 2.1.1.2 is employed to generate synthetic mobility data. The vehicle module is the PyChrono's build-in High Mobility Multipurpose Wheeled Vehicle (HMMWV) of Figure 2.6, which is a module for an off-road military Humvee.

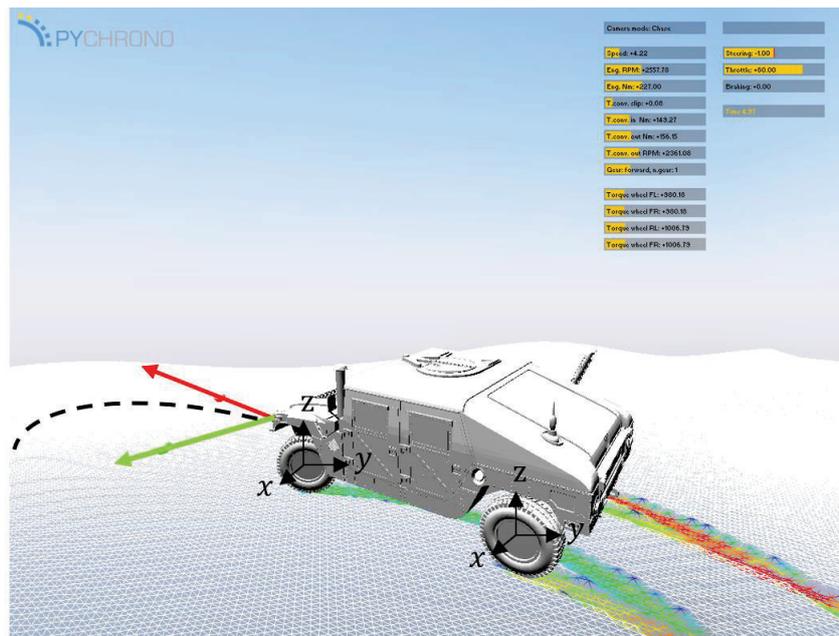


Figure 2.6 Simulation of HMMWV for the case study

The key vehicle model parameters are

- Vehicle mass = 2422.74 kg
- Wheelbase = 3.378 m
- Wheel track = 1.819 m
- Driveline Type: all-wheel drive (AWD)

During the simulation, the vehicle uses a proportional–integral–derivative (PID) controller with preset gains to keep it on the specified path, and a fixed throttle value of

100%. The terrain is described as a Gaussian random field with a mean of 0 meters, a standard deviation of 0.26 meters, and a correlation length of 0.5 meters. These parameters generate a random field that closely resembles realistic off-road terrains in terms of variation of hills and bumps in relation to the terrain size. Figure 2.7 presents 10 different random terrain maps generated using the KL expansion method (Eqs. (2.2) and (2.3)). A zoomed-in view of one of the terrain maps is given in Figure 2.8 indicating a maximum variability of the vertical height of about 2 meters.

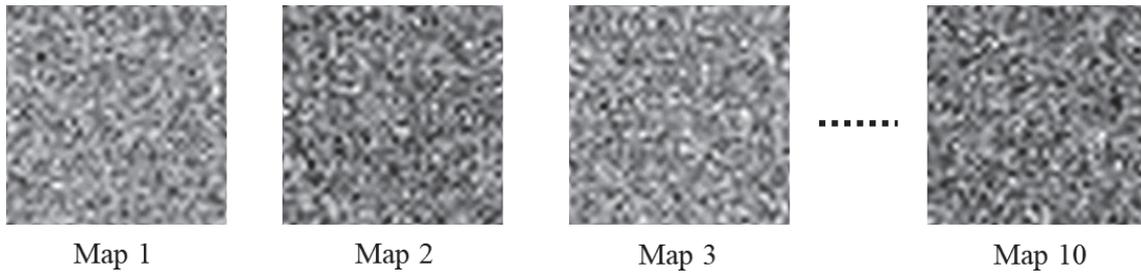


Figure 2.7 Ten different terrain height maps generated using KL expansion

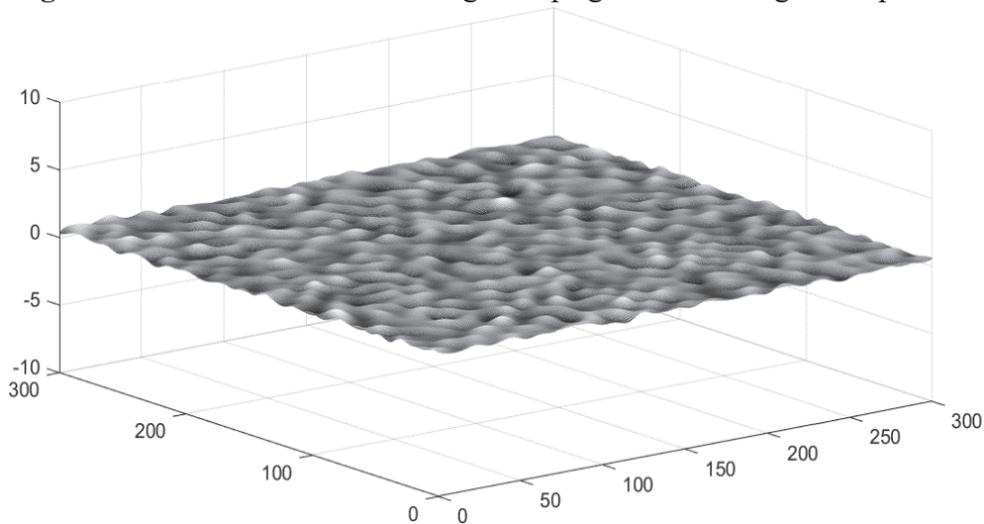


Figure 2.8 A realization of terrain map described as random field

The terrain is assumed rigid and the smooth (penalty) contact method (SMC) is used in the vehicle mobility simulation with terrain friction = 0.9, terrain restitution = 0.01, and terrain Young's modulus = 2×10^7 Pa. In order to investigate the impact of the number of training paths on the performance of different methods, two sets of synthetic vehicle mobility data are generated.

- In the *first* set of data, five paths are generated from each terrain map resulting in 50 paths in total (referred as Scenario 1).

- For the *second* set of data, nine paths are generated from each map generating therefore 90 paths in total (referred as Scenario 2).

For mobility data collection, the vehicle is guided by its controller during the simulation as it follows a specified path. The key metrics of interest recorded during the simulation at each time step are the terrain elevation and the vehicle's vertical acceleration at the center of mass. Figure 2.9 shows the selected five paths to collect synthetic mobility data for Scenario 1. The starting and end points ($S_i, T_i, i = 1, \dots, 5$) of the five paths are

$$\begin{cases} S_i = (-140, d_2) \\ T_i = (140, d_2) \end{cases}, d_2 = \pm 140, 0. \quad (2.36)$$

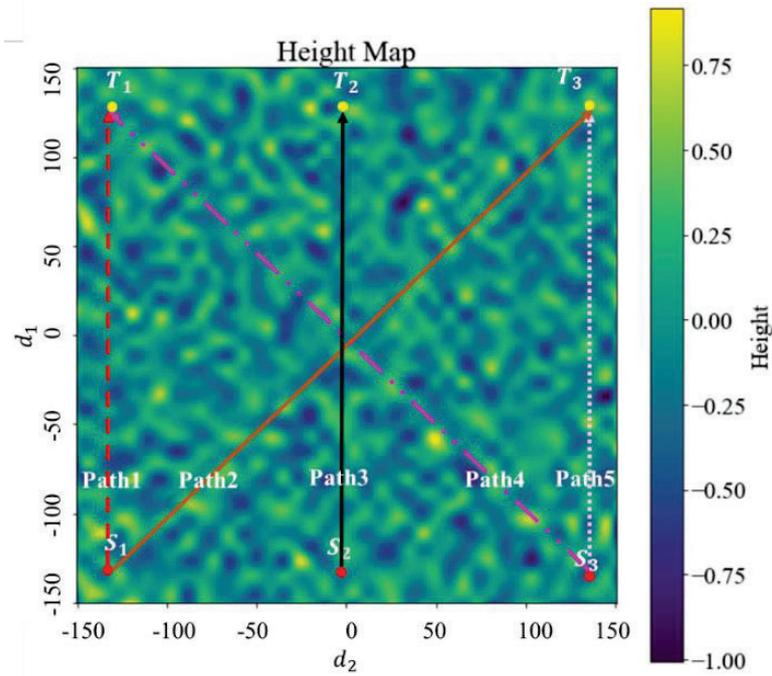


Figure 2.9 Five paths of each terrain map for data collection in Scenario 1

The nine paths of each terrain map are generated similarly for Scenario 2. Figure 2.10 shows an example (path 1) of the generated synthetic data indicating that the vertical acceleration fluctuates considerably over the path.

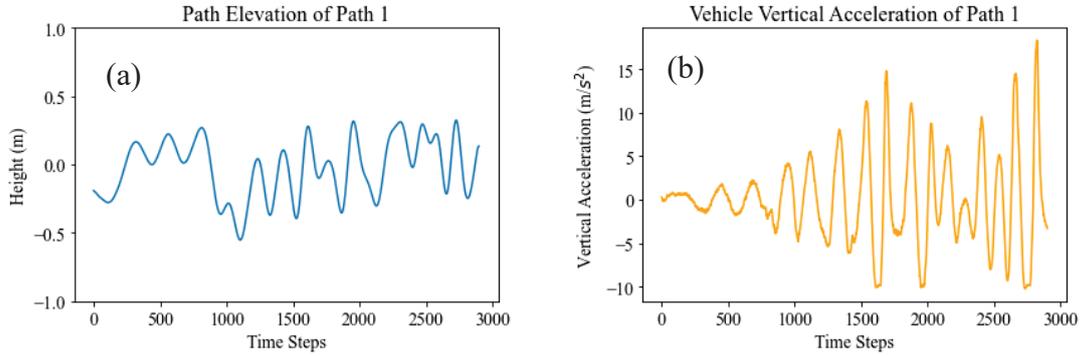


Figure 2.10 (a) Path elevation and (b) vertical acceleration for Path 1 of Figure 2.9

Section 2.3.2 compares the accuracy of different methods including GP-NARX, NARX-Net, DENA-GP, DENA-Net, DENA-DIFF-GP, DENA-DIFF-Net, and LSTM. For Scenario 1, 40 out of the 50 paths are randomly selected for training and the remaining 10 paths are used for testing to compare the performance of different methods. For Scenario 2, 80 out of 90 are randomly selected for training and 10 are used for testing.

2.3.2. Comparison of mobility prediction using different surrogate models

For the training of GP-NARX, NARX-Net, DENA-GP, DENA-Net, and LSTM, the number of lags is optimized for each model to obtain the best performance. For DENA-DIFF-GP and DENA-DIFF-Net, the number of lags is optimized for each cluster. As discussed in Section 3.1, the Dirichlet Process is employed to determine the optimal number of clusters in DENA and DENA-DIFF. Figure 2.11 shows the clustering iterative process for Scenario 1 with 40 training paths and the resulting optimal estimate of the number of clusters, $Q = 3$. Following that, Figure 2.12 presents the clusters of training data for terrain elevation and vehicle vertical acceleration for Scenario 1 with 40 training paths and 10 testing paths. As shown in this figure, even though there is an overlapping between different clusters, the shapes of the clusters are different indicating the difference in local dynamic behaviors of the vehicle. Figure 2.12 is only a 2-D plot of two out of the 40 inputs comprising 20 lags of the terrain elevation and 20 lags of the vehicle acceleration. To better illustrate the difference of clusters, Figure 2.13 shows the clustering of different regions for a testing path in the dynamic ensemble. It indicates that the prediction regions are classified into different regions representing different local dynamic behaviors.

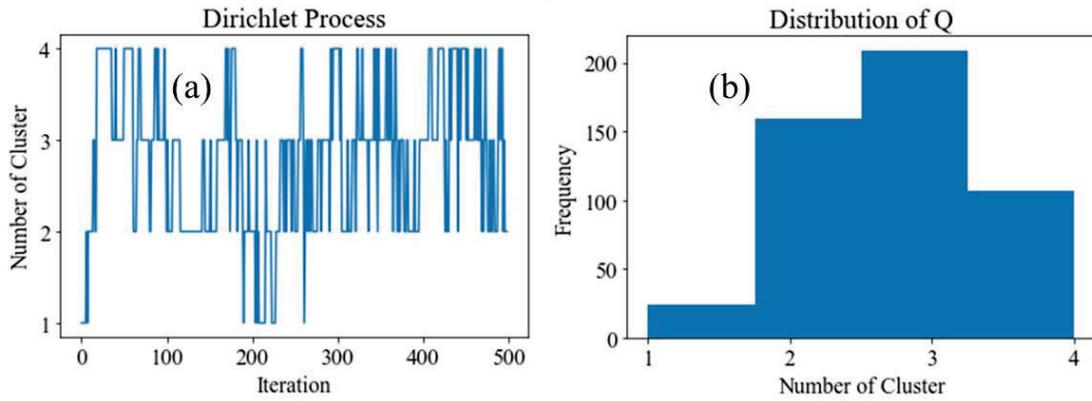


Figure 2.11 Dirichlet process to estimate the optimal number of clusters: (a) Dirichlet iterative process; (b) posterior distribution of number of clusters

Table 2.1 lists the optimal number of lags for DENA-DIFF-GP and DENA-DIFF-Net for Scenario 1. The optimal number of lags is different for different clusters and different NARX models.

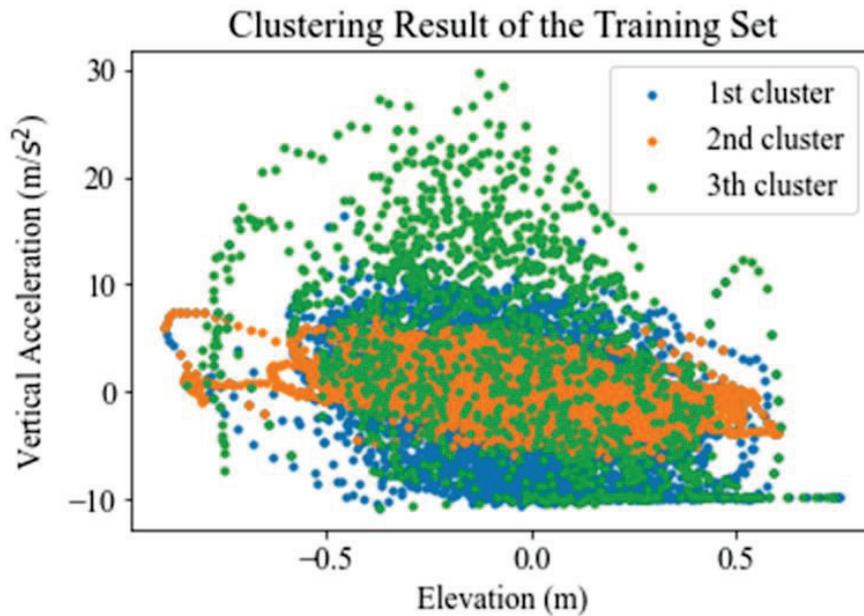


Figure 2.12 Clustering of the training dataset (Scenario 1)

Table 2.1 Optimal number of lags for DENA-DIFF (Scenario 1)

Method	Number of Lags		
	Cluster 1	Cluster 2	Cluster 3
DENA-DIFF-Net	20	15	5
DENA-DIFF-GP	19	18	20

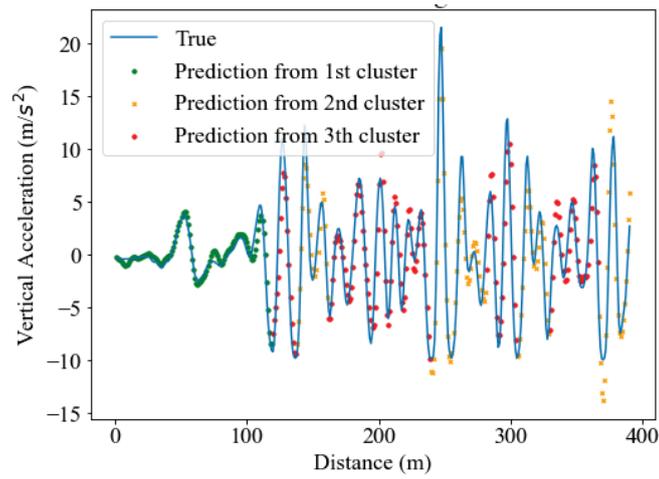


Figure 2.13 Clustering of regions for a test path 1

Figure 2.14 through 2.16 compare the prediction of different methods for a testing path against the true vertical acceleration in Scenario 1. Following that, Figure 2.17 through 2.19 show the prediction error of different methods for this particular path.

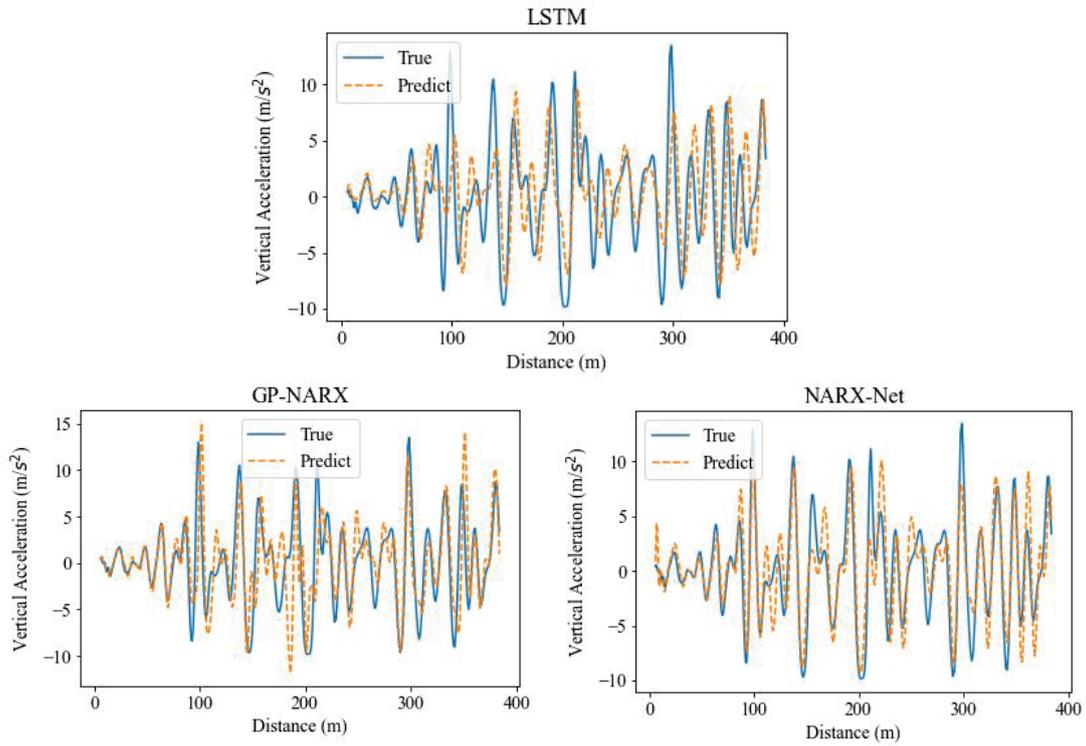


Figure 2.14 Comparison of GP-NARX, LSTM, and NARX-Net for a testing path

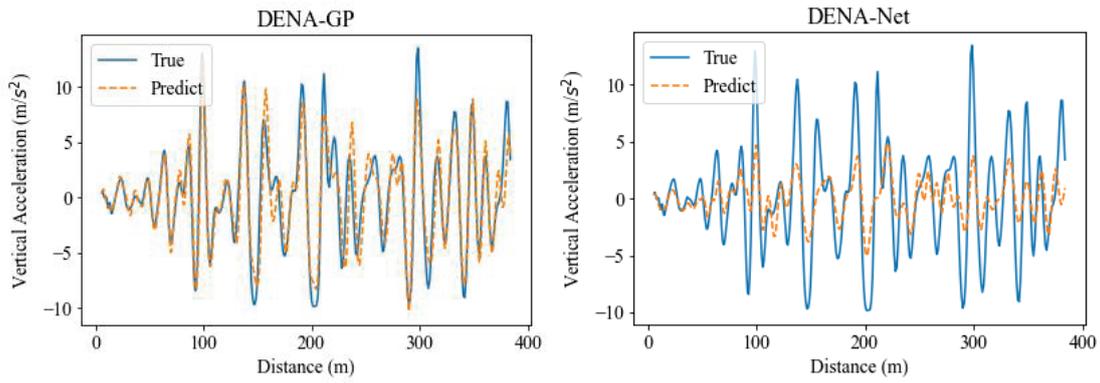


Figure 2.15 Comparison of different DENA methods for a testing path

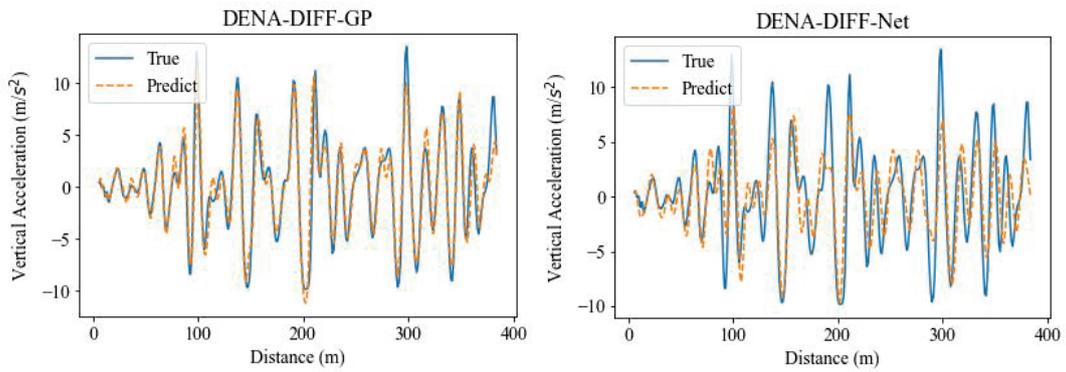


Figure 2.16 Comparison of different DENA-DIFF methods for a testing path

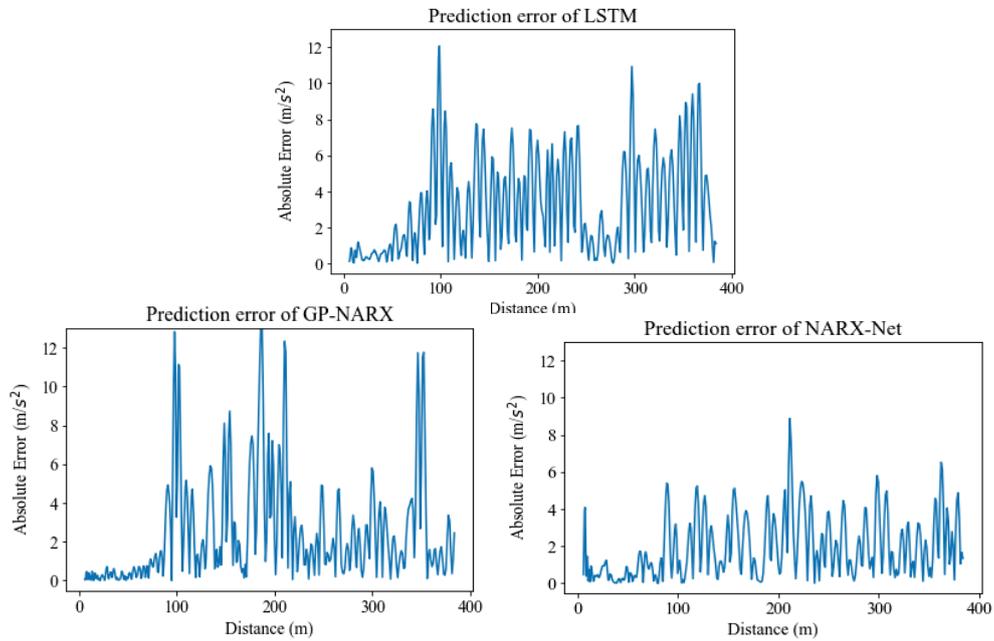


Figure 2.17 Absolute prediction error comparison of GP-NARX, LSTM, and NARX-Net for a testing path

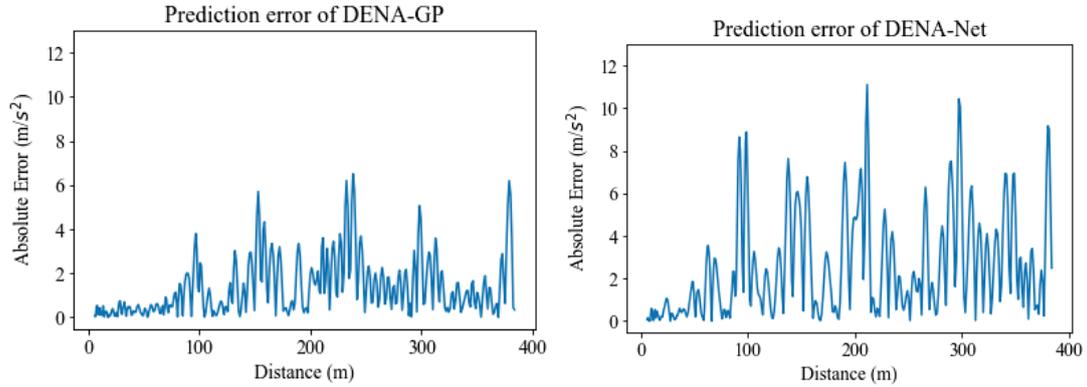


Figure 2.18 Absolute prediction error comparison of DENA methods for a testing path

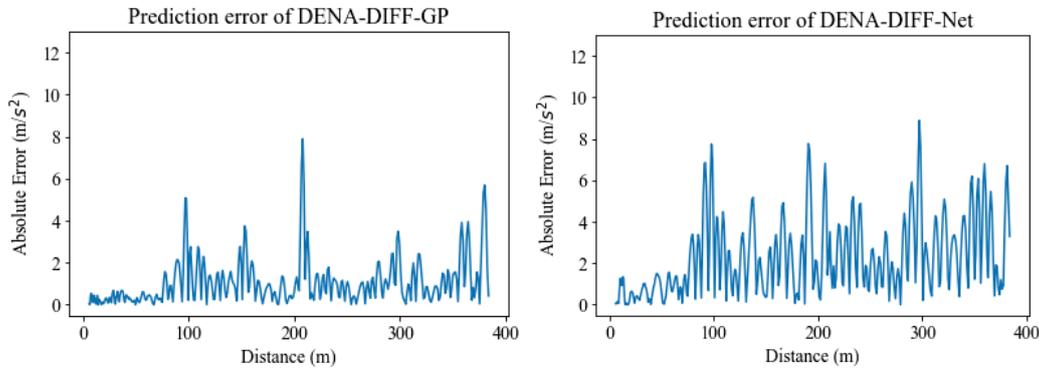


Figure 2.19 Absolute prediction error comparison of DENA-DIFF methods for a testing path

Figure 2.20 compares the prediction error distribution for different methods for this path. It shows that the DENA-DIFF-GP method performs the best. Subsequently, Table 2.2 summarizes the root mean square error (RMSE) and the average absolute error of different methods for ten testing paths in the studied two scenarios (i.e. one with 40 training paths and the other one with 80 training paths). The results show that the prediction accuracy of NARX model can be improved using the proposed dynamic ensemble of different NARX models. For both scenarios (i.e. 40 and 80 training paths), the accuracy of DENA-GP-DIFF is the best with DENA-GP and NARX-Net following. Additionally, as the number of training paths increases from 40 to 80, the performance of all methods improves. The accuracy of different methods also gets closer to each other as the number of training paths becomes high. This implies that the ensemble method may not be necessary for vehicle mobility prediction if there is a large number of training paths. A direct NARX-Net model may satisfy the decision maker's accuracy requirement. The proposed DENA methods are

more suitable for applications with a smaller number of available training paths.

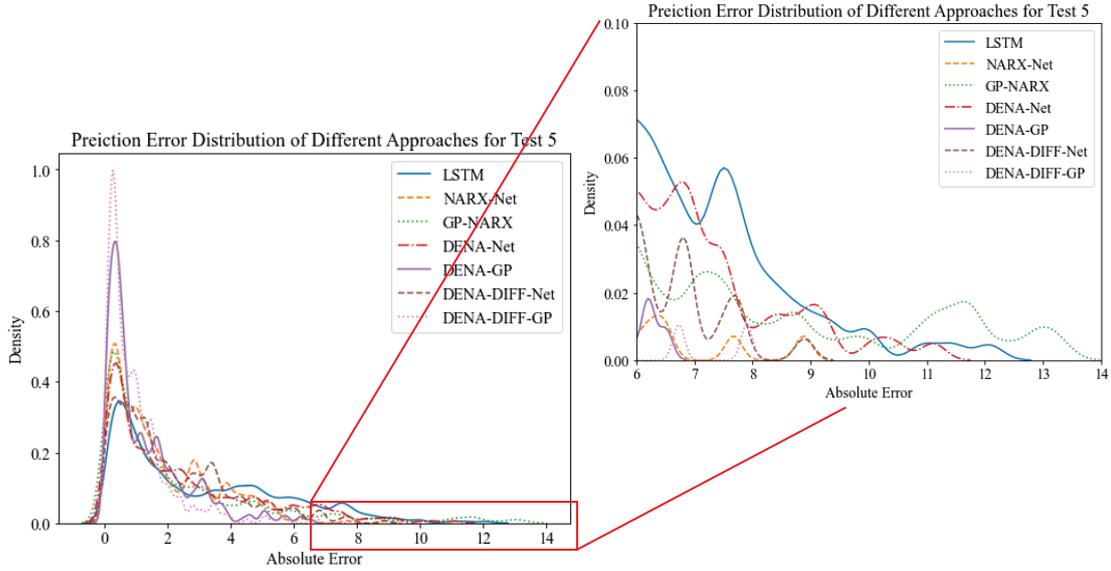


Figure 2.20 Prediction error distribution of different methods for test path 5

Table 2.2 Comparison of mean RMSE and absolute errors of different approaches for ten testing paths using 40 and 80 training paths

Metrics	RMSE		Average Absolute Error	
	40	80	40	80
Number of training paths				
LSTM	4.194	3.517	3.027	2.550
GP-NARX	3.818	3.822	2.606	2.613
NARX-Net	2.938	2.036	2.116	1.402
DENA-GP	2.301	1.921	1.512	1.314
DENA-Net	3.723	2.749	2.717	2.031
DENA-DIFF-GP	2.003	1.910	1.306	1.236
DENA-DIFF-Net	2.942	2.563	2.717	1.926

2.4. Summary

Building a cheap surrogate model to substitute the original computationally expensive off-road AGV simulation model plays an important role in supporting real-time decision making and mission planning under uncertainty. It is observed that a single surrogate model cannot accurately capture the complicated nonlinear dynamics of an AGV over time. This paper overcomes this challenge by developing a dynamic ensemble framework for different NARX models. The prediction domain is first partitioned into different dynamic regions

using a Gaussian mixture model-based clustering method. Based on the partition, different NARX models are constructed and then dynamically ensembled together over time. To account for the dynamic behavior being potentially different at different time steps, the dynamic ensemble framework is also extended to include NARX models with different number of lags. The results of a case study showed that the proposed DENA framework can improve the prediction accuracy of NARX models for mobility prediction of off-road AGVs in an off-road environment. Even though the developed approach is for off-road AGVs, it is also applicable to other dynamic systems which have complicated nonlinear dynamics.

Chapter 3

Simulation-Based Mission Mobility Reliability Analysis of Off-Road Autonomous Vehicles

This chapter develops a new simulation-based mission mobility reliability (MMR) analysis framework for off-road ground vehicles. In contrast to the current approaches which only focus on mobility prediction uncertainty at specific locations, the proposed framework not only investigates effects of mobility dependence over space on mission planning, but also overcomes the computational challenges in mobility prediction under uncertainty without sacrificing the accuracy. The main contributions can be summarized as: (1) definition of MMR for the first time to quantify effects of mobility prediction uncertainty on mission planning; (2) employment of a single-loop Kriging surrogate modeling method to efficiently evaluate MMR by refining the Kriging surrogate model in regions which are critical for MMR analysis; and (3) development of a Bayesian updating scheme which allows for dynamical updating of MMR over the course of a mission based on vehicle operational data. The synthesis of the above three contributions enables the off-road ground vehicle to proactively avoid rare events of immobility during a mission, and thus guarantee the safety of the vehicle on the battlefield. This part has been published at:

- [56] Liu, Y., Jiang, C., Mourelatos, Z. P., Gorsich, D., Jayakumar, P., Fu, Y., Majcher, M., and Hu, Z., 2021, "Simulation-based mission mobility reliability analysis of off-road ground vehicles," *Journal of Mechanical Design*, 143(3).

3.1. Background

3.1.1. Off-Road Vehicle Mobility Model

Off-road vehicle mobility model is an analytical or simulation model that is used to predict the vehicle mobility characteristics for given off-road conditions and vehicle design

[3, 6, 7]. At TARDEC (now called GVSC) or cooperating with TARDEC, many mobility models have been developed in the past, such as the Bekker's Derived Terrain-mechanics Model [8], the Army NATO Reference Mobility Model (NRMM), the absorbed power criteria, and the Dynamic Analysis and Design System (DADS) [5]. Different mobility models may have different quantities of interest. For instance, in the MMP model, the MMP is related to the vehicle parameters as follows [12].

$$MMP = 1.26Wm_w b p_w d_w, \quad (3.1)$$

where W is weight of vehicle, m_w is the total number of road-wheels, b is track width, p_w is track plate length and d_w is road-wheel diameter.

In the GO/NO-GO condition model [57], the decision-making parameter $CL_{L-wheels}$ is given by

$$CL_{L-wheels} = 1.85W \cdot 2n_a b_a^{0.8} d_a^{0.8} \delta^{0.4}, \quad (3.2)$$

in which n_a is the number of axles, b_a and d_a are the overall width and diameter of the tire inflated but unloaded, and δ is the tire deflection when loaded.

In Bekker's Terramechanics Model [8], the key parameter displacement x is given by

$$x = A_1 e^{\left(-b_x + \sqrt{(b_x^2 - 1)}\right)\omega t} + A_2 e^{\left(-b_x - \sqrt{(b_x^2 - 1)}\right)\omega t}, \quad (3.3)$$

where b_x is the coefficient of damping, ωt is the natural time frequency of an aperiodic vibration, and A_1 and A_2 are coefficients related to soil stress and soil deformation.

Among the developed models, NRMM is one of the most commonly used models. Its development started in the 1950s [58]. There are also other upgrades such as NRMM II, in order to increase the performance of the mobility prediction model [20, 59]. Since NRMMs were developed several decades ago, the performance of these models is limited by the analysis methods and computing capabilities. With the development of powerful computational tools such as finite element analysis (FEA), high-performance computation, and simulation techniques, and with the need of developing advanced vehicle systems such as autonomous vehicles, there is an urgent need to develop the Next Generation NRMM (NG-NRMM) [60]. One of the important aspects of the NG-NRMM is to account for the uncertainty in the vehicle mobility prediction.

No matter what kind of off-road mobility model is used, the inputs to the models

(NRMM or NG-NRMM) can always be classified into two groups, namely vehicle-related parameters and terrain-related parameters. The vehicle-related parameters include vehicle dimension, weight, power, movement type (wheel or track), and so on. Terrain-related parameters include slope, cohesive strength, friction coefficient, bulk density and etc. In addition, terrain-related parameters are usually space-dependent (i.e. vary with spatial coordinates). Let $X \in \mathbb{R}^{1 \times n}$ be a vector of vehicle parameters, $Y \in \mathbb{R}^{1 \times m}$ be a vector of terrain parameters which are functions of spatial coordinates \mathbf{d} , a vehicle mobility model can be generalized as

$$V = G(\mathbf{X}, \mathbf{Y}(\mathbf{d})), \quad (3.4)$$

where $V \in \mathbb{R}$ is the vehicle mobility variable (e.g. the maximum attainable speed in this chapter).

3.1.2. Mission Mobility Reliability

As mentioned above, accounting for the uncertainty in vehicle mobility prediction is one of the most important aspects of NG-NRMM. The inherent uncertainty in the terrain and soil properties and variability of the vehicle performances lead to the uncertainty of the vehicle mobility. As shown in Figure 3.1, if a vehicle design and a soil region are selected, a soil map can be used in conjunction with the vehicle mobility model to generate a mobility map. Due to the uncertainty in both the vehicle systems and the soil map, there is uncertainty in the vehicle mobility for any given spatial location \mathbf{d} .

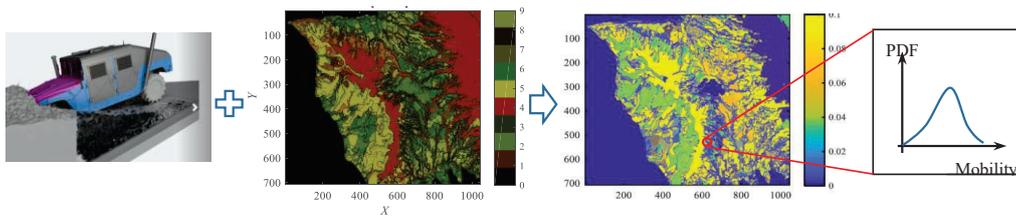


Figure 3.1 Uncertainty propagation of vehicle mobility

In order to quantify effects of uncertainty sources on the vehicle mobility, at a specific spatial location \mathbf{d} , the off-road vehicle mobility reliability $R_M(\mathbf{d})$ is defined as follows

$$R_M(\mathbf{d}) = \iint_{G(\mathbf{X}, \mathbf{Y}(\mathbf{d})) \geq e} f_X(\mathbf{x}) f_{Y(\mathbf{d})}(\mathbf{y}) d\mathbf{x} d\mathbf{y}, \quad (3.5)$$

where e is the threshold of decision-making parameter, if $V = G(\mathbf{X}, \mathbf{Y}(\mathbf{d})) \geq e$, it means that the vehicle can achieve the minimum required maximum attainable speed at this

location, otherwise the vehicle will be immobile (i.e. failed); $f_{\mathbf{x}}(\mathbf{x})$ and $f_{Y(\mathbf{d})}(\mathbf{y})$ are respectively the probability density function (PDF) of vehicle parameters and terrain parameters at \mathbf{d} .

The above mobility reliability, which is also referred as the ‘‘probability of mobility’’, has been employed in the literature to generate the probability of mobility maps [17]. Even though the mobility reliability is able to effectively quantify the uncertainty of mobility at any given spatial location, it cannot be directly applied to reliability-based mission planning, due to the fact that the stochastic mobility map does not provide information about the correlation of mobility over space. The correlation of vehicle mobility stems from correlations of soil properties over space. For a particular soil type and a certain soil property, the soil property needs to be modeled as a random field over space instead of a random variable. In this chapter, the soil properties are assumed to be Gaussian random fields for the sake of illustration. The developed framework can also be extended to account for the situation that the random fields are non-Gaussian. Similar to many other random field problems, one way to model/simulate the variability of soil property over space is to employ the Karhunen–Loève (KL) expansion method [61]. For a given terrain parameter or soil property over a spatial domain $\mathbf{d} \in \Omega$, $Y(\mathbf{d})$ is modeled using the KL expansion method as follows

$$Y(\mathbf{d}) = \mu_Y(\mathbf{d}) + \sigma_Y(\mathbf{d}) \sum_{j=1}^{N_{KL}} \sqrt{\lambda_j} \xi_j \eta_j(\mathbf{d}), \quad (3.6)$$

in which $\mu_Y(\mathbf{d})$ and $\sigma_Y(\mathbf{d})$ are respectively the mean and standard deviation of $Y(\mathbf{d})$, λ_j and $\eta_j(\mathbf{d})$ are the eigenvalues and eigenvectors of $\boldsymbol{\rho} \triangleq \{\rho(\mathbf{d}_i, \mathbf{d}_j), \forall i, j = 1, 2, \dots, N_d\}$, N_d is the number of discretized spatial points on a mission path Ω (as shown in Figure 3.2), $\xi_j, j = 1, 2, \dots, N_{KL}$ are standard normal random variables, N_{KL} is the number of the largest eigenvalues that makes 95% of the summation of all eigenvalues, and the correlation $\rho(\mathbf{d}_i, \mathbf{d}_j)$ between two locations is given by

$$\rho(\mathbf{d}_i, \mathbf{d}_j) = \exp\left[-\beta_1(d_{1i} - d_{1j})^2 - \beta_2(d_{2i} - d_{2j})^2\right], \quad (3.7)$$

where $\mathbf{d}_i = [d_{1i}, d_{2i}]$ are the two spatial coordinates at location i , β_1 and β_2 are

respectively the correlation length parameters of coordinates 1 and 2. Note that the developed framework is not limited to above squared exponential function. The other types of correlation functions such as constant, linear, Ornstein–Uhlenbeck, etc [62] can also be employed to describe the correlation between two spatial points. The best correlation function needs to be determined according to the geostatistical data of the soil.

Furthermore, this chapter defines the *mission mobility reliability* (MMR) to relate the variability and correlation of vehicle mobility over space with mission planning. For a given mission plan Ω as shown in Figure 3.2, the MMR, $R(\Omega)$, is defined as

$$R(\Omega) = \Pr\{V = G(\mathbf{X}, \mathbf{Y}(\mathbf{d})) \geq e, \forall \mathbf{d} \in \Omega\}, \quad (3.8)$$

where “ \forall ” stands for “for all” and $\Pr\{\cdot\}$ is a probability operator. It implies that only if the speed of a vehicle is always greater than or equal to the threshold e on the mission path, the mission is considered as successful, otherwise is considered as failed.

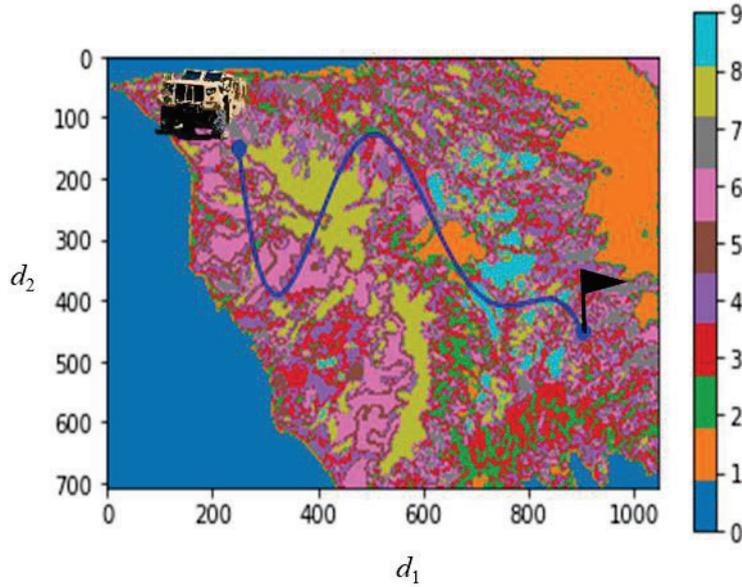


Figure 3.2 A mission path over a terrain map

The corresponding mission mobility failure probability is given by

$$p_f(\Omega) = \Pr\{V = G(\mathbf{X}, \mathbf{Y}(\mathbf{d})) < e, \exists \mathbf{d} \in \Omega\}, \quad (3.9)$$

in which “ \exists ” means “there exists”.

The above MMR quantifies the probability that the vehicle completes the mission path without immobility. Once an event of immobility happens, it implies that the mission is failed. Evaluating this MMR plays a vital role in guaranteeing the safety of the off-road

ground vehicle in the battlefield and avoiding events of immobility, especially for autonomous ground vehicles.

Due to the involvement of computer simulation models of vehicle-terrain interaction and soil maps, the evaluation of Eq. (3.8) or (3.9) is not straightforward. There are two main challenges that need to be addressed to perform MMR analysis.

- *First*, the mobility prediction model given in Eqs. (3.8) and (3.9) are usually high-fidelity M&S. The evaluation of MMR needs to execute the high-fidelity M&S numerous times, which is computationally prohibitive.
- *Second*, there are many sources of uncertainty presented in the mobility analysis. Some of the uncertainty sources, such as the variability in the soil properties, can be reduced when a particular vehicle starts to proceed on a pre-specified mission path. As the vehicle proceeds over the path, more and more field information about the vehicle mobility will be collected. The probability of accomplishing the mission (i.e. MMR) will then change over time and needs to be updated. How to leverage the field information in the MMR analysis is the second challenging issue that needs to be addressed.

In order to tackle above challenges, in the subsequent section (Section 3.2), a simulation-based off-road ground vehicle MMR analysis approach is proposed.

3.2.Simulation-Based Off-Road Ground Vehicle MMR Analysis

In this section, we first provide a brief overview of the proposed simulation-based MMR analysis framework. Following that, we explain the proposed framework in details.

3.2.1. Overview

Figure 3.3 shows an overview of the proposed framework. As shown in this figure, there are two phases of off-road ground vehicle MMR analysis, namely (1) mission-planning phase and (2) during-mission phase. The two phases are connected through the mobility M&S as indicated in the figure. The main objectives of the two phases are explained as below.

- (1) Phase 1-mission-planning phase: In the mission-planning phase, the MMR of a given mission path is evaluated with the consideration of the natural variability of

soil properties over the mission path. The main objective is to address the challenges in solving Eq. (3.8) to provide the probability of completing a mission in the early planning stage.

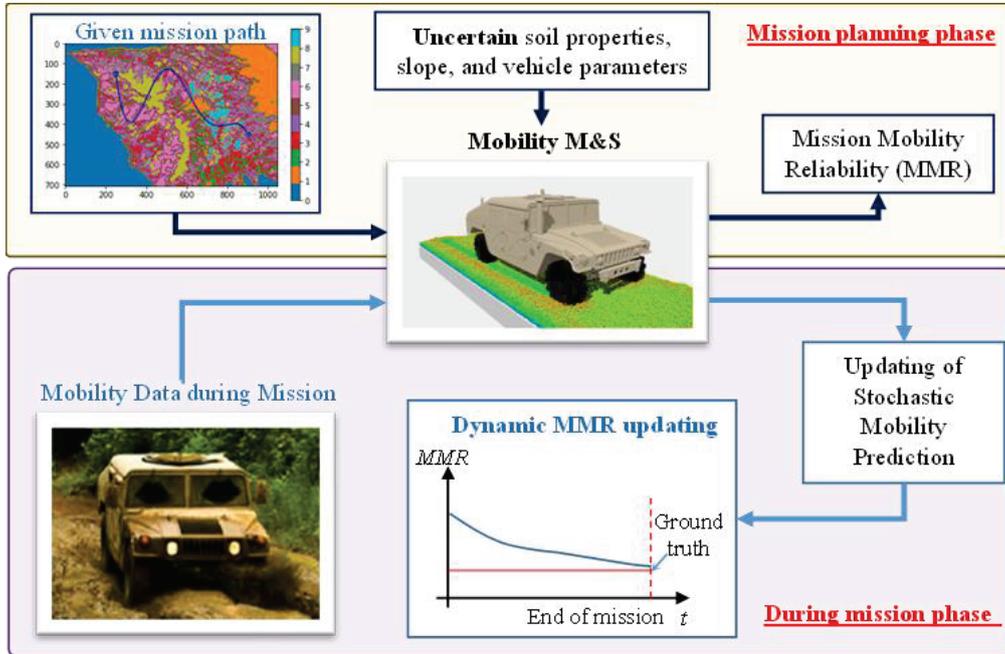


Figure 3.3 Overview of the proposed framework for simulation-based off-road ground vehicle MMR analysis

- (2) Phase 2-during-mission phase: In the mission-planning phase, the MMR analysis is purely based on the M&S and soil maps. Even though there is uncertainty in the soil properties and vehicle system, the soil properties and vehicle system parameters are deterministic values in reality for a particular vehicle and a specific mission path. As a result, a mission can only have two states: success or fail. In the during-mission phase, vehicle mobility measurement data collected through sensors on the vehicle are used to update the MMR estimate dynamically based on the MMR analysis approach developed in Phase 1. This dynamic updating of MMR allows us to predict the failure (i.e. immobility) with high confidence before it happens and thus proactively avoid the situation of immobility. As illustrated in Figure 3.3, the MMR gets close to the ground truth (i.e. success or fail) as the vehicle proceeds on the mission path.

Next, we provide more details about the two phases of the simulation-based MMR analysis.

3.2.2. Simulation-Based MMR Analysis in Mission-Planning Phase

In the mission-planning phase, the simulation-based MMR analysis is very similar to the time-dependent reliability analysis problems [63, 64]. In recent years, various time-dependent reliability analysis approaches, such as first-order reliability method-based methods [65, 66], sampling-based approaches [67, 68], and adaptive surrogate modeling-based methods [69-73], have been proposed to overcome the computational issue in reliability analysis. For MMR analysis, the reliability is space-dependent instead of time-dependent. Moreover, the MMR needs to be updated in the during-mission phase using Bayesian method. Considering the fact that the M&S is shared by both the mission-planning and during-mission phases, the surrogate modeling-based method is employed. More specifically, the single-loop Kriging (SILK) surrogate modeling approach [74] is employed in this chapter, since SILK can easily handle the presence of both random variables and stochastic processes in reliability analysis.

The employment of SILK in the mission planning phase, however, is not very straightforward. Two issues need to be addressed in order to perform MMR analysis with SILK. The first issue is the simulation of soil properties over the mission path, and the second one is the generation of training points for surrogate modeling. In what follows, we discuss how to tackle these two issues in details.

3.2.2.1. Simulation of soil properties over the mission path

Let $\mathbf{Y}(\mathbf{d}) \triangleq [Y_1(\mathbf{d}), Y_2(\mathbf{d}), \dots, Y_m(\mathbf{d})]$, $\forall \mathbf{d} \in \Omega$, be the space-dependent soil properties or terrain parameters, where m is the number of (slope or soil) property variables, for a given mission path Ω as indicated in Figure 3.2, $\mathbf{Y}(\mathbf{d})$ may consist of multiple slope types or soil types which are described as random fields. It implies that even for a single soil property variable, $Y_i(\mathbf{d}), \forall \mathbf{d} \in \Omega$, it consists of multiple random fields with different

statistical properties. Taking the terrain slope as an example, to generate random realizations of $Y_i(\mathbf{d}), \forall \mathbf{d} \in \Omega, i = 1, 2, \dots, m$ for MMR analysis, as illustrated in Figure 3.4, the coordinates $\mathbf{d}_{(p)}^{\text{Slope}}$ corresponding to the p -th slope ID are first identified from Ω as

$$\mathbf{d}_{(p)}^{\text{Slope}} = \{\mathbf{d} : \text{where } S(\mathbf{d}) = p, \forall \mathbf{d} \in \Omega\}, \quad p = 1, 2, \dots, N_{\text{slope}}, \quad (3.10)$$

where $S(\mathbf{d})$ is the slope ID at location \mathbf{d} obtained from the terrain map (see Section 3.3.1 for a terrain map example), N_{slope} is the number of slope IDs, and we have

$$\{\mathbf{d}_{(1)}^{\text{Slope}} \cup \dots \cup \mathbf{d}_{(N_{\text{slope}})}^{\text{Slope}}\} = \Omega. \quad (3.11)$$

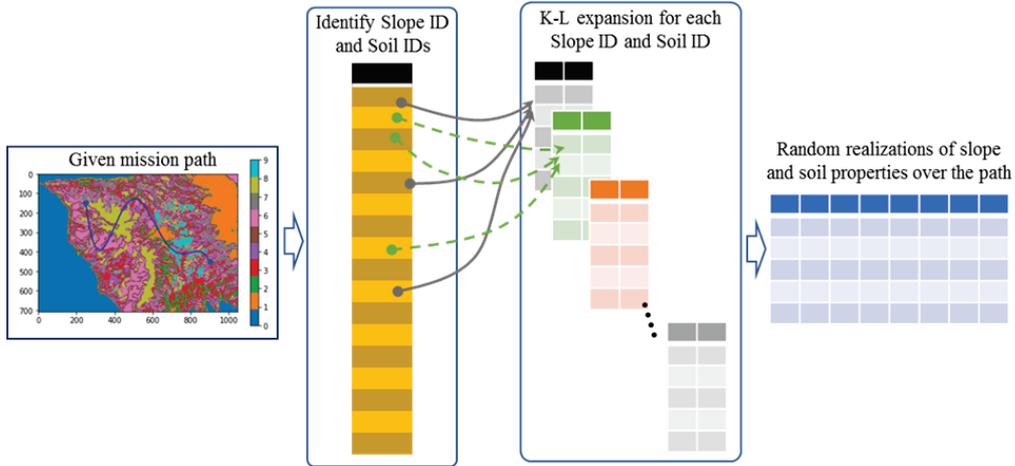


Figure 3.4 Simulation of a space-dependent variables with different slope ID or soil types
Similarly, the coordinates $\mathbf{d}_{(h)}^{\text{Soil}}$ of the h -th soil ID are identified as

$$\mathbf{d}_{(h)}^{\text{Soil}} = \{\mathbf{d} : \text{where } SL(\mathbf{d}) = h, \forall \mathbf{d} \in \Omega\}, \quad h = 1, 2, \dots, N_{\text{soil}}, \quad (3.12)$$

in which $SL(\mathbf{d})$ is the soil ID obtained from the soil map (see Section 3.3.1), N_{soil} is the number of soil types, and $\{\mathbf{d}_{(1)}^{\text{Soil}} \cup \dots \cup \mathbf{d}_{(N_{\text{soil}})}^{\text{Soil}}\} = \Omega$.

After partitioning the coordinates Ω on the mission path into $\mathbf{d}_{(1)}^{\text{Slope}}, \mathbf{d}_{(2)}^{\text{Slope}}, \dots, \mathbf{d}_{(N_{\text{slope}})}^{\text{Slope}}$ for slope IDs and $\mathbf{d}_{(1)}^{\text{Soil}}, \mathbf{d}_{(2)}^{\text{Soil}}, \dots, \mathbf{d}_{(N_{\text{soil}})}^{\text{Soil}}$ for soil types, the slope or soil property corresponding to a particular slope ID or soil ID is described using KL expansion given in Eq. (3.6). Taking $Y_i(\mathbf{d}), \forall \mathbf{d} \in \Omega$ as an example, according to the partition of the

coordinates on the mission path, $Y_i(\mathbf{d}), \forall \mathbf{d} \in \Omega$ is expressed as a union of multiple random fields as follows

$$Y_i(\mathbf{d}) = \{ \mathbf{Y}_{i1} \cup \dots \cup \mathbf{Y}_{iN_T} \}, \forall \mathbf{d} \in \Omega, \quad (3.13)$$

in which $\mathbf{Y}_{ij} \triangleq \{Y_{ij}(1), Y_{ij}(2), \dots, Y_{ij}(N_s(j))\}$ where $Y_{ij}(k), k = 1, 2, \dots, N_s(j)$ represents $Y_i(\mathbf{d})$ associated with the j -th slope ID or soil ID at the k -th location, $N_s(j)$ is the number of spatial points associated with the j -th slope or soil ID (i.e. the number of elements in $\mathbf{d}_{(j)}^{\text{Slope}}$ or $\mathbf{d}_{(j)}^{\text{Soil}}$), and N_T is given by

$$N_T = \begin{cases} N_{slope}, & \text{if } Y_i(\mathbf{d}) \text{ is a slope-related variable} \\ N_{soil}, & \text{if } Y_i(\mathbf{d}) \text{ is a soil type-related variable} \end{cases}, \quad (3.14)$$

$Y_{ij}(k), \forall j = 1, 2, \dots, N_T; k = 1, 2, \dots, N_s(j)$ are modeled with KL expansion as follows

$$Y_{ij}(k) = \mu_{ij} + \sigma_{ij} \sum_{q=1}^{N_{ij}} \sqrt{\lambda_q} \xi_q \eta_q(k), \quad (3.15)$$

where μ_{ij} and σ_{ij} are respectively the mean and standard deviation of $Y_{ij}(k)$, N_{ij} is the number of expansion terms, and λ_q and η_q are the eigenvalues and eigenvectors of the following correlation matrix

$$\boldsymbol{\rho}_{ij} = \begin{bmatrix} 1 & \rho_{ij}(\mathbf{d}_{(1)}^{\text{S}}, \mathbf{d}_{(2)}^{\text{S}}) & \dots & \rho_{ij}(\mathbf{d}_{(1)}^{\text{S}}, \mathbf{d}_{(N_s(j))}^{\text{S}}) \\ \rho_{ij}(\mathbf{d}_{(2)}^{\text{S}}, \mathbf{d}_{(1)}^{\text{S}}) & 1 & \dots & \rho_{ij}(\mathbf{d}_{(2)}^{\text{S}}, \mathbf{d}_{(N_s(j))}^{\text{S}}) \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{ij}(\mathbf{d}_{(N_s(j))}^{\text{S}}, \mathbf{d}_{(1)}^{\text{S}}) & \rho_{ij}(\mathbf{d}_{(N_s(j))}^{\text{S}}, \mathbf{d}_{(2)}^{\text{S}}) & \dots & 1 \end{bmatrix}_{N_s(j) \times N_s(j)}, \quad (3.16)$$

in which $\rho_{ij}(\cdot, \cdot)$ is the correlation function (i.e. Eq. (3.7)) of $Y_{ij}(k)$, $\mathbf{d}_{(k)}^{\text{S}} = \mathbf{d}_{(j)}^{\text{Slope}}(k), \forall k = 1, 2, \dots, N_s(j)$ if $Y_{ij}(k)$ is a slope-related variable; Otherwise, $\mathbf{d}_{(k)}^{\text{S}} = \mathbf{d}_{(j)}^{\text{Soil}}(k), \forall k = 1, 2, \dots, N_s(j)$ (i.e. if $Y_{ij}(k)$ is a soil type related variable).

Using Eqs. (3.10) through (3.16), random realizations can be generated for $\mathbf{Y}_{ij}, \forall j = 1, 2, \dots, N_T$. As illustrated in Figure 3.4, the random realizations of \mathbf{Y}_{ij} are then assembled together according to the coordinates on the mission path to obtain random realizations of $Y_i(\mathbf{d}), \forall \mathbf{d} \in \Omega$. The random realizations of $Y_i(\mathbf{d}), \forall \mathbf{d} \in \Omega$ are used as the inputs in the SILK method [74] for: (1) selection of new training points in refining the

surrogate model of the mobility model, and (2) MMR analysis. Next, we will discuss how to generate initial training points for the surrogate modeling of the mobility model $V = G(\mathbf{X}, \mathbf{Y}(\mathbf{d}))$.

3.2.2.2. Generation of training points for surrogate modeling

a) Generation initial training points

For any specific coordinates \mathbf{d} , Figure 3.5 shows the relationships between the map coordinates, soil type (i.e. ID), slope type/ID, and space-dependent parameters $\mathbf{Y}(\mathbf{d})$.

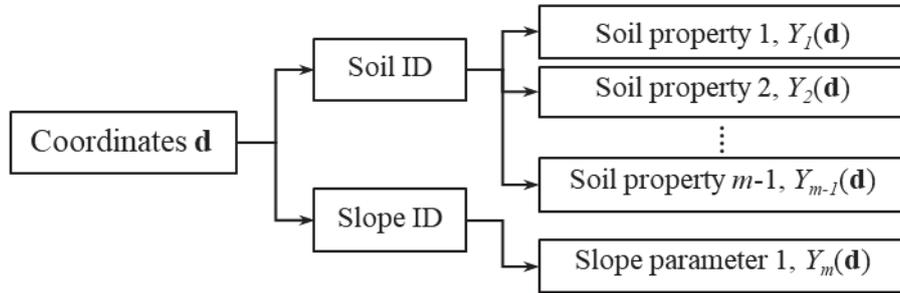


Figure 3.5 Relationship between coordinates, soil type, slope parameters, and soil properties

As shown in above figure, the soil property parameters share the same soil ID for any given spatial point, \mathbf{d} . The soil ID represented as integer values governs the statistical parameters of soil properties (e.g. bulk density, friction coefficient, and cohesive strength). It indicates that there are strong dependences among the soil-type related parameters. In order to maintain the strong dependences in generating initial training points for $\mathbf{Y}(\mathbf{d})$, in this chapter, two sets of initial training points are first generated using Latin Hypercube sampling method [75].

The two sets of training points are respectively training samples of soil and slope IDs (i.e. integer) and training samples in the uniform domain of $[0, 1]$. Let the integer training samples of soil/slope IDs be $\boldsymbol{\alpha} = [\alpha^{(1)}, \alpha^{(2)}, \dots, \alpha^{(N_{in})}]$ where N_{in} is the number of initial training points and the uniform domain training points be $\mathbf{u} = [\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \dots, \mathbf{u}^{(N_{in})}]$, \mathbf{u} are then transformed into training points of $\mathbf{Y}(\mathbf{d})$ as follows

$$\mathbf{y}_t^{(l)} = f_s(\mathbf{u}^{(l)}, \boldsymbol{\theta}(\alpha^{(l)})), \forall l = 1, 2, \dots, N_{in}, \quad (3.17)$$

where $f_s(\cdot, \cdot)$ is a function mapping $\mathbf{u}^{(l)}$ to training point $\mathbf{y}_t^{(l)}$ of $\mathbf{Y}(\mathbf{d})$ and $\boldsymbol{\theta}(\alpha^{(l)})$ are the statistical parameters (e.g. mean and standard deviation) related to soil/slope ID

$\alpha^{(l)}$. Note that $\mathbf{y}_t^{(l)}$ here represents multiple slope/soil property parameters that share a same soil/slope ID $\alpha^{(l)}$.

For instance, if $\boldsymbol{\theta}(k) = [\boldsymbol{\theta}_L(k), \boldsymbol{\theta}_U(k)]$, $k = 1, 2, \dots, N_{soil}$ are the lower and upper bounds of the training interval for a given slope/soil ID k , we have $\mathbf{y}_t^{(l)}$ as

$$\mathbf{y}_t^{(l)} = \boldsymbol{\theta}_L(\alpha^{(l)}) + \mathbf{u}^{(l)}(\boldsymbol{\theta}_U(\alpha^{(l)}) - \boldsymbol{\theta}_L(\alpha^{(l)})), \forall l = 1, 2, \dots, N_{in}. \quad (3.18)$$

After defining the initial training points of \mathbf{X} and $\mathbf{Y}(\mathbf{d})$ as \mathbf{x}_t and \mathbf{y}_t , we then obtain the corresponding mobility responses as \mathbf{v}_t . Using $[\mathbf{x}_t, \mathbf{y}_t]$ and \mathbf{v}_t , a surrogate model is constructed for the computationally expensive vehicle mobility model $V = G(\mathbf{X}, \mathbf{Y}(\mathbf{d}))$ using Kriging surrogate modeling method as below [74, 76]

$$V = \hat{G}(\mathbf{X}, \mathbf{Y}(\mathbf{d})), \quad (3.19)$$

in which $V = \hat{G}(\mathbf{X}, \mathbf{Y}(\mathbf{d}))$ stands for the Kriging surrogate model which is an approximation of the original high-fidelity mobility model.

According to the property of Kriging surrogate model, we have

$$V \sim N(\mu_V(\mathbf{X}, \mathbf{Y}(\mathbf{d})), \sigma_V^2(\mathbf{X}, \mathbf{Y}(\mathbf{d}))), \quad (3.20)$$

where $\mu_V(\mathbf{X}, \mathbf{Y}(\mathbf{d}))$ and $\sigma_V(\mathbf{X}, \mathbf{Y}(\mathbf{d}))$ are respectively the mean and standard deviation of the Kriging surrogate model prediction, and $N(\cdot, \cdot)$ stands for normal distribution.

Since $V = \hat{G}(\mathbf{X}, \mathbf{Y}(\mathbf{d}))$, may not accurately represent the original high-fidelity mobility model, it could lead to large error in the MMR analysis. To improve the accuracy of reliability analysis, many adaptive surrogate modeling methods have been proposed. As mentioned above, the SILK method is employed in this chapter for MMR analysis because it is capable to handle time-dependent problem which is similar to the space-dependent problem. Next, we discuss how to adaptively adding new training points to refine the surrogate model.

b) Generation of new training points through adaptive training

In SILK [74], the surrogate model is adaptively refined in important regions to guarantee the accuracy of reliability analysis. Let the random samples of \mathbf{X} be $\mathbf{x}^{MCS} \triangleq \{\mathbf{x}_i^{MCS}, i = 1, 2, \dots, n_{MCS}\}$ and the random realizations generated for $\mathbf{Y}(\mathbf{d})$ using

Eqs. (3.10) through (3.16) be $\mathbf{y}^{MCS} \triangleq \{\mathbf{y}_{ij}^{MCS}, i=1, 2, \dots, n_{MCS}; j=1, 2, \dots, N_d\}$ where n_{MCS} is the number of realizations in Monte Carlo simulation (MCS), N_d is the number of discretized spatial points on the mission path Ω , and \mathbf{y}_{ij}^{MCS} indicates the i -th realization of $\mathbf{Y}(\mathbf{d})$ at j -th spatial location, the new training point for \mathbf{X} and $\mathbf{Y}(\mathbf{d})$ is identified in SILK as follows [74]

$$\mathbf{x}^{new}, \mathbf{y}^{new} = [\mathbf{x}_{i^*}^{MCS}, \mathbf{y}_{i^*(j^*|i^*)}^{MCS}], \quad (3.21)$$

where i^* and $j^* | i^*$ are identified by

$$i^* = \arg \min_{i=1, 2, \dots, n_{MCS}} \{Q_{\min}(i)\}, \quad (3.22)$$

$$Q_{\min}(i) = \begin{cases} Q_e, & \text{if } \mu_V(\mathbf{x}_i^{MCS}, \mathbf{y}_{ij}^{MCS}) < e \text{ and } Q(\mathbf{x}_i^{MCS}, \mathbf{y}_{ij}^{MCS}) \geq 2, \exists j=1, 2, \dots, N_d \\ \min_{j=1, 2, \dots, N_d} \{Q(\mathbf{x}_i^{MCS}, \mathbf{y}_{ij}^{MCS})\}, & \text{otherwise} \end{cases}, \quad (3.23)$$

in which Q_e is a value greater than 2 and $Q(\mathbf{x}_i^{MCS}, \mathbf{y}_{ij}^{MCS})$ is given by [74]

$$Q(\mathbf{x}_i^{MCS}, \mathbf{y}_{ij}^{MCS}) = \frac{|\mu_V(\mathbf{x}_i^{MCS}, \mathbf{y}_{ij}^{MCS}) - e|}{\sigma_V(\mathbf{x}_i^{MCS}, \mathbf{y}_{ij}^{MCS})}, \quad (3.24)$$

and

$$j^* | i^* = \arg \min_{j=1, 2, \dots, N_d} \{Q(\mathbf{x}_{i^*}^{MCS}, \mathbf{y}_{i^*j}^{MCS})\}. \quad (3.25)$$

After the new training point $\mathbf{x}^{new}, \mathbf{y}^{new}$ is identified, the corresponding mobility response is obtained. The new training point is then added to the training point dataset and the surrogate model $V = \hat{G}(\mathbf{X}, \mathbf{Y}(\mathbf{d}))$, is re-trained. The process continues until the accuracy criterion is satisfied. In SILK, the accuracy criterion is defined as below [74]

$$\mathcal{E}_r^{\max} < 0.05, \quad (3.26)$$

where

$$\mathcal{E}_r^{\max} = \max_{N_{f2}^* \in [0, N_2]} \left\{ \frac{|N_{f2} - N_{f2}^*|}{N_{f1} + N_{f2}^*} \right\}, \quad (3.27)$$

in which
$$N_2 = \sum_{i=1}^{n_{MCS}} I_d \{ Q_{\min}(i) < 2 \} ,$$

$$N_{f1} = \sum_{i=1}^{n_{MCS}} I_d \{ (\mu_V(i, j) < e, \exists j = 1, 2, \dots, N_d) \cap Q_{\min}(i) \geq 2 \} ,$$

$N_{f2} = I_d \{ (\mu_V(i, j) < e, \exists j = 1, 2, \dots, N_d) \cap Q_{\min}(i) < 2 \}$, where $\mu_V(i, j) \triangleq \mu_V(\mathbf{x}_i^{MCS}, \mathbf{y}_{ij}^{MCS})$ and $I_d \{ E \}$ is an indicator function. $I_d \{ E \} = 1$, if event E is true. Otherwise, $I_d \{ E \} = 0$.

When the accuracy criterion given in Eq. (3.26) is satisfied, we terminate the adaptive training of $V = \hat{G}(\mathbf{X}, \mathbf{Y}(\mathbf{d}))$. After that, the MMR of a given mission path Ω is estimated as follows

$$R(\Omega) \approx \frac{1}{n_{MCS}} \sum_{i=1}^{n_{MCS}} I_d \{ \mu_V(i, j) \geq e, \forall j = 1, 2, \dots, N_d \}, \quad (3.28)$$

where $\mu_V(i, j) \triangleq \mu_V(\mathbf{x}_i^{MCS}, \mathbf{y}_{ij}^{MCS})$ is the mean prediction obtained by plugging \mathbf{x}^{MCS} and \mathbf{y}^{MCS} into the surrogate model $V = \hat{G}(\mathbf{X}, \mathbf{Y}(\mathbf{d}))$.

Until now, we have discussed all the details for simulation-based MMR analysis in the mission-planning phase. By overcoming the computational challenge in this phase, we are able to efficiently perform MMR analysis and ensure the MMR in mission planning. While the planned mission path usually has a high MMR, there is still a risk of having immobility (i.e. failure) events when the vehicle proceeds on the path. The dynamic updating of MMR in the during-mission phase (as discussed in Sec. 3.2.3) allows us to sense the immobility events before it happens and thus proactively avoid rare events of failures (i.e. immobility).

3.2.3. Dynamic Updating of MMR in the During-Mission Phase

The goal of dynamic updating of MMR in the during-mission phase is to proactively forecast the failure based on vehicle measurement data given that a vehicle has already accomplished part of the mission path (as illustrated in Figure 3.6).

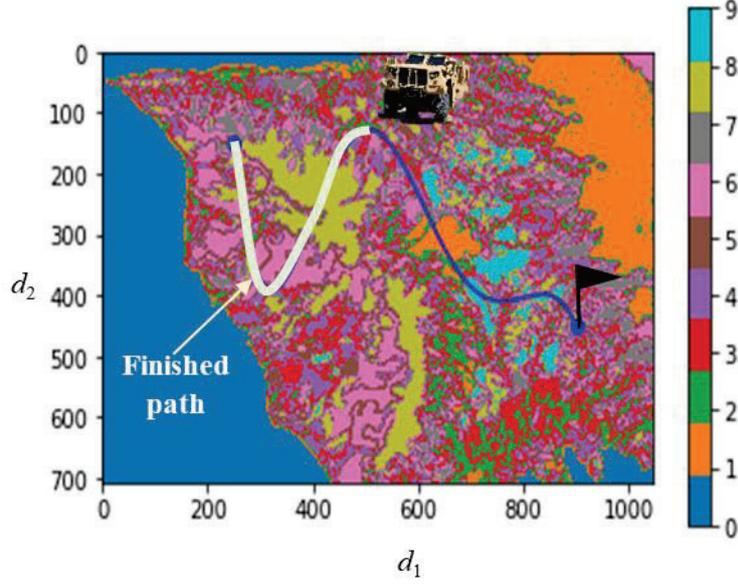


Figure 3.6 Illustration of a vehicle proceeds on a mission path

Denoting the spatial points of the finished part and unfinished part of the mission path as respectively Ω_f and Ω_{uf} , we have $\Omega_f \cup \Omega_{uf} = \Omega$. Let the measured vehicle mobility data of the finished part Ω_f be $\mathbf{v}_{lc} \triangleq \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_c\}, \forall c = 1, 2, \dots, N_d$, where $\mathbf{v}_i = \mathbf{v}(\mathbf{d}_i)$ are the mobility data collected at spatial point \mathbf{d}_i , and \mathbf{d}_c is the current location, the task is to dynamically estimate the following MMR conditioned on \mathbf{v}_{lc}

$$R(\Omega) | \mathbf{v}_{lc} = \Pr\{V = \hat{G}(\mathbf{X} | \mathbf{v}_{lc}, \mathbf{Y}(\mathbf{d}) | \mathbf{v}_{lc}) \geq e, \forall \mathbf{d} \in \Omega_{uf}\}, \quad (3.29)$$

where $V = \hat{G}(\mathbf{X}, \mathbf{Y}(\mathbf{d}))$, is the refined surrogate model from Phase 1, $\mathbf{X} | \mathbf{v}_{lc}$ and $\mathbf{Y}(\mathbf{d}) | \mathbf{v}_{lc}$ are respectively \mathbf{X} and $\mathbf{Y}(\mathbf{d}), \mathbf{d} \in \Omega_{uf}$ conditioned on \mathbf{v}_{lc} .

A critical step in evaluating Eq. (3.29) is to update \mathbf{X} and $\mathbf{Y}(\mathbf{d}), \mathbf{d} \in \Omega_{uf}$ based on \mathbf{v}_{lc} . The challenge of performing this updating is that there is no direct relationship between the random variables \mathbf{X} and $\mathbf{Y}(\mathbf{d}), \mathbf{d} \in \Omega_{uf}$ and \mathbf{v}_{lc} . Fortunately, the vehicle mobility M&S provides such a bridge between these variables and the measurement data. In this section, a sequential updating scheme is proposed to dynamically update MMR (as shown in Figure 3.7). It consists of three main steps, namely (1) Bayesian inference of

uncertainty parameters, (2) uncertainty sources updating, and (3) MMR updating and remaining mobilable distance estimation.

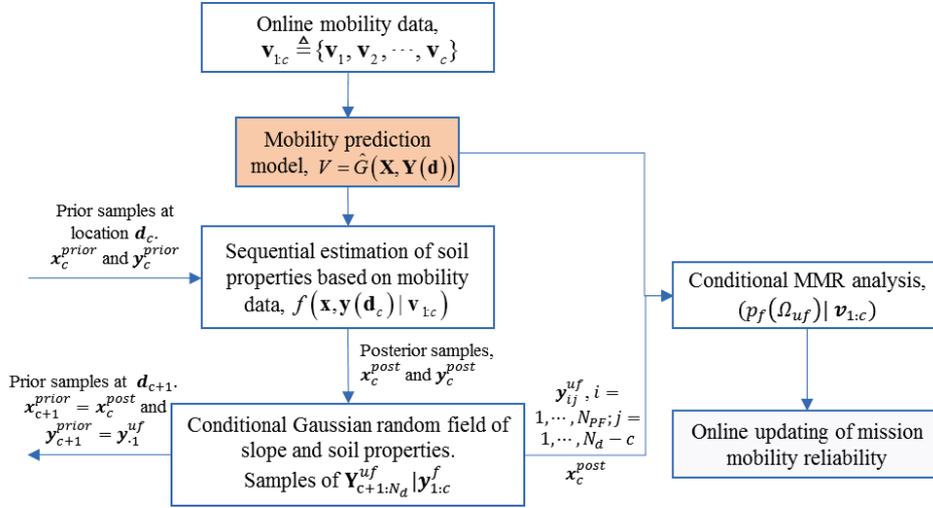


Figure 3.7 Flowchart of the dynamic updating of mission mobility reliability

3.2.3.1. Bayesian inference of uncertain parameters in the mobility model

The proposed framework starts with the online mobility measurement data $\mathbf{v}_{1:c}$ and sequentially estimates \mathbf{x} and $\mathbf{y}(\mathbf{d})$ based on $\mathbf{v}_{1:c}$. The estimated \mathbf{x} and $\mathbf{y}(\mathbf{d})$ are then used to update \mathbf{X} and $\mathbf{Y}(\mathbf{d})$, $\mathbf{d} \in \Omega_{uf}$ using conditional Gaussian random field. For given $\mathbf{v}_{1:c}$, the posterior PDF of \mathbf{X} and $\mathbf{Y}(\mathbf{d})$ at \mathbf{d}_c is estimated using the recursive Bayesian method as below

$$f(\mathbf{x}, \mathbf{y}(\mathbf{d}_c) | \mathbf{v}_{1:c}) = \frac{f(\mathbf{v}_c | \mathbf{x}, \mathbf{y}(\mathbf{d}_c))f(\mathbf{x}, \mathbf{y}(\mathbf{d}_c) | \mathbf{v}_{1:c-1})}{\iint f(\mathbf{v}_c | \mathbf{x}, \mathbf{y}(\mathbf{d}_c))f(\mathbf{x}, \mathbf{y}(\mathbf{d}_c) | \mathbf{v}_{1:c-1})d\mathbf{x}d\mathbf{y}(\mathbf{d}_c)} \quad (3.30)$$

$$\propto f(\mathbf{v}_c | \mathbf{x}, \mathbf{y}(\mathbf{d}_c))f(\mathbf{x}, \mathbf{y}(\mathbf{d}_c) | \mathbf{v}_{1:c-1}),$$

where “ \propto ” stands for “proportional to”, $f(\mathbf{v}_c | \mathbf{x}, \mathbf{y}(\mathbf{d}_c))$ is the likelihood function of observing \mathbf{v}_c for given \mathbf{x} and $\mathbf{y}(\mathbf{d}_c)$ at \mathbf{d}_c , and $f(\mathbf{x}, \mathbf{y}(\mathbf{d}_c) | \mathbf{v}_{1:c-1})$ is given by

$$f(\mathbf{x}, \mathbf{y}(\mathbf{d}_c) | \mathbf{v}_{1:c-1}) = \int f(\mathbf{y}(\mathbf{d}_c) | \mathbf{y}(\mathbf{d}_{c-1}))f(\mathbf{x}, \mathbf{y}(\mathbf{d}_{c-1}) | \mathbf{v}_{1:c-1})d\mathbf{y}(\mathbf{d}_{c-1}), \quad (3.31)$$

in which $f(\mathbf{y}(\mathbf{d}_c) | \mathbf{y}(\mathbf{d}_{c-1}))$ is the PDF of $\mathbf{y}(\mathbf{d}_c)$ conditioned on $\mathbf{y}(\mathbf{d}_{c-1})$.

As illustrated in Figure 3.7, the mobility prediction model $V = \hat{G}(\mathbf{X}, \mathbf{Y}(\mathbf{d}))$, is employed to connect mobility measurement data \mathbf{v}_c with \mathbf{x} and $\mathbf{y}(\mathbf{d}_c)$. $f(\mathbf{v}_c | \mathbf{x}, \mathbf{y}(\mathbf{d}_c))$ is thus computed by

$$f(\mathbf{v}_c | \mathbf{x}, \mathbf{y}(\mathbf{d}_c)) = \prod_{j=1}^{N_c} \phi \left(\frac{v_{cj} - \mu_V(\mathbf{x}, \mathbf{y}(\mathbf{d}_c))}{\sqrt{\sigma_\varepsilon^2 + \sigma_V^2(\mathbf{x}, \mathbf{y}(\mathbf{d}_c))}} \right), \quad (3.32)$$

where $\phi(\cdot)$ is the PDF of a standard normal variable, $\mathbf{v}_c = [v_{c1}, v_{c2}, \dots, v_{cN_c}]$, σ_ε is the standard deviation of the observation noise, N_c is the number of observations at location \mathbf{d}_c , $\mu_V(\mathbf{x}, \mathbf{y}(\mathbf{d}_c))$ and $\sigma_V(\mathbf{x}, \mathbf{y}(\mathbf{d}_c))$ are respectively the mean and standard deviation of the vehicle mobility prediction obtained from the surrogate model $V = \hat{G}(\mathbf{X}, \mathbf{Y}(\mathbf{d}))$, (i.e. Eq. (3.20)).

The equation given in Eqs. (3.30) and (3.31) are analytically intractable. In this chapter, the Particle Filtering (PF) method is employed to solve Eqs. (3.30) and (3.31) recursively [77]. Since the goal of during-mission phase is to perform online updating of MMR, the PF method can also be replaced with unscented Kalman filter method [78] to further improve the efficiency of online updating. Let the prior samples of \mathbf{X} and $\mathbf{Y}(\mathbf{d})$ at \mathbf{d}_c be $\mathbf{x}_c^{prior} = [\mathbf{x}_c^{(1)}, \mathbf{x}_c^{(2)}, \dots, \mathbf{x}_c^{(N_{PF})}]$ and, where N_{PF} is the number of particles in PF, the posterior samples $\mathbf{x}_c^{post} = [\mathbf{x}_c^{post}(1), \mathbf{x}_c^{post}(2), \dots, \mathbf{x}_c^{post}(N_{PF})]$ and $\mathbf{y}_c^{post} = [\mathbf{y}_c^{post}(1), \mathbf{y}_c^{post}(2), \dots, \mathbf{y}_c^{post}(N_{PF})]$ are obtained by resampling \mathbf{x}_c^{prior} and \mathbf{y}_c^{prior} based on the following weights

$$w_j = \frac{f(\mathbf{v}_c | \mathbf{x}_c^{(j)}, \mathbf{y}_c^{(j)})}{\sum_{i=1}^{N_{PF}} f(\mathbf{v}_c | \mathbf{x}_c^{(i)}, \mathbf{y}_c^{(i)})}, \quad \forall j = 1, 2, \dots, N_{PF}, \quad (3.33)$$

where $f(\mathbf{v}_c | \mathbf{x}_c^{(i)}, \mathbf{y}_c^{(i)})$, $i = 1, 2, \dots, N_{PF}$ is the likelihood function computed using Eq. (3.32).

The posterior samples \mathbf{x}_c^{post} will be directly used as the prior samples for \mathbf{X} at the next spatial point \mathbf{d}_{c+1} since random variables are constant over space. However, \mathbf{y}_c^{post} cannot be directly used as prior samples of $\mathbf{Y}(\mathbf{d})$ at \mathbf{d}_{c+1} due to the spatially varying characteristics of $\mathbf{Y}(\mathbf{d})$. In addition, the discontinuity of soil type and slope ID also complicates the generation of prior samples for $\mathbf{Y}(\mathbf{d})$ at \mathbf{d}_{c+1} . This chapter develops a numerical procedure (as depicted in Figure 3.8) to tackle these challenges.

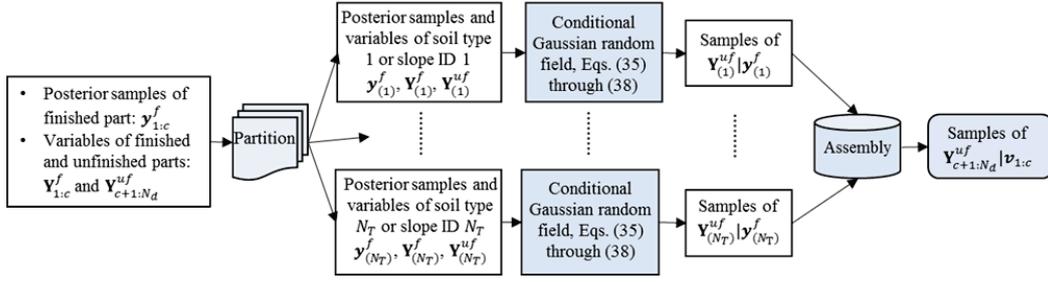


Figure 3.8 Flowchart of generating samples for the unfinished part of the mission path

3.2.3.2. Uncertainty sources updating

Let the posterior samples of the finished part of the mission path (as indicated in Figure 3.6) be $\mathbf{y}_{1:c}^f \triangleq \{\mathbf{y}_1^{post}, \mathbf{y}_2^{post}, \dots, \mathbf{y}_c^{post}\}$, where $\mathbf{y}_i^{post} = [\mathbf{y}_i^{post}(1), \mathbf{y}_i^{post}(2), \dots, \mathbf{y}_i^{post}(N_{PF})]$, $\forall i = 1, 2, \dots, c$, the soil properties and slope, $\mathbf{Y}(\mathbf{d})$, $\forall \mathbf{d} \in \Omega_f$ of the finished part of the mission path be $\mathbf{Y}_{1:c}^f$, and $\mathbf{Y}(\mathbf{d})$, $\forall \mathbf{d} \in \Omega_{uf}$ of the unfinished part of the mission path be $\mathbf{Y}_{c+1:N_d}^{uf}$, we first partition $\mathbf{y}_{1:c}^f$ into $\mathbf{y}_{(1)}^f, \mathbf{y}_{(2)}^f, \dots, \mathbf{y}_{(N_{soil})}^f$ if they are soil type-related or $\mathbf{y}_{(1)}^f, \mathbf{y}_{(2)}^f, \dots, \mathbf{y}_{(N_{slope})}^f$ if they are slope ID-related. Similarly, we partition $\mathbf{Y}_{1:c}^f$ and $\mathbf{Y}_{c+1:N_d}^{uf}$ according the slope ID or soil type as follows

$$\begin{aligned} \mathbf{Y}_{1:c}^f &\stackrel{\text{Partition}}{\Rightarrow} \{\mathbf{Y}_{(1)}^f, \mathbf{Y}_{(2)}^f, \dots, \mathbf{Y}_{(N_{soil})}^f\} \text{ or } \{\mathbf{Y}_{(1)}^f, \mathbf{Y}_{(2)}^f, \dots, \mathbf{Y}_{(N_{slope})}^f\}, \\ \mathbf{Y}_{c+1:N_d}^{uf} &\stackrel{\text{Partition}}{\Rightarrow} \{\mathbf{Y}_{(1)}^{uf}, \mathbf{Y}_{(2)}^{uf}, \dots, \mathbf{Y}_{(N_{soil})}^{uf}\} \text{ or } \{\mathbf{Y}_{(1)}^{uf}, \mathbf{Y}_{(2)}^{uf}, \dots, \mathbf{Y}_{(N_{slope})}^{uf}\}. \end{aligned} \quad (3.34)$$

After the partition of $\mathbf{y}_{1:c}^f$, $\mathbf{Y}_{1:c}^f$, and $\mathbf{Y}_{c+1:N_d}^{uf}$, for the i -th slope ID or soil type and given sample of $\mathbf{y}_{1:c}^f$ as below

$$\begin{aligned} \mathbf{y}_{1:c}^f(j) &\triangleq \{\mathbf{y}_1^{post}(j), \mathbf{y}_2^{post}(j), \dots, \mathbf{y}_c^{post}(j)\} \\ &= \{\mathbf{y}_{(1)}^f(j) \cup \dots \cup \mathbf{y}_{(N_T)}^f(j)\}, \quad j = 1, 2, \dots, N_{PF}, \end{aligned} \quad (3.35)$$

the conditional mean $\boldsymbol{\mu}_{(i)}^{uf} | \mathbf{y}_{(i)}^f(j)$ and covariance matrix $\boldsymbol{\Sigma}_{(i)}^{uf} | \mathbf{y}_{(i)}^f(j)$ of $\mathbf{Y}_{(i)}^{uf} | \mathbf{y}_{(i)}^f(j)$ are obtained as follows

$$\boldsymbol{\mu}_{(i)}^{uf} | \mathbf{y}_{(i)}^f(j) = \boldsymbol{\mu}_{fu}^{(i)} + \boldsymbol{\Sigma}_{fu}^{(i)} (\boldsymbol{\Sigma}_{ff}^{(i)})^{-1} (\mathbf{y}_{(i)}^f(j) - \boldsymbol{\mu}_f^{(i)}), \quad (3.36)$$

$$\boldsymbol{\Sigma}_{(i)}^{uf} | \mathbf{y}_{(i)}^f(j) = \boldsymbol{\Sigma}_{uu}^{(i)} - \boldsymbol{\Sigma}_{fu}^{(i)} (\boldsymbol{\Sigma}_{ff}^{(i)})^{-1} (\boldsymbol{\Sigma}_{fu}^{(i)})^T, \quad (3.37)$$

where $i = 1, 2, \dots, N_T$, $N_T = N_{soil}$ if the variable is a soil type-related variable and

$N_T = N_{slope}$ if it is a slope-related variable, $\boldsymbol{\mu}_{uf}^{(i)}$ is the unconditional mean values of $\mathbf{Y}_{(i)}^{uf}$, $\boldsymbol{\mu}_f^{(i)}$ is the unconditional mean values of $\mathbf{Y}_{(i)}^f$, $\boldsymbol{\Sigma}_{fu}^{(i)}$ is the covariance matrix between $\mathbf{Y}_{(i)}^f$ and $\mathbf{Y}_{(i)}^{uf}$, $\boldsymbol{\Sigma}_{ff}^{(i)}$ is the auto-covariance matrix of $\mathbf{Y}_{(i)}^f$, and $\boldsymbol{\Sigma}_{uu}^{(i)}$ is the auto-covariance matrix of $\mathbf{Y}_{(i)}^{uf}$. The covariance matrices are obtained similarly to Eq. (3.16), according to the coordinates of the spatial points.

Based on $\boldsymbol{\mu}_{(i)}^{uf} | \mathbf{y}_{(i)}^f(j)$ and $\boldsymbol{\Sigma}_{(i)}^{uf} | \mathbf{y}_{(i)}^f(j)$, we can then generate a group of samples for $\mathbf{Y}_{(i)}^{uf} | \mathbf{y}_{(i)}^f(j)$ using multivariate Gaussian distribution as follows

$$\mathbf{y}_{(i)}^{uf}(j) = [y_{(i)}^{uf}(1, j), y_{(i)}^{uf}(2, j), \dots, y_{(i)}^{uf}(N_{ui}, j)], \quad (3.38)$$

where N_{ui} is the number of variables in $\mathbf{Y}_{(i)}^{uf}$.

Implementing Eqs. (3.35) through (3.38) for all $i = 1, 2, \dots, N_T$ and $j = 1, 2, \dots, N_{PF}$, we obtain N_{PF} samples for $\mathbf{Y}_{(1)}^{uf}, \mathbf{Y}_{(2)}^{uf}, \dots, \mathbf{Y}_{(N_{soil})}^{uf}$ or $\mathbf{Y}_{(1)}^{uf}, \mathbf{Y}_{(2)}^{uf}, \dots, \mathbf{Y}_{(N_{slope})}^{uf}$. The samples are then assembled together according to their coordinates (as indicated in Figure 3.8) to form random samples of $\mathbf{Y}_{c+1:N_d}^{uf} | \mathbf{y}_{1:c}^f$. Defining the obtained samples as $\mathbf{y}_{ij}^{uf}, i = 1, 2, \dots, N_{PF}; j = 1, 2, \dots, (N_d - c)$, $\mathbf{y}_{\cdot 1}^{uf} = [\mathbf{y}_{11}^{uf}, \mathbf{y}_{21}^{uf}, \dots, \mathbf{y}_{N_{PF}1}^{uf}]$ is then the prior samples of $\mathbf{Y}(\mathbf{d})$ at spatial point \mathbf{d}_{c+1} . By implementing the above procedure recursively, we are able to sequentially estimate the soil properties and vehicle parameters based on the vehicle mobility measurement data.

3.2.3.3. MMR updating and remaining mobilable distance estimation

Since \mathbf{y}_{lc}^f are estimated posterior samples of \mathbf{Y}_{lc}^f conditioned on \mathbf{v}_{lc} , $\mathbf{y}_{ij}^{uf}, i = 1, 2, \dots, N_{PF}; j = 1, 2, \dots, (N_d - c)$ are also samples of $\mathbf{Y}_{c+1:N_d}^{uf} | \mathbf{v}_{lc}$. As indicated in Figure 3.7, the samples of $\mathbf{Y}_{c+1:N_d}^{uf} | \mathbf{v}_{lc}$ and \mathbf{x}_c^{post} are then used as inputs to Eq. (3.29) to estimate the conditional MMR $R(\Omega) | \mathbf{v}_{lc}$ as below

$$R(\Omega) | \mathbf{v}_{lc} \approx \frac{1}{N_{PF}} \sum_{i=1}^{N_{PF}} I_d \{ \mu_V(i, j) \geq e, \forall j = 1, 2, \dots, N_d - c \}, \quad (3.39)$$

where $\mu_V(i, j)$ is the mean mobility prediction $\mu_V(\mathbf{x}_c^{post}(i), \mathbf{y}_{ij}^{uf})$ obtained by plugging

$\mathbf{x}_c^{post}(i)$ and \mathbf{y}_{ij}^{uf} into the mobility surrogate model, $v = \hat{G}(\mathbf{x}_c^{post}(i), \mathbf{y}_{ij}^{uf})$.

Alternatively, the mission mobility failure probability is computed by

$$p_f(\Omega) | \mathbf{v}_{lc} \approx \frac{1}{N_{PF}} \sum_{i=1}^{N_{PF}} I_d \{ \mu_V(i, j) < e, \exists j = 1, 2, \dots, N_d - c \}. \quad (3.40)$$

In addition to $R(\Omega) | \mathbf{v}_{lc}$ and $p_f(\Omega) | \mathbf{v}_{lc}$, we can also obtain the cumulative density function (CDF) of the remaining mobilable distance (RMD) at current location \mathbf{d}_c as

$$\Pr\{RMD \leq N_e | \mathbf{v}_{lc}\} = \frac{1}{N_{PF}} \sum_{i=1}^{N_{PF}} I_d \{ \mu_V(i, j) < e, \exists j = 1, 2, \dots, N_e \}, \quad (3.41)$$

in which $N_e \leq N_d - c$ is a specific threshold of RMD and $I_d\{\cdot\}$ is an indicator function defined in Eq. (3.27). The above CDF value presents the probability that the vehicle will be immobilable in the future N_e spatial points conditioned on current vehicle measurements \mathbf{v}_{lc} .

As the vehicle proceeds on the mission pass, $R(\Omega) | \mathbf{v}_{lc}$ and $\Pr\{RMD \leq N_e | \mathbf{v}_{lc}\}$ will be updated. If the updated MMR shows that the $R(\Omega) | \mathbf{v}_{lc}$ is too low, the decision maker needs to decide whether the vehicle should continue this path or a new path needs to be identified based on $\Pr\{RMD \leq N_e | \mathbf{v}_{lc}\}$. Next, a case study is used to demonstrate the two phases of the proposed simulation-based MMR analysis.

3.3. Case Study

In this section, an off-road map is used to demonstrate the efficacy of the proposed framework. The application problem is first introduced in Section 3.3.1. Following that, Section 3.3.2 implements the MMR analysis in mission-planning phase, and Section 3.3.3 demonstrates the dynamic updating of MMR during the mission.

3.3.1. Description of the MMR Analysis Problem

As shown in Figure 3.9, a map is taken from ArcGIS/ENVI database [79] and US Geological Survey database [80], and a part of the map is selected to demonstrate the

proposed framework. The slope and soil maps of the selected area are shown in Figure 3.10 accordingly, which are discretized by 1046 points in the coordinate d_1 and 707 points in the coordinate d_2 .

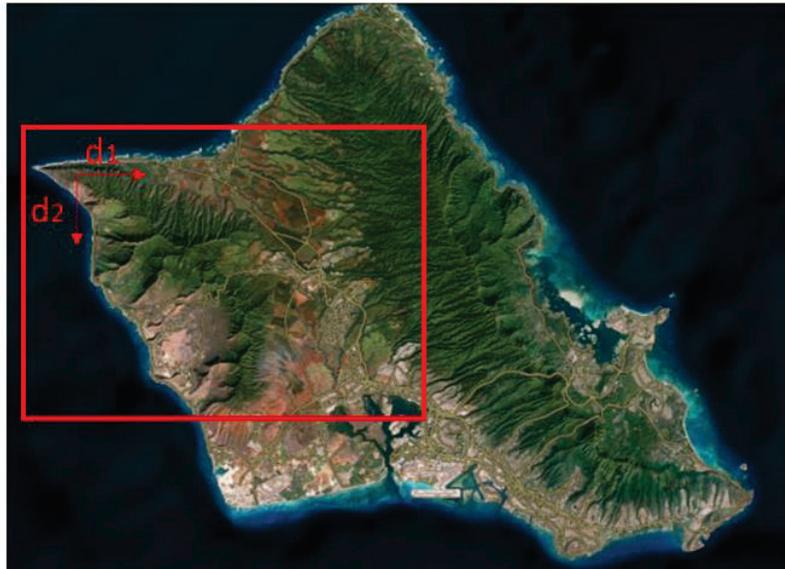


Figure 3.9 Original map from the satellite [79]

As shown in Figure 3.10, there are 10 different slope intervals and soil types represented by different IDs, with slope ID zero and soil ID zero represents water. Additionally, each slope ID is associated with a slope parameter, and each soil ID is associated with three different soil parameters, i.e. cohesive strength, friction coefficient, and bulk density. In this chapter, all these parameters are considered as stationary Gaussian random field. Table 3 presents statistical information of different parameters corresponding to different slope/soil IDs, which are assumed based on data obtained from the US Geological Survey database [80].

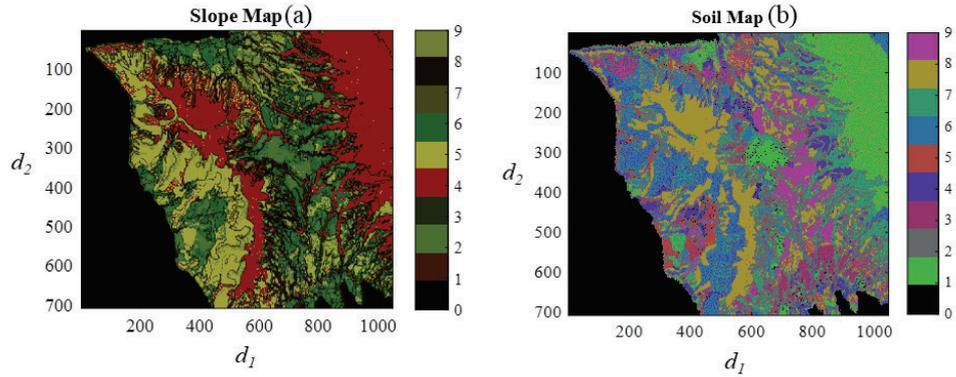


Figure 3.10 The slope and soil IDs of the selected area

Table 3.1 Distribution and correlation parameters of slope and soil properties

Slope/ Soil ID	Slope		Soil					
			Cohesive strength		Friction coefficient		Bulk density	
	μ^*/σ^*	β_1^*/β_2^*	μ/σ	β_1/β_2	μ/σ	β_1/β_2	μ/σ	β_1/β_2
0	Water		Water					
1	5/0.2	3.3/3.0	0.2/0.01	40/39	0.01/0.001	33/32	0.05/0.001	27/28
2	10/0.5	3.4/3.6	2/0.2	42/43	0.4/0.05	28/33	1.2/0.1	23/27
3	12/0.5	4.3/2.2	5/0.4	41/41	0.7/0.02	28/27	2.1/0.1	20/25
4	14/1	3.6/4.4	6/0.2	41/39	0.76/0.02	31/29	2.3/0.1	26/24
5	16/2	3.5/5.1	2/0.1	40/42	0.56/0.01	33/32	1.6/0.04	27/21
6	18/2	2.8/5.0	11/0.2	38/37	0.8/0.02	27/27	2.1/0.05	26/28
7	22/2	3.2/2.2	5/0.02	39/42	0.7/0.03	28/30	1.2/0.02	24/28
8	24/1	4.7/4.0	4/0.1	37/39	0.45/0.005	29/28	1.45/0.01	24/27
9	28/1	4.5/6.7	8/0.4	42/35	0.78/0.03	32/32	2.35/0.05	26/24

* μ is mean, σ is standard deviation, β_1 is the correlation length in coordinate 1, and β_2 is the correlation length in coordinate 2.

In this chapter, the failure event is defined as the maximum attainable speed is less than 2 m/s (i.e. $e=2$ in Eq. (3.9)). The off-road mobility model is assumed to be

$$\begin{aligned}
 V(\mathbf{d}) &= G(\mathbf{X}, \mathbf{Y}(\mathbf{d})) \\
 &= \exp\left(0.013Y_2^2(\mathbf{d}) + 0.8\cos\left(\pi \cdot (0.5Y_1(\mathbf{d}) - 1.5Y_3(\mathbf{d}))/180\right) + \sin\left(\pi Y_4(\mathbf{d})/45\right)\right) \\
 &\quad + Y_3(\mathbf{d})Y_4(\mathbf{d}) - 0.207,
 \end{aligned} \tag{3.42}$$

where $V(\mathbf{d})$ is the vehicle speed at a spatial point \mathbf{d} , $\mathbf{Y}(\mathbf{d}) = [Y_1(\mathbf{d}), Y_2(\mathbf{d}), Y_3(\mathbf{d}), Y_4(\mathbf{d})]$

is a vector of terrain parameters which has 4 parameters, i.e. slope ($Y_1(\mathbf{d})$), cohesive

strength ($Y_2(\mathbf{d})$), friction coefficient ($Y_3(\mathbf{d})$), bulk density ($Y_4(\mathbf{d})$). For the sake of illustration, the uncertainty of vehicle parameters \mathbf{X} is not considered in this case study.

For the mission plan of the ground vehicle at the selected map, a path is selected from coordinates (250,150) to (900,450) as shown in Figure 3.11. As a result, there are 400 spatial points (coordinates) on the selected route. Figure 3.12 gives the slope IDs and soil IDs of each point.

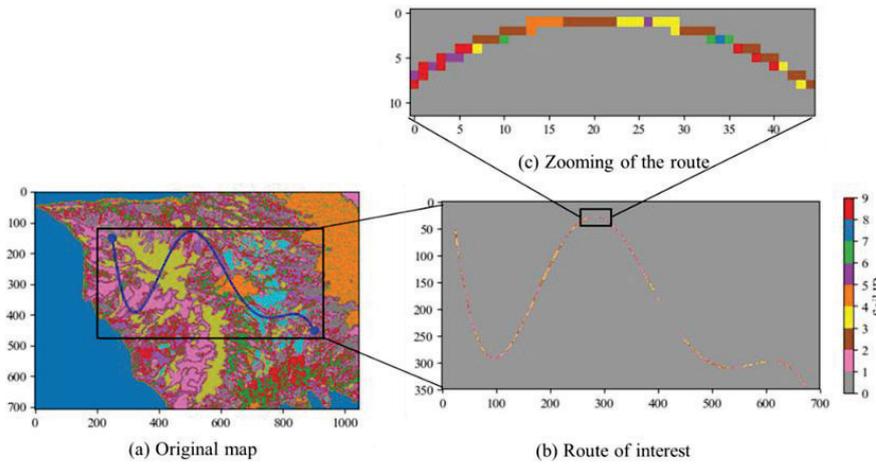


Figure 3.11 Route chosen from the map (Soil)

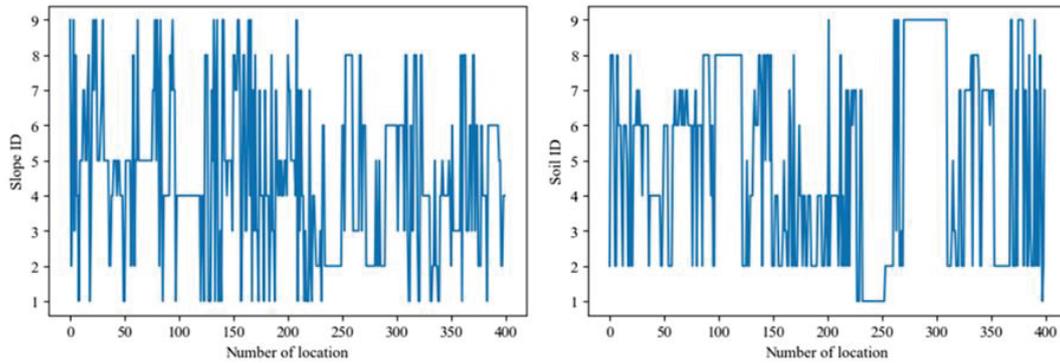


Figure 3.12 Slope and Soil ID of each coordinate on the path

Next, we will demonstrate how to perform surrogate modeling-based MMR analysis in the mission-planning phase and how to update the MMR in the during-mission phase.

3.3.2. Mission-Planning Phase: Surrogate-Modeling Based MMR Analysis

As discussed above, the parameters $Y_i(\mathbf{d}), i = 1, 2, 3, 4$ are space-dependent random

fields. Before estimating MMR, random realizations of these parameters are generated using the KL expansion method using Eqs. (3.13) through (3.16). In addition, 15 initial training points are generated for $Y_i(\mathbf{d}), i=1,2,3,4$ using the approach discussed in Sec. 3.2.2. After that, an initial surrogate $V = \hat{G}(\mathbf{X}, \mathbf{Y}(\mathbf{d}))$ is constructed based on the training points.

The surrogate model is then adaptively refined using the SILK method as discussed in Sec. 3.2.2. Figure 3.13 shows the predicted mission mobility failure probability ($p_f(\Omega)$) with respect to the added number of function evaluations. The result shows that the SILK-based MMR analysis method only needs additional 4 function evaluations to get a very accurate estimate of the mission mobility failure probability. In this figure, the true mission mobility failure probability is estimated using the brutal-force MCS method with $N_{MCS} = 10^4$ samples. This demonstrates the high efficiency of adaptive surrogate modeling-based MMR analysis method.

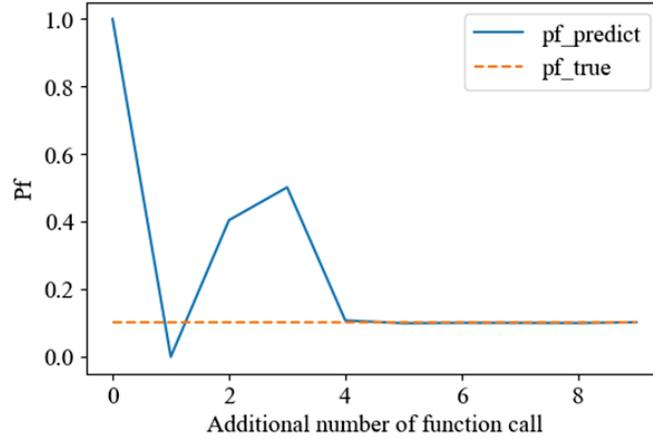


Figure 3.13 Mission mobility failure probability w.r.t. the added number of function evaluations

3.3.3. During-Mission Phase: Dynamic Updating of MMR

From the mission-planning phase, as depicted in Figure 3.13, it shows that the mission mobility failure probability of the selected mission path (i.e. Figure 3.11) is 0.1022 from both the MCS and adaptive Kriging surrogate modeling-based method. Even though the failure probability is quite low, rare events of immobility is still possible to happen. To proactively avoid the rare events, MMR is dynamically updated based on the online

mobility data as discussed in Sec. 3.2.3.

In order to demonstrate the dynamic updating process, we first select a realization of the terrain properties that leads to failure of the mission. The selected terrain properties are then assumed to be unknown during the updating process. Figure 3.14 depicts the vehicle speed corresponds to the selected realization of terrain properties at each point. It shows that if the vehicle proceeds on the mission path (i.e. Figure 3.11), an immobility event (i.e. maximum attainable speed < 2 m/s) will happen at the 232-th point. If the proposed framework can predict the immobility event before the vehicle reaches to the 232-th point, it could save the vehicle from the immobility event.

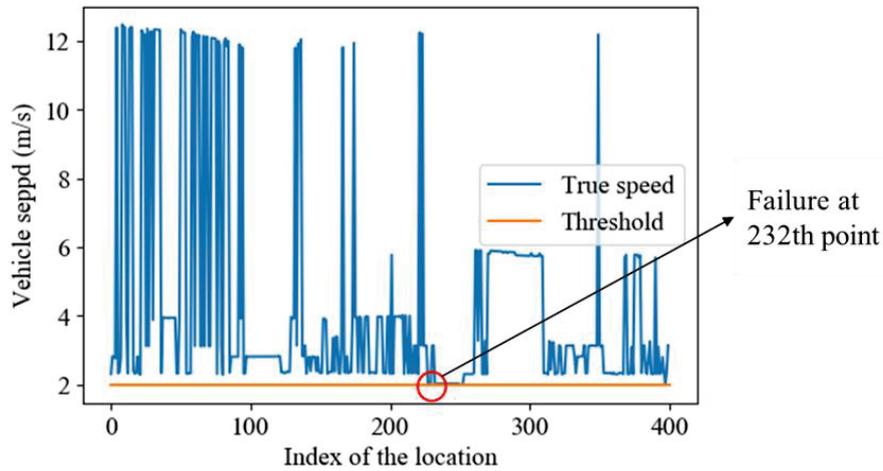


Figure 3.14 Plot of vehicle speed vs. coordinates

In this chapter, the slope at each location is assumed to be directly measurable when a vehicle passes a certain point, while the other soil properties are not directly measurable. Using the Bayesian method discussed in Sec. 3.2.2, we can estimate the posterior distributions of the soil properties as mobility data of the vehicle are collected. Figure 3.15 shows the estimated posterior distributions of soil parameters at the first point, when the vehicle starts to proceed on the selected mission path. Comparing with the prior distribution, it is observed that the uncertainty in the soil property parameters is reduced. Based on the Bayesian inference at the first point, we can then update the distribution of the remaining points on the mission path by following the procedure as shown in Figure 3.8. Based on

the updated distributions, we can further calculate the mission mobility failure probability of the remaining part of the mission path. As discussed in Sec. 3.2.2, the updated distributions of the parameters are also used as the prior distributions for the next spatial point. The aforementioned process continues as the vehicle proceeds on the mission path.

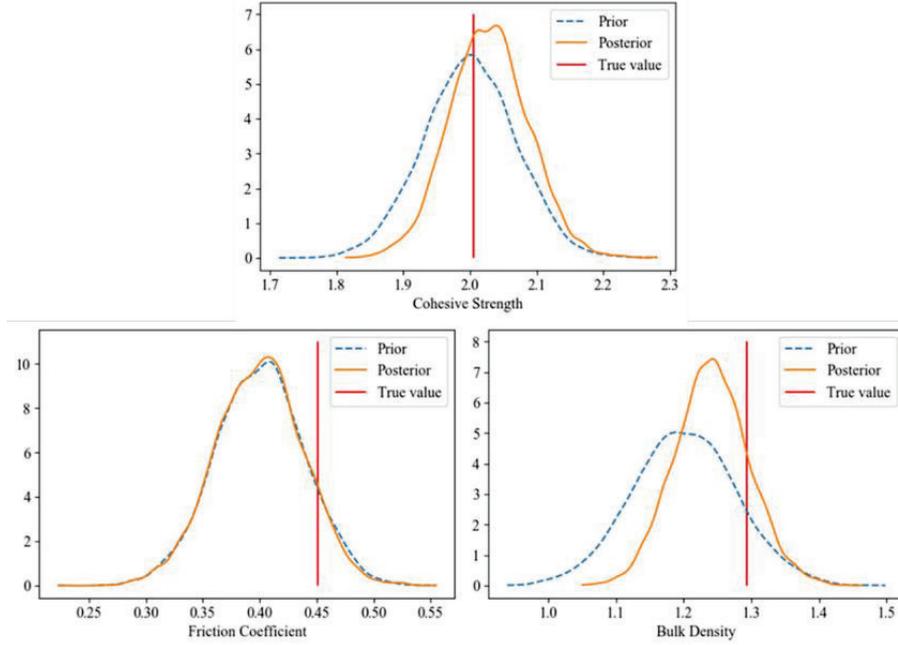


Figure 3.15 Distribution of the soil property parameters at the first point

Figure 3.16 presents the 95% confidence interval and mean value of the posterior distributions of the soil property parameters after the vehicle proceeds 50 points on the mission path. The results show that the mean values of the posterior distribution of soil parameters are close to the underlying true values which are assumed to be unknown. Following that, Figure 3.17 presents the updating history of the mission mobility failure probability as the vehicle proceeds over the mission path. It shows that $p_f(\Omega) | \mathbf{v}_{1:c}$ goes up close to 1 at a point close to the 30-th point which is way ahead the 232-th point where the failure (i.e. immobility) actually happens. Figure 3.18 presents the updating history of the RMD over the mission path as the vehicle proceeds on the route. It can see that the proposed dynamic updating framework is able to accurately estimate the RMD as the vehicle proceeds on the route and more observations of the vehicle mobility are collected.

This demonstrates the effectiveness of proposed framework in proactively identifying the rare events of immobility before it happens.

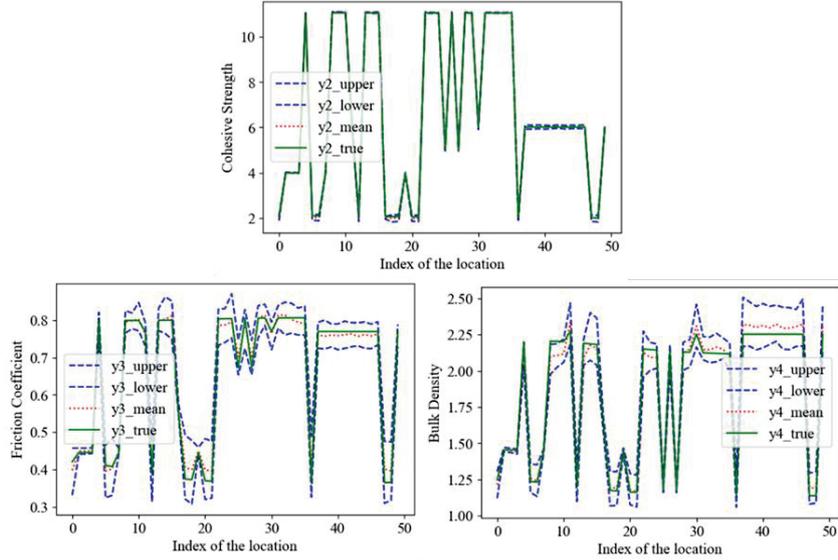


Figure 3.16 Comparison of the estimated soil property parameters and true values

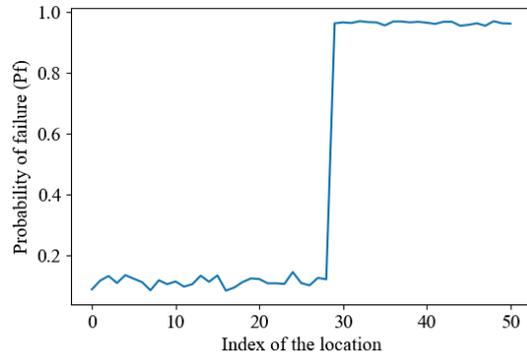


Figure 3.17 Updating history of $p_f(\Omega) | \mathbf{v}_{1c}$ as the vehicle proceeds on the mission path

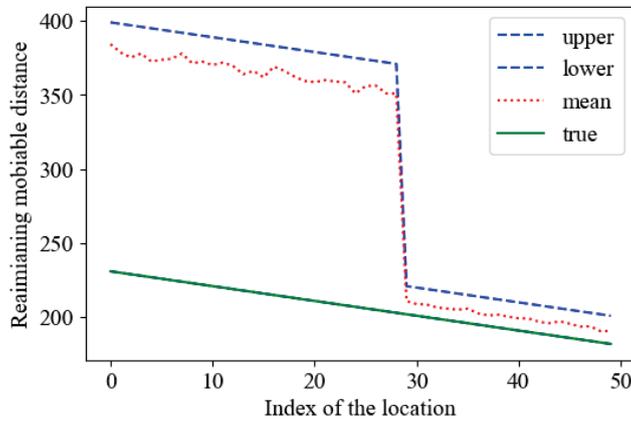


Figure 3.18 Updating history of the remaining mobilable distance (RMD) on a path

3.4. Summary

This chapter proposes a simulation-based MMR analysis framework to account for the uncertainty and improve the prediction accuracy of off-road ground vehicle mobility. The framework has two phases, including a mission-planning phase and a during-mission phase. For the mission-planning phase, a single-loop Kriging surrogate modeling approach is employed to reduce the computational cost by selecting the most critical training points as well as maintaining the prediction accuracy of MMR. For the during-mission phase, a Bayesian updating framework is proposed to reduce the uncertainty of terrain parameters with observed vehicle mobility information. Using conditional Gaussian process, we can also reduce the uncertainty of the path that has not been accomplished and thus dynamically update the MMR. The dynamic updating of MMR allows us to proactively avoid the rare occurrence of failures before it happens. A case study using data from ArcGIS/ENVI database and US Geological Survey database demonstrates the efficacy of the proposed framework.

Chapter 4

Reliability-Based Multi-Vehicle Path Planning Under Uncertainty Using a Physarum-Based Approach

The goal of this chapter is to develop a model-based mission planning framework to guarantee the reliability of AGV operation in the early design stage. As shown in Figure 4.1, a proper model-based off-road mission planning process mainly consists of two phases: (1) *transforming the target map into a vehicle mobility map* using the target map information and the off-road mobility predictive model, and (2) *finding the optimal path* under predefined mission requirements based on the vehicle mobility map.

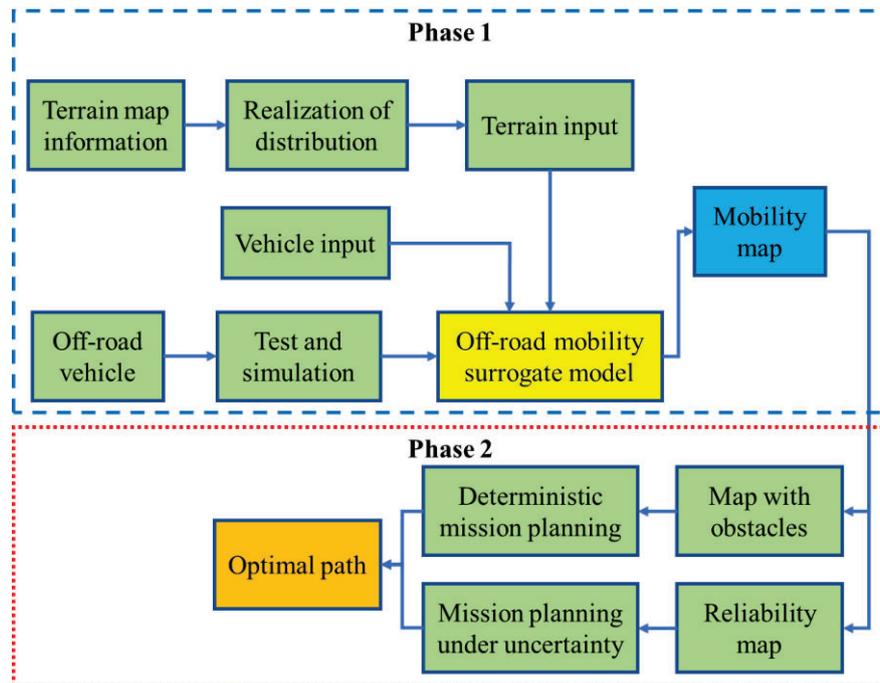


Figure 4.1 Process of mission planning for off-road vehicles

This chapter proposed a novel path planning approach based on a bio-inspired Physarum-based algorithm. We first use an open-source high-fidelity dynamic simulation model named PyChrono to predict vehicle mobility in uncertain off-road environment. To

overcome the computational challenge in mobility prediction uncertainty caused by the PyChrono simulations, we employ an adaptive surrogate modeling method to construct an efficient and yet accurate surrogate model of the original mobility simulation model. The constructed surrogate model is then used to generate a probabilistic vehicle mobility map by considering various uncertainty sources in the terrain and soil parameters. After that, we transform the original mobility map into a navigation map using a navigation mesh to further reduce the computational time in mission planning by reducing the nodes and edges in a network used to represent the map. Finally, we develop single vehicle and multi-vehicle path planning subjected to a reliability constraint using a novel Physarum-based algorithm. The main contributions of this chapter are summarized as: (1) a systematic reliability-based multi-vehicle mission planning framework is developed for AGVs in uncertain off-road environments using model-based approach in the early design; (2) synthesis of an adaptive surrogate modeling method and a navigation mesh is used to drastically reduce the computational time in reliability-based mission planning; and (3) a novel bio-inspired Physarum algorithm is used to efficiently perform multi-vehicle mission planning in a dynamic environment. Even though the proposed method is demonstrated using an off-road AGV, it is also applicable to path planning of unmanned aerial vehicles in uncertain environment. This part has been published at:

- [81] Liu, Y., Jiang, C., Zhang, X., Mourelatos, Z. P., Barthlow, D., Gorsich, D., Singh, A., and Hu, Z., 2022, "Reliability-Based Multivehicle Path Planning Under Uncertainty Using a Bio-Inspired Approach," *Journal of Mechanical Design*, 144(9), p. 091701.

4.1. Background

4.1.1. Off-Road Vehicle Mobility Analysis Model

Although many approaches have been developed in the past decades, it is still a

challenging issue to accurately predict the off-road mobility of an AGV due to the complex physical interactions between the vehicle and terrain [82], especially for deformable terrain conditions. For example, Rubinstein et al. [83] developed a detailed multi-body dynamic simulation model for tracked off-road vehicles. Liang et al. [84] proposed a semi-empirical approach of tire modeling to simulate the tire/terrain interaction. Krenn et al. [85] combined soft soil contact models with Bekker's terramechanics theory to create an efficient approach for multi-body dynamic simulation. Senatore et al. [86] used rigid wheels and flexible tires to build a tire model for off-road dynamic simulation.

In this chapter, the open source multi-physics simulation package, PyChrono [35, 87], is employed to predict vehicle mobility for any given terrain/soil conditions and to generate a vehicle mobility map (Phase 1 in Figure 4.1). PyChrono is the Python version of Chrono which is programmed in C++. The advantages of PyChrono are that it is simple to use and to be combined with other python libraries. More specifically, the Pychrono: Vehicle module is used in this chapter. It provides different types of vehicles running on a rigid, granular, or deformable terrain. A driving vehicle controller can also be used for closed-loop or interactive vehicle behavior control. Figure 4.2 shows two examples in PyChrono.

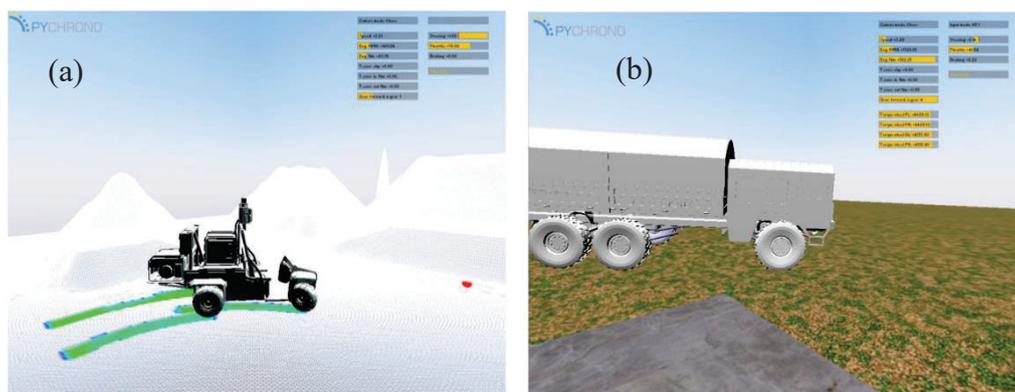


Figure 4.2 Two simulation examples of PyChrono: Vehicle: (a) gator UTV under deformable terrain, (b) MAN truck under rigid terrain.

The vehicle frames in the PyChrono module are defined using ISO 8855:2011. Figure 4.3 presents an overview flowchart of the interactions of different modules during the

simulation. As shown in Figure 4.3, all subsystems can be customized in the simulation. For example, the vehicle type, the powertrain, and tires can be switched for different predefined vehicle types in the module, and the driver and terrain can be defined by assigning different values.

In this chapter, the SCMDDeformableTerrain module is used for the terrain to better represent the interaction between tire and soil. The model is defined by Gallina et al. [88] and its input parameters are summarized in Table 4.1.

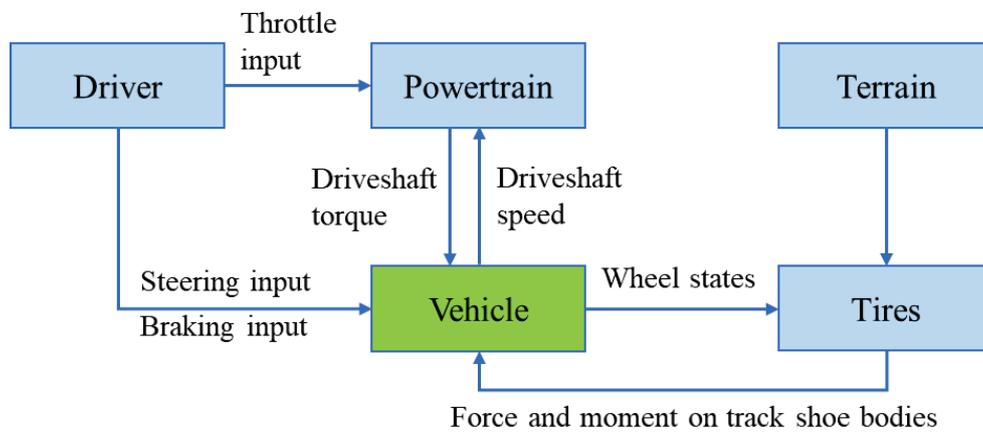


Figure 4.3 Simulation flowchart of PyChrono::Vehicle

Table 4.1 Input parameters of the SCMDDeformableTerrain module

Map parameter	Size parameter	Soil parameter
Height map	sizeX	Bekker_Kphi
	sizeY	Bekker_Kc
	hMin	Bekker_n
	hMax	Mohr_cohesion
		Mohr_friction
		Janosi_shear

Figure 4.4 presents an example of the inputs and output of the off-road vehicle mobility prediction model. As indicated in the figure, the model takes a height map, soil parameters, and the coordinates of the path as input. Based on these inputs, it predicts the performance of a specific vehicle following a path in a defined area. Various vehicle characteristics including vehicle acceleration, vehicle speed along the path, and

displacement of four wheels, can be simulated.

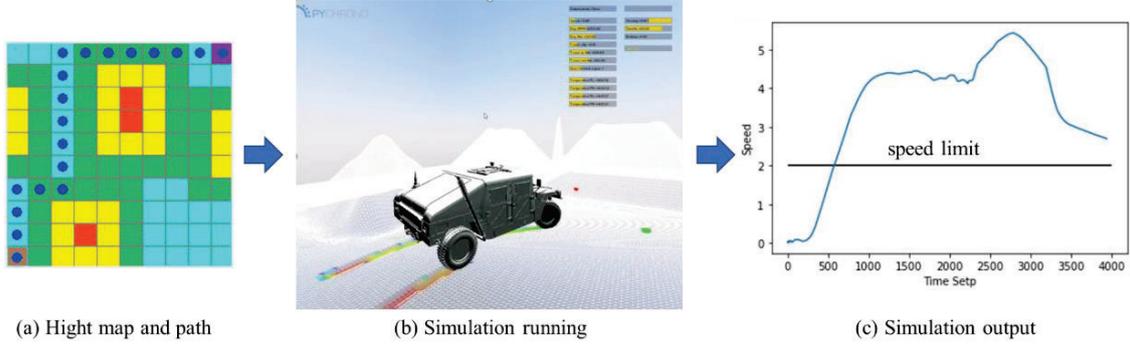


Figure 4.4 A simulation model of off-road AGV in PyChrono

In general, for a given vehicle design Λ such as vehicle type, powertrain, tires, etc. as shown in Figure 4.3, the mobility model of an off-road AGV can be represented as

$$V(\mathbf{d}) = G(\mathbf{X}(\Lambda), \mathbf{Y}(\mathbf{d}), \Lambda), \quad (4.1)$$

where \mathbf{d} are the spatial coordinates, $V(\mathbf{d})$ is the vehicle mobility at spatial location \mathbf{d} , $G(\cdot)$ is the mobility simulation model, $\mathbf{X}(\Lambda)$ are the vehicle-related parameters, such as vehicle weight, size, and power, and \mathbf{Y} a function of spatial coordinate \mathbf{d} representing the terrain-related parameters, such as slope, friction coefficient, cohesive limit, etc.

4.1.2. Deterministic Mission Planning

In deterministic mission planning, there are two types of subjects in a map \mathbf{M} : the free space \mathbf{F} , and the obstacle space \mathbf{O} , so that $\mathbf{M} = \mathbf{F} \cup \mathbf{O}$. The objective of mission planning is to generate an optimal path Ω_{opt} with the shortest distance while not crossing any obstacles as

$$\Omega_{opt} = \arg \min_{\Omega \in \Omega} l(\Omega), \quad \mathbf{d} \notin \mathbf{O} \quad \forall \mathbf{d} \in \Omega, \quad (4.2)$$

where Ω is a feasible path in the candidate path set Ω , $l(\Omega)$ is the length of Ω , and \mathbf{d} represents the coordinates of the path.

If there are no hard obstacles, such as big rocks or ravines, the off-road mobility model

can provide a GO/NO-GO map based on the mean value of the terrain properties. The GO and NO-GO areas are respectively the free space and obstacle space as represented as:

$$\mathbf{F} = \{\mathbf{d} \in \mathbf{M} \mid G(\bar{\mathbf{X}}(\Lambda), \bar{\mathbf{Y}}(\mathbf{d}), \Lambda) \geq e\}, \quad (4.3)$$

$$\mathbf{O} = \{\mathbf{d} \in \mathbf{M} \mid G(\bar{\mathbf{X}}(\Lambda), \bar{\mathbf{Y}}(\mathbf{d}), \Lambda) < e\}, \quad (4.4)$$

where $\bar{\mathbf{X}}(\Lambda)$ and $\bar{\mathbf{Y}}(\mathbf{d})$ are the mean values of vehicle-related parameters and terrain-related parameters respectively, and e is the threshold of speed to determine whether the vehicle is stuck or not stuck at a certain location.

4.1.3. Off-Road AGV Mobility Reliability

4.1.3.1. Uncertainty sources in vehicle mobility analysis

As shown in Figure 4.5, the uncertainty in the vehicle mobility analysis using Eq. (4.1) stems from two major sources, namely (1) uncertainty in vehicle modeling and simulation (M&S) and (2) uncertainty in the terrain-related parameters. The uncertainty sources in M&S usually remain constant over a target map, whereas the uncertainty of terrain-related parameters varies with the slope/soil type over space.

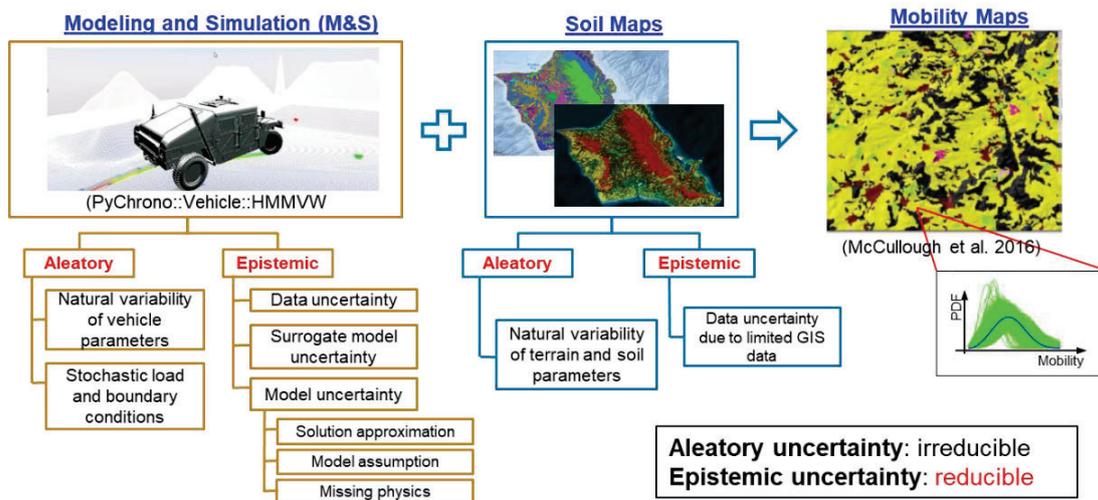


Figure 4.5 Uncertainty sources in terrain-related parameters

Furthermore, the uncertainty sources in M&S and soil maps can be classified into

aleatory and epistemic uncertainty. Aleatory uncertainty represents the natural variability such as natural variability of vehicle physical parameters across a population of vehicles, variability of soil properties at different locations, etc. Epistemic uncertainty refers to the uncertainty due to lack of knowledge which is reducible if more information is available. Some representative examples of epistemic uncertainty in off-road AGV mobility analysis include model form uncertainty of the mobility prediction model, solution approximations, and data uncertainty in soil properties due to limited data.

Due to the heterogenous uncertainty sources in the M&S and soil properties, the vehicle mobility at each location is uncertain (see Figure 4.5). The uncertainty of vehicle mobility can be quantified by propagating various uncertainty sources in both M&S and soil maps through a vehicle mobility simulation model. To propagate the uncertainty, various uncertainty sources must be modeled, which is not a trivial task since it requires a large volume of data from both vehicle systems and various soil samples. Some related data of the soil properties are available in [88]. Vehicle-related uncertainty sources, however, vary with vehicle types and manufactures. In this chapter, for the sake of illustration, vehicle-related uncertainty sources are not considered, since it is known that the variability of vehicle parameters are vehicle-dependent and are negligible compared to soil-related parameters[22]. Thus, we mainly focus on the uncertain terrain slope and soil properties as listed in Table 4.1. The developed framework, however, can be easily extended to incorporate other uncertainty sources if their statistical information is available.

As shown in Figure 4.6, the soil type varies at different locations. For example, it may be sand (e.g., soil ID= l) at a location, or grass (e.g., soil ID= n) at a different location. Accordingly, the distributions of soil properties (e.g., friction or cohesive strength) are different for a sandy area or a grassland, for example. Moreover, there are several different soil properties for each soil type resulting in several different maps of soil parameters (see Figure 4.6).

For any given spatial location \mathbf{d} (Figure 4.6), the slope/soil type is first identified and the statistical distribution of the uncertain soil/slope properties is determined. For example, the cumulative distribution function (CDF) of a soil parameter $Y_1(\mathbf{d})$ at location \mathbf{d} can be represented as

$$\Pr\{Y_1(\mathbf{d}) \leq y\} = \int_{-\infty}^y f_{Y_1}(y, \boldsymbol{\theta}(\mathbf{d})) dy, \quad (4.5)$$

in which $f_{Y_1}(y, \boldsymbol{\theta}(\mathbf{d}))$ is the probability density function (PDF) in terms of the distribution parameters $\boldsymbol{\theta}(\mathbf{d})$ which are statistical parameters determined by the slope/soil type at spatial location \mathbf{d} .

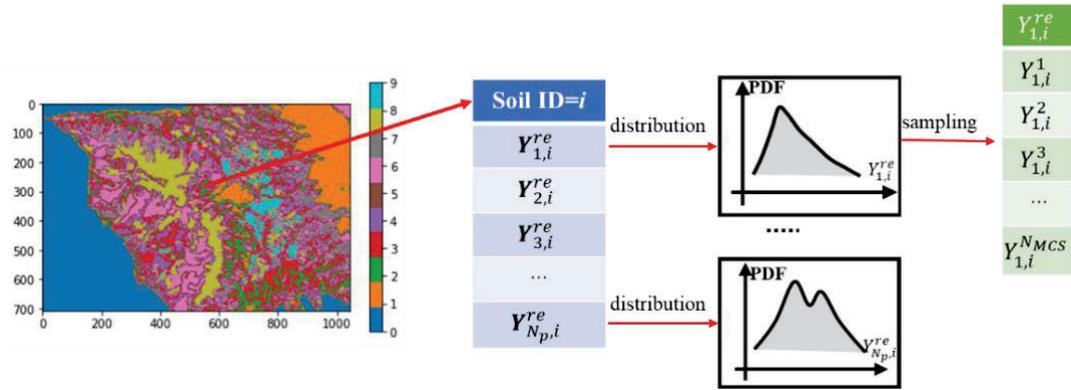


Figure 4.6 Uncertainty sources in terrain-related parameters

As mentioned previously, the distribution parameters $\boldsymbol{\theta}(\mathbf{d})$ can also be uncertain due to data uncertainty caused by limited geographic information system (GIS) data. In this chapter, it is assumed that the distribution parameters $\boldsymbol{\theta}(\mathbf{d})$ are known, as shown in the numerical example section. We concentrate on how to identify a reliable path for a given soil property statistical distributions.

4.1.3.2. Vehicle mobility reliability

To quantitatively quantify the impact of various uncertainty sources on vehicle mobility, Jiang et. al. [31] defined two types of vehicle mobility reliability, namely vehicle state mobility reliability (SMR) and vehicle mission reliability (MMR).

SMR is defined as the probability that the maximum attainable speed $V(\mathbf{d})$ of an

AGV at a certain location is greater than or equal to a predefined threshold e . It quantifies the probability that an AGV will not get stuck in a deformable soil or equivalently, the probability that a vehicle can remain mobile at a specific location. Mathematically, SMR at any location \mathbf{d} or a given vehicle design Λ is given by

$$\begin{aligned} SMR(\mathbf{d}) &= \Pr \{V(\mathbf{d}) = G(\mathbf{X}(\Lambda), \mathbf{Y}(\mathbf{d}), \Lambda) \geq e\}, \\ &= \iint_{G(\mathbf{X}(\Lambda), \mathbf{Y}(\mathbf{d}), \Lambda) \geq e} f_{\mathbf{X}}(\mathbf{x}) f_{\mathbf{Y}}(\mathbf{y}, \boldsymbol{\theta}(\mathbf{d})) d\mathbf{x} d\mathbf{y}, \end{aligned} \quad (4.6)$$

where $\Pr\{\cdot\}$ denotes probability, $f_{\mathbf{X}}(\mathbf{x})$ is the joint PDF of $\mathbf{X}(\Lambda)$, and e is a threshold of vehicle mobility.

MMR quantifies the probability of completing a mission considering the spatial dependence of various uncertainty sources on a given mission path Ω [31]. For a given Ω , MMR is defined as

$$MMR(\Omega) = \Pr \{V(\mathbf{d}) = G(\mathbf{X}(\Lambda), \mathbf{Y}(\mathbf{d}), \Lambda) \geq e, \forall \mathbf{d} \in \Omega\}, \quad (4.7)$$

in which \forall means “for all”.

SMR and MMR quantify the mobility of a given vehicle design from different perspectives. The evaluation of these two types of reliability metrics is also different. Even though MMR is able to more comprehensively reflect the capability of an AGV than SMR from a mission perspective, the high computational complexity brings challenges to both mission planning and practical applications. In this chapter, a bio-inspired approach is developed based on SMR to guarantee the reliability of AGVs on a path. The integration of the proposed method with MMR is even more complicated and will be investigated in future work.

The evaluation of the SMR in Eq. (4.6) is computationally challenging because of the high-dimensional integration. If the brute-force Monte Carlo simulation (MCS) is employed to estimate SMR, we can first generate N_{mcs} random realizations of soil

parameters according to their spatial coordinates and statistical distributions. Based on that, the uncertainty of a soil property over a target map can be obtained by assembling together the random realizations at different locations. By propagating the N_{mcs} random realizations of soil properties through the vehicle mobility model, we obtain N_{mcs} realizations of $V(\mathbf{d})$ as $V^{(i)}(d)$, $i=1,2,\dots,N_{mcs}$. Based on that, the SMR at \mathbf{d} can be approximated as

$$SMR(\mathbf{d}) \approx \frac{1}{N_{mcs}} \sum_{i=1}^{N_{mcs}} I(V_{re}^i(\mathbf{d}) \geq e), \quad \begin{cases} I = 0, & \text{if } V_i < e \\ I = 1, & \text{if } V_i \geq e \end{cases}. \quad (4.8)$$

The MCS-based method, however, is computationally prohibitive. In the next section, we first discuss how to overcome the computational challenge in vehicle mobility reliability analysis using an adaptive surrogate modeling method. Subsequently, we identify an optimal path based on the SMR analysis using a novel Physarum method.

4.2. Reliability-Based Mission Planning Using a Bio-Inspired Method

In order to incorporate post-design operation into design and guarantee the reliability of AGV during operation in early design stages, the following reliability-based design optimization model must be solved

$$\begin{aligned} & \min C(\Lambda, \Omega), \\ & s.t. \\ & R(\Lambda, \Omega) \geq [R], \end{aligned} \quad (4.9)$$

where $C(\Lambda, \Omega)$ is a cost function of vehicle design Λ and path Ω controlling the post-design operation, $R(\Lambda, \Omega)$ is the vehicle mobility reliability, and $[R]$ is a mobility reliability requirement. The value of $[R]$ is determined by the decision maker according to the risk level he/she can accept. It is usually related to the economic loss of a vehicle failure event. For example, a high reliability is usually required if the consequence of

failure is catastrophic.

In this chapter, we fix the vehicle design and employ SMR as the mobility reliability constraint as discussed above. We will also include Λ in the design variables in future research. For given Λ , Eq. (4.9) reduces to the following optimization model

$$\begin{aligned} & \min_{\Omega} C(\Lambda, \Omega), \\ & s.t. \\ & \Pr\{V(\mathbf{d}) = G(\mathbf{X}(\Lambda), \mathbf{Y}(\mathbf{d}), \Lambda) \geq e\} \geq [R], \forall \mathbf{d} \in \Omega. \end{aligned} \quad (4.10)$$

Solving the above optimization model requires tackling two major challenges: (1) how to efficiently evaluate the probabilistic SMR constraint; and (2) how to solve the optimization model to identify the optimal path. In this chapter, the first challenge is solved using an adaptive surrogate modeling method (Sec. 4.2.1) and the second challenge is addressed using a bio-inspired approach (Sec. 4.2.2).

4.2.1. Adaptive Surrogate Modeling for Model-Based Mobility Reliability Analysis

Evaluating the mobility reliability requires a large number of high-fidelity computer simulations, which is computationally very expensive. In order to reduce the required computational cost in reliability analysis, various adaptive surrogate modeling methods have been developed in recent years to build an efficient yet accurate surrogate model of the original simulation model [89, 90]. The basic idea is to adaptively add new training data for surrogate modeling to improve the prediction accuracy in regions which are critical for reliability analysis. It has shown promising accuracy and efficiency for reliability analysis in various applications [91, 92]. For model-based mobility reliability analysis, our goal is to build a surrogate model of $V(\mathbf{d}) = G(\mathbf{X}(\Lambda), \mathbf{Y}(\mathbf{d}), \Lambda)$ to replace the original model.

We start with a Latin Hypercube sampling (LHS) of soil properties based on the low and upper bounds provided in [88] (Table 4.2). For each LHS sample, we use PyChrono to

create a simulation model to predict the maximum attainable vehicle speed. As shown in Figure 4.7, a terrain geometry is first created based on a terrain slope. The terrain is created long enough such that the vehicle can reach the maximum attainable speed. The deformable soil parameters are the values of a specific LHS sample. The vehicle is then commanded to accelerate from rest until the maximum steady vehicle speed is reached. Based on that, we obtained the maximum attainable speed for specific terrain slope and values of soil properties.

Table 4.2 Input parameters of the SCMDDeformableTerrain module

Parameters		Lower Bound	Upper Bound
Slope-related	Slope	0	45°
Soil-related	Bekker_Kphi	1	10 ⁹
	Bekker_Kc	1	10 ¹⁰
	Bekker_n	0.8	2
	Mohr_cohesion	22.5°	37.5°
	Mohr_friction	0.001	0.05
	Janosi_shear	0	1000

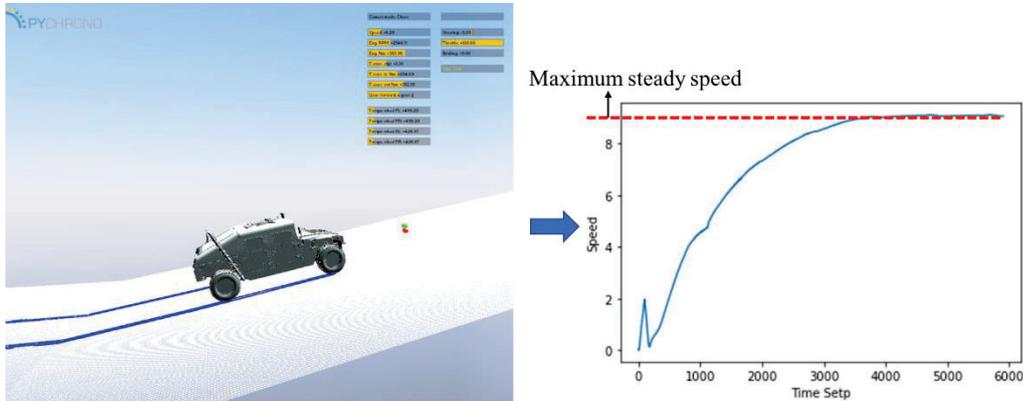


Figure 4.7 Vehicle mobility simulation model to predict the maximum attainable speed

Based on a number of initial samples, we can build a surrogate model $V(\mathbf{d}) = \hat{G}(\mathbf{X}, \mathbf{Y}(\mathbf{d}))$ using a Gaussian process (GP) surrogate modeling method to replace the original PyChrono simulation model. Note that since the vehicle design Λ is fixed in this chapter, Λ is omitted in the surrogate modeling for the sake of explanation. For any new sample $\mathbf{X} = \mathbf{x}$ and $\mathbf{Y}(\mathbf{d}) = \mathbf{y}$, we predict the maximum attainable speed using the GP

model as

$$V = \hat{G}(\mathbf{x}, \mathbf{y}) \sim N(\mu(\mathbf{z}), \sigma^2(\mathbf{z})), \quad (4.11)$$

where $\mathbf{z} \triangleq [\mathbf{x}, \mathbf{y}]$, $N(\mu(\mathbf{z}), \sigma^2(\mathbf{z}))$ stands for normal distribution with a mean value of $\mu(\mathbf{z})$ and a standard deviation of $\sigma(\mathbf{z})$.

Since the initial surrogate model $V(\mathbf{d}) = \hat{G}(\mathbf{X}, \mathbf{Y}(\mathbf{d}))$ may not accurately represent the original simulation model, an adaptive Kriging Monte Carlo simulation (AK-MCS) method is then employed to improve the surrogate prediction accuracy in mobility reliability analysis [56, 93, 94]. Figure 4.8 shows the flowchart of the adaptive surrogate modeling method. A large number of MCS samples are first generated for the variables given in Table 4.2. If the generated samples are $\mathbf{z}^{(i)}, i = 1, 2, \dots, N_{mcs}$, we compute the U value of every sample using the following U learning function

$$U(\mathbf{z}^{(i)}) = \left| \frac{\hat{L}(\mathbf{z}^{(i)})}{\sigma(\mathbf{z}^{(i)})} \right| = \left| \frac{\mu(\mathbf{z}^{(i)}) - e}{\sigma(\mathbf{z}^{(i)})} \right|, \quad \forall i = 1, 2, \dots, N_{mcs}, \quad (4.12)$$

where $\mu(\mathbf{z}^{(i)})$ and $\sigma(\mathbf{z}^{(i)})$ are respectively the mean and standard deviation of the GP prediction (see Eq. (4.11)). The U function aims to concentrate resources on the sample in the proximity of the boundary corresponding to the threshold while with high uncertainty. The lower the U value is, the closer the sample is to the threshold e with lower prediction accuracy, which means this sample should be tested in simulation and used as a training point for the surrogate model to improve the prediction accuracy for mobility reliability.

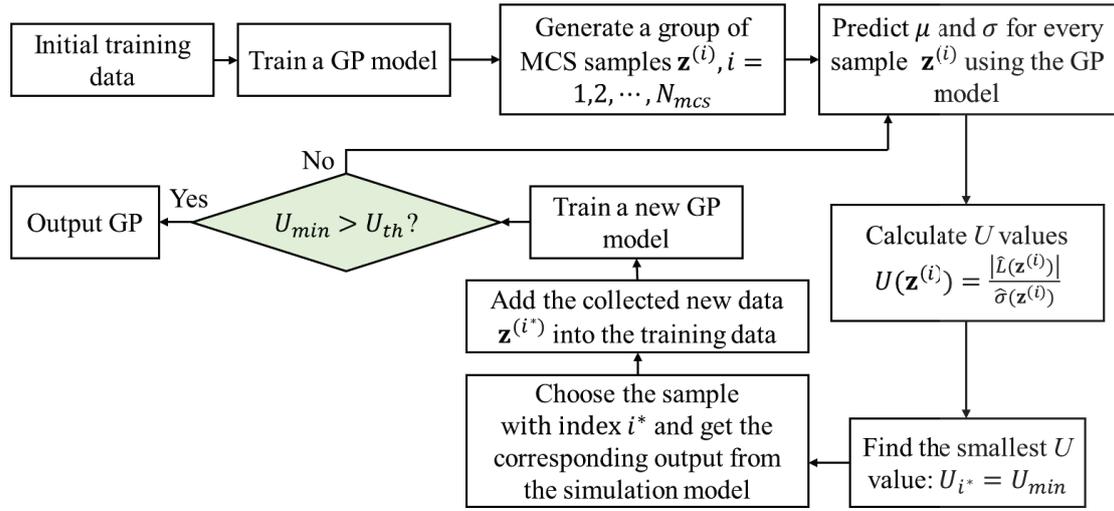


Figure 4.8 Flowchart of adaptive surrogate modeling-based reliability analysis

Based on the U values from Eq. (4.12), a new training sample $\mathbf{z}^{(i)}$ is identified as

$$i^* = \arg \min_i \{U(\mathbf{z}^{(i)})\}. \quad (4.13)$$

The identified new training sample is then added to the training data pool and the GP surrogate model is retrained using the updated training dataset. The process continues until the smallest U value is greater than a certain threshold.

After we have an accurate surrogate model of the off-road AGV mobility, the surrogate model is used to predict the SMR according to the slope type and soil type at each location using Eq. (4.8). Since the SMR constraint of Eq. (4.10) requires that $SMR(\mathbf{d})$ at each location of a path Ω should be greater than $[R]$, we can rewrite the constraint given in Eq. (4.10) as follows

$$\begin{aligned} & \Pr \{V(\mathbf{d}) = G(\mathbf{X}(\Lambda), \mathbf{Y}(\mathbf{d}), \Lambda) \geq e\} \geq [R], \forall \mathbf{d} \in \Omega. \\ & \triangleq \Omega \in \Gamma, \text{ where } \Gamma = \{\forall \mathbf{d} \mid \Pr \{V(\mathbf{d}) = G(\mathbf{X}(\Lambda), \mathbf{Y}(\mathbf{d}), \Lambda) \geq e\} \geq [R]\}, \end{aligned} \quad (4.14)$$

in which Γ represents a set of coordinates \mathbf{d} where $SMR(\mathbf{d}) \geq [R]$ is satisfied.

The above equation indicates that the optimization model of Eq. (4.10) can be solved in two sequential steps. In the *first* step, the coordinates \mathbf{d} where the SMR constraint is

satisfied are identified. To be consistent with the deterministic path planning, all the coordinates over a map \mathbf{M} are classified into GO/NO-GO according to Eq. (4.14) as

$$\mathbf{d}_{GO} = \{\mathbf{d} \in \mathbf{M} \mid SMR(\mathbf{d}) \geq [R]\}, \quad (4.15)$$

$$\mathbf{d}_{NO-GO} = \{\mathbf{d} \in \mathbf{M} \mid SMR(\mathbf{d}) < [R]\}. \quad (4.16)$$

Based on the classification of the GO/NO-GO using the SMR constraint, Eq. (4.10) is written as a new model in the *second* step as

$$\begin{aligned} & \min_{\Omega} C(\Lambda, \Omega), \\ & s.t. \quad \Omega \in \mathbf{d}_{GO}. \end{aligned} \quad (4.17)$$

In this chapter, a bio-inspired approach is introduced to solve Eq. (4.17). It allows us to identify a path that satisfies the SMR constraint using model-based approach. Figure 4.9 presents the flowchart of reliability-based mission planning. It starts with generating a mobility reliability map by identifying the soil/slope ID of each coordinate in the map. For a certain ID, random realizations of the slope and soil parameters are generated according to their statistical distributions. Using the trained mobility surrogate model, random realizations of vehicle mobility are obtained for each coordinate of the target map. Based on the random realizations, a SMR map is generated, and subsequently a GO/NO-GO map is obtained using Eqs. (4.15) and (4.16).

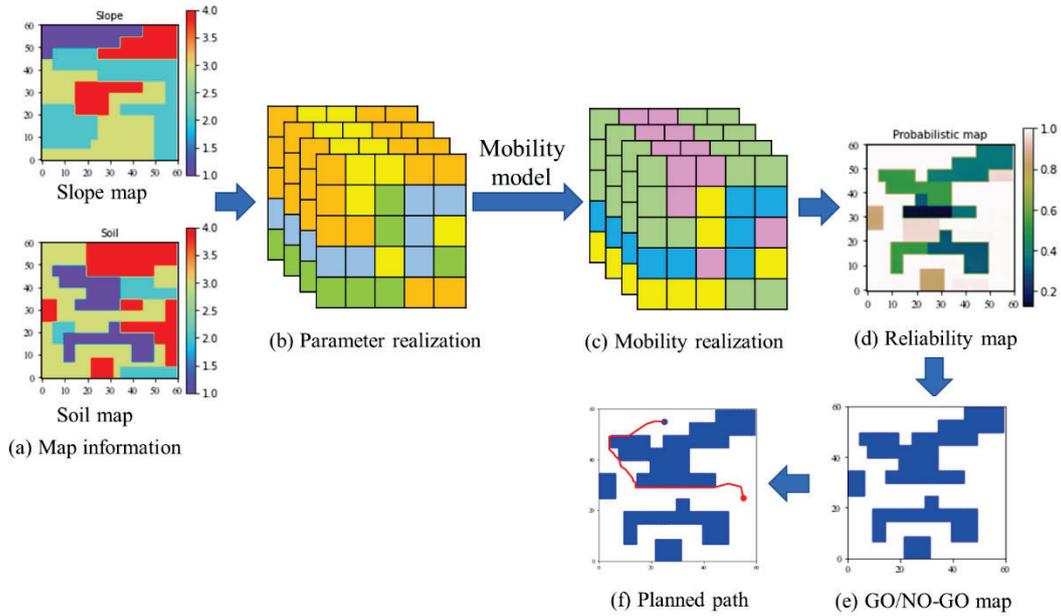


Figure 4.9 Flowchart of mobility reliability-based mission planning

Treating the NO-GO coordinates as obstacles, the next question is how to efficiently solve Eq. (4.17) to identify an optimal path for an off-road AGV to go from a starting point to a target point (see Figure 4.9(e)-(f)). It is worth mentioning that the “obstacles” considered here are different from the conventional “hard” obstacles in path planning of robotics. In common seen robotics path planning problems, the “hard” obstacles are detected using cameras or lidars if the robots are close to the obstacles. Because the obstacles are detectable, paths of robots can be re-planned. In such a case, a globally optimal and reliable path may not be very important. For the off-road AGV, however, the obstacles are not real/hard. Instead, they represent the probability that a vehicle may get stuck at the obstacle location. They cannot be directly detected by a camera or a lidar. Taking a mud pond as an example, a vehicle can sometimes pass the mud pond and sometimes get stuck in it. Even if a camera can detect the existence of a mud pond, it cannot tell if a vehicle can pass it or get stuck in it. It would be too late if we wait for the vehicle to get stuck in the mud pond and then perform the re-planning. This is the reason a globally optimal and reliable path is particularly important for an off-road AGV.

For reliable path planning of off-road AGVs, approaches have been developed using

RRT* [31]. Another widely used approach is the Dijkstra-based method [95]. Even though these approaches have shown promising performance in identifying an optimal path, we noticed that the required computational time is still high especially for multi-vehicle path planning problems where the algorithms must be run multiple times. In this chapter, we suggest an alternative path planning algorithm to solve the optimization model of Eq. (4.17) for off-road AGVs using a bio-inspired Physarum method [96]. To the best of our knowledge, this is the first time a bio-inspired approach is employed for path planning of off-road AGVs. As shown in the numerical example section, the Physarum approach shows better performance than the Dijkstra-based method. Sec. 4.2.2 provides more details of the Physarum-based mission planning method.

4.2.2. *Physarum solver-based mission planning*

In order to integrate the Physarum method with the GO/NO-GO map generated from the mobility reliability analysis of Sec. 4.2.1 (i.e. solve Eq. (4.17)) and to further reduce the computational effort in mission planning, we first need to convert the GO/NO-GO map corresponding to any mobility reliability requirement into a network described by nodes and edges. To accomplish such a conversion, the following navigation mesh method is employed.

4.2.2.1. *Navigation mesh*

There are many ways to convert the GO/NO-GO map into a network. A straightforward way of doing that is to connect every non-obstacle coordinate with all its non-obstacle neighborhood coordinates. This method is usually referred to as *uniform mesh* (Figure 4.10(b)). The uniform mesh allows to cover all locations in a map and thus achieves high accuracy in path planning. However, the number of nodes and edges increases exponentially with the size of the target map, which significantly increases the required computational cost of path planning and hinders the practical application of the uniform

mesh in off-road AGV mission planning. This disadvantage of uniform mesh exists for both the commonly used Dijkstra algorithm and the physarum algorithm presented in Sec. 4.2.2.2.

As an alternative, the navigation mesh is widely used for path finding in game design [97, 98]. It partitions a map into polygons, and a network is built by connecting the center or the vertex of the polygons. Figure 4.10 compares the uniform and navigation meshes for a GO/NO-GO map given in Figure 4.10(a).

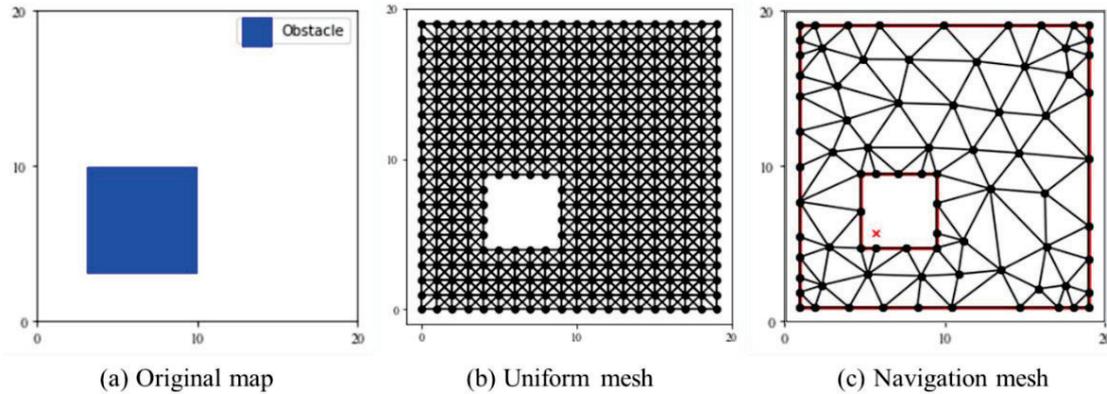


Figure 4.10 Comparison of uniform mesh and navigation mesh

As shown in Figure 4.10, the navigation mesh drastically reduces the number of nodes and edges of the resulting network. Therefore, it can significantly reduce the computational time in mission planning. The disadvantage of navigation mesh, however, is that it sacrifices accuracy. The identified paths may not be the truly globally optimal, since the path depends on the defined edges and nodes. However, this disadvantage can be overcome by tuning the relevant parameters in generating the mesh. In this chapter, a Python package named Triangle [99] is employed to convert a GO/NO-GO map into a network for mission planning. It is built based on Jonathan Richard Shewchuk’s two-dimensional quality mesh generator and Delaunay triangulator library [100]. The employment of a navigation mesh in conjunction with the adaptive surrogate modeling method drastically reduces the required computational effort of mission planning from different perspectives.

Next, we discuss how to identify optimal paths for a single vehicle and multi-vehicle

scenarios by using a bio-inspired physarum solver and the navigation mesh.

4.2.2.2. Physarum solver

Physarum polycephalum is a bio-inspired path planner originally proposed by Nakagaki et al. [101], who demonstrated that Physarum could disassemble and reassemble the tube structure and tube thickness over time in response to the change of external conditions (availability of food sources) to identify the shortest path connecting the food sources. As shown in Figure 4.11, this organism can form a tube forming the shortest path connecting two food sources located at the source point (s) and the sink point (t) in the maze. The mathematical model of the path generator in the network-based map was proposed by Tero et al. [96].

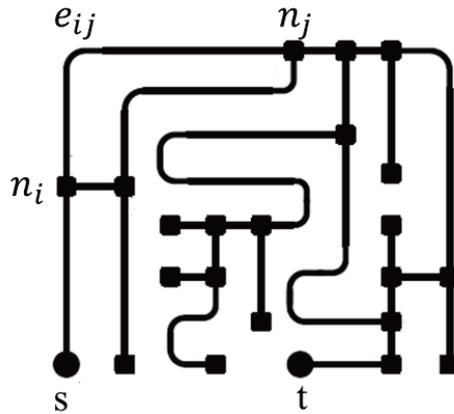


Figure 4.11 Graphical representation of the maze using a network

Suppose we represent the maze with a graph G , in which two nodes s and t are designated as the starting and ending nodes. The other nodes are labelled as n_1, n_2, \dots, n_i etc. The edge between nodes n_i and n_j is represented by e_{ij} . The set of all edges is denoted by E . For a network-based map as shown in Figure 4.11, the flux Q_{ij} through edge e_{ij} is formulated following the Poiseuille flow assumption as

$$Q_{ij} = \frac{D_{ij}}{L_{ij}} (p_i - p_j), \quad (4.18)$$

where p_i is the pressure at node n_i , L_{ij} is the length of edge e_{ij} , and D_{ij} is its conductivity.

Since the inflow and outflow should be balanced, Kirchhoff's law yields:

$$\begin{cases} \sum_i Q_{ij} + I_0 = 0, & j = s, \\ \sum_i Q_{ij} - I_0 = 0, & j = t, \\ \sum_{\forall (i,j) \in E} Q_{ij} = 0, & \text{otherwise} \end{cases}, \quad (4.19)$$

where I_0 is the flux from the source node.

To simplify the problem, we let $I_0 = 1$ [102]. By combining Eqs. (4.18) and (4.19), we have [96]:

$$\sum_i \frac{D_{ij}}{L_{ij}} (p_i - p_j) = \begin{cases} -1, & j = s, \\ 1, & j = t, \\ 0, & \text{otherwise} \end{cases}, \quad (4.20)$$

where all p_i can be determined by solving the linear system of equations (4.20).

Afterwards, the flux through each edge e_{ij} is derived using Eq. (4.18) accordingly.

In order to formulate the conductivity change for each edge, we have

$$\frac{dD_{ij}}{dt} = f(|Q_{ij}|) - rD_{ij}, \quad (4.21)$$

where $f(x)$ is an increasing function satisfying $f(0) = 0$.

To save computational effort, we let $f(x) = x$ and $r = 1$, so that

$$\frac{D_{ij}^{m+1} - D_{ij}^m}{\delta t} = |Q_{ij}^m| - D_{ij}^{m+1}, \quad (4.22)$$

where m represents the m^{th} iteration and $\delta t = 1$. Based on the above equations, an optimal

path can be identified for any given map [96].

After generating the navigation mesh, Figure 4.12 shows the flowchart of the Physarum-based path planning approach. In the figure, th is the threshold to determine whether the conductivity matrix \mathbf{D} has converged.

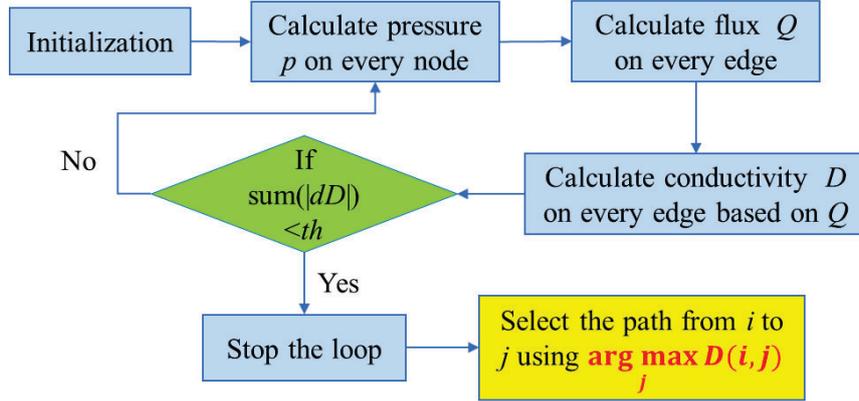


Figure 4.12 Flowchart of the Physarum solver algorithm

In order to calculate the pressure at each node, we must solve Eq. (4.20). The matrix expression of Eq. (4.20) is

$$\mathbf{A} \cdot \mathbf{p} = \mathbf{C}, \quad (4.23)$$

where $\mathbf{A} \in \mathbb{R}_{n \times n}$ is given by

$$\mathbf{A} = \frac{\mathbf{D}}{\mathbf{L}} = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & A_{22} & \cdots & A_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n1} & A_{n2} & \cdots & A_{nn} \end{bmatrix} = \begin{bmatrix} \frac{D_{11}}{L_{11}} & \cdots & \frac{D_{1n}}{L_{1n}} \\ \vdots & \ddots & \vdots \\ \frac{D_{n1}}{L_{n1}} & \cdots & \frac{D_{nn}}{L_{nn}} \end{bmatrix}, \quad A_{ii} = -\sum_j A_{ij}, \quad (4.24)$$

$\mathbf{D} \in \mathbb{R}_{n \times n}$ is the conductivity matrix, $\mathbf{L} \in \mathbb{R}_{n \times n}$ is the length matrix of the network, and \mathbf{D} ,

\mathbf{L} , and \mathbf{p} are given by

$$\mathbf{D} = \begin{bmatrix} D_{11} & D_{12} & \cdots & D_{1n} \\ D_{21} & D_{22} & \cdots & D_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ D_{n1} & D_{n2} & \cdots & D_{nn} \end{bmatrix}. \quad (4.25)$$

$$\mathbf{L} = \begin{bmatrix} L_{11} & L_{12} & \cdots & L_{1n} \\ L_{21} & L_{22} & \cdots & L_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ L_{n1} & L_{n2} & \cdots & L_{nn} \end{bmatrix}, L_{ij} = +\infty \text{ when no connection between } i, j. \quad (4.26)$$

$$\mathbf{p} = [p_1, p_2, \dots, p_n]^T, \quad (4.27)$$

and

$$\mathbf{C} = [C_1, C_2, \dots, C_n]^T, C_i = -1 \text{ when } i = s, C_i = 1 \text{ when } i = t. \quad (4.28)$$

Once the conductivity matrix \mathbf{D} reaches a steady state, an optimal path can be identified by selecting the edges with the largest conductivity that form a path connecting the starting point with the target point.

Algorithm 1 shows the general procedure of Physarum-based method, including the updating of the conductivity matrix and the identification of the optimal path after the conductivity matrix converges.

Algorithm 4.1 Physarum algorithm for single vehicle path planning

-
- 1 Data: \mathbf{L} is the length matrix given in Eq. (4.26), s is the start point, t is the target point, and n is the size of the network
 - 2 Initialization: $D_{ij} = 0.5$, $Q_{ij} = 0$, $p_i = 0$, $\forall i, j = 1, 2, \dots, n$.
 - 3 While $\text{sum}(|d\mathbf{D}|) > th$ do
 - 4 Calculate pressure using Eq. (4.20)

$$\sum_i \frac{D_{ij}}{L_{ij}} (p_i - p_j) = \begin{cases} -1, & j = s \\ 1, & j = t \\ 0 & \text{otherwise} \end{cases}$$
 - 5 $Q_{ij} = \frac{D_{ij}}{L_{ij}} (p_i - p_j)$, // Using Eq. (4.18)
 - 6 $D_{ij} = 0.5(|Q_{ij}| + D_{ij})$ // Using Eq. (4.21)

```

7       $d\mathbf{D} = |\mathbf{D}^{n+1} - \mathbf{D}^n|$  // Calculate the change of the conductivity
8       $path = [], ind = s, path.append(ind)$ 
9      While  $ind \neq t$  do
10      $ind_{next} = \arg \max_i (\mathbf{D}_{ind, \cdot}),$  where  $\mathbf{D}_{ind, \cdot} = [D_{ind,1}, D_{ind,2}, \dots, D_{ind,n}]$ 
11      $path.append(ind_{next})$ 
12      $D_{ind_{next}, path[-1]} = 0,$  where  $path[-1]$  represents the last element
13     of  $path$ , and  $\mathbf{D}_{\cdot, ind} = 0,$  where  $\mathbf{D}_{\cdot, ind} = [D_{1,ind}, D_{2,ind}, \dots, D_{n,ind}]$ 
13      $ind = ind_{next}$ 

```

This algorithm can also handle dynamic path planning problem. If there are new edges between nodes i and j , or new obstacles that cut the edges between node i and j , we simply need to change L_{ij} from infinity to l or from l to infinity. Using the previous conductivity matrix \mathbf{D} , the knowledge of the remaining part of the map that has not changed is carried into the new calculation saving computational time.

Moreover, the Physarum-based method can naturally accommodate the task of multi-vehicle mission planning including automatic vehicle assignments and identification of optimal path for each vehicle. The procedure is similar to single vehicle path planning. The difference is that there are more than one starting point and target point. As a result, \mathbf{C} in Eq. (4.23) becomes

$$\mathbf{C} = [C_1, C_2, \dots, C_n]^T, C_i = -1 \text{ when } i \in \mathbf{s}, C_i = 1 \text{ when } i \in \mathbf{t}, \quad (4.29)$$

where \mathbf{s} and \mathbf{t} are the group of starting points and target points.

After the conductivity matrix converges, we can identify an optimal path by starting with any starting point and using the path identification part in Algorithm 1. For the remaining starting points, we must update the conductivity matrix based on the edges

selected by the last path as

$$D_{ij}^* = \begin{cases} D_{ij} - 1, & \text{if } D_{ij} > 1 \\ 0.8D_{ij}, & \text{otherwise} \end{cases}, \forall ij \text{ satisfies } e_{ij} \in \Omega_{last}, \quad (4.30)$$

where D_{ij}^* represents the updated conductivity value of edge e_{ij} , and Ω_{last} is the last optimal path found by Algorithm 1. Noted that it includes multiple edges.

Finally, we repeat the procedure in Algorithm 1 using the updated conductivity matrix to identify another optimal path, until every starting point is assigned to a target point.

4.2.3. Implementation Procedure

Figure 4.13 summarizes the overall implementation procedure of the proposed reliability-based mission planning framework. It includes off-road vehicle mobility modeling, reliability-based map generation, network-based meshing, and Physarum-based path planning.

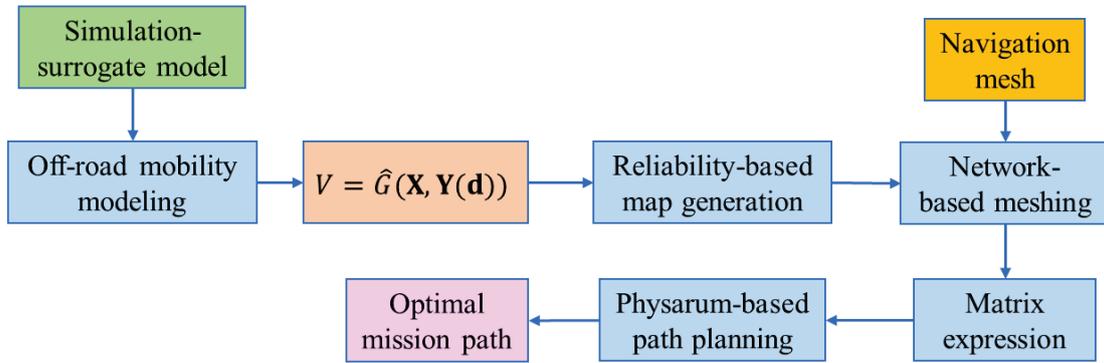


Figure 4.13 Flowchart of the proposed approach

In the Sec. 4.3, we use a case study to demonstrate the efficacy of the proposed method for path planning under uncertainty.

4.3. Numerical Example

In this numerical example, we first construct a surrogate model of the off-road mobility model using the approach of Sec. 4.2. Then, we convert a soil and slope map of

interest into a mobility reliability map and the GO/NO-GO map. After that, two case studies are used to demonstrate the effectiveness of the proposed Physarum solver-based approach in conjunction with a navigation mesh for both a single vehicle and multi-vehicle mission planning in off-road environments.

4.3.1. Maps of Interest

Figure 4.14 presents the slope map and a soil map of interest for mission planning. These maps are created manually to demonstrate the proposed framework. They are discretized using 60 points in both coordinate d_1 and d_2 . Also, there are 6 different slope intervals and soil types represented by different slope and soil IDs. Each slope ID represents a slope parameter and each soil ID represents 6 soil parameters in the PyChrono simulation model, including Bekker_Kphi (k_ϕ), Bekker_Kc (k_c), Bekker_n (B_n), Mohr_cohesion (c), Mohr_friction (ϕ), and Janosi_shear (J). In this chapter, all parameters are assumed to follow the Gaussian distribution for the sake of illustration. The proposed framework is applicable to any type of statistical distributions. Table 4.3 shows the statistical information of the parameters corresponding to different slope/soil IDs.

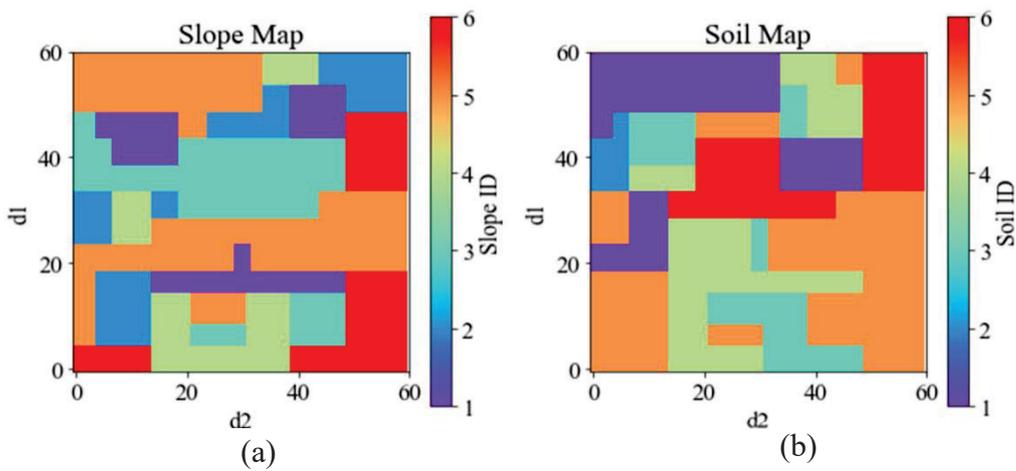


Figure 4.14 (a) Slope map and (b) Soil map of interest

Table 4.3 Statistical information of different slope/soil parameters

Slope/Soil ID	Slope		Soil											
	Slope ($^{\circ}$)		k_{ϕ}		k_c		B_n		c (Pa)		Φ ($^{\circ}$)		J (m)	
	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
1	39	1	1×10^9	1000	5×10^8	500	2.8	0.01	950	1	37.5	0.5	0.048	0.001
2	36.5	1.5	5×10^8	1500	1×10^8	750	2.6	0.05	800	2.5	35	1	0.04	0.002
3	34.5	3	1×10^8	2000	5×10^7	2500	2.2	0.1	650	5	32.5	1.5	0.036	0.002
4	32	2.5	5×10^7	500	1×10^7	1500	2.2	0.02	500	10	30	1	0.032	0.003
5	26.5	2	1×10^7	1000	5×10^6	1000	2.0	0.1	450	15	27.5	2	0.029	0.002
6	17.5	1.5	5×10^6	2000	1×10^6	2000	1.8	0.01	300	5	25	0.5	0.024	0.001

* μ is the mean value, and σ is the standard deviation

4.3.2. Generation of Mobility Reliability Map

In order to predict the vehicle mobility under different off-road environment considering various uncertainty sources, we first construct a surrogate model for the mobility prediction using the adaptive surrogate modeling method presented in Section 4.2.1. In the PyChrono simulation, the HMMWV vehicle system is used including the vehicle body, powertrain and tires as shown in Figure 4.15. The vehicle uses full double wishbone suspensions and a Pitman arm steering mechanism [103].

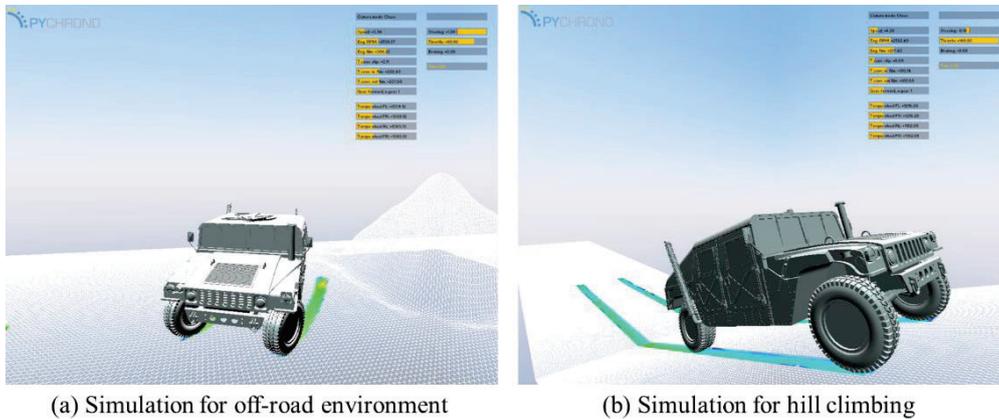


Figure 4.15 HAMMV vehicle used in this numerical example

In the adaptive surrogate modeling, we first generated 100 training points as the initial training points. A radial basis function kernel plus a white noise kernel is used in the Gaussian process modeling. The surrogate model is then adaptively refined in critical regions using the learning function given in Eq. (4.9) with parameter $e = 2m/s$. Figure

4.16(a) presents the convergence history of the U values in the adaptive training. As shown, U_{\min} reached the threshold $U_{th} = 2$ after 60 iterations. To verify the effectiveness of the adaptive surrogate modeling method, we compare its prediction accuracy of mobility reliability for a certain soil type with its counterparts obtained from: (1) a global surrogate modeling with 1300 training points and (2) surrogate modeling by adaptively minimizing the variance. Figure 4.16(b) shows that the adaptive surrogate modeling using the learning function given in Eq. (4.9) is more accurate and converges faster than the commonly used variance minimization approach. This demonstrates the efficacy of adaptive surrogate modeling in efficiently constructing a mobility surrogate model for vehicle mobility reliability analysis.

After the surrogate model is properly trained, it was used to predict vehicle mobility for different slopes and soil parameters. For the maps in Figure 4.14, $N_{MCS} = 10000$ Monte Carlo samples are generated for each slope/soil parameters according to their statistical distributions. Based on the samples and the mobility surrogate model, a probabilistic mobility map and a mobility reliability map was obtained. Figure 4.17(a) shows the mobility map obtained using the mean values of the soil properties. Figure 4.17(b) shows the state mobility reliability (SMR) map obtained using Eq. (4.7).

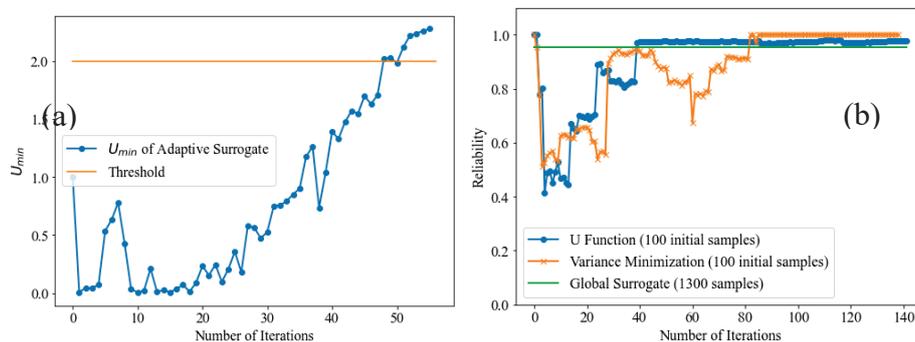


Figure 4.16 (a) Convergence history of U_{\min} in adaptive surrogate modeling; (b)

Convergence history comparison of adaptive surrogate modeling and the commonly used variance minimization-based surrogate modeling method

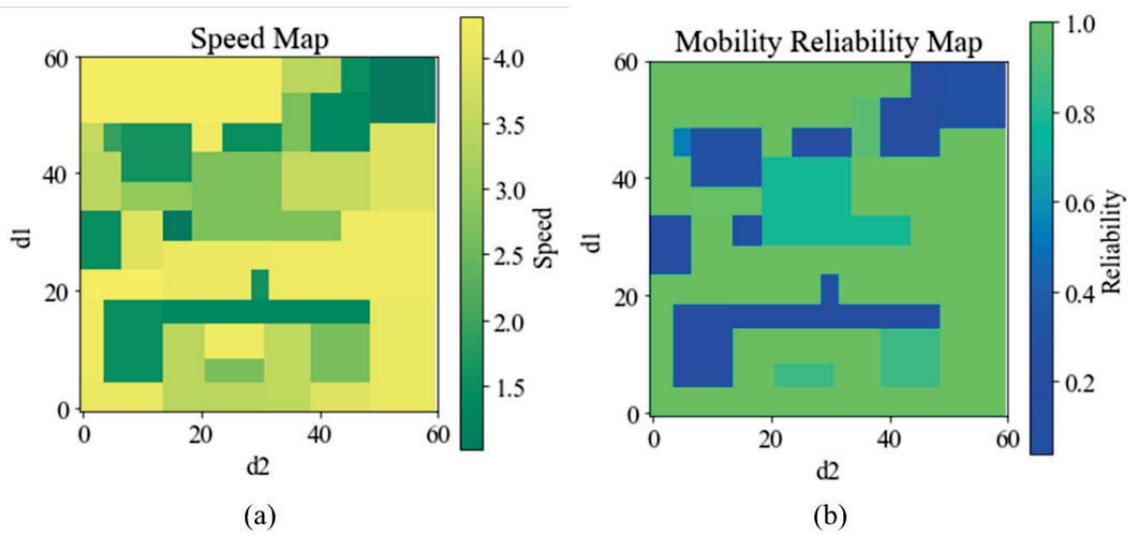


Figure 4.17 (a) Mean speed map and (b) SMR map

With the SMR map, the GO/NO-GO map can be easily generated using Eqs. (4.11) and (4.12). Figure 4.18 shows three different maps for different value of R_t .

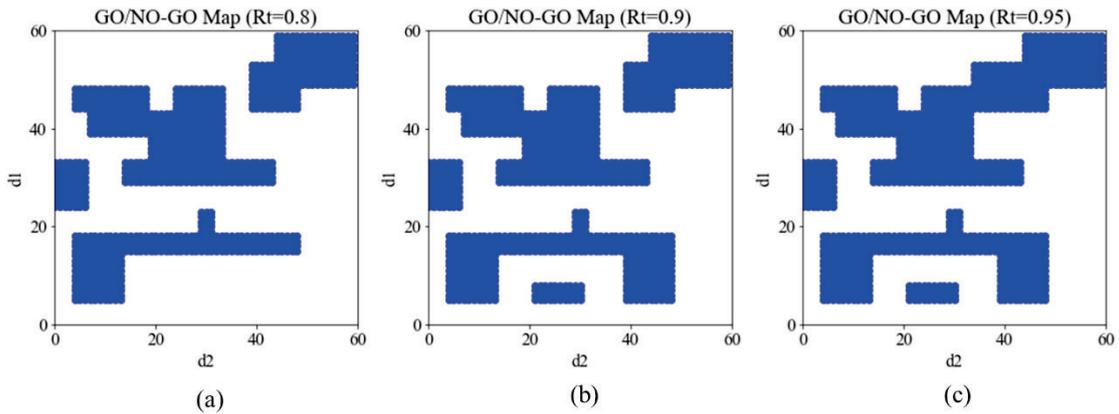


Figure 4.18 GO/NO-GO map (a) $R_t = 0.8$ (b) $R_t = 0.9$ and (c) $R_t = 0.95$

As Figure 4.18 shows, changing the reliability threshold could lead to different GO/NO-GO maps. The number of obstacles increases with reliability level. An increased reliability level will lead to an increased length of the shortest path. This implies that the vehicle must sacrifice travel time to ensure mobility reliability. The decision maker needs to determine an optimal reliability requirement by considering the consequence of failure to achieve a tradeoff between travel time and mobility reliability. By setting $R_t = 0.9$, we

then converted the corresponding GO/NO-GO map into the navigation mesh (Figure 4.19). Now the map is ready to be used for mission planning subject to the constraint that SMR should be greater than 0.9. Thus, the paths identified based on the map satisfy the mobility reliability requirement of at least 0.9. Next, we investigate two case studies, namely a single vehicle and multi-vehicles, using the obtained navigation mesh.

4.3.3. Case Study 1: Path Planning for a Single Vehicle

Figure 4.20 (a) shows the starting point (S) and target point (T) for this case study. The Physarum solver-based approach must find the shortest path between S and T. As Figure 4.20 (b) shows, while the vehicle follows the shortest path and reaches the middle of the path (M), a new obstacle appears in the map. The vehicle must dynamically update the path based on the new map starting from the current point M. Also, it needs to update the shortest path between the initial starting point S and target point T for future vehicles.

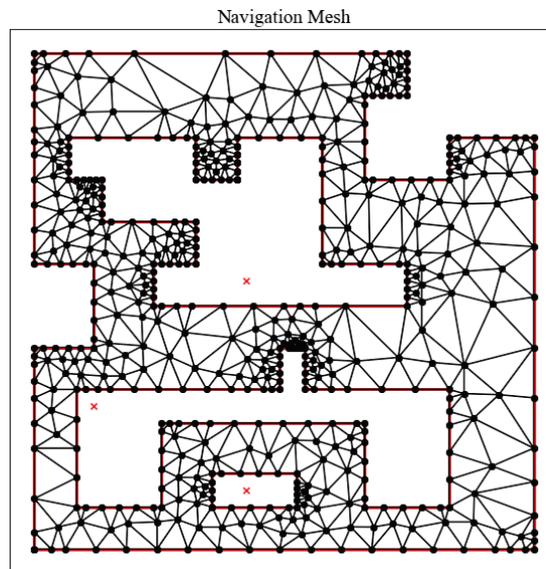


Figure 4.19 Navigation mesh of the map

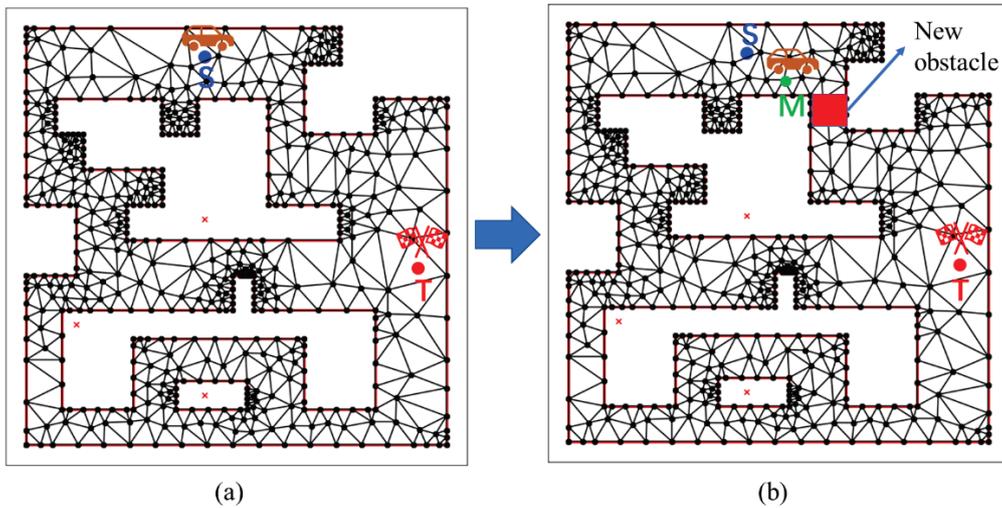


Figure 4.20 (a) Original map and (b) Dynamic obstacle in the map

Figure 4.21 presents the identified optimal paths that satisfy the reliability requirement. The paths include (a) a shortest path connecting the starting and target points, (b) an optimal path for the situation of dynamic path planning, and (c) an optimal path if new obstacle appears on the map for future vehicles.

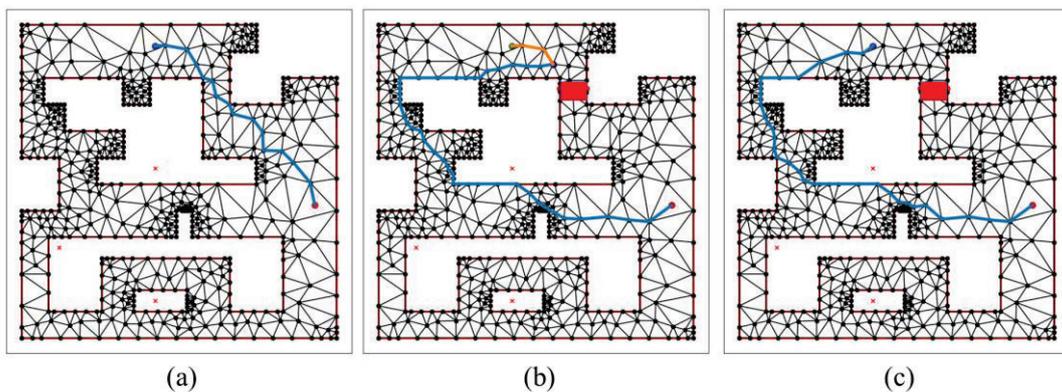


Figure 4.21 (a) Shortest path, (b) Dynamic path planning from the middle, and (c) Dynamic path planning from the starting point

Table 4.4 compares the proposed Physarum algorithm and the widely used Dijkstra approach for this first case study which identifies a path connecting the starting and target points. The results show that even though both approaches are able to identify the optimal paths for different scenarios, the Dijkstra-based approach is faster than the Physarum algorithm for a single vehicle path planning under uncertainty.

Table 4.4 Comparison between different approaches for single vehicle path planning

Method	Situation	Length of the path	Calculation time (s)
Physarum	Original	48.19	0.66
	Dynamic-middle	98.21	0.94
	Dynamic-all	91.48	0.31
Dijkstra	Original	48.19	0.14
	Dynamic-middle	98.21	0.12
	Dynamic-all	91.48	0.12

4.3.4. Case Study 2: Mission Assignment and Planning for Multiple Vehicles

As shown in Figure 4.22, in this case, there are three starting points and three target points. The decision maker must find the best combination between starting and target points in order to minimize the overall length. A dynamic obstacle also appears in the map as shown in Figure 4.22(b). The approach must re-plan the path from the starting point if a new obstacle appears and still minimize the overall length.

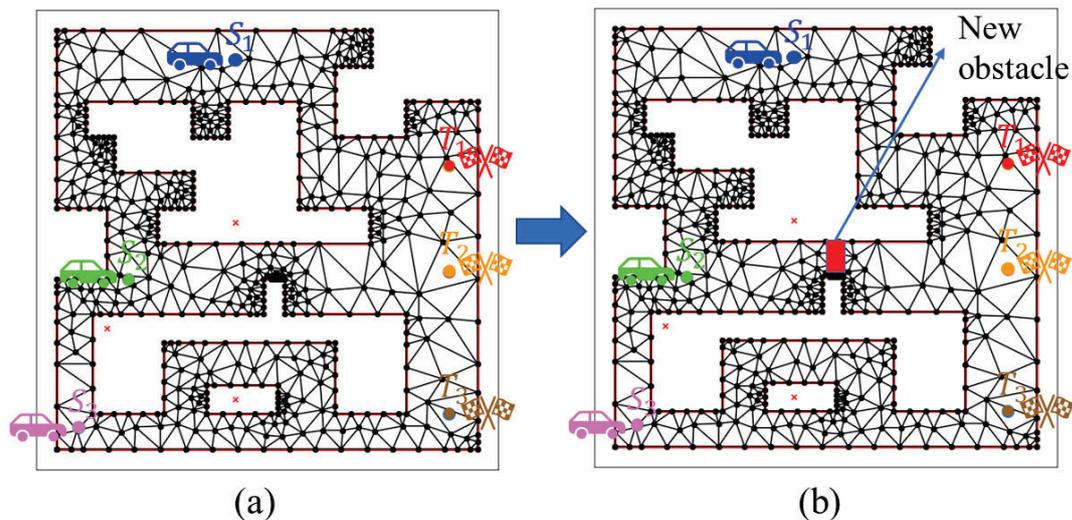


Figure 4.22 (a) Original map and (b) Dynamic obstacle of the map

The Physarum algorithm only needs to generate three different paths for each starting point, while the Dijkstra algorithm must generate all nine possible pairs. Figure 4.23 presents the results of the Physarum algorithm for the original path planning and the dynamic path planning. Table 4.5 compares the Physarum and Dijkstra algorithms. Based on the results in the table, we observe that the proposed Physarum algorithm reduces the

computational time by directly providing the optimal solution. It outperforms the Dijkstra algorithm for the problem of multi-vehicle assignment under uncertainty.

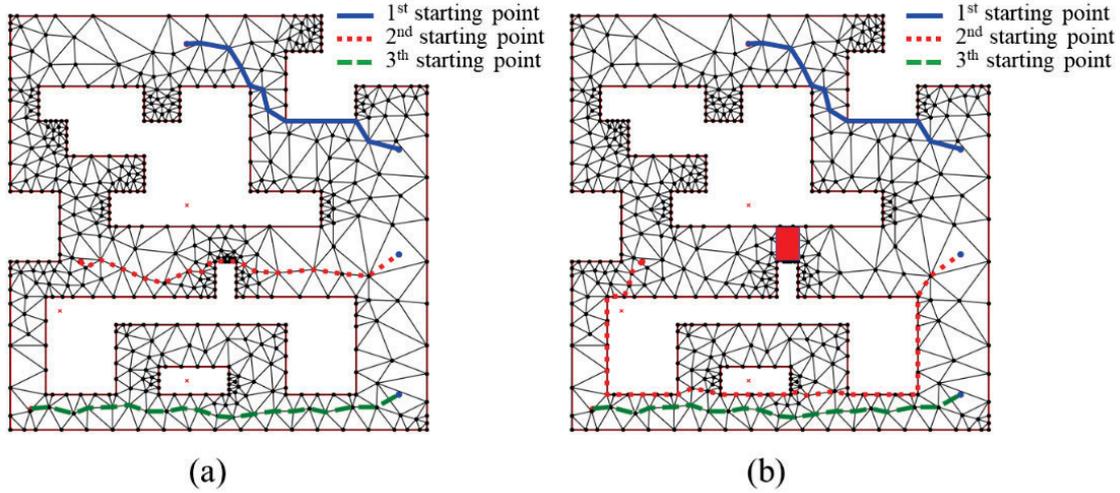


Figure 4.23 (a) original mission assignment and (b) dynamic mission assignment

Table 4.5 Comparison between different approaches for multi-vehicle path planning

Method	Situation	Overall length of all paths	Time of graph building (s)	Calculation time (s)
Physarum	Original	134.39	0.02	0.69
	Dynamic	173.85	0.02	0.59
Dijkstra	Original	134.39	0.01	1.06
	Dynamic	173.85	0.01	1.24

4.4. Summary

This chapter proposes a Physarum algorithm for model-based path planning under uncertainty subject to a mobility reliability constraint. A simulation model is first used to generate the training data to build a surrogate model for the vehicle mobility. An adaptive surrogate modeling approach is employed to further improve the accuracy of the surrogate model. Using the vehicle mobility surrogate model, a GO/NO-GO map is generated by specifying a vehicle state mobility reliability requirement. Finally, a Physarum algorithm is combined with a navigation mesh to perform mission planning of single and multiple vehicles in a dynamic environment with a specific reliability requirement. According to the results of the two case studies, the proposed method is capable of finding the shortest path

in a network-based map. Although the Physarum algorithm is slower than the Dijkstra algorithm for a single vehicle path planning, it is faster than Dijkstra for multi-vehicle path planning problems. Moreover, the results show that if the number of vehicles increases from 1 to 3, the planning time of the Dijkstra algorithm increases from 0.14s to 1.06s (around 8 times) while that of Physarum algorithm changes only from 0.66s to 0.69s (basically the same). The proposed method can also handle dynamic path planning problems very well.

Chapter 5

Reliability-Based Path Planning Under Uncertainty with Multiple Constraints Using RRT*-Based Approach

This chapter presents a novel Rapidly-exploring Random Trees Star (RRT*) based approach for model-based vehicle mission planning under uncertainty with multiple mobility reliability requirements for off-road AGV. A physics-based vehicle dynamics simulation model is first employed to predict vehicle mobility (i.e., maximum attainable speed and vehicle vertical acceleration) for any given terrain conditions. Then multiple surrogate models are constrained to obtain different types of mobility performance. After that, the mission mobility reliability for the AGV is defined using different mobility performances. Finally, an RRT* based approach is used to identify an optimal path satisfying all reliability requirements. Then a GO/NO-GO map is generated using the state mobility reliability to furtherly reduce the computational time of the algorithm. A case study is used to demonstrate the effectiveness of the proposed methods and algorithms. The results show that the proposed method can identify the optimal path while maintain the reliability of all mobility performances.

5.1. Background

5.1.1. *Off-Road Vehicle Mobility Model*

Global path planning of off-road AGVs is to find the shortest path in a certain area from a start point to a target point while satisfying specific requirements. Physics-based M&S plays a critical role in predicting the mobility and global path planning of AGVs. As shown in Figure 5.1, M&S-based global path planning of AGVs in general consists of four main steps, namely

- 1) First, the target area map is obtained either from satellite or other sources.
- 2) Second, the environment in the target area is characterized using topographic information (height map) and soil information (soil map). The soil map is additionally described by soil parameters, such as cohesive strength, friction angle, bulk density, etc. At the same time, vehicle and corresponding vehicle parameters are determined.
- 3) Third, the height information, soil parameters, and vehicle parameters are used to construct a physics-based mobility model to describe the interaction between soil and vehicle to predict the vehicle mobility.
- 4) Last, the mobility model is combined with path planning algorithm to find the optimal path in the map from the start point to the end point.

Several concepts are defined here and used for path planning. The target map is called a configuration space denoted by Ω . In the configuration space, the space with obstacles is called obstacle space, denoted by Ω_{ob} . The space that AGVs can fulfill the mission is called free space represented by Ω_{fr} . The arbitrary location in the configuration space has two dimensions and is given by $\mathbf{x} = (x_1, x_2)$. The start point is also called the initial point represented by \mathbf{x}_{ini} . The end point is the target to achieve, which is denoted by \mathbf{x}_{tr} . The connection between any arbitrary nodes is called edge. The planned path of the mission is defined by Γ .

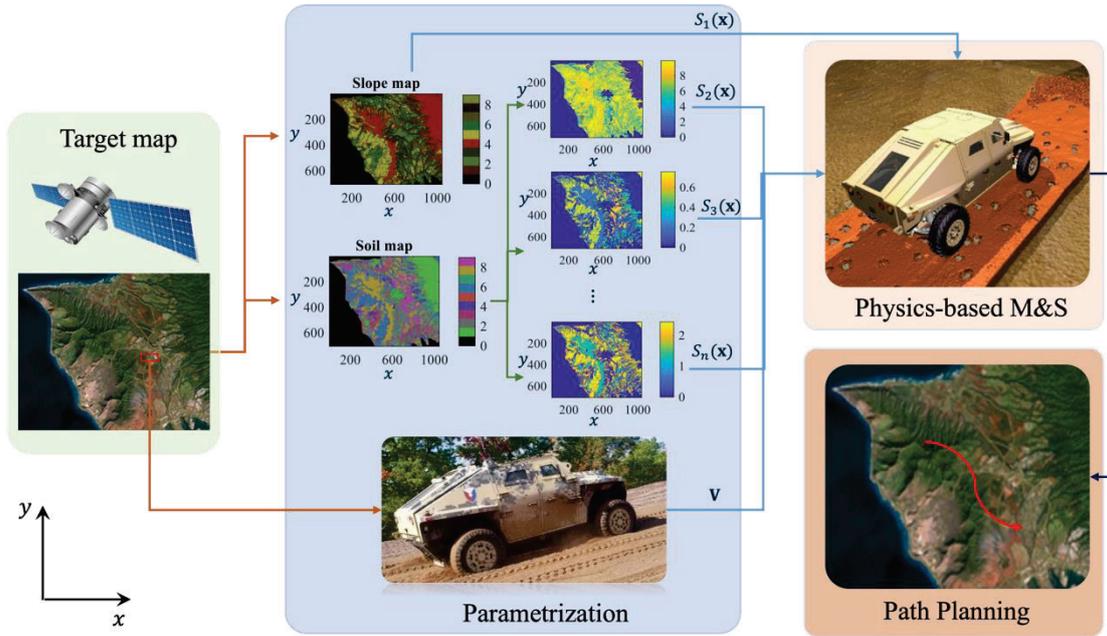


Figure 5.1 Illustration of path planning for Off-road AGV

5.1.2. Physics-based M&S

The mobility of an AGV is related to the vehicle and terrain conditions as shown in Figure 5.1. A generalized formulation for the mobility model is given by

$$\mathbf{Y}(\mathbf{x}) = M(\mathbf{V}, \mathbf{S}(\mathbf{x})), \mathbf{x} \in \Omega, \quad (5.1)$$

where $\mathbf{Y}(\cdot)$ is the output or quantities of interest representing the vehicle mobility at location \mathbf{x} , e.g., the maximum attainable speed and maximum vertical acceleration, $M(\cdot, \cdot)$ is the mobility model in general, which could be an empirical model or a simulation model, \mathbf{V} is a vector of vehicle-related parameters, and $\mathbf{S}(\cdot) = [S_1(\mathbf{x}), S_2(\mathbf{x}), \dots, S_n(\mathbf{x})]^T$ is a vector representing the surrounding conditions of the vehicle (e.g., slope and soil parameters).

With the mobility model, M&S is combined with the path planning algorithm (RRT*) to identify the optimal path while satisfying physical constraints.

5.1.3. Path Planning by RRT*

RRT* finds the optimal path by iterations of a growing tree structure starting from \mathbf{x}_{ini} . A random node ($\mathbf{x}_{rand} \in \Omega$) is generated using a sampling strategy. If $\mathbf{x}_{rand} \in \Omega_{ob}$, \mathbf{x}_{rand} is rejected. The sampling process continues until a random sample is in the free space; namely, $\mathbf{x}_{rand} \in \Omega_{fr}$. Then the nearest node (\mathbf{x}_{nst}) to \mathbf{x}_{rand} in tree \mathbf{T} is identified. If \mathbf{x}_{rand} is accessible to \mathbf{x}_{nst} , meaning that there is no obstacle between \mathbf{x}_{rand} and \mathbf{x}_{nst} , \mathbf{x}_{rand} is inserted to the tree by connecting \mathbf{x}_{nst} and \mathbf{x}_{rand} . Otherwise, a new node \mathbf{x}_{new} is generated using a steering function and added to the tree by connecting it to \mathbf{x}_{nst} . A collision check is performed to ensure that no obstacle between \mathbf{x}_{new} and \mathbf{x}_{nst} .

After \mathbf{x}_{new} passes the collision check, the rewiring operation is used to modify the tree [104], which consists of four steps as shown in Figure 5.2. The numbers on the edges represent distances between two connected nodes. The four steps of the rewiring operation are reviewed as follows:

- 1) First, a radius is given to find the neighbor nodes within the sphere as shown in Figure 5.2(a). The radius is defined as

$$r = \gamma \left(\frac{\log \mathcal{G}}{\mathcal{G}} \right)^{\frac{1}{dim}}, \quad (5.2)$$

where γ is a constant defined according to the environment, \mathcal{G} is the number of iterations, and dim is the dimension of configuration space. Within the area in Figure 5.2(a), four candidate parent nodes are identified, which are labeled as 4, 5, 6, and 8.

- 2) Second, the best parent with the lowest cost from \mathbf{x}_{ini} to \mathbf{x}_{new} is identified. From Figure 5.2(b), it is easily observed that the route $0 \rightarrow 1 \rightarrow 5 \rightarrow 9$ has the lowest cost in distance which is $3+5+3=11$. Hence, the parent node \mathbf{x}_p of node 9 is node

5.

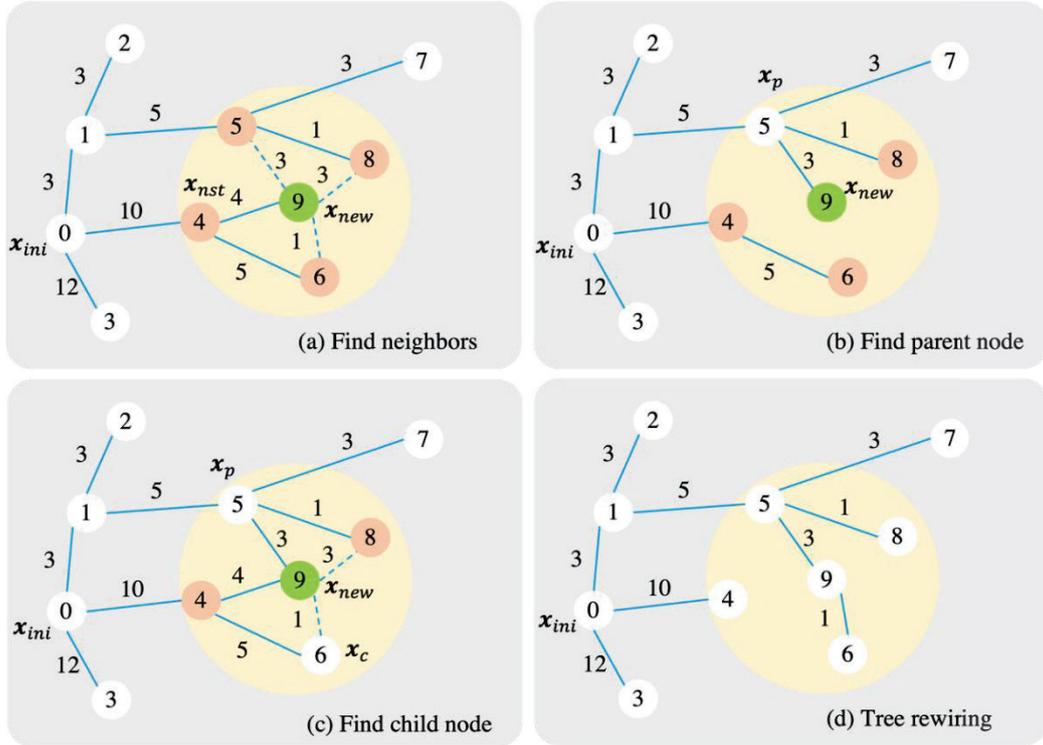


Figure 5.2 Illustration of tree-rewriting in RRT*

- 3) Third, the child node \mathbf{x}_c is identified from the remaining neighbor nodes 4, 6, and 8. Starting from \mathbf{x}_{ini} and passing through \mathbf{x}_p and \mathbf{x}_{new} , the lowest cost is $0 \rightarrow 1 \rightarrow 5 \rightarrow 9 \rightarrow 6$ with a total distance of 12 according to Figure 5.2(c). The child of node 9 is identified as node 6.
- 4) Last, the edges $9 \rightarrow 4$ and $4 \rightarrow 6$ are removed, new edges $5 \rightarrow 9$ and $9 \rightarrow 1$ are constructed. The rewired tree is obtained as shown in Figure 5.2(d).

5.1.4. Challenges in Path Planning of Off-Road AGVs

Reliable path planning of off-road AGVs is challenging for various reasons. Three of them are addressed in this work. The first challenge is the highly uncertain working environment of AGVs due to uncertain soil conditions, which leads to variation of AGV mobility as shown in Figure 5.3. In a certain region with deformable soil, an AGV could have a certain probability of either passing or getting stuck. Understanding the effect of

uncertainty on mobility performance is critical for path planning of off-road AGVs.

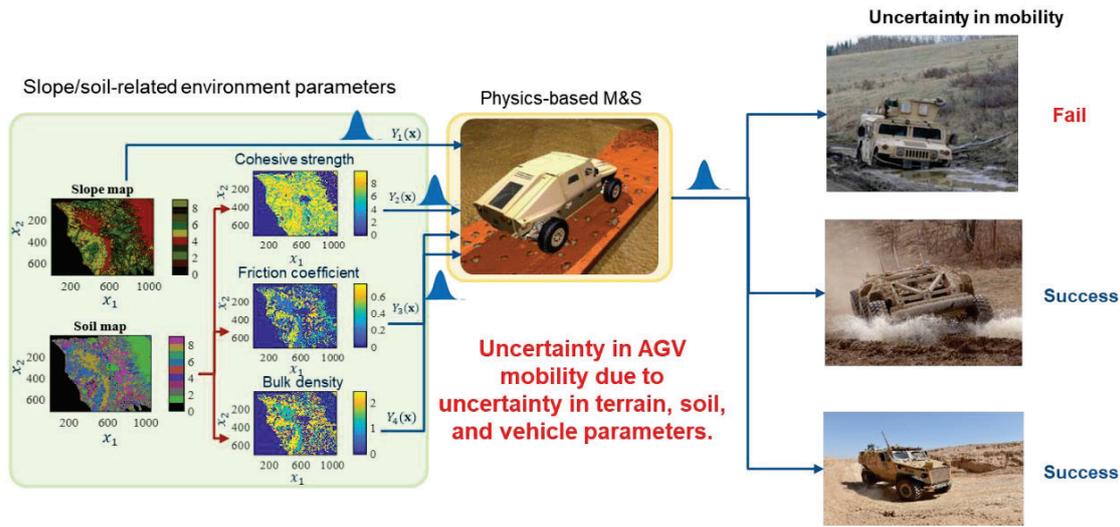


Figure 5.3 Uncertainty sources and its impact on AGV mobility

Another challenge is that AGVs could lose mobility in many ways, but current methods only consider mobility in terms of the speed of AGVs, called speed-made-good reliability. Besides the maximum attainable speed, AGVs could fail in other ways. For instance, a highly possible failure is the vehicle and supplies being damaged due to extreme vibration. Considering mobility from multiple perspectives is critical to ensure the success of a mission.

The third challenge is the balance of computational efficiency and accuracy. The semi-empirical model is very efficient, but with large prediction errors. On the other hand, the high-fidelity simulation model is accurate but takes a very long time to run. For path planning with multiple reliability constraints, the reliability constraints need to be evaluated thousands of times. If the high-fidelity simulation model is used for M&S, it is computationally prohibitive. Modeling methods with high efficiency and accuracy are therefore needed for reliability-based path planning with multiple failure modes.

To address the above challenges, this work develops a path planning method with multiple reliability constraints under uncertainty. An adaptive surrogate model and a dynamic ensemble-based dynamic surrogate modeling method are used to efficiently assess

reliability from two different perspectives. The detailed methodology is provided in the next section.

5.2. Reliability-Based Path Planning of Off-Road AGVs Subject to Multiple Failure Modes

In this section, we first provide an overview of the proposed framework. After that, we explain each element of the proposed framework in detail.

5.2.1. Overview

As discussed previously, without accounting for the prevalent uncertainty in path planning could lead to a high risk of failure of the off-road AGV. Many failure modes could happen, such as vehicle getting stuck on a path, vehicle overturning, running out of power, or goods damage due to severe vertical acceleration. In this paper, we focus on two main failure modes that can be simulated using M&S, namely maximum attainable speed and maximum vertical acceleration. To account for these failure modes in path planning of off-road AGVs, a generalized reliability-based path planning model is formulated as

$$\begin{aligned}
 & \min C(\Gamma) \\
 & s.t. MMR_i \geq R_i, i = 1, 2, \dots, n_r, \\
 & \Gamma \in \Omega_{fr}
 \end{aligned} \tag{5.3}$$

where Γ is the designed path; $C(\Gamma)$ is cost in terms of Γ to minimize; MMR represents the mission mobility reliability, whose subscript denotes the i -th reliability consideration when n_r failure modes exist (i.e., $n_r = 2$ in this paper).

Figure 5.4 shows an overview of the proposed framework. As shown in this figure, the proposed framework consists of three main modules.

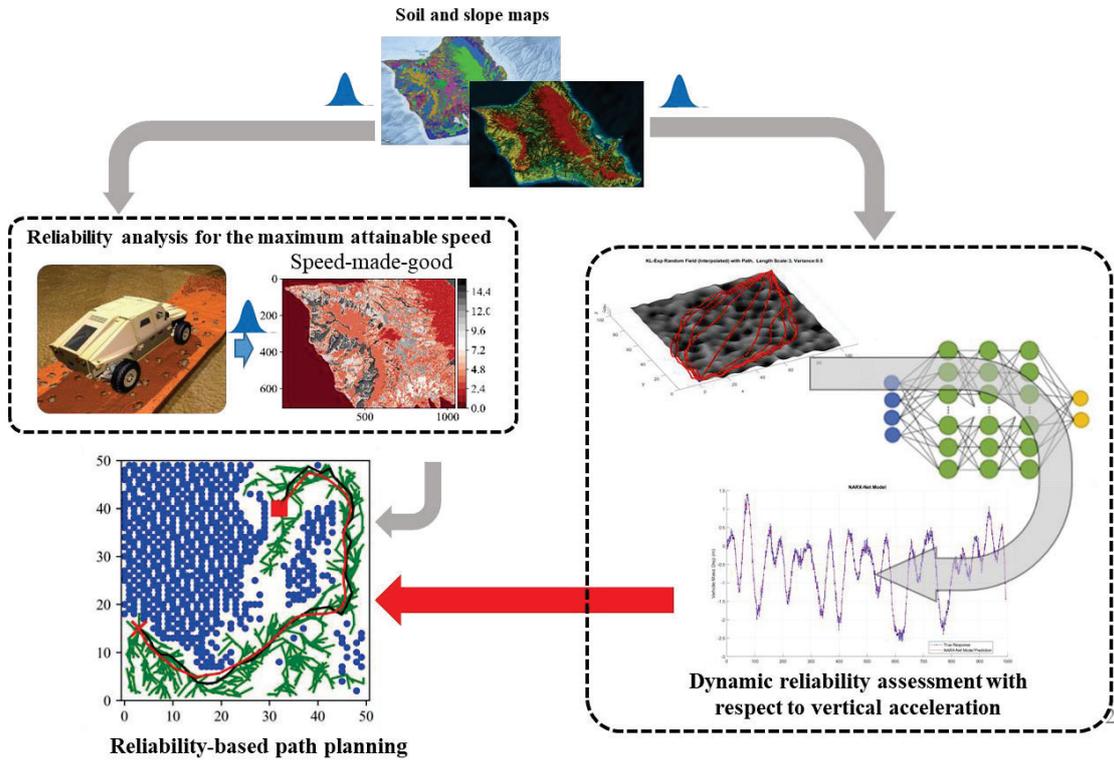


Figure 5.4 Overview of the proposed framework

The *first* module focuses on mobility reliability analysis of off-road AGVs with respect to the maximum attainable speed, which is the maximum steady-state speed that a vehicle can achieve for a given terrain and soil condition. The *second* module concentrates on mission mobility reliability analysis for the maximum vertical acceleration. It is worth noting that reliability analyses for maximum attainable speed and vertical acceleration require different approaches. This is attributed to the fact that the vertical acceleration is a dynamic quantity while the maximum attainable speed is a quasi-static quantity. Based on the reliability analysis methods developed in the first two modules, the third module integrates the reliability constraints into RRT*-based path planning algorithm to identify the path that is shortest and satisfies the two reliability constraints.

In what follows, we explain the three modules of the proposed framework in detail.

5.2.2. Reliability Analysis for the Maximum Attainable Speed

In order to operate successfully in the off-road environment, an AGV needs to maintain

a certain minimum speed to avoid getting stuck in the deformable terrain or for safety purpose in the battlefield. Based on this consideration, the reliability with respect to the maximum attainable speed is defined as

$$MMR_v(\Gamma) = \Pr\{v_m(\mathbf{S}(\mathbf{x})) > v_e, \forall \mathbf{x} \in \Gamma\}, \quad (5.4)$$

where $v_m(\cdot)$ is the maximum attainable speed, v_e is a speed threshold, “ \forall ” stands for "for all", and $\mathbf{S}(\mathbf{x})$ is a vector of terrain- and soil-related parameters at spatial location \mathbf{x} .

In this paper, the maximum attainable speed $v_m(\mathbf{S}(\mathbf{x}))$ for surrogate model construction is predicted using a high-fidelity simulation software called PyChrono [35, 87]. Figure 5.5 presents an example of the dynamic simulation of off-road AGV motion in PyChrono. The considered terrain- and soil-related parameters $\mathbf{S}(\mathbf{x})$ in the simulation include the height of slope (h), soil properties such as Bekker coefficients (k_ϕ, k_c, B_n), soil cohesion (c), friction angle (ϕ), and Janosi shear displacement (J_s) [88]. Thus, we have input parameters $S(\mathbf{x}) = (h(\mathbf{x}), k_\phi(\mathbf{x}), k_c(\mathbf{x}), B_n(\mathbf{x}), c(\mathbf{x}), \phi(\mathbf{x}), J_s(\mathbf{x}))$ as illustrated in Figure 5.3. The uncertainty in the soil properties results in uncertainty in AGV mobility, such as the maximum attainable speed. To account for the uncertainty in terrain and soil properties, we model $\mathbf{S}(\mathbf{x})$ as space-dependent Gaussian random fields. The spatial correlation between any arbitrary two locations in the map is described by an auto-correlation function as

$$k_{12} = \exp\left\{-\left(\frac{\mathbf{x}_1 - \mathbf{x}_2}{\theta_x}\right)^2\right\}, \quad (5.5)$$

where θ_x is the correlation length along space; $\mathbf{x}_1, \mathbf{x}_2$ are two arbitrary points in the map, and k_{12} is the correlation between the two points.

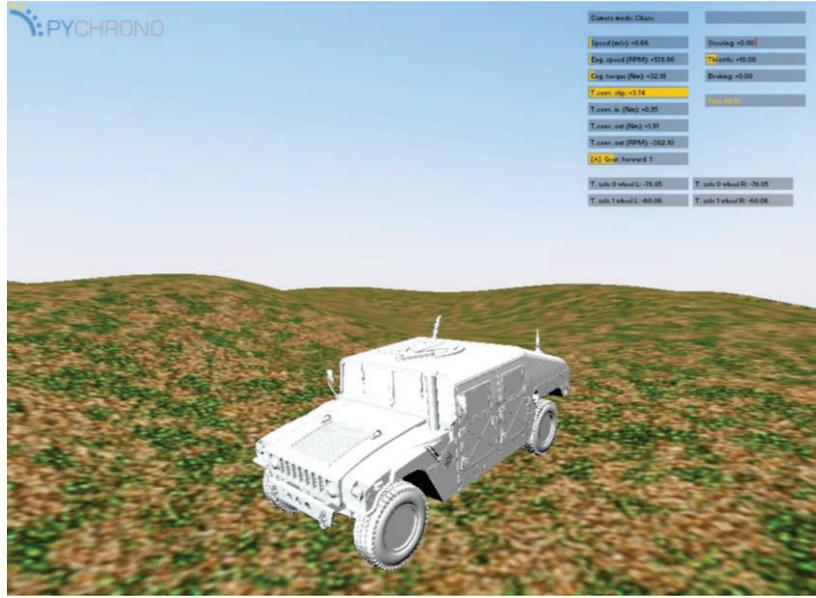


Figure 5.5 An example of simulating movement of an AGV in off-road environment in PyChrono

In order to generate random realizations of the random fields for reliability analysis, the truncated Karhunen-Loeve (K-L) expansion [105] is employed as follows

$$R_f(\mathbf{x}) = \mu(\mathbf{x}) + \sum_{i=1}^m \sqrt{\lambda_i} \varphi_i(\mathbf{x}) \xi_i, \quad (5.6)$$

where $R_f(\mathbf{x})$ is a generated realization of a random field; $\mu(\mathbf{x})$ is the mean function in terms of the spatial coordinate \mathbf{x} of the random field; m is the truncation number; λ_i and $\varphi_i(\mathbf{x})$ is the eigenvalue and eigenvector by performing eigen-analysis of the correlation matrix, respectively; and $\xi_i, i = 1, 2, \dots, m$ are independent standard Gaussian random variables.

A direct way of performing reliability analysis is to propagate the realizations of the random field through the PyChrono simulation model as illustrated in Figure 5.3. However, such a direct Monte Carlo simulation (MCS) is computationally prohibitive. To address this challenge, an adaptive surrogate model is adopted. The adaptive surrogate modeling method has been proven that it can accurately find the failure boundary with high efficiency

[90, 93, 94, 106-110]. We first generate N initial training points by Latin Hypercube sampling denoted by $\mathbf{S} = [\mathbf{S}_1, \mathbf{S}_2, \dots, \mathbf{S}_N]^T$. Their responses are obtained from PyChrono as $\mathbf{v}_m = [v_{m1}, v_{m2}, \dots, v_{mN}]^T$. Given the training data, an initial Gaussian Process model (GP) is obtained as

$$\hat{v}_m = \mathbf{f}(\mathbf{S})^T \boldsymbol{\beta} + \varepsilon(\mathbf{S}), \quad (5.7)$$

where \hat{v}_m is the predicted maximum attainable speed by GP; $\mathbf{f}(\cdot)$ and $\boldsymbol{\beta}$ are vectors representing basis functions and corresponding coefficients, respectively; and $\varepsilon(\cdot)$ is an error term. The prediction by GP is not a single value but a normal distribution; namely, $\hat{v}_m \sim N(\mu_v(\mathbf{S}), \sigma_v^2(\mathbf{S}))$, where μ_v is the mean prediction and σ_v is the standard deviation denoting the prediction uncertainty.

In addition to the training data, we also generate a group of input samples using MCS as $\mathbf{S}_{MCS} = \{\mathbf{S}^i\}_1^{N_{MCS}}$. The corresponding predicted responses from the initial GP are denoted as $\hat{v}_m^i \sim N(\mu_v^i(\mathbf{S}^i), (\sigma_v^i(\mathbf{S}^i))^2)$. Based on the MCS samples, we then identify new training points for the refinement of the GP model given in Eq. (5.7) by using the following active learning function

$$U(\mathbf{S}^i) = \frac{|\mu_v^i(\mathbf{S}^i) - v_e|}{\sigma_v^i(\mathbf{S}^i)}, i = 1, 2, \dots, N_{MCS}, \quad (5.8)$$

where \mathbf{S}^i is a point from \mathbf{S}_{MCS} ; The value of $U(\mathbf{S}^i)$ indicates the probability of misclassification. A lower value of $U(\mathbf{S}^i)$ means a higher probability that the sample is misclassified. We therefore identify the samples of \mathbf{S}_{MCS} in the vicinity of the failure boundary with lower $U(\mathbf{S}^i)$ value, until all $U(\mathbf{S}^i)$ from \mathbf{S}_{MCS} are greater than 2; namely,

$(\min\{\mathbf{S}^i\}_1^{N_{MCS}}) \geq 2$. Then, the active learning converges, and we have 95% confidence that the samples are right classified.

The new training sample \mathbf{S}^* at each iteration is identified by

$$\mathbf{S}^* = \arg \min \{U(\mathbf{S}^i)\}_1^{N_{MCS}}, \quad (5.9)$$

The identified new training sample is added to the training set to refine the GP model at each iteration. After the algorithm converges, the safe and failure region of the AGV maximum attainable speed is identified. MCS can be used to obtain the vehicle mission mobility reliability in terms of maximum attainable speed (i.e. Eq. (5.4)) as

$$MMR_v(\Gamma) \approx \frac{1}{N_{MCS}} \sum_{i=1}^{N_{MCS}} I_v, \begin{cases} I_v = 0, & \text{if } \min_{x \in \Gamma} e_v(x) < 0 \\ I_v = 1, & \text{if } \min_{x \in \Gamma} e_v(x) \geq 0 \end{cases} \quad (5.10)$$

Now we have the first reliability constraint (MMR_v). Next, we discuss the details of the second reliability constraint in terms of vertical acceleration.

5.2.3. Dynamic Reliability Analysis with Respect to Vertical Acceleration

A major difference between the maximum attainable speed and the vertical acceleration is that the former is only affected by the terrain and soil properties at a specific spatial location while the latter is dependent on not only the terrain and soil at current location, but also that of previous locations that an AGV passes through. Because of this, a different reliability analysis method is required for the failure mode of vertical acceleration. Moreover, from the results of PyChrono simulations, it is found that the vertical acceleration is insensitive to the soil parameters but closely related to the slope height. As shown in Figure 5.6, given a path, different soils are used to simulate the vertical acceleration behavior along the path. It is shown that the soil type has a negligible impact on the vertical acceleration.

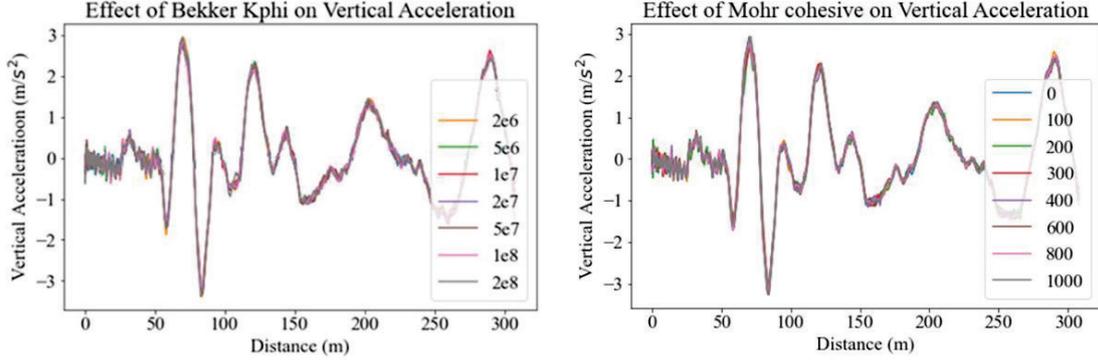


Figure 5.6 Effects of soil parameters on the vertical acceleration

Hence, we can simplify the vertical acceleration model as a function of the terrain elevation as

$$a_i = f(h_i, h_{i-1}, \dots, h_{i-p}, a_{i-1}, a_{i-2}, \dots, a_{i-q}), \quad (5.11)$$

where a_i is the vertical acceleration at spatial location \mathbf{x}_i , $h_i = h(\mathbf{x}_i)$ is the elevation at \mathbf{x}_i , $\forall i = 1, 2, \dots, p$, and $f(\cdot)$ is a nonlinear function. This model is also called a Nonlinear Auto Regressive exogenous (NARX) model.

Due to the complicated off-road environment and the nonlinear vehicle dynamics, there is no analytical solution available for Eq. (5.11). In this section, we learn Eq. (5.11) using the synthetic mobility data collected from PyChrono simulations, and a recently developed dynamic ensemble of NARX Models (DENA) method is employed to accomplish this task [32]. The basic idea of DENA is first to train multiple NARX models with different segments divided from the whole prediction area using the Gaussian mixture model (GMM) [111, 112]. Then, the NARX models are ensembled dynamically over paths to predict the vertical acceleration along the path.

More specifically, we first generate N_p paths for the off-road AGV. The heights (terrain elevation) of the N_p paths are denoted by $h_j^i, i = 1, 2, \dots, N_p; j = 1, 2, \dots, N_q$, where N_i is the number of coordinates for i -th path. Given the path information, the vertical acceleration is obtained by PyChrono simulation and is denoted by $a_j^i, i = 1, 2, \dots, N_p; j = 1, 2, \dots, N_q$. We then convert the data into the NARX format as

$$\mathbf{H}_T = \begin{bmatrix} \mathbf{h}^1 & \mathbf{a}^1 \\ \mathbf{h}^2 & \mathbf{a}^2 \\ \vdots & \vdots \\ \mathbf{h}^{N_T} & \mathbf{a}^{N_T} \end{bmatrix}, \quad (5.12)$$

where \mathbf{H}_T is the training data from the paths generated; $\{\mathbf{h}^i \in \mathbb{R}^{1 \times p}\}_1^{N_T}$ are heights from the segments of the paths with the length of p points; $\{\mathbf{a}^i \in \mathbb{R}^{1 \times q}\}_1^{N_T}$ are the corresponding vertical acceleration from the same segments of the paths. Based on the first q points, we predict the next vertical acceleration $\{\mathbf{a}^{(q+1)i} \in \mathbb{R}^{1 \times 1}\}_1^{N_T}$. We let $T=q+1$, then the corresponding prediction of the training points from Eq. (5.12) is given by

$$\mathbf{A}_T = \begin{bmatrix} a^{T1} \\ a^{T2} \\ \vdots \\ a^{TN_T} \end{bmatrix}, \quad (5.13)$$

Next, we partition all the training points and labels into different segments using Gaussian mixture model (GMM) to capture the dynamic behavior of AGVs in different dynamic regions. GMM approximates the joint PDF of \mathbf{H}_T as a weighted sum of multivariate Gaussian components as below

$$f(\mathbf{H}_T) = \sum_{k=1}^Q w_k \phi(\mathbf{H}_T | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \quad (5.14)$$

where Q is the number of Gaussian components; w_k is the weight of the k -th Gaussian component; $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ is the mean vector and the covariance matrix of the multivariate Gaussian distribution, respectively; and $\phi(\cdot)$ is the PDF of the multivariate Gaussian distribution.

After that, the probability that the i -th ($j = 1, \dots, N_T$) training point belongs to k -th ($k = 1, \dots, Q$) cluster is given by

$$p_k(\mathbf{H}_{T,i}) = p(\mathbf{H}_{T,i}, k | w_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \quad (5.15)$$

where $\mathbf{H}_{T,i}$ represent the i -th ($j = 1, \dots, N_T$) training point in \mathbf{H}_T ; $p_k(\mathbf{H}_{T,i})$ denotes the probability that the training point belongs to the k -th ($k = 1, \dots, Q$) cluster; and $w_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$ are the learned parameters from GMM. The summation of the probability $p_k(\mathbf{H}_{T,i}), k = 1, \dots, Q$ is 1 as

$$\sum_{k=1}^Q P(\mathbf{H}_{T,i}, k | w_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = 1, \quad (5.16)$$

Now we can divide the whole training points \mathbf{H}_T into Q groups. Each of the group is denoted by

$$\mathbf{H}_{g,k} = \{\mathbf{H} | I_c(\mathbf{H}_{T,i}, k) = 1, \forall \mathbf{H} \in \mathbf{H}_T\}, k = 1, \dots, Q, \quad (5.17)$$

Similarly, the corresponding predicted vertical accelerations in Eq. (5.13) are divided into the same Q groups, which is denoted by

$$\mathbf{A}_{g,k} = \{\mathbf{A} | I_c(\mathbf{A}_{T,i}, k) = 1, \forall \mathbf{A} \in \mathbf{A}_T\}, k = 1, \dots, Q, \quad (5.18)$$

It is noted that a sample \mathbf{H} and the corresponding \mathbf{A} could belong to multiple groups, which means the training points of multiple NARX models could overlap. The samples and corresponding labels of each group are a subset of \mathbf{H}_T and \mathbf{A}_T , respectively. The union of all subsets are \mathbf{H}_T and \mathbf{A}_T , which is denoted by

$$\mathbf{H}_T = \{\mathbf{H}_{g,1} \cup \dots \cup \mathbf{H}_{g,Q}\}, \quad (5.19)$$

$$\mathbf{A}_T = \{\mathbf{A}_{g,1} \cup \dots \cup \mathbf{A}_{g,Q}\}, \quad (5.20)$$

Recall that every subset of training points is used to build a NARX model. Each NARX model is denoted by

$$\hat{a}_k = \hat{g}_{NX,k}(\mathbf{h}, \mathbf{a}), k = 1, 2, \dots, Q, \quad (5.21)$$

Where $\mathbf{h} = [h_d, h_{d-1}, \dots, h_{d-q}]$ and $\mathbf{a} = [a_d, a_{d-1}, \dots, a_{d-q}]$ represent the height and vertical acceleration first $q+1$ and q steps, respectively; $\hat{g}_{NX,k}(\cdot)$ denotes the NARX model; and

\hat{a}_k is the prediction at coordinate d by k -th NARX model. In this paper, the GPR model is employed to learn $\hat{g}_{NX,k}(\cdot)$ based on data.

After all the NARX models are available, we assemble them dynamically over prediction length to predict the vertical acceleration in a given path. The prediction at d coordinate by assembled DENA model is given by [32]

$$\hat{a}(d) = \sum_{k=1}^Q w(\mathbf{H}_{T,d}, k) \hat{g}_{NX,k}(\mathbf{h}, \mathbf{a}), \quad (5.22)$$

where $w(\mathbf{H}_{T,d}, k)$ is denoted by

$$w(\mathbf{H}_{T,d}, k) = \frac{\hat{p}_k(\mathbf{H}_{T,d})}{\sum_{k=1}^Q \hat{p}_k(\mathbf{H}_{T,d})}, \quad (5.23)$$

in which $\hat{p}_k(\mathbf{H}_{T,d})$ is obtained by

$$\hat{p}_k(\mathbf{H}_{T,d}) = \begin{cases} \hat{p}_k(\mathbf{H}_{T,d}), & \text{if } \hat{p}_k(\mathbf{H}_{T,d}) \geq \frac{0.05}{Q-1}, \\ 0, & \text{otherwise} \end{cases} \quad (5.24)$$

where $\hat{p}_k(\mathbf{H}_{T,d})$ is solved by Eq. (5.15) using GMM.

Since Gaussian Process regression is used to fit the dynamic behavior, we have \hat{a}_k in Eq. (5.21) following a normal distribution as given below.

$$\hat{a}_k \sim N(\mu_{\hat{a}_k}, \sigma_{\hat{a}_k}^2), \quad (5.25)$$

where $\mu_{\hat{a}_k}$ and $\sigma_{\hat{a}_k}$ are the mean and standard deviation of \hat{a}_k . By substituting Eq. (5.25) into Eq. (5.22), we have the mean prediction and standard deviation of DENA as shown below.

$$\mu_{\hat{a}}(d) = \sum_{k=1}^Q w(\mathbf{H}_{T,d}, k) \mu_{\hat{a}_k}(\mathbf{h}, \mathbf{a}), \quad (5.26)$$

$$\sigma_{\hat{a}}(d) = \sqrt{\sum_{k=1}^Q w^2(\mathbf{H}_{T,d}, k) \sigma_{\hat{a}_k}^2(\mathbf{h}, \mathbf{a})}, \quad (5.27)$$

Once we have the DENA model constructed, the vertical acceleration along any given path can be predicted recursively. Figure 5.7 summarizes the overall flowchart of DENA for off-road AGV vertical acceleration prediction.

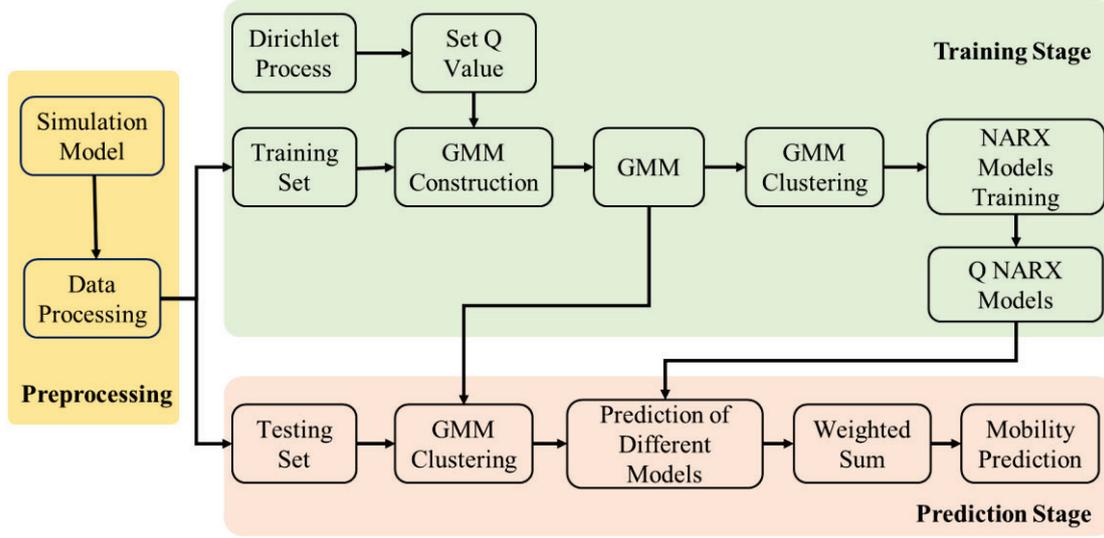


Figure 5.7 Flowchart of Dynamic Ensemble of NARX model (DENA)

As discussed previously, the severe vibration could cause damage to the goods and supply inside the AGV as well as some components of the AGV. Therefore, we set a threshold for vertical acceleration. Once the vertical acceleration along the path exceeds the threshold, the mission is assumed to be failed. Mathematically, we have,

$$I_a = \begin{cases} 1, & \text{if } \max_{x \in \Gamma} \hat{a}(x) \leq a_{th} \\ 0, & \text{if } \max_{x \in \Gamma} \hat{a}(x) > a_{th} \end{cases}, \quad (5.28)$$

where $\hat{a}(x)$ is the predicted acceleration by DENA along the planned path; a_{th} is the threshold vertical acceleration.

The reliability in terms of vertical acceleration by MCS is then computed by

$$MMR_a(\Gamma) \approx \frac{1}{N_{MCS}} \sum_{i=1}^{N_{MCS}} I_a, \quad (5.29)$$

Next, we discuss how to integrate the above presented reliability constraints with the RRT* algorithm for reliability-based path planning of off-road AGVs.

5.2.4. Reliability-Based Path Planning Subject to Multiple Failure Modes

In this subsection, we discuss how to integrate the reliability constraints into the RRT* algorithm for path planning. We name the modified RRT* as MC-RRT*, which means RRT* subject to multiple reliability constraints.

5.2.4.1. MC-RRT*

As discussed in Sec. 5.1.3, we first initialize a tree denoted by \mathbf{T} which contains all the nodes generated during the path exploration. For the first step, only the start point \mathbf{x}_{ini} and the target \mathbf{x}_{tr} are in \mathbf{T} . Next, we randomly generate a node $\mathbf{x}_{rand} \in \Omega_{fr}$ and find the nearest node (\mathbf{x}_{nst}) in the tree to \mathbf{x}_{rand} . It is noted for the first random node, the nearest node is \mathbf{x}_{ini} . The function used to find the nearest node is given by

$$\mathbf{x}_{nst} = Nearest(\mathbf{T}, \mathbf{x}_{rand}) = \arg \min_{\forall \mathbf{x} \in \mathbf{T}} \|\mathbf{x}_{rand} - \mathbf{x}\|, \quad (5.30)$$

After the random node is found, a steering function is used to generate a new node \mathbf{x}_{new} , which is denoted by

$$\mathbf{x}_{new} = steer(\mathbf{x}_{rand}, \mathbf{x}_{nst}), \quad (5.31)$$

where $steer(\cdot)$ denotes the steering function. Then, the reliability MMR_v and MMR_a in Eqs. (5.10) and (5.29) from \mathbf{x}_{ini} to \mathbf{x}_{new} are checked. If the path satisfies the reliability targets, the neighbor nodes within a radius are found by Eq. (5.2). The neighbor nodes are denoted by

$$\mathbf{x}_{nbor} = Near(\mathbf{T}, \mathbf{x}_{new}), \quad (5.32)$$

After that, the best parent node for \mathbf{x}_{new} is chosen from \mathbf{x}_{nbor} with the lowest cost. The algorithm performs rewiring to rebuild the path to \mathbf{x}_{new} . The above operations execute

iteratively until \mathbf{x}_{new} arrives the region of the target point. The detailed algorithm of MC-RRT* is provided in Algorithm 5.1.

Algorithm 5.1: MC-RRT* for reliability-based path planning of off-road AGVs

```

1  Input Parameters:  $R_v$ ,  $R_a$ ,  $\mathbf{x}_{ini}$ ,  $\mathbf{x}_{tr}$ ,  $h$ ,  $\mathbf{S}$ 
2   $\mathbf{N} \leftarrow \text{InitializeNodes}(\mathbf{x}_{ini})$ ,  $\mathbf{E} \leftarrow \text{InitializeEdge}(\emptyset)$ ,  $\mathbf{T} \leftarrow (\mathbf{N}, \mathbf{E})$ ;
3  For  $i = 1$  to  $i = N_{iter}$ :
4  |  $\mathbf{x}_{rand} \leftarrow \text{SampleFree}$ ;
5  |  $\mathbf{x}_{nst} \leftarrow \text{Nearest}(\mathbf{T}, \mathbf{x}_{rand})$ ;
6  |  $\mathbf{x}_{new} \leftarrow \text{steer}(\mathbf{x}_{rand}, \mathbf{x}_{nst})$ ;
7  | If  $\text{MMR}_v(\mathbf{x}_{new}, \mathbf{x}_{ini}, h, \mathbf{S}) > R_v$  &
8  |    $\text{MMR}_a(\mathbf{x}_{new}, \mathbf{x}_{ini}, h, \mathbf{S}) > R_a$  then:
9  |   |  $\mathbf{X}_{nbor} \leftarrow \text{Near}(\mathbf{T}, \mathbf{x}_{new})$ ;
10 |   |  $\mathbf{x}_{parent} \leftarrow \mathbf{X}_{nbor}$ ;
11 |   |  $c_{min} \leftarrow \text{Cost}(\mathbf{x}_{ini}, \mathbf{x}_{parent}, \mathbf{x}_{new})$ ;
12 |   | For  $\mathbf{x}$  in  $\mathbf{X}_{nbor}$ :
13 |   |   | If  $\text{MMR}_v(\mathbf{x}_{new}, \mathbf{x}_{ini}, h, \mathbf{S}) > R_v$  &
14 |   |      $\text{MMR}_a(\mathbf{x}_{new}, \mathbf{x}_{ini}, h) > R_a$  &
15 |   |        $\text{Cost}(\mathbf{x}_{ini}, \mathbf{x}, \mathbf{x}_{new}) < c_{min}$  then:
16 |   |       |  $\mathbf{x}_{parent} \leftarrow \mathbf{x}$ ;
17 |   |       |  $c_{min} \leftarrow \text{Cost}(\mathbf{x}_{ini}, \mathbf{x}, \mathbf{x}_{new})$ ;
18 |   |   End
19 |   End
20 |    $\mathbf{N} \leftarrow \mathbf{N} \cup \mathbf{x}_{new}$ ,  $\mathbf{E} \leftarrow \mathbf{E} \cup (\mathbf{x}_{new}, \mathbf{x}_{parent})$ ;
21 |   For  $\mathbf{x}$  in  $\mathbf{X}_{nbor}$ :
22 |   |  $c_{nbor} \leftarrow \text{Cost}(\mathbf{x}_{ini}, \mathbf{x}_{parent}, \mathbf{x})$ ;
23 |   | If  $\text{MMR}_v(\mathbf{x}_{new}, \mathbf{x}_{ini}, h, \mathbf{S}) > R_v$  &
24 |      $\text{MMR}_a(\mathbf{x}_{new}, \mathbf{x}_{ini}, h) > R_a$  &
25 |        $\text{Cost}(\mathbf{x}_{ini}, \mathbf{x}, \mathbf{x}_{new}) < c_{nbor}$  then:
26 |       |  $c_{nbor} \leftarrow \text{Cost}(\mathbf{x}_{ini}, \mathbf{x}, \mathbf{x}_{new})$ ;
27 |       End
28 |        $\mathbf{E} \leftarrow (\mathbf{E} \setminus \{(\mathbf{x}, \mathbf{x}_{parent})\}) \cup \{(\mathbf{x}_{new}, \mathbf{x})\}$ ;
29 |   End
30 End

```

25 | **End**
 26 **End**
 27 **Return** $\mathbf{T} = (\mathbf{N}, \mathbf{E}), \Gamma;$

5.2.4.2.MC-RRT*-2

To further improve the efficiency of MC-RRT*, we propose the second algorithm called MC-RRT*-2 by using the GO/NO-GO map. The idea is to reduce the search domain for path planning before reliability analysis. As shown in Fig. 1.3, for the same location, the slope angle of the AGV passed depends on the orientations. Different orientations could result in different slope angles. For the same slope, the AGV could lose mobility in specific moving directions. Therefore, we use the worst case (the maximum slope) in predicting the maximum attainable speed. The min-max approach could provide higher robustness for path planning.

In this case, we simply use the maximum slope to replace the slope information in Eq. (5.7) in predicting the mobility in terms of maximum attainable speed as below

$$e_{v,wrst} = \hat{v}_{m,wrst} - v_{th} = \mathbf{f}(s_{max}, \bar{\mathbf{S}})^T \boldsymbol{\beta} + \varepsilon(s_{max}, \bar{\mathbf{S}}) - v_{th}, \quad (5.33)$$

where s_{max} is the maximum slope; $\bar{\mathbf{S}}$ represents the other parameters in \mathbf{S} in Eq. (5.7), thereby, $\mathbf{S} = [s_{max}, \bar{\mathbf{S}}]$. Using state mobility reliability (SMR) [31], we have the GO/NO-GO map as follows.

$$\Omega_{GO} = \{\mathbf{x} \in \Omega \mid SMR_v(\mathbf{x}) \geq R_v\}, \quad (5.34)$$

$$\Omega_{NO-GO} = \{\mathbf{x} \in \Omega \mid SMR_v(\mathbf{x}) < R_v\}, \quad (5.35)$$

where Ω_{GO} and Ω_{NO-GO} denote the feasible and infeasible area in map, respectively, R_v is the reliability target in terms of maximum attainable speed, and SMR_v is obtained using

MCS as

$$SMR_v(\Omega) \approx \frac{1}{N_{MCS}} \sum_{i=1}^{N_{MCS}} I_{v,wrst}^i \begin{cases} I_{v,wrst} = 0, & \text{if } \min_{x \in \Omega} e_{v,wrst} < 0 \\ I_{v,wrst} = 1, & \text{if } \min_{x \in \Omega} e_{v,wrst} \geq 0 \end{cases} \quad (5.36)$$

After the GO/NO-GO map is available, we can use an obstacle check function to avoid the unnecessary computational cost before solving MMR_v and MMR_a . The obstacle check passes, if there is no NO-GO area between the two nodes. The details of MC-RRT*-2 are given in Algorithm 5.2.

Algorithm 5.2: MC-RRT*-2 for reliability-based path planning of off-road AGVs

```

1  Input Parameters:  $R_v, R_a, \mathbf{x}_{ini}, \mathbf{x}_{tr}, h, \mathbf{S}$ 
2   $\mathbf{N} \leftarrow \text{InitializeNodes}(\mathbf{x}_{ini}), \mathbf{E} \leftarrow \text{InitializeEdge}(\emptyset), \mathbf{T} \leftarrow (\mathbf{N}, \mathbf{E});$ 
3  For  $i = 1$  to  $i = N_{iter}$ :
4  |  $\mathbf{x}_{rand} \leftarrow \text{SampleFree};$ 
5  |  $\mathbf{x}_{nst} \leftarrow \text{Nearest}(\mathbf{T}, \mathbf{x}_{rand});$ 
6  |  $\mathbf{x}_{new} \leftarrow \text{steer}(\mathbf{x}_{rand}, \mathbf{x}_{nst});$ 
7  | If  $\text{obstacleFree}(\mathbf{x}_{nst}, \mathbf{x}_{new})$  &
8  |    $MMR_v(\mathbf{x}_{new}, \mathbf{x}_{ini}, h, \mathbf{S}) > R_v$  &
9  |    $MMR_a(\mathbf{x}_{new}, \mathbf{x}_{ini}, h, \mathbf{S}) > R_a$  then:
10 | |  $\mathbf{X}_{nbor} \leftarrow \text{Near}(\mathbf{T}, \mathbf{x}_{new});$ 
11 | |  $\mathbf{x}_{parent} \leftarrow \mathbf{X}_{nbor};$ 
12 | |  $c_{min} \leftarrow \text{Cost}(\mathbf{x}_{ini}, \mathbf{x}_{parent}, \mathbf{x}_{new});$ 
13 | | For  $\mathbf{x}$  in  $\mathbf{X}_{nbor}$ :
14 | | | If  $\text{obstacleFree}(\mathbf{x}, \mathbf{x}_{new})$  &
15 | | |    $MMR_v(\mathbf{x}_{new}, \mathbf{x}_{ini}, h, \mathbf{S}) > R_v$  &
16 | | |    $MMR_a(\mathbf{x}_{new}, \mathbf{x}_{ini}, h) > R_a$  &
17 | | |    $\text{Cost}(\mathbf{x}_{ini}, \mathbf{x}, \mathbf{x}_{new}) < c_{min}$  then:
18 | | | |  $\mathbf{x}_{parent} \leftarrow \mathbf{x};$ 
19 | | | |  $c_{min} \leftarrow \text{Cost}(\mathbf{x}_{ini}, \mathbf{x}, \mathbf{x}_{new});$ 
20 | | | End
21 | | End
22 | End
23 |  $\mathbf{N} \leftarrow \mathbf{N} \cup \mathbf{x}_{new}, \mathbf{E} \leftarrow \mathbf{E} \cup (\mathbf{x}_{new}, \mathbf{x}_{parent});$ 
24 | For  $\mathbf{x}$  in  $\mathbf{X}_{nbor}$ :

```

```

19   |   |   |  $c_{nbor} \leftarrow Cost(\mathbf{x}_{ini}, \mathbf{x}_{parent}, \mathbf{x});$ 
    |   |   |
    |   |   | If  $obstacleFree(\mathbf{x}, \mathbf{x}_{new}) \ \&$ 
    |   |   |    $MMR_v(\mathbf{x}_{new}, \mathbf{x}_{ini}, h, \mathbf{S}) > R_v \ \&$ 
20   |   |   |    $MMR_a(\mathbf{x}_{new}, \mathbf{x}_{ini}, h) > R_a \ \&$ 
    |   |   |    $Cost(\mathbf{x}_{ini}, \mathbf{x}, \mathbf{x}_{new}) < c_{nbor}$  then:
    |   |   |   |  $c_{nbor} \leftarrow Cost(\mathbf{x}_{ini}, \mathbf{x}, \mathbf{x}_{new});$ 
    |   |   |   | End
    |   |   |   | E  $\leftarrow (\mathbf{E} \setminus \{(\mathbf{x}, \mathbf{x}_{parent})\}) \cup \{(\mathbf{x}_{new}, \mathbf{x})\};$ 
21   |   |   |   |
22   |   |   |   | End
    |   |   |   |
23   |   |   |   | E  $\leftarrow (\mathbf{E} \setminus \{(\mathbf{x}, \mathbf{x}_{parent})\}) \cup \{(\mathbf{x}_{new}, \mathbf{x})\};$ 
    |   |   |   |
24   |   |   |   | End
    |   |   |   |
25   |   |   |   | End
    |   |   |   |
26   |   |   |   | End
    |   |   |   |
27   |   |   |   | Return  $\mathbf{T} = (\mathbf{N}, \mathbf{E}), \Gamma;$ 

```

In the next section, we use a case study to demonstrate the proposed reliability-based path planning algorithm in comparison with the conventional RRT* without any reliability considerations.

5.3. Case Study

In this example, we use a $200m \times 200m$ target map to demonstrate the proposed path planning method. The map is obtained from the ARCGIS/ENVI database with uncertainty. High-fidelity simulation by PyChrono is used to obtain the training data for the construction of the two surrogate models. The proposed MC-RRT* and MC-RRT*-2 are applied to find the optimal path with required reliability targets. The optimal path found is validated using PyChrono simulation. The detailed map and soil parameters information and results are provided below.

5.3.1. Map Information and Soil Properties

As discussed previously, uncertainty exists in the target area due to the complex surrounding environment. We use the Gaussian random field to model the uncertainty over space. A realization of the random fields (height and soil parameters) is shown in Fig. 5.8. Note that the map given in Fig. 5.8 has been artificially modified in order to demonstrate

the differences between different path planning algorithms. Table 5.1 gives the uncertainty parameters of the target map. We assume that five different soils could exist in the target map. The properties of each soil are described using six parameters, which are Bekker coefficients (k_ϕ, k_c, B_n), soil cohesion (c), friction angle (ϕ), and Janosi shear displacement (J_s). All the soil parameters together with height are represented by the Gaussian random field with a correlation length of 30 m.

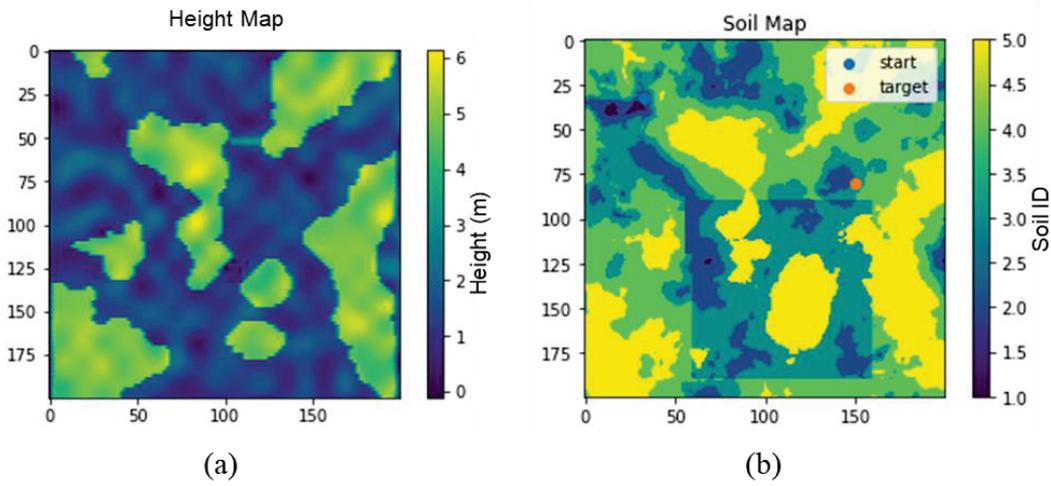


Figure 5.8 (a) Height map and (b) Soil map of interest

5.3.2. Construction of Mobility Model

As discussed in Secs. 5.2.2 and 5.2.3, we use adaptive GP and DENA as surrogate models to predict the maximum attainable speed and vertical acceleration respectively. For adaptive GP, we use 100 initial training points from Latin Hypercube Sampling (LHS) to construct an initial GP model. Active learning then refines the GP model iteratively until satisfying accuracy is achieved.

Table 5.1 Statistical information of different slope/soil parameters

Height		Soil ID	Soil											
h (m)			k_ϕ		k_c		B_n		c (Pa)		ϕ ($^\circ$)		J_s (m)	
μ	σ		μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
1.3	0.45	1	1×10^9	1000	5×10^8	500	2.8	0.01	950	1	37.5	0.5	0.048	0.001

	2	5×10^8	1500	1×10^8	750	2.6	0.05	800	2.5	35	1	0.04	0.002
	3	1×10^8	2000	5×10^7	2500	2.2	0.1	650	5	32.5	1.5	0.036	0.002
	4	5×10^7	500	1×10^7	1500	2.2	0.02	500	10	30	1	0.032	0.003
	5	1×10^7	1000	5×10^6	1000	2.0	0.1	450	15	27.5	2	0.029	0.002

The convergence history and accuracy of adaptive GP are shown in Figure 5.9. From Figure 5.9(a), we see that after 45 iterations, the minimum value of U from Eq. (5.8) is larger than 2. The accuracy of adaptive GP is compared with the ground truth by Monte Carlo Simulation (MCS) and variance minimization method as shown in Figure 5.9(b). It shows that the active learning based adaptive GP converges faster with good accuracy.

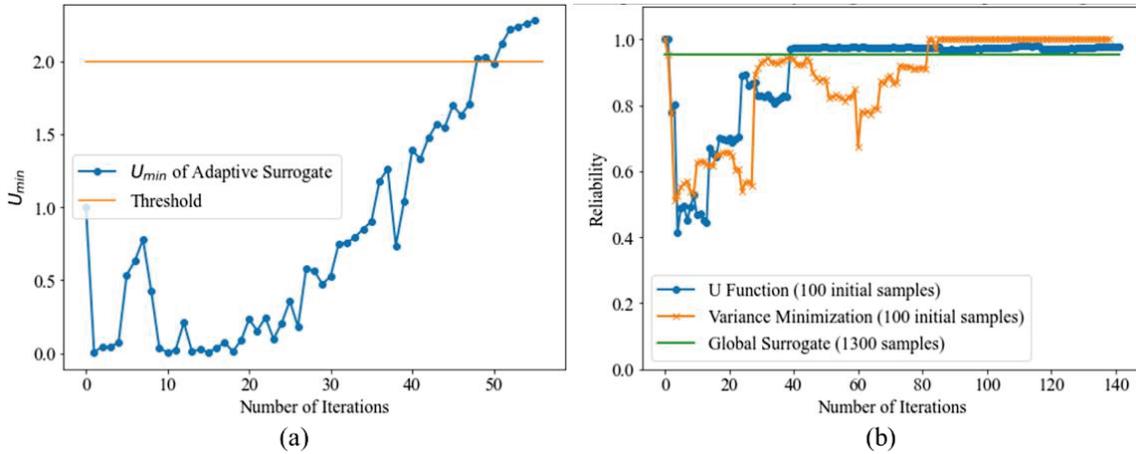


Figure 5.9 The results of adaptive GP and comparison with other methods

To obtain the DENA model for vertical acceleration, we use the training data collected from a deformable terrain of the target map. The terrain is a realization of a Gaussian random field. From the realization, we generate eight paths in which the data (heights and vertical acceleration) of seven paths are used to train the DENA model, and the data from another eighth path is for test. Figure 5.10 shows the 3-D terrain and paths.

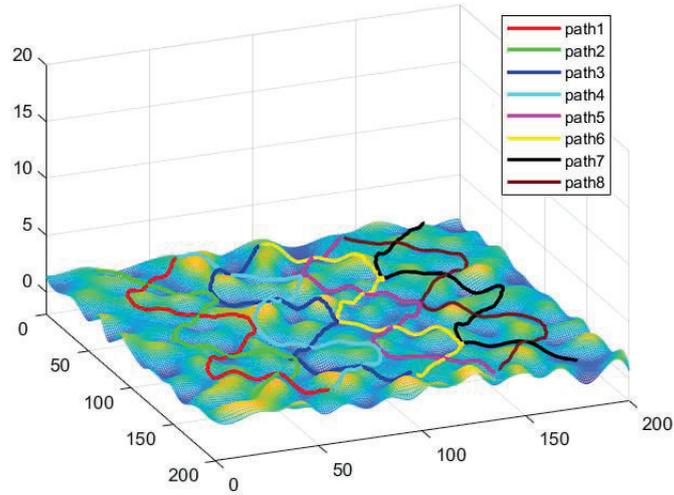


Figure 5.10 The height contour and generated eight paths

Using the method discussed in Sec. 5.2.3, the cluster number Q is determined to be two and the optimal number of lags for NARX models is identified as five. The test result of path eight is given in Figure 5.11. We compared the results of DENA and GP-NARX. It shows that DENA can capture the dynamic behavior of the vertical velocity better than GP-NARX.

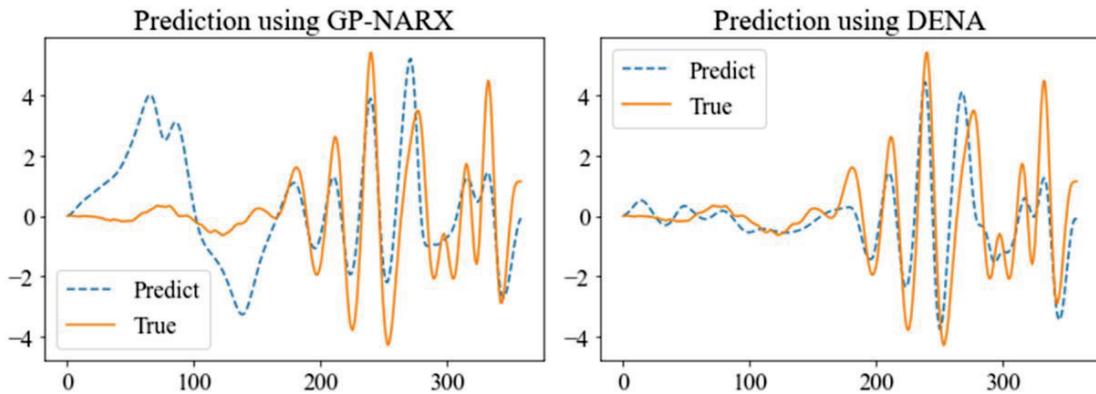


Figure 5.11 Comparison between DENA and GP-NARX

5.3.3. Results of Path Planning

After the surrogate models are constructed, we can obtain reliability in terms of maximum attainable speed and vertical acceleration for any given path. The RRT* and reliability constraints from surrogate models are combined for path planning.

We evaluate four different methods, which are RRT* with GO/NO-GO map (RRT*), RRT* with one reliability constraint (R-RRT*), MC-RRT*, and MC-RRT*-2. The same evaluation criteria are used for performance assessment, including the length of the generated path, reliability in terms of maximum attainable speed and vertical acceleration, and computation time. The reliability is evaluated using MCS with 5,000 samples. The detailed results for different methods are provided below. Figure 5.12 shows the paths obtained from different methods.

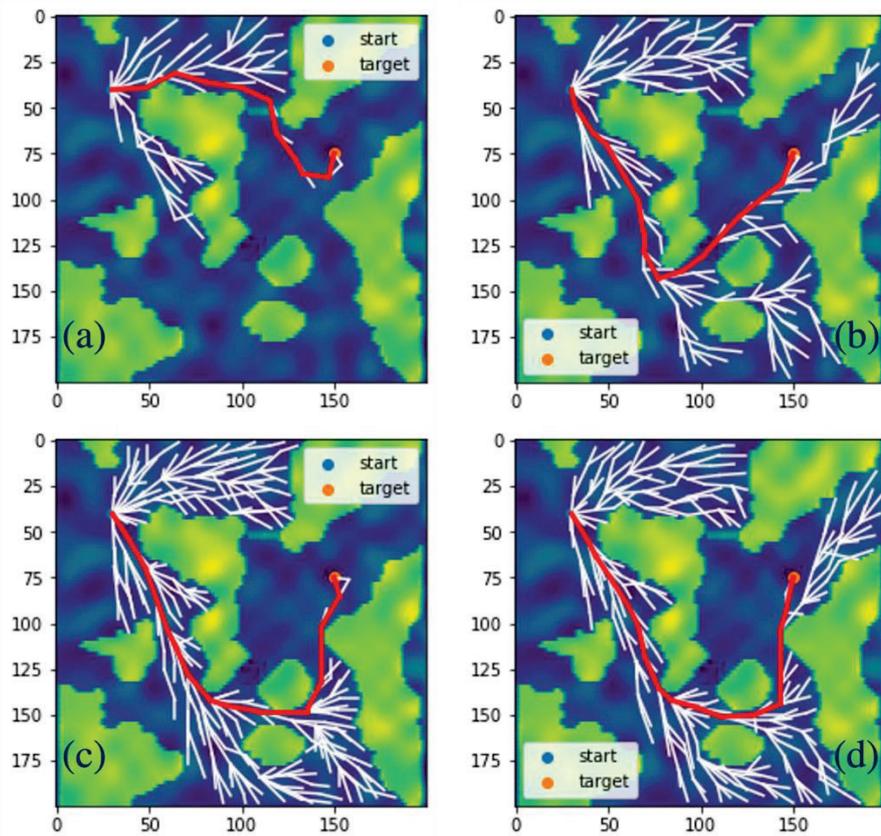


Figure 5.12 Path planning by different methods; (a) RRT*; (b) R-RRT*; (c) MC-RRT*; and (d) MC-RRT*-2.

From the figure, we can observe that RRT* has the shortest path as it does not consider any reliability constraints. MC-RRT* and MC-RRT*-2 have similar paths visually. R-RRT* considers the reliability in terms of maximum attainable speed. After reliability analysis, as shown in Figure 5.13, it is found that the AGV has a high chance of losing mobility if it crosses the bump as indicated in this figure. Since there is no reliability

consideration in RRT*, the identified path from RRT* (i.e., Figure 5.12(a)) has a high risk of failure. This is the reason why the path identified from R-RRT* (i.e., Figure 5.12(b)) bypasses the bump highlighted in Figure 5.13, and is longer than the path obtained from RRT* in Figure 5.12(a).

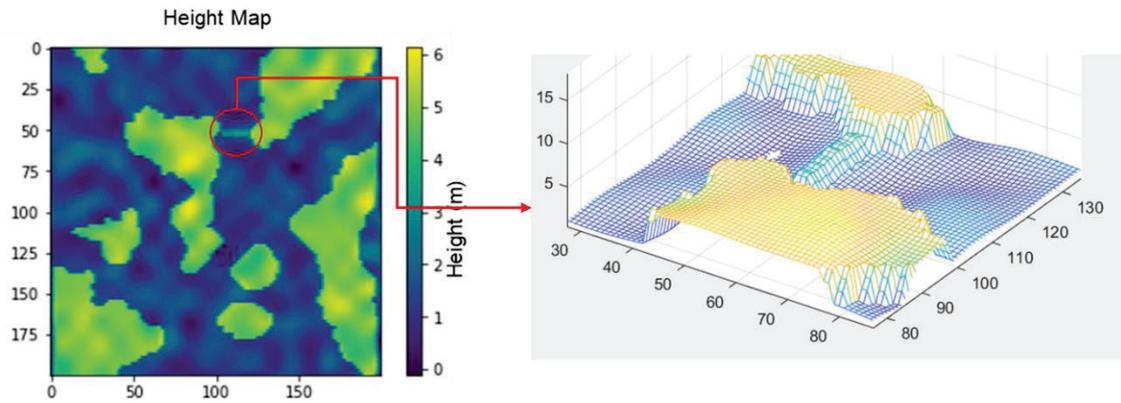


Figure 5.13 Illustration of a high-risk area of mobility loss

In Figure 5.12(c), MC-RRT* chooses to avoid the valley as illustrated in Figure 5.14, which results in a longer path. This is because MC-RRT* has more reliability considerations than R-RRT*. As shown in Figure 5.14, the terrain condition of the valley is very bumpy, which could cause high vertical accelerations of AGVs and damage the supplies or the vehicle. With more reliability considerations, MC-RRT* (i.e., Figure 5.12(c)) identifies a longer path than R-RRT* in Figure 5.12(b).

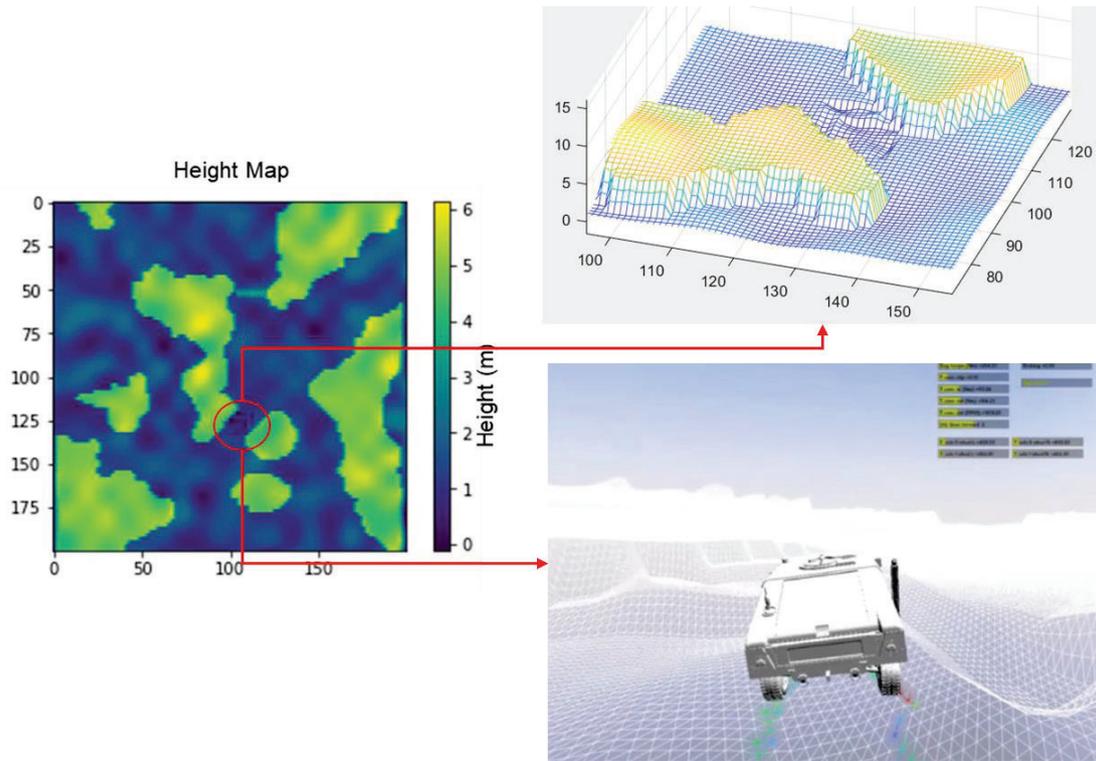


Figure 5.14 Illustration of bumpy terrain condition

To further compare different methods, MCS is used to evaluate the reliability. For illustration, we only provide the detailed simulation results of RRT*, which is shown in Figure 5.15. We assume the AGV loses mobility if the speed is lower than 2 m/s and the AVG or the inside supply could be damaged if the vertical acceleration exceeds 5 m/s². It is obvious that the AGV fails to maintain mobility for some simulations and the AGV or the inside supply is damaged for most of the simulations. The other methods are also evaluated using the same MCS samples, but their figures are not provided due to limited space.

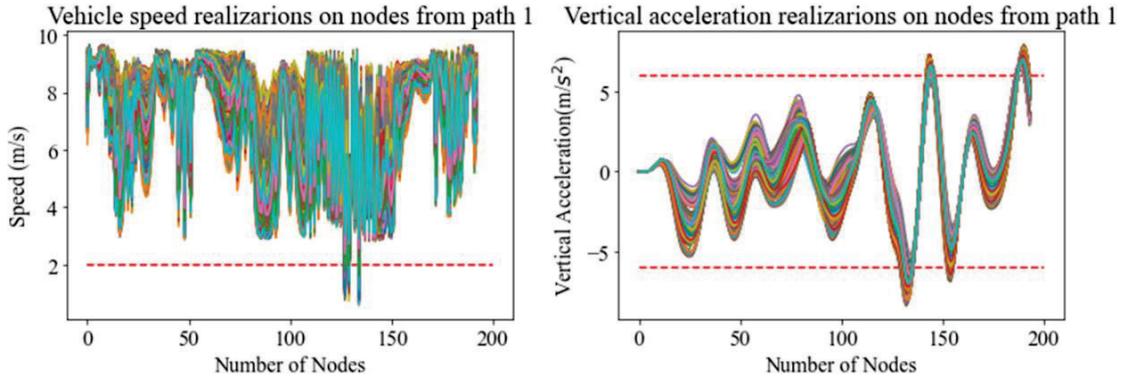


Figure 5.15 The MCS results of RRT* with 5,000 samples

Table 5.2 gives the results of different methods quantitatively. As expected, path identified from RRT* has the lowest reliability which is consistent with the results in Fig. 16, although it has the shortest path. R-RRT* has low reliability in the vertical acceleration and it is more efficient than MC-RRT* and MC-RRT*-2. MC-RRT*-2 improves the efficiency by decreasing computation time by 1.17 hours compared with MC-RRT* with only a bit of sacrifice in R_a . Both paths identified from MC-RRT* and MC-RRT*-2 satisfy the reliability requirement of 0.95.

Table 5.2 Comparison of different methods

Method	Length of the path	MMR_v	$MMRR_a$	Computation time (h)
RRT*	146.56	0.84	0	0.15
R-RRT*	225.24	0.96	0.58	1.12
MC-RRT*	246.71	0.99	0.98	3.37
MC-RRT*-S	248.31	0.99	0.97	2.5

5.4. Summary

This paper proposes a reliability-based global path planning approach for off-road autonomous vehicles (AGVs) using the rapidly-exploring random trees Star (RRT*) method while considering two reliability constraints. Due to prevalent uncertainty in the off-road environment, the high-fidelity simulation model is computationally too expensive to use for reliability analysis. To improve the efficiency, two surrogate models including an adaptive GP and a DENA-GP, are created based on the data from limited high-fidelity simulations for reliability analysis. We combine RRT* with the reliability constraints from

the surrogate models, which results in the proposed, MC-RRT* and MC-RRT*-2 path planning algorithms. The case study shows that both MC-RRT* and MC-RRT*-2 can find the target in the map with satisfying reliability. MC-RRT*-2 is more efficient than MC-RRT* as GO/NO-GO map can avoid infeasible locations before reliability analysis.

Even though MC-RRT*-2 is computationally cheaper than MC-RRT*, the required computational effort is too high for real-time path planning due to the consideration of reliability constraints. In our future work, we will further reduce the required computational time by integrating the proposed method with upcrossing rate-based reliability analysis method [113].

Chapter 6

Conclusion and Future Works

For off-road autonomous ground vehicles (AGV), the corresponding reliability quantification and path planning under uncertainty is an important topic that needs a lot of research. In the dissertation, the work provides a whole procedure of finding a reliable mission path under uncertain environment. The procedure includes mobility modeling, reliability analysis and path planning algorithms. Although for each separate part of the procedure, various approaches have been proposed as reviewed in Chapter 1, there are still challenges that needs to be addressed. Also, each part needs to be assembled together to complete the whole story.

As presented in Chapter 2, the focus is on the surrogate modeling of vehicle mobility with non-stationary dynamic behaviors. A dynamic ensemble framework of NARX models (DENA) and a dynamic ensemble framework of NARX models using a variable number of lags (DENA-DIFF) are proposed. Multiple NARX models are trained for different dynamic behaviors based on clustering of the training data. The NARX models are then assembled together dynamically over time to predict the dynamic prediction of AGV mobility over a long time period.

As presented in Chapter 3, a new simulation-based mission mobility reliability (MMR) analysis framework is developed for off-road ground vehicles. By using KL expansion for random realization of soil properties and adaptive sampling, the framework can reduce the computational effort of reliability prediction while maintain the accuracy. Also, a Bayesian updating scheme can update the reliability and predict future failure when new observations are given. The approach enables the off-road ground vehicle to proactively avoid rare

events of immobility during a mission, and thus guarantee the safety of the vehicle on the battlefield.

As presented in Chapter 4, a physics-based vehicle dynamics simulation model (Pychrono) is employed to predict vehicle mobility for any given terrain and soil conditions. Based on physics-based simulations, the vehicle SMR in operation is then analyzed using an adaptive surrogate modeling method to overcome the computational challenges in mobility reliability analysis by adaptively constructing a surrogate. Subsequently, a bio-inspired approach called Physarum-based algorithm is used in conjunction with a navigation mesh to identify an optimal path satisfying a specific mobility reliability requirement. The developed Physarum-based framework can handle both single-vehicle and multiple-vehicle scenarios with dynamic changes of the mission area.

As presented in Chapter 5, the approaches developed in Chapter 2 to 4 are ensembled together with RRT* algorithm to provide a reliability-based path planning algorithm with multiple mobility constraints (MC-RRT* and MC-RRT*-S). These algorithms remove the slope information which can only be obtained after the path is generated and uses the height information instead. The mission mobility reliability (MMR) is always maintained during the exploration steps of the RRT* algorithm which guarantee the overall mission reliability.

Due to the limit of the resources and effort, the following directions are worth exploring in the future if the current work needs to be further improved and expand.

- The mobility data is collected through the simulation model. If the data from a real testing vehicle can be obtained, the corresponding surrogate model will be more persuasive.
- The DENA can be expanded to ensemble different dynamic models such as LSTM and can also be improved by physic informed model.
- Reinforcement learning along with high-fidelity simulation can be used for path planning.

Bibliography

- [1] Ciobotaru, T., 2009, "Semi-empiric algorithm for assessment of the vehicle mobility," Leonardo electronic journal of practices and technologies, 8(15), pp. 19-30.
- [2] McCullough, M., and Haug, E., 1986, "Dynamics of high mobility track vehicles."
- [3] McCullough, M., Jayakumar, P., Dasch, J., and Gorsich, D., 2017, "The next generation NATO reference mobility model development," Journal of Terramechanics, 73, pp. 49-60.
- [4] Petrick, E., Janosi, Z., and Haley, P., 1981, "The use of the nato reference mobility model in military vehicle procurement," No. 0148-7191, SAE Technical Paper.
- [5] Singh, A., 2012, "Mobility of Military Vehicles at TARDEC," Army Tank Automotive Research Development and Engineering Center Warren MI.
- [6] Bradbury, M., Dasch, J., Gonzalez, R., Hodges, H., Jain, A., Iagnemma, K., Letherwood, M., McCullough, M., Priddy, J., And Wojtysiak, B., 2016, "Next-Generation NATO Reference Mobility Model (NG-NRMM)," Tank Automotive Research, Development and Engineering Center (TARDEC) Warren MI.
- [7] McCullough, M., Jayakumar, P., Dasch, J., and Gorsich, D., 2016, "Developing the Next Generation NATO Reference Mobility Model," US Army TARDEC Warren United States.
- [8] Laughery, S., Gerhart, G., and Goetz, R., 1990, "Bekker's terramechanics model for off-road vehicle research," Tacom Research Development And Engineering Center Warren MI.
- [9] Recuero, A., Serban, R., Peterson, B., Sugiyama, H., Jayakumar, P., and Negrut, D., 2017, "A high-fidelity approach for vehicle mobility simulation: Nonlinear finite element tires operating on granular material," Journal of Terramechanics, 72, pp. 39-54.
- [10] Serban, R., Olsen, N., Negrut, D., Recuero, A., and Jayakumar, P., 2017, "A co-simulation framework for high-performance, high-fidelity simulation of ground vehicle-terrain interaction," North Atlantic Treaty Organization, Brussels, Belgium, Report No. Avt-265.
- [11] Korlath, G., 2007, "Mobility analysis of off-road vehicles: Benefits for development, procurement and operation," Journal of Terramechanics, 44(5), pp. 383-393.

- [12] Hetherington, J., 2001, "The applicability of the MMP concept in specifying off-road mobility for wheeled and tracked vehicles," *Journal of terramechanics*, 38(2), pp. 63-70.
- [13] Choi, K., Jayakumar, P., Funk, M., Gaul, N., and Wasfy, T. M., 2019, "Framework of reliability-based stochastic mobility map for next generation nato reference mobility model," *Journal of Computational and Nonlinear Dynamics*, 14(2).
- [14] González, R., Jayakumar, P., and Iagnemma, K., 2017, "Stochastic mobility prediction of ground vehicles over large spatial regions: a geostatistical approach," *Autonomous Robots*, 41(2), pp. 311-331.
- [15] Gonzalez, R., Jayakumar, P., and Iagnemma, K., 2017, "Generation of stochastic mobility maps for large-scale route planning of ground vehicles: A case study," *Journal of Terramechanics*, 69, pp. 1-11.
- [16] Detweiler, Z. R., and Ferris, J. B., 2010, "Interpolation methods for high-fidelity three-dimensional terrain surfaces," *Journal of Terramechanics*, 47(4), pp. 209-217.
- [17] Hu, Z., Mourelatos, Z. P., Gorsich, D., Jayakumar, P., and Majcher, M., 2020, "Testing Design Optimization for Uncertainty Reduction in Generating Off-Road Mobility Map Using a Bayesian Approach," *Journal of Mechanical Design*, 142(2).
- [18] Lessem, A., Ahlvin, R., Mason, G., and Mlakar, P., 1992, "Stochastic Vehicle Mobility Forecasts Using the NATO Reference Mobility Model. Report 1. Basic Concepts and Procedures," Army Engineer Waterways Experiment Station Vicksburg Ms Geotechnical Lab.
- [19] Lessem, A., Mason, G., and Ahlvin, R., 1996, "Stochastic vehicle mobility forecasts using the NATO reference mobility model," *Journal of Terramechanics*, 33(6), pp. 273-280.
- [20] Priddy, J. D., 1995, "Stochastic Vehicle Mobility Forecasts Using the NATO Reference Mobility Model. Report 3. Database Development for Statistical Analysis of the NRMM II Cross-Country Traction Empirical Relationships," Army Engineer Waterways Experiment Station Vicksburg Ms Geotechnical Lab.
- [21] Haug, E. J., Adkins, F. A., and Coroian, D., 1998, "Domains of mobility for a planar body moving among obstacles," *Journal of Mechanical Design*, 120(3), pp. 462-467.
- [22] Choi, K., Jayakumar, P., Funk, M., Gaul, N., and Wasfy, T. M., 2019, "Framework of reliability-based stochastic mobility map for next generation nato reference mobility model," *ASME J Comput Nonlin Dyn*, 14(2).
- [23] Jayakumar, P., and Mechergui, D., 2019, "Efficient generation of accurate mobility maps using machine learning algorithms," US Army TARDEC Warren United States.

- [24] Yang, D., 1993, "Collision-free path planning by using nonperiodic B-spline curves," *Journal of Mechanical Design*, 115(3), pp. 679-684.
- [25] Melchior, N. A., and Simmons, R., "Particle RRT for path planning with uncertainty," *Proc. Proceedings 2007 IEEE International Conference on Robotics and Automation*, IEEE, pp. 1617-1624.
- [26] Liu, Q., Zhao, L., Tan, Z., and Chen, W., 2017, "Global path planning for autonomous vehicles in off-road environment via an A-star algorithm," *International Journal of Vehicle Autonomous Systems*, 13(4), pp. 330-339.
- [27] Sun, W., van den Berg, J., and Alterovitz, R., 2016, "Stochastic extended LQR for optimization-based motion planning under uncertainty," *IEEE Transactions on Automation Science and Engineering*, 13(2), pp. 437-447.
- [28] Van Den Berg, J., Abbeel, P., and Goldberg, K., 2011, "LQG-MP: Optimized path planning for robots with motion uncertainty and imperfect state information," *The International Journal of Robotics Research*, 30(7), pp. 895-913.
- [29] Zhang, B., Mao, Z., Liu, W., and Liu, J., 2015, "Geometric reinforcement learning for path planning of UAVs," *Journal of Intelligent & Robotic Systems*, 77(2), pp. 391-409.
- [30] Chao, N., Liu, Y.-k., Xia, H., Ayodeji, A., and Bai, L., 2018, "Grid-based RRT* for minimum dose walking path-planning in complex radioactive environments," *Annals of Nuclear Energy*, 115, pp. 73-82.
- [31] Jiang, C., Hu, Z., Mourelatos, Z. P., Gorsich, D., Jayakumar, P., Fu, Y., and Majcher, M., 2021, "R2-RRT*: reliability-based robust mission planning of off-road autonomous ground vehicle under uncertain terrain environment," *IEEE Transactions on Automation Science and Engineering*, 19(2), pp. 1030-1046.
- [32] Liu, Y., Barthlow, D., Mourelatos, Z. P., Zeng, J., Gorsich, D., Singh, A., and Hu, Z., 2022, "Mobility Prediction of Off-Road Ground Vehicles Using a Dynamic Ensemble of NARX Models," *Journal of Mechanical Design*, 144(9), p. 091709.
- [33] Close, C. M., Frederick, D. K., and Newell, J. C., 2001, *Modeling and analysis of dynamic systems*, John Wiley & Sons.
- [34] Brunton, S. L., and Kutz, J. N., 2019, *Data-driven science and engineering: Machine learning, dynamical systems, and control*, Cambridge University Press.
- [35] Tasora, A., Serban, R., Mazhar, H., Pazouki, A., Melanz, D., Fleischmann, J., Taylor, M., Sugiyama, H., and Negrut, D., "Chrono: An open source multi-physics dynamics engine," *Proc. International Conference on High Performance Computing in Science and Engineering*, Springer, pp. 19-49.

- [36] Liu, Y., Jiang, C., Zhang, X., Mourelatos, Z. P., Barthlow, D., Gorsich, D., Singh, A., and Hu, Z., 2021, "Reliability-Based Multi-Vehicle Path Planning Under Uncertainty Using a Bio-Inspired Approach," *Journal of Mechanical Design*, pp. 1-44.
- [37] Mazhar, H., Heyn, T., Pazouki, A., Melanz, D., Seidl, A., Bartholomew, A., Tasora, A., and Negrut, D., 2013, "Chrono: a parallel multi-physics library for rigid-body, flexible-body, and fluid dynamics," *Mechanical Sciences*, 4(1), pp. 49-64.
- [38] Worden, K., Becker, W., Rogers, T., and Cross, E., 2018, "On the confidence bounds of Gaussian process NARX models and their higher-order frequency response functions," *Mechanical Systems and Signal Processing*, 104, pp. 188-223.
- [39] Khalid, A., Sundararajan, A., and Sarwat, A. I., "An arima-narx model to predict lion state of charge for unknown charge/discharge rates," *Proc. 2019 IEEE Transportation Electrification Conference (ITEC-India)*, IEEE, pp. 1-4.
- [40] Li, M., and Wang, Z., 2022, "LSTM-augmented deep networks for time-variant reliability assessment of dynamic systems," *Reliability Engineering & System Safety*, 217, p. 108014.
- [41] Siegelmann, H. T., Horne, B. G., and Giles, C. L., 1997, "Computational capabilities of recurrent NARX neural networks," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 27(2), pp. 208-215.
- [42] Sum, J., Kan, W.-K., and Young, G. H., 1999, "A Note on the Equivalence of NARX and RNN," *Neural computing & applications*, 8(1), pp. 33-39.
- [43] Menezes Jr, J. M. P., and Barreto, G. A., 2008, "Long-term time series prediction with the NARX network: An empirical evaluation," *Neurocomputing*, 71(16-18), pp. 3335-3343.
- [44] Olah, C., "Understanding LSTM Networks."
- [45] Zhang, X., and Mahadevan, S., 2020, "Bayesian neural networks for flight trajectory prediction and safety assessment," *Decision Support Systems*, 131, p. 113246.
- [46] Xiong, Y., Chen, W., Apley, D., and Ding, X., 2007, "A non-stationary covariance-based Kriging method for metamodelling in engineering design," *International Journal for Numerical Methods in Engineering*, 71(6), pp. 733-756.
- [47] Chung, J., Ahn, S., and Bengio, Y., 2016, "Hierarchical multiscale recurrent neural networks," *arXiv preprint arXiv:1609.01704*.
- [48] Hu, Z., and Mourelatos, Z., 2018, "Efficient Global Surrogate Modeling Based on Multi-Layer Sampling," *SAE International Journal of Materials and Manufacturing*, 11(4), pp. 385-400.

- [49] Ghazali, R., Hussain, A. J., Nawi, N. M., and Mohamad, B., 2009, "Non-stationary and stationary prediction of financial time series using dynamic ridge polynomial neural network," *Neurocomputing*, 72(10-12), pp. 2359-2367.
- [50] Reynolds, D. A., 2009, "Gaussian mixture models," *Encyclopedia of biometrics*, 741(659-663).
- [51] Papadimitriou, D. I., Mourelatos, Z. P., and Hu, Z., 2019, "Reliability analysis using second-order saddlepoint approximation and mixture distributions," *Journal of Mechanical Design*, 141(2).
- [52] Murty, M. N., and Devi, V. S., 2015, *Introduction to pattern recognition and machine learning*, World Scientific.
- [53] Blei, D. M., Griffiths, T. L., Jordan, M. I., and Tenenbaum, J. B., "Hierarchical topic models and the nested Chinese restaurant process," *Proc. NIPS*.
- [54] Escobar, M. D., and West, M., 1995, "Bayesian density estimation and inference using mixtures," *Journal of the american statistical association*, 90(430), pp. 577-588.
- [55] Murphy, K. P., 2012, *Machine learning: a probabilistic perspective*, MIT press.
- [56] Liu, Y., Jiang, C., Mourelatos, Z. P., Gorsich, D., Jayakumar, P., Fu, Y., Majcher, M., and Hu, Z., 2021, "Simulation-based mission mobility reliability analysis of off-road ground vehicles," *Journal of Mechanical Design*, 143(3).
- [57] Maclaurin, E., 1997, "Proposed revisions to MMP based on the results of tractive performance trials with single pneumatic tyres and a modular track system," DERA/LS4/TR970122/1.0. Defence Evaluation and Research Agency.
- [58] Priddy, J. D., 1999, "Improving the Traction Prediction Capabilities in the NATO Reference Mobility Model (NRMM)," Army Engineer Waterways Experiment Station Vicksburg Ms Geotechnical Lab.
- [59] Jones, R. A., Price, S. J., and Ahlvin, R. B., 2004, "Mission level mobility analysis of the US Marine Corps HIMARS vehicles," Engineer Research And Development Center Vicksburg Ms Geotechnical and Structures Lab.
- [60] Shoop, S., Kestler, K., and Haehnel, R., 2006, "Finite element modeling of tires on snow," *Tire science and technology*, 34(1), pp. 2-37.
- [61] Huang, S., Mahadevan, S., and Rebba, R., 2007, "Collocation-based stochastic finite element analysis for random field problems," *Probabilistic engineering mechanics*, 22(2), pp. 194-205.
- [62] Gneiting, T., 2002, "Compactly supported correlation functions," *Journal of*

Multivariate Analysis, 83(2), pp. 493-508.

[63] Hu, Z., and Du, X., 2013, "Time-dependent reliability analysis with joint upcrossing rates," *Structural and Multidisciplinary Optimization*, 48(5), pp. 893-907.

[64] Zhang, D., and Han, X., 2020, "Kinematic reliability analysis of robotic manipulator," *Journal of Mechanical Design*, 142(4).

[65] Du, X., and Hu, Z., 2012, "First Order Reliability Method With Truncated Random Variables," *Journal of Mechanical Design*, 134(9).

[66] Hu, Z., and Du, X., 2014, "First order reliability method for time-variant problems using series expansions," *Structural and Multidisciplinary Optimization*, 51(1), pp. 1-21.

[67] Mourelatos, Z. P., Majcher, M., Pandey, V., and Baseski, I., 2015, "Time-dependent reliability analysis using the total probability theorem," *Journal of Mechanical Design*, 137(3).

[68] Hu, Z., and Du, X., 2013, "A sampling approach to extreme value distribution for time-dependent reliability analysis," *Journal of Mechanical Design*, 135(7), p. 071003.

[69] Liu, Y., Zhao, Y., Hu, Z., Mourelatos, Z. P., and Papadimitriou, D., 2019, "Collision-Avoidance Reliability Analysis of Automated Vehicle Based on Adaptive Surrogate Modeling," *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part B: Mechanical Engineering*, 5(2).

[70] Jiang, C., Qiu, H., Gao, L., Wang, D., Yang, Z., and Chen, L., 2020, "Real-time estimation error-guided active learning Kriging method for time-dependent reliability analysis," *Applied Mathematical Modelling*, 77, pp. 82-98.

[71] Hu, Z., and Du, X., 2015, "Mixed Efficient Global Optimization for Time-Dependent Reliability Analysis," *Journal of Mechanical Design*, 137(5), p. 051401.

[72] Wang, Z., and Wang, P., 2013, "A new approach for reliability analysis with time-variant performance characteristics," *Reliability Engineering & System Safety*, 115, pp. 70-81.

[73] Zhang, D., Han, X., Jiang, C., Liu, J., and Li, Q., 2017, "Time-dependent reliability analysis through response surface method," *Journal of Mechanical Design*, 139(4).

[74] Hu, Z., and Mahadevan, S., 2016, "A single-loop kriging surrogate modeling for time-dependent reliability analysis," *Journal of Mechanical Design*, 138(6).

[75] Stein, M., 1987, "Large sample properties of simulations using Latin hypercube sampling," *Technometrics*, 29(2), pp. 143-151.

- [76] Simpson, T. W., Mauery, T. M., Korte, J. J., and Mistree, F., 2001, "Kriging models for global approximation in simulation-based multidisciplinary design optimization," *AIAA journal*, 39(12), pp. 2233-2241.
- [77] Arulampalam, M. S., Maskell, S., Gordon, N., and Clapp, T., 2002, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Transactions on signal processing*, 50(2), pp. 174-188.
- [78] Wan, E. A., and Van Der Merwe, R., "The unscented Kalman filter for nonlinear estimation," *Proc. Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No. 00EX373)*, Ieee, pp. 153-158.
- [79] Clausen, A., Wang, F., Jensen, J. S., Sigmund, O., and Lewis, J. A., 2015, "Topology optimized architectures with programmable Poisson's ratio over large deformations," *Advanced Materials*, 27(37), pp. 5523-5527.
- [80] Larsen, U. D., Sigmund, O., and Bouwsta, S., 1997, "Design and fabrication of compliant micromechanisms and structures with negative Poisson's ratio," *Journal of microelectromechanical systems*, 6(2), pp. 99-106.
- [81] Liu, Y., Jiang, C., Zhang, X., Mourelatos, Z. P., Barthlow, D., Gorsich, D., Singh, A., and Hu, Z., 2022, "Reliability-Based Multivehicle Path Planning Under Uncertainty Using a Bio-Inspired Approach," *Journal of Mechanical Design*, 144(9), p. 091701.
- [82] Serban, R., Olsen, N., Negrut, D., Recuero, A., and Jayakumar, P., "A co-simulation framework for high-performance, high-fidelity simulation of ground vehicle-terrain interaction," *Proc. Conference: NATO AVT-265 Specialists' Meeting, Vilnius, Lithuania (May 2017)*, p. 24.
- [83] Rubinstein, D., and Hitron, R., 2004, "A detailed multi-body model for dynamic simulation of off-road tracked vehicles," *Journal of Terramechanics*, 41(2-3), pp. 163-173.
- [84] Liang, C.-Y., Allen, R. W., Rosenthal, T. J., Chrstos, J. P., and Nunez, P., 2004, "Tire modeling for off-road vehicle simulation," *SAE transactions*, pp. 1063-1073.
- [85] Krenn, R., and Gibbesch, A., 2011, "Soft soil contact modeling technique for multi-body system simulation," *Trends in computational contact mechanics*, pp. 135-155.
- [86] Senatore, C., and Sandu, C., 2011, "Off-road tire modeling and the multi-pass effect for vehicle dynamics simulation," *Journal of Terramechanics*, 48(4), pp. 265-276.
- [87] Serban, R., Taylor, M., Negrut, D., and Tasora, A., 2019, "Chrono:: Vehicle: template-based ground vehicle modelling and simulation," *International Journal of Vehicle Performance*, 5(1), pp. 18-39.

- [88] Gallina, A., Krenn, R., Scharringhausen, M., Uhl, T., and Schäfer, B., 2014, "Parameter identification of a planetary rover wheel–soil contact model via a Bayesian approach," *Journal of Field Robotics*, 31(1), pp. 161-175.
- [89] Hu, Z., Mahadevan, S. J. S., and Optimization, M., 2016, "Global sensitivity analysis-enhanced surrogate (GSAS) modeling for reliability analysis," 53(3), pp. 501-521.
- [90] Hu, Z., and Mahadevan, S., 2018, "Adaptive surrogate modeling for time-dependent multidisciplinary reliability analysis," *Journal of Mechanical Design*, 140(2).
- [91] Kaminsky, A. L., Wang, Y., and Pant, K., 2021, "An Efficient Batch K-Fold Cross-Validation Voronoi Adaptive Sampling Technique for Global Surrogate Modeling," *Journal of Mechanical Design*, 143(1), p. 011706.
- [92] Zhang, Y., Kim, N. H., and Haftka, R. T., 2020, "General-surrogate adaptive sampling using interquartile range for design space exploration," *Journal of Mechanical Design*, 142(5).
- [93] Echard, B., Gayton, N., and Lemaire, M., 2011, "AK-MCS: an active learning reliability method combining Kriging and Monte Carlo simulation," *Structural Safety*, 33(2), pp. 145-154.
- [94] Fauriat, W., and Gayton, N., 2014, "AK-SYS: an adaptation of the AK-MCS method for system reliability," *Reliability Engineering & System Safety*, 123, pp. 137-144.
- [95] Deng, Y., Chen, Y., Zhang, Y., and Mahadevan, S., 2012, "Fuzzy Dijkstra algorithm for shortest path problem under uncertain environment," *Applied Soft Computing*, 12(3), pp. 1231-1237.
- [96] Tero, A., Kobayashi, R., and Nakagaki, T., 2006, "Physarum solver: A biologically inspired method of road-network navigation," *Physica A: Statistical Mechanics and its Applications*, 363(1), pp. 115-119.
- [97] Xu, X., Huang, M., and Zou, K., 2011, "Automatic generated navigation mesh algorithm on 3D game scene," *Procedia Engineering*, 15, pp. 3215-3219.
- [98] Hartley, T. P., and Mehdi, Q. H., "In-game adaptation of a navigation mesh cell path," *Proc. 2012 17th International Conference on Computer Games (CGAMES)*, IEEE, pp. 230-236.
- [99] "Triangle 20200404 documentation," <https://rufat.be/triangle/>.
- [100] Shewchuk, J. R., 2008, "A two-dimensional quality mesh generator and Delaunay triangulator," *Computer Science Division University of California at Berkeley, Berkeley, California*, <http://www.cs.cmu.edu/quake/triangle.html>, pp. 94720-91776.

- [101] Nakagaki, T., Yamada, H., and Tóth, Á., 2000, "Maze-solving by an amoeboid organism," *Nature*, 407(6803), pp. 470-470.
- [102] Zhang, X., Chan, F. T., Yang, H., and Deng, Y., 2017, "An adaptive amoeba algorithm for shortest path tree computation in dynamic graphs," *Information Sciences*, 405, pp. 123-140.
- [103] "PYCHRONO: A Multi-physics Simulation Engine," <https://projectchrono.org/pychrono/>.
- [104] Noreen, I., Khan, A., and Habib, Z., 2016, "Optimal path planning using RRT* based approaches: a survey and future directions," *International Journal of Advanced Computer Science and Applications*, 7(11).
- [105] Yin, J., Shen, D., Du, X., and Li, L., 2022, "Distributed Stochastic Model Predictive Control With Taguchi's Robustness for Vehicle Platooning," *IEEE Transactions on Intelligent Transportation Systems*.
- [106] Yin, J., and Du, X., 2022, "Active learning with generalized sliced inverse regression for high-dimensional reliability analysis," *Structural Safety*, 94, p. 102151.
- [107] Wang, D., Qiu, H., Gao, L., and Jiang, C., 2021, "A single-loop Kriging coupled with subset simulation for time-dependent reliability analysis," *Reliability Engineering & System Safety*, 216, p. 107931.
- [108] Jiang, C., Qiu, H., Gao, L., Wang, D., Yang, Z., and Chen, L., 2020, "EEK-SYS: system reliability analysis through estimation error-guided adaptive Kriging approximation of multiple limit state surfaces," *Reliability Engineering & System Safety*, 198, p. 106906.
- [109] Yang, M., Zhang, D., Jiang, C., Han, X., and Li, Q., 2021, "A hybrid adaptive Kriging-based single loop approach for complex reliability-based design optimization problems," *Reliability Engineering & System Safety*, 215, p. 107736.
- [110] Meng, Z., Zhang, Z., Zhang, D., and Yang, D., 2019, "An active learning method combining Kriging and accelerated chaotic single loop approach (AK-ACSLA) for reliability-based design optimization," *Computer Methods in Applied Mechanics and Engineering*, 357, p. 112570.
- [111] McLachlan, G. J., and Basford, K. E., 1988, *Mixture models: Inference and applications to clustering*, M. Dekker New York.
- [112] Kim, T., and Song, J., 2018, "Generalized reliability importance measure (GRIM) using Gaussian mixture," *Reliability Engineering & System Safety*, 173, pp. 105-115.
- [113] Jiang, C., Liu, Y., Mourelatos, Z. P., Gorsich, D., Fu, Y., and Hu, Z., 2022, "Efficient

Reliability-Based Mission Planning of Off-Road Autonomous Ground Vehicles Using an Outcrossing Approach," *Journal of Mechanical Design*, 144(4).