DEVELOPMENT OF ROBOTIC SYSTEMS FOR BRIDGE DECK INSPECTION AND REHABILITATION

by

FEI LIU

A dissertation submitted to the

School of Graduate Studies

Rutgers, The State University of New Jersey

In partial fulfillment of the requirements

For the degree of

Doctor of Philosophy

Graduate Program in Mechanical And Aerospace Engineering

Written under the direction of

Jingang Yi

And approved by

____

____

____

____

New Brunswick, New Jersey

May, 2019

ABSTRACT OF THE DISSERTATION

Development of Robotic Systems for Bridge Deck Inspection and Rehabilitation

By FEI LIU

Dissertation Director:

Jingang Yi

The condition of civil infrastructure such as bridges is of utmost importance for the safety of traveling public and sustainability of the economic activity. The bridge decks deteriorate faster than other bridge components due to their direct exposure to traffic and environmental loads. Effective health monitoring, maintenance, repair, rehabilitation and replacement of the deteriorating civil infrastructure components are necessary to ensure the transportation safety. Current assessment of concrete bridge decks still relies on visual inspection and use of simple nondestructive and destructive evaluations which are not capable to detect defect in early stage. More advanced nondestructive evaluation (NDE) technologies, which can provide more comprehensive assessment, are not used on a regular basis due to lower speed of manual data collection. On the other hand, the current practice of repair of bridge deck only happen at the late stage resulting in

extremely high cost. Also, there is currently no available system to treat early stage defect such as delamination and internal cracking.

The goal of this dissertation is to provide a integrated solution for efficient and effective bridge deck inspection and maintenance with emphasis on five interlaced topics: (i) development of an autonomous bridge deck inspection platform, (ii) automated data processing for bridge deck image data, (iii) development of an autonomous bridge deck rehabilitation platform focusing on early stage delamination, (iv) modeling of the bit-concrete interaction for the rehabilitation procedure, (v) strategies for simultaneously deployment of the bridge deck inspection and rehabilitation robots. In the first part, we present a robotic system for bridge deck data collection. The robot integrates multiple NDE techniques that enable the characterization of three most common deterioration types in concrete bridge decks: rebar corrosion, delamination, and concrete degradation. The autonomous navigation and precise data registration are enable by a robust localization system that fusing two GPS and wheel odometry through Extended Kalman Filter (EKF). In the second part, we present a new automated image mosaicing system for bridge deck surface reconstruction. By combining the navigation data and feature-based image registration in the graph optimization framework, our proposed approach inherits the drift-less nature from GPS while still maintains local accuracy of feature-based image registration. In the third part, we develop a robotic system for non-destructive rehabilitation (NDR) targeting the early delamination on bridges such as internal cracking. The NDR system is composed of an omni-directional mobile base, a 5 degree of freedom manipulator and a custom-made end-effector that performs the rehabilitation procedures including drilling and filling. Motion planning algorithm is developed for the mobile manipulator to perform GPS guided rehabilitation procedures.In the fourth part, we present

a dynamic model of pure percussive drilling for autonomous robotic rehabilitation for concrete bridge decks. We derive the minimum static force to enable effective percussive drilling which provide us guidance for the mobile manipulator drilling in the previous part. A dry friction-based pure percussive drilling model is then presented to describe the drilling process characteristics and to capture the influence of drilling conditions and parameters on the penetration rate. In the fifth part, we present the strategies to simultaneously deploy the inspection and rehabilitation robot on the bridge decks. We adopt the Gaussian process approach to generate the global and local delamination map online. The inspection robot dynamically determine the step size based on the local prediction uncertainty that accelerate the data collection. Moreover, we design a target planning algorithm based on the global delamination map for the rehabilitation robot to choose the next target to repair. The algorithms proposed are validated through a multi-robot simulation system that could take real bridge inspection data.

Acknowledgements

First, I am grateful to my Ph.D. adviser Jingang Yi for his support and guidance. Prof. Jingang Yi helped me shaping the research direction as a graduate student and shaping my skills for my future career paths. Besides the knowledge I learned from him, I have always been encouraged by his enthusiasm about research work and enlightened by his vision on research topics.

I would like to thank my committee members: Professor Nenad Gucunski, Professor Qingze Zou and Professor Aaron Mazzeo for reviewing my work and providing valuable feedback to my work. I would like to thank Professor Nenad Gucunski in particular for his guidance through many in-depth discussions and kind support along the journey.

I am also grateful to all my lab mates at Robotics, Automation and Mechatronics Laboratory, including Mitja, Kuo, PengCheng, Kaiyan, Yizhai, Chaoke, Moiz, Yongbin, Siyu and others for all helpful discussions and sharing their knowledge throughout these years at Rutgers.

Finally, I would like to thank my family, especially to my wife Shanshan for her encouragement and support and my daughter Serenna for being my lucky star. Special thanks goes to my parents for constantly believing in my potential and unconditional love along the long journey.

Dedication

This dissertation is dedicated to my family

Table of Contents

List of Tables

List of Figures

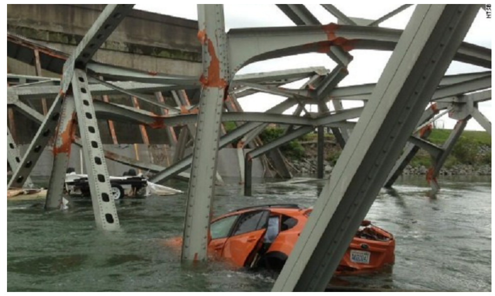
Chapter 1
Introduction

1.1 Motivation

One of the biggest challenges the United States faces today is infrastructure like-bridges inspection and maintenance. The threat to safety of aging bridges has been recognized as a growing problem of national concern to the general public. There are currently more than 600,000 bridges in the U.S. [5] with average age of 45 years and four in 10 are 50 years or older. The condition of those bridges are critical for the safety of the traveling public and economic vitality of the country. According to the National Bridge Inventory there are about 150,000 bridges through the U.S. that are structurally deficient or functionally obsolete due to various mechanical and weather conditions, inadequate maintenance, and deficiencies in inspection and evaluation [37], and this number is growing. Numerous bridges collapsed recently have raised a strong call for efficient bridge inspection and evaluation [75]. The cost of maintenance and rehabilitation of the deteriorating bridges is immense. The cost of repairing and replacing deteriorating highway bridges in U.S. was estimated to be $123 billions in 2017 [5].

Concrete bridge deck deteriorate faster than other bridge components due to their direct exposure to environment and traffic loads. Between 50 and 85% of bridge maintenance funds are spent to repair or replace portions of the nation's

(a)



(b)

Figure 1.1: Bridge collapse accident: (a) At I-35W Mississippi River bridge, Minnesota, 2007; (b) At I-5 Skagit River Bridge, Washington, 2013[75]

2.8 billion square feet of bridge decks [124]. Conservative estimate is that more than $5 billion is spent annually to maintain, repair and replace bridge decks. Therefore, condition monitoring and timely implementation of maintenance and rehabilitation procedures are needed to reduce future costs associated with bridge management.

More economical management of bridges can be achieved through early problem detection and mitigation. The most common cause of deterioration is corrosion that will typically lead to concrete delamination and spalling [51]. As shown in Figure 1.2. the deterioration start from rebar corrosion which could due to reasons such as moisture penetration. The oxide from the corrosion lead to expansion and start causing small crack in the neighbor region around the rebar area. The repeated traffic load further expand the cracks and form more severe delaminations. Only at the late stage the delamination develop into visually observable defects such as spalling. The ability to detect and fix deterioration before the late stage is crucial in fighting the hidden risk and avoiding hectic cost in bridge repair in the late stage.

Non-Destructive Evaluation (NDE) and Non-Destructive Rehabilitation (NDR) has emerged as a promising solution for the early bridge deck defect detection and

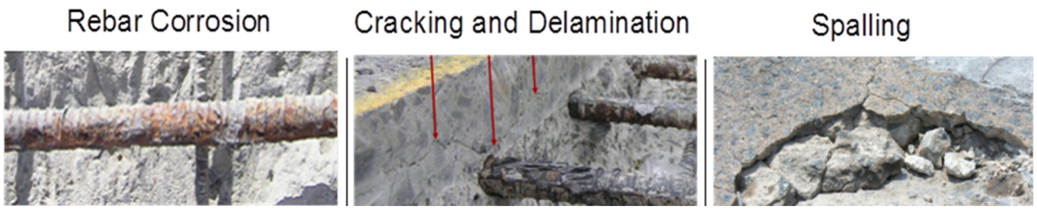


Figure 1.2: Typical concrete bridge deck deterioration and damage: (left) Rebar corrosion; (middle) Delamination; (right) Deck spalling.[51]

mitigation. NDE approach has been proven in the last decade as it could provide much more accurate assessment for the bridge deck condition [52]. However, most of NDE and NDR techniques are still used as manual through a group of engineers in a closed section of road. Those practices can't meet the increasing demand for frequent bridge deck inspection and rehabilitation and therefore limit the deployment of NDE techniques. Despite the manual

The goal of this dissertation is to address these challenges with five intertwined research directions. More specifically, this dissertation focuses on: (i) the development of autonomous bridge deck inspection robot; (ii) automated data processing of large volume of inspection data; (iii) the development of autonomous bridge deck rehabilitation robot; (iv) understanding percussive drilling process for effective rehabilitation (v) the development of strategies to enable simultaneously deployment of the bridge deck inspection and rehabilitation robots.

1.2 Background

The autonomous bridge deck inspection and rehabilitation is a novel field that brings the innovative solution from different fields. Each of these five research topics will be discussed and covered individually in the following chapters. Here we introduce the background of the research and the related work of each topic.

1.2.1 Autonomous bridge deck inspection

As the most weathered part, the bridge deck deteriorate over time as a result of excessive loading, environmental exposure, material aging and inadequate maintenance. Therefore, bridge deck inspection is conducted to identify the bridge deterioration condition and further facilitate the appropriate maintenance or rehabilitation procedures. The current practice for concrete bridge deck evaluation in the large part still relies on human visual inspection or use simple evaluation tools. Figure 1.3 shows common bridge evaluation practice such as visual inspection, chain dragging and hammer sounding. As pointed out in [48], there are justified reservation regarding the accuracy and objectivity of those practices. Non-Destructive Evaluation (NDE) has emerged as a promising approach in the last decade as it could provide much more accurate assessment for the bridge deck condition[52]. There are a number of the NDE technologies that have been currently used for bridge health evaluation, such as ground penetrating radar (GPR)[28], impact echo[54], electrical resistivity[6] and high-definition camera [116]. Each NDE techniques has its strength and limitation and is best suited for characterization particular defect type. During evaluation, each NDE technique is conducted manually through a group of engineers in a closed section of road as showed in Figure 1.4. The manual inspection procedure is not only time-consuming causing extended slowdown for the traffic flow, but also pose potential safety risk for the human inspectors. Those drawbacks further limited the extensive deployment of NDE method for bridge inspection.

To overcome the limitation of manual NDE, automated bridge deck inspection become a promising approach in civil infrastructure application. Automated bridge inspection would be efficient and reliable in the data collection and safe for the inspector. There are several attempts to bring the automation and robotics into the bridge deck inspection. German Federal Institute for Material Research

Figure 1.3: State of practice in bridge deck inspection: (left) Visual inspection; (middle) Chain dragging; (right) Hammer sounding

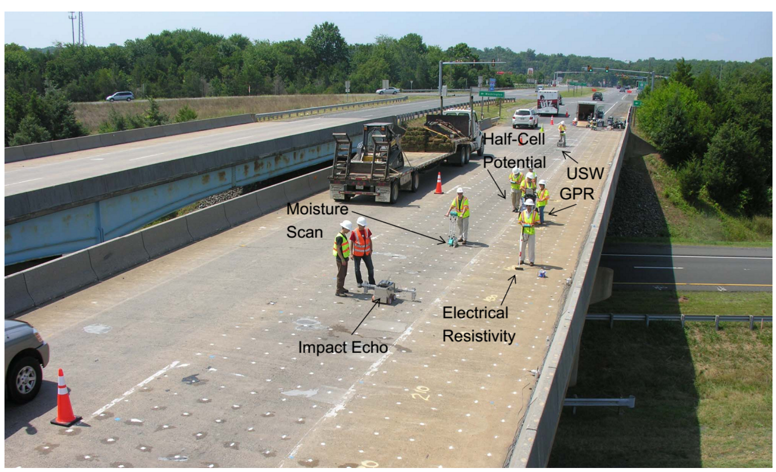


Figure 1.4: Manual NDE bridge deck data collection through a group of engineers

and Testing (BAM) have attempt to automate data collection through developing NDT-Stepper [128]. The NDT-Stepper is an automated cart that equipped with single pneumatically actuated impact echo and ultrasonic probes. It moves in constant increments at the speed of 2–3 m/min and collect the data at each stop. BAM has later developed a robotic system BETOSCAN based on the widely used Mobile Robot Pioneer platform for inspection of reinforced concrete slabs [129, 117]. BETOSCAN implements multiple NDE techniques including ultrasonic, potential mapping, microwaves, cover meter and thermometers, therefore could scan the slabs for presence of delamination, corrosion activity and moisture. [84] used the same Pioneer platform to develop a system using vision that can automatically detect and map cracks in concrete slabs. [81] develops a mobile robot by integrating single channel ground penetrating radar on the Seekur Jr platform which is slightly larger than the Pioneer platform. In view of all the previous effort to integrate only limited NDE techniques, we develop the RABIT robot that bringing complementary NDE sensor arrays to a heavy payload robotic system. The RABIT robot also fuses the multiple navigation sensor to provide a robust pose estimation which enable the fully autonomous inspection. That enables us to provide a comprehensive view of the bridge deck efficiently and effectively.

1.2.2 Automatic data processing for collected image

To provide a comprehensive view of health condition of the bridge, the RABIT robot collects multiple non-destructive evaluation (NDE) sensors including impact echo, ground penetrating radar, resistivity probe and high definition camera. One of challenges faced in building the bridge deck health system is to process large volume of inspection data. Among all the techniques, a detailed panorama image of the bridge provides the cornerstones for the civil engineers to assess the bridge condition. A well formed panorama image can help researchers and engineers to identify the area with deterioration and cross-validate with other NDE sensor measurement.

Cameras are widely used in bridge deck or road scanning [18]. The scan is done using either line scan camera [91] or area camera [99]. Line scan cameras are known for its short exposure time and fast data acquisition rate, thus often integrated onto specialized vehicles to scan the road surface [91]. The line scan cameras are often triggered by encoder pulses and require synchronization between the the camera and registration data. One drawback for line scan cameras is that it is difficult to attain perfect match due to no overlap between line scans. In contrast with line cameras, area cameras, which are widely available, take discrete snapshot of the ground plane. The images taken sequentially usually have overlapped area. An image registration algorithm is later performed to assemble multiple images into a panorama image. This process is referred as image stitching, or in the planar case image mosaicing.

Image mosaicing could be categorized as a special case of scene reconstruction where images are related by planar homography only. In [44], an in-depth review is provided for the existing image mosaicing algorithms. Mosaicing involves two major step in processing: registration and blending. Registration refers to the process of align multiple images to the reference coordinate based on the calculated geometric transformation. Though attempts have been made to overcome the registration errors by utilizing blending, the significance of accurate registration in image mosaicing still remains unquestionable [44]. There are two major ways for frame to frame image registration: area-based registration and feature-based registration [123]. Feature-based method, such as [12], relies on the extracted salient feature from the images to compute the geometric transformation between images. It is considered more robust than the area-based method in case of illumination variation and doesn't have the convergence range problem.

But in real application scene, it is necessary to choose between different blob-like features, e.g. SURF [9] and corner-like features, e.g. ORB [118] to achieve good result.

Navigation information could also be helpful for the registration since it contains the relative transformation information between two frames. But navigation information alone is not accurate enough for image stitching in most case. It is sometimes used jointly with image registration, such as in aerial imaging [131] or seabed imaging [36]. In [131], position is only used heuristically to determine the potential overlapped image. In [36], navigation information is used as the initial guess for the optimization to align the images.

Another closely related field is visual SLAM which has received significant advancement in the past decades. In [101], a SLAM system using ORB feature was proposed and represents the most reliable SLAM system up to date. A graph optimization framework g^2o [74] was used to carry the local and global bundle adjustment. SLAM usually rely on video input which assumes relative small frame to frame movement.

In this dissertation, we propose an image mosaicing system that specifically tailored for creating panorama image from the collected bridge deck images. By incorporating the navigation data, our proposed approach inherits the drift-less nature from GPS while still maintains local accuracy of feature-based image registration.

1.2.3 Autonomous bridge deck rehabilitation

The current practice of repairs of concrete bridge decks often happens in the late stages of delamination. The partial- or full-depth repairs involve labor intensive and expensive process including removal of damaged concrete and other deteriorating materials from reinforcing steel as shown in Figure 1.5a and placement of

(a) (b)

Figure 1.5: State of practice in bridge deck rehabilitation: (a) Removal of damaged concrete; (b) Placement of new repair material

the repair material as shown in Figure 1.5b. We introduce RABIT, the bridge inspection robot, that could detect early deterioration and this should complement with early mitigation approach for the maximum benefit. It is estimated that early intervention could lead to a longer bridge deck life and hence increase the current average of 20-25 years to about 50 years of deck life [34]. Therefore, we develop the non-destructive rehabilitation (NDR) system is aiming at delivering a non-destructive, rapid, cost effective rehabilitation at an early stage of deterioration.

Delaminations are basically horizontal cracks in the concrete that occur mainly due to rusting of steel rebars. Since they normally do not have any surface openings before the late stage such as spalling, they are very difficult to detect and repair in the early stage. [71] [70] demonstrated that certain mixes of alkali alumina-silicate matrices reinforced with nano/micro fibers could give a material with good mechanical properties and had desired flow for the hair-line cracks as thin as 0.03 inches. But their method is manual prone and has special requirement for injection fixture which limit the large scale deployment. We're aiming for an automated and minimally invasive procedure that could replace the current state of practice.

Robotics and automation technologies have increasingly gained attention for bridge inspection, maintenance, and rehabilitation. Mobile robot- or vehicle-based inspection and maintenance systems are developed for vision based crack detection and maintenance of highways and tunnels[126][92][136]. Robotic rehabilitation systems that remove defective concrete have also been reported in [17]. A robotic system was developed for removing the paint of bridge bottom. In [90], a similar system was used for imaging the bottom surface of bridge. Those systems usually are the integration of industrial robot with the peeper crane which still require human intervention. Most of these work mainly focus on the use of robots for inspection or simple maintenance such as painting etc., rather than the complex robotic hammer drilling for repairing concrete defects.

1.2.4 Percussive drilling model

The NDR system use a minimal invasive procedure to rehabilitate the defect area. One critical step for the rehabilitation procedure is to drill holes to reach the defect by using a modified rotary hammer on a manipulator. The rotary hammer uses impact, rather than thrust forces or torques, as the main source to crush the bristle concrete materials. For high-quality robotic drilling on concrete, modeling of the drilling process is a critical step to design the robotic control systems. However, understanding the mechanisms in concrete drilling is a challenging task due to the complicated energy transfer and complex bit-concrete interactions during impact.

Although studies of metal drilling process, including the robotic manipulator drilling, are reported extensively in the past several decade[106], there is few work that discusses the robotic drilling model and control for concrete or rocks. On the other hand, understanding and modeling of percussive drilling in rocks and concrete is not a new subject and the early studies are reported about four decades ago. In [62, 65, 27, 39], empirical percussive drilling models are proposed to capture the drill bit impact interactions with rocks. A hysteresis relationship between the drill bit penetration and applied force is commonly assumed known in these models. Computational approach, such as finite element method or other impact energy-based simulation, are also used to study the percussive drilling in [109, 107, 22, 21]. Several analytical models are proposed to capture the impact energy as wave transmission between the drill bit and the rock [95, 94]. In these models, both the penetration-force relationship and the impact wave form are needed to completely solve the percussive drilling problem. In [108, 29], computational approach is used to calculate the energy and impact interactions between various components in hammer drills used in practice.

For viewpoint of control system design of hammer drill bit-concrete interactions, all of the above mentioned percussive drilling models are not desirable. The empirical model cannot give the physical interpretation and connection with drilling process parameters in practice, the computational models are too complicated for control design purposes, while the impact wave propagation models are too simplified for capturing the actual hammer drill systems. Instead, we propose a dry friction-based percussive drilling model that is inspired and extended from the model in [73] and the work in [109]. The proposed model is compact in mathematical representation and therefore, is desirable for use of designing control systems for drill bit-concrete interactions. Moreover, the model captures the penetration-force relationship through the dry friction characteristics and can readily be used to interpret the rock crush/chipping phenomena [109].

1.2.5 Simultaneous bridge deck inspection and rehabilitation

Bridge deck inspection should be accompanied by bridge deck rehabilitation to increase the life span of the bridge. Having bridge deck problems exposed and fixed at the same time could further reduce the interruption of traffic and labor cost. However, simultaneous deployment of the bridge deck inspection and rehabilitation robot is not trivial in three aspect. First, the rehabilitation requires a high precision delamination map that could be generated online while the inspection is still in process. Second, high precision usually requires the inspection robot to stop more frequently which largely increase the inspection time. Third, there are always resources conflict problem when deploy multi-robot system[132].

We design our strategies to tackle the delamination mapping and planning problem around the Gaussian process regression[130]. Gaussian Processes have been long used to model temperatures and other spatial phenomena[26]. It becomes a popular approach in the robotics society as it provide posterior estimation with uncertainty that facilitate the stochastic motion planning and control. [57][58] use Gaussian process to model the uncertainty of the ship hull and plan the optimal path for the underwater inspection vehicle. [96] present an informative planning algorithm with Gaussian process to enable an autonomous marine vehicle to perform persistent ocean monitoring. [119] propose an algorithm for exploration with Gaussian process in unknown environments.

We want our inspection robot to stop less frequently without sacrificing the precision of delamination map. This is closely related to the problem of adaptive sampling [58] which the goal is to choose observation locations that maximize the information gain and minimizing prediction uncertainty. Early work such as tackling the next-best-view problem [25] focus on the geometry approach for searching for the informative views. More recent approach has been adopt the probabilistic modeling, such as information gain [138] and Gaussian process[96].

The coordination between the inspection and rehabilitation is related to field of multi-robot systems. Our system could be categorized as heterogeneous and cooperative multi-robot system as pointed out in [132]. Similar system such as [14][121] use ground and aerial robot together to search for a target. Despite the difference, the central problem lies in the multi-robot system is the resource conflict[132].We implement an framework similar to lead-follower scheme to avoid resource conflict such as collision between robots.

1.3 Dissertation outline and contribution

There are seven chapters in this dissertation. Chapter 1 presents the introduction and background. In Chapter 2, we presents the development of the bridge inspection robot. In Chapter 3, we presents the an image mosaicing system specially tailored for images collected from the bridge inspection robot. In chapter 5, we present the development of the bridge rehabilitation robot that focus on early delamination mitigation. We present the strategies that enable simultaneous deployment of the bridge inspection robot and rehabilitation robot in Chapter 6. Conclusion of the dissertation and discussion of the future work are presented in Chapter 7. The content of each chapter is described as follows.

In chapter 2, we present the design and development of the autonomous bridge inspection robot. We first introduce the non-destructive evaluation (NDE) technologies implemented including GPR, impact echo, and electrical resistivity. We then present the hardware and software integration of robotic system and NDE technologies. After introducing the whole system, we focus on the robust pose estimation system that enable the autonomous inspection. The pose estimation is done through fuse multiple navigation sensors information and provide a robust estimation when the noise present. We demonstrate the performance of the inspection robot through extensive field deployment.

In chapter 3, we present a new image mosaicing system for the bridge deck surface reconstruction. By fusing the navigation data with feature-based image registration in the graph optimization framework, our proposed approach inherits the drift-less nature from GPS while still maintaining local accuracy of feature-based image registration. We evaluate the accuracy through quantitative test on real bridges and show our system is robust to interference in the outdoor environment such as illumination variation.

In chapter 4, we present the design and development of autonomous bridge rehabilitation robot that focus on provide minimal invasive rehabilitation for early bridge deck delamination. We first present the hardware design of the robot including the omni-directional mobile robot platform, a 5 degree of freedom manipulator and a custom made end-effector for drilling and filling procedures. We then discuss planning algorithm for the mobile manipulator and drilling and filling procedures. The robotic system performance was validated through extensive experimental testing and field deployment.

In chapter 5, we present a mathematical model of a pure percussive drilling process that is critical procedure for the rehabilitation process. A modified dry friction-based drilling model was presented to capture three major phenomenon in the drilling process: the elastic deformations, crushing and chipping of the penetrated material. We analyzed the drilling model and presented a set of analytical formulation for the critical drill bit kinetic energy and the penetration rate per impact. The model parameters were physically interpreted with the experimental testing and the values of these parameters were estimated experimentally. Finally, we validated the model prediction with experiments through extensive drilling tests.

In chapter 6, we present the approach to simultaneously deploy the inspection and rehabilitation on the bridge decks. We solve the online delamination map generation by Gaussian process regression. We take a multi-threading approach to create two training sessions: a local Gaussian process training with only local data to fulfill real-time requirement and a global Gaussian process training with all the data available. In order to reduce the stop frequency of the inspection robot, we implement an adaptive step size approach to dynamically determine the step size based on the prediction uncertainty in the front path of the inspection robot. Moreover, we design a target planning algorithm based on the global delamination map for the rehabilitation robot to choose the next target to repair. The algorithms proposed are validated through a custom simulation system that could take real bridge inspection data.

The main contribution of the dissertation are described in details as follows.

1. A novel autonomous bridge inspection robot is developed to provide a comprehensive view of bridge health condition efficiently and effectively. The developed platform improved the data collection speed dramatically compared to conventional NDE deployment which will further enable the frequent bridge inspection in a large scale. During the development of the robot, we address two major challenges which is a robust navigation system and a robust NDE data collection.

2. A new automatic image mosaicing system is developed for bridge deck surface reconstruction. This system is specially tailored for the bridge inspection task to enable the automatic processing of large volume image data. This is realized by combining the image and navigation information in the graph optimization frameworks.

3. A novel autonomous bridge deck rehabilitation robot is developed to provide minimal invasive repair for the early defect such as delamination. The system features a mobile manipulator and a custom-made end-effector could function as drilling and filling modules. To our best knowledge, this represent the first autonomous robotic system targeting at early defect mitigation.

4. A new percussive drilling model is proposed to describe the complicated energy transfer and complex bit-concrete interactions during concrete drilling. This model is mathematically compact therefore it's suitable for designing and optimizing the control of drilling process in concrete or rocks. The model also provides a means to further design, optimize and enhance the drilling performance for applications such as robotic bridge deck rehabilitation.

5. A set of new strategies is presented to solve the problem of simultaneously deployment of inspection and rehabilitation robot. We propose a separate local and global Gaussian process training scheme to overcome the training speed issue and achieve online delamination map generation. A new adaptive step determination based on local uncertainty is proposed to accelerate the inspection progress. A new target planning algorithm based on global delamination map is proposed for the rehabilitation robot to determine the next target while avoiding resource conflict with the inspection robot. We also create a simulator that could take in real bridge data to validate our algorithms.

Chapter 2

Bridge Deck Inspection Robot

2.1 Introduction

The condition of bridges are essential to ensure the transportation safety. As the most weathered part, the bridge deck deteriorate over time as a result of excessive loading, environmental exposure, material aging and inadequate maintenance. Therefore, bridge deck inspection is conducted to identify the bridge deterioration condition and further facilitate the appropriate maintenance or rehabilitation procedures. The current practice for concrete bridge deck evaluation in the large part still relies on human visual inspection or use simple evaluation tools. There are justified reservation regarding the accuracy and objectivity of those practices as pointed in [48]. Non-Destructive Evaluation (NDE) has emerged as a promising approach in the last decade as it could provide much more accurate assessment for the bridge deck condition. Besides the comprehensive overview, the NDE techniques also enable the accurate monitoring of deterioration progression [52]. There are several NDE technologies that have been currently used for bridge deck health evaluation, such as ground penetrating radar, impact echo, electrical resistivity and high-definition camera. Each NDE techniques has its strength and limitation and is best suited for characterization particular defect type. During evaluation, each NDE technique is conducted manually through a group of engineers in a closed section of road as showed in fig 2.1. Those practices can't not meet the increasing demand for cost-effective and safe evaluation as in the following aspect. First, the speed of manual NDE is slow leading to extended slowdown for the traffic flow. Second, due to speed, in most cases only one NDE technique is deployed which subject to the limitation of that technique and could not provide comprehensive evaluation of the bridge. Third, significant labor work are required to perform the manual NDE test resulting in high inspection cost. Fourth, the human inspector are exposed to the vehicles runs in the adjacent lanes which poses potential safety risk. Fifth, the manual NDE techniques are still prone to human error and subject to inspector experience.

To overcome the limitation of manual NDE, automated bridge deck inspection become a promising approach in civil infrastructure application. Automated bridge inspection would be efficient and reliable in data collection and safe for the inspector. There are several attempts to bring the automation and robotics into the bridge deck inspection. German Federal Institute for Material Research and Testing (BAM) have attempt to automate data collection through developing NDT-Stepper [128]. The NDT-Stepper is an automated cart that equipped with single pneumatically actuated impact echo and ultrasonic probes. It moves in constant increments at the speed of 2–3 m/min and collect the data at each stop. BAM later developed a robotic system BETOSCAN based on the widely used Pioneer platform from Adept MobileRobots for inspection of reinforced concrete slabs [129, 117]. BETOSCAN implemented multiple NDE techniques including ultrasonic, microwaves, cover meter and thermometers, therefore could scan the slabs for presence of delamination, corrosion activity and moisture. [84] used the same Pioneer platform to develop a system using vision that can automatically detect and map cracks in concrete slabs.

Inspired by previous effort, the Robotics Assisted Bridge Inspection Tool (RABIT) is developed for fully autonomous bridge deck inspection by bringing multiple NDE sensor arrays to a much bigger robotic system [50, 75, 79, 77, 76, 47].

There are two major challenges in the RABIT development. One is to seamless integrate NDE sensors with the mobile robot base for automated data collection. This involves deploying the contact or contact-less NDE sensors through the electro-pneumatic system. This also involves design effective and coordinated robot movement to enable reliable data collection. The other major challenge comes from building an accurate and robust system for localization and navigation. Since the robot need to navigate on the narrow bridge deck, it need a localization system up to centimeter grade. Although high-accuracy GPS could reach the requirement with the real-time kinematic (RTK) correction, the GPS signals are not always reliable and robust especially on bridges with supporting structure such as steel cable and truss elements.

The rest of the chapter is organized as follows. In the next section, we give an overview of the robotic system and its software hardware integration. In Section 2.3, we introduce the NDE sensors integrated with the robot. In Section 2.4, we introduce the robot hardware and software design for the robotic integration of the NDE sensors. In Section 2.5, we present the robust localization algorithm that fused multiple navigation sensors and the planning and control. The system and the proposed localization algorithm is validated through outdoor experiment in Section 2.6.

2.2 Overview

The RABIT robot could be divided into the NDE system and robotic system. The system overview diagram is shown in Figure 2.2. The NDE system is responsible for triggering the NDE sensor data collection while doing the online post-processing. The robotic system is responsible for the navigation, motion planning and various actuation of the pneumatic and electrical parts of the robot. Those

Figure 2.1: Manual practice of NDE through a group of engineers

two systems need to work seamlessly to ensure successful data collection.

The robotic system composed of one industrial standard embedded computer running Linux, the omni-directional Seekur mobile base, the navigation sensor suite including two Novatel RTK GPS and wheel odometry, and the electro-pneumatic control system that placing the NDE sensors to the right data collection position. The robotic system is responsible for the navigation, motion planning, mobile base movement and actuation of the electro-pneumatic control system for placing NDE sensors.

The NDE system composed of two industrial standard embedded computers and NDE sensors including impact echo, ground penetrating radar, resipot and high definition surface imaging camera. The NDE sensor arrangement is shown in Figure 2.3The two computers are running NDE data collection software in Windows system that perform interfacing with the NDE sensors, post-processing and visualization. In addition to NDE system, the van transporting the RABIT also doubles as the monitor station as shown in 2.4. The human inspector is able to inspect the data collection progress and quality.

The successful data collection requires the close coordination between the NDE system and robotic system. Due to software driver availability, the NDE

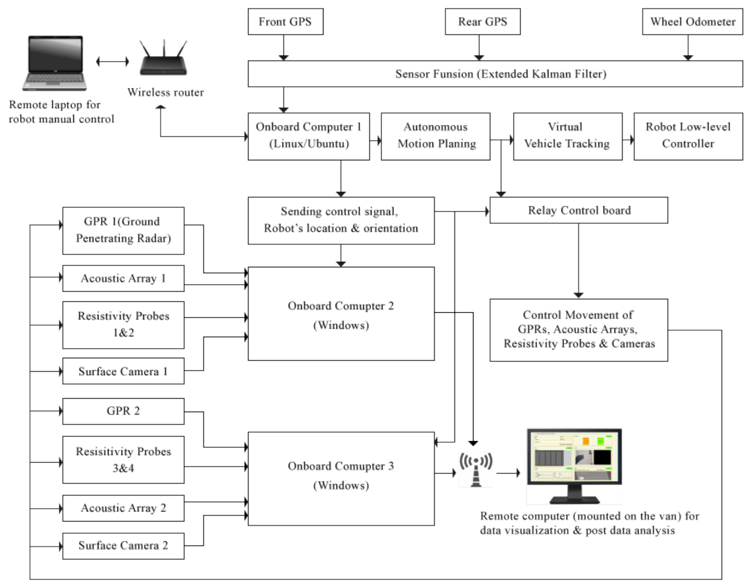


Figure 2.2: An overview of the bridge deck inspection robot RABIT

system software is implemented in windows, while the robotic system software is wrote in Linux system. The intra-system communication is enabled through Ethernet for those two different operating system .

In each data collection attempt, the robotic system first stop the robot and actuate the electro-pneumatic system to place the sensors in contact with the ground. The robotic system computer then send command through TCP/IP protocol to the NDE system computer indicating the NDE sensors are ready for data collection. The command also contains the current pose of the the robot. The NDE system computers then send command through various NDE sensor interfaces and collect the response data. Once all the data collection is completed, a confirmation message is sent to the robotic system computer. At the same time, the NDE computers post-process the data and send to the transportation van for the human inspector to check the data quality and progress.

Figure 2.3: RABIT robot NDE sensors and Navigation sensors. Yellow tiles indicate the NDE sensors and blue tiles indicate robot navigation sensors. Top-left: the front view of the RABIT; Top-right: the back view of RABIT; Bottom-left: a close-up of the impact echo array in the data collection position; Bottom-right: the folding position for transportation convenience.

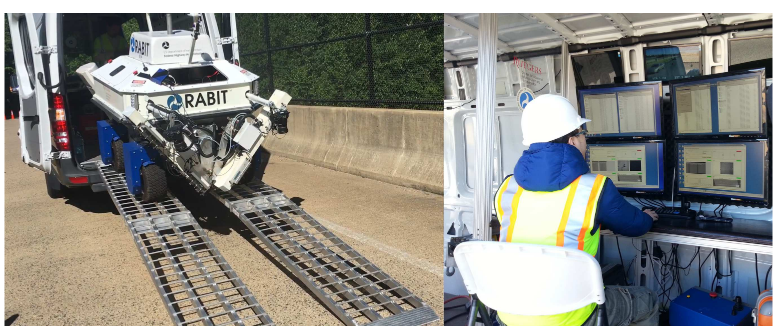


Figure 2.4: Transportation van and command center. Left: unloading the robot from the van for data collection; Right: human monitor center inside the van

2.3 NDE sensors

2.3.1 Ground penetrating radar (GPR)

Ground penetrating radar is a geophysical method that uses electromagnetic waves to inspect the subsurface condition. The GPR based condition assessment of concrete bridge decks has been described in many publications [98, 7, 113, 53, 127]. When the electromagnetic wave emitted from the source encounter metallic objects such as rebars, the reflection waves could be detected by a receiving antenna. Based on the attenuation of electromagnetic waves, GPR can provide a qualitative condition assessment of bridge decks on the top rebar level. The GPR is also used as a quality assurance tool for new construction or rehabilitation.

The RABIT robot equipped with two Hi-Bright ground-coupled GPR arrays manufactured by Ingegneria Dei Sistemi (IDS), Italy. The two Hi-Bright GPR arrays have 32 bow-tie type antenna with 2.0 GHz center frequency. Each GPR array box contains eight pairs of dual-polarization antennas in the orthogonal orientation, as illustrated in 2.5. Dual polarization antennas can facilitate GPR data analysis in the situation that the top rebar is not in the preferred orientation. There is 10 cm spacing between antennas which six times higher spatial resolution than 0.6 m spacing that required by FHWA Program protocols [47]. A minor loss of spatial resolution with the current antenna arrangement is the spacing between the end antennas of the two arrays, which is about 25 cm.

During scanning, the two GPR arrays mounted on the rear end of the RABIT are pressed by pneumatic mechanism to be in close contact with the ground. When the RABIT moves forward, the encoder attach to the mobile base wheel generate pulses that trigger the data collection of the ground penetrating radar. 2.6 shows the continuous imaging of one channel. The parabola shape in the image are reflection of the rebar which is used to access the rebar corrosion condition

Figure 2.5: IDS ground penetrating radar array and layout[47]

[28].

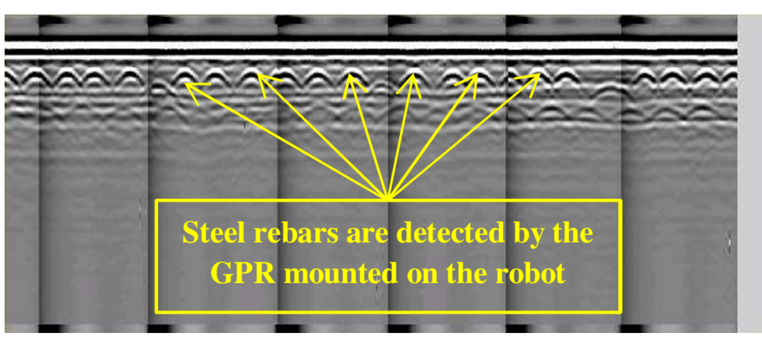


Figure 2.6: Visualization of ground penetrating radar that shows the steel rebar

2.3.2 Impact echo and ultrasonic surface wave

2.3.2.1 Impact echo

The impact echo method is used to detect discontinuities in concrete and measure the concrete thickness. Delamination is the horizontal cracking that are common defect in bridge deck. As an elastic-wave based method, the impact echo are used to detect and characterize delamination in the concrete bridge decks in terms of depth, spread, and severity [86, 120, 16, 45, 49]. It can be also used to detect debonding of overlays on bridge decks[77].

The impact echo method measures the transient vibration response of a mechanical impact on a plate-like structure[78]. The mechanical impact generates longitudinal and transverse body waves, and surface-guided waves such as Lamb and Rayleigh surface waves propagated in the plate. The transient time response of the solid structure is commonly measured with a contact sensor such as accelerometer that is close to the impact source. As shown in Figure 2.10, the frequency response of the measured transient time-signal is obtained through Fast Fourier Transform (FFT). The peak frequencies on the amplitude spectrum corresponds to particular resonance modes as shown in Figure 2.8. To interpret the severity of the delamination in a concrete deck with the IE method, a test point is described as solid if the dominant frequency corresponds to the thickness stretch modes (Lamb waves) family [45]. In that case, the frequency of the fundamental thickness stretch mode is the zero-group-velocity frequency of the first symmetric (S_1) Lamb mode, or also called the IE frequency (f_{IE}). The frequency can be related to the thickness of a plate H for a known P-wave velocity C_p of concrete by

$$H = \frac{\beta_1 C_p}{f_{IE}}$$

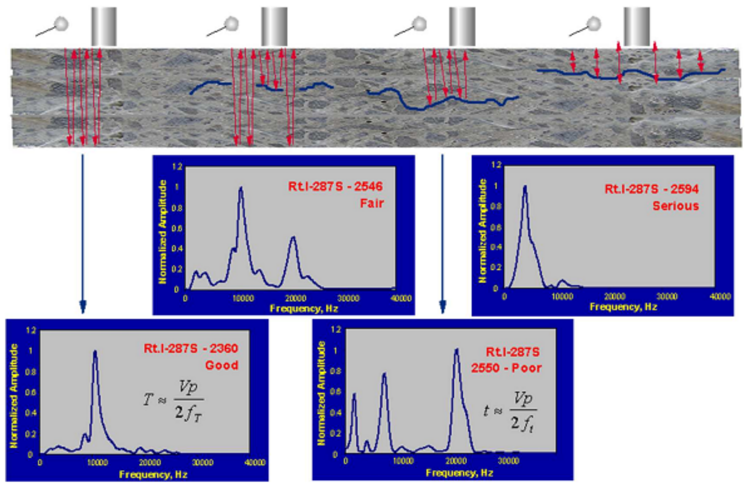


Figure 2.7: Physical interpretation of how impact echo method identify various delamination condition[78]

where β is a correction factor that depends on Poisson's ratio of concrete, ranging from 0.945 to 0.957 for the normal range of concrete. A delaminated point in the deck will theoretically demonstrate a shift in the thickness stretch mode toward higher values because the wave reflections occur at shallower depths. Depending on the extent and continuity of the delamination, the partitioning of the wave energy reflected from the bottom of the deck and the delamination may vary. Progressed delamination is characterized by a single peak at a frequency corresponding to the depth of the delamination. In case of wide or shallow delaminations, the dominant response of the deck to an impact is characterized by a low frequency response of flexural mode oscillations of the upper delaminated portion of the deck. The typical way of interpreting the severity of the delamination in a concrete deck is shown in 2.7.

2.3.2.2 Ultrasonic surface waves (USW)

The USW test is utilized to assess concrete quality and, thus, possible concrete degradation, by measuring concrete modulus [102]. Instances of significant drops in the measured modulus will often be an indication of presence of delamination or other major defects [137]. Low modulus is often related to concrete degradation or delamination. However, lower modulus values are also observed in new decks as a result of material variability and concrete placement procedures. Therefore, only periodical assessment of concrete modulus can be used to detect a deterioration process [78].

The USW technique is the Spectral Analysis of Surface Waves (SASW) to evaluate material properties (elastic modulus) in the near surface area. As shown in 2.8, the SASW uses the phenomenon of surface wave dispersion (i.e., velocity of propagation as a function of frequency and wave length, in layered systems to obtain the information about layer thickness and elastic modulus). A SASW test consists of recording the response of the deck, at two receiver locations, to an impact on the surface of the deck. The surface wave velocity can be obtained by measuring the phase difference between two different sensors as

$$C = 2\pi f \frac{d}{\Delta\phi}$$

where f is frequency, d is distance between two sensors. The USW test is identical to the SASW, except that the frequency range of interest is limited to a narrow high-frequency range in which the surface wave penetration depth does not exceed the thickness of the tested object. Significant variation in the phase velocity will be an indication of the presence of a delamination or other anomaly. In cases of relatively homogeneous materials, the velocity of the surface waves does not vary significantly with frequency. The surface wave velocity can be precisely related to the material modulus, or concrete modulus in the case of bridge decks, using either the measured or assumed mass density, or Poisson's ratio of the material. In the case of a sound and homogeneous deck, the velocity of the surface waves will show little variability. An average velocity is used to correlate it to the concrete modulus. Significant variation in the phase velocity will be an indication of the

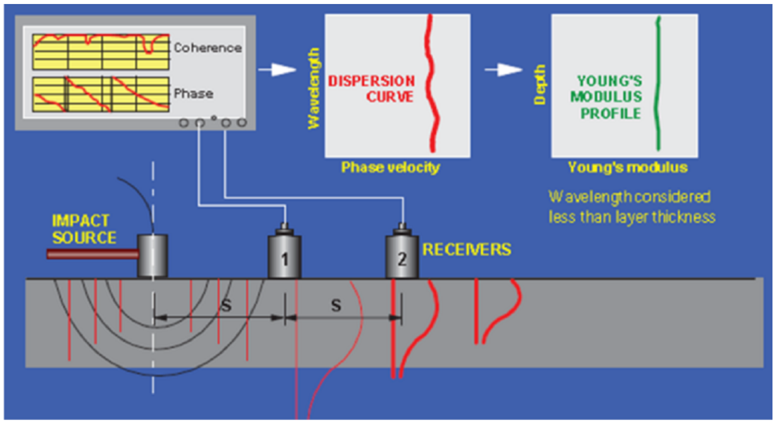


Figure 2.8: Physical interpretation of how ultrasonic surface wave measure concrete modulus[78]

presence of a delamination or other anomaly.

2.3.2.3 Implementation

The developed robotic system integrates more than a dozen of the impact echo sensors. The acoustic arrays are two boxes of size 0.9m×0.2m, each containing seven accelerometers and four impact sources. The arrangement of the sources and receivers is shown in 2.9. The sources are linear solenoid type impactors, while the receivers are accelerometers. The acoustic arrays were designed and manufactured by Geomedia Research and Development, Inc. As illustrated in 2.9, each acoustic array enables the conduct of eight impact echo (IE) and up to six ultrasonic surface waves (USW) tests. The spacing between a source and near receiver is 7.5 cm (3 in.). The spacing between the sensors allows delamination assessment with a resolution of 15 cm (0.5 ft) in the deck's transverse direction, which is four times higher than the one according to the LTBP Program protocols for data collection (0.6 m) using manual IE devices.

The RABIT stop every 2 feet to collect the acoustic data. During each stop, the acoustic arrays are pneumatically pressed against the deck surface to achieve uniform coupling between the sensors and deck surface. The hammering action

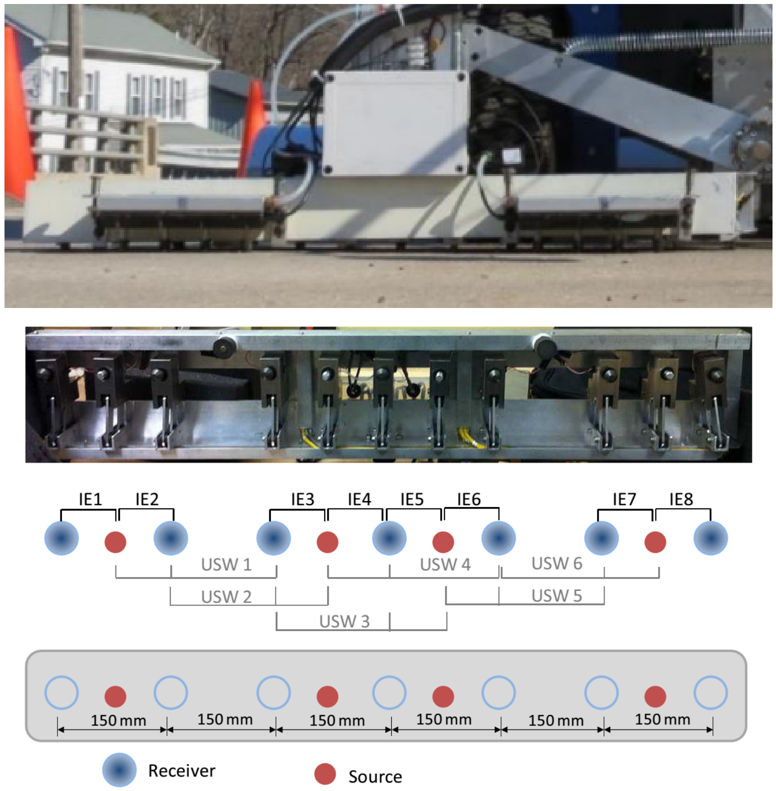


Figure 2.9: Impact echo array and sensor layout. The red dot are the the source generating the hammering action. The blue dot are sensors that measure the reflection wave.

is triggered from the source in consequence and the accelerator picks up the reflected sound wave. In 2.10, the measured transient time signal on the left hand are transformed to frequency domain on the right hand side through fast Fourier transform. The frequency response represent particular resonance modes. The SASW takes the same time signals recorded at two receiver locations for a single impact on the surface of the deck.

2.3.3 Electrical resistivity

Electrical resistivity sensor measures concrete's electrical resistivity, which is a reflection of the corrosive environment of the bridge deck [13, 46]. The presence of water, chlorides, salts, or other contaminants reduces concrete's resistivity,

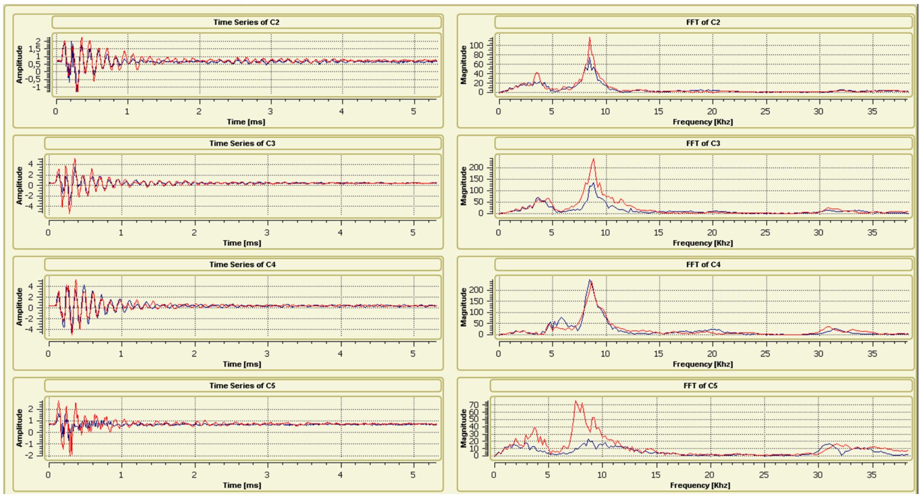
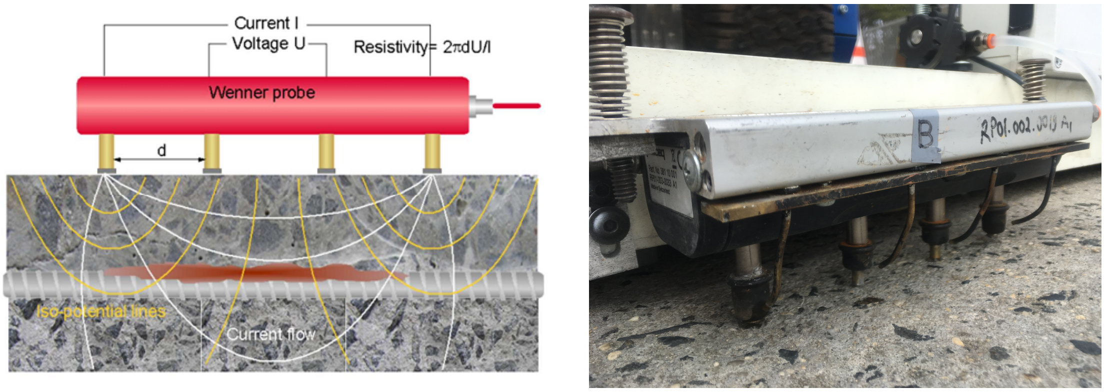


Figure 2.10: Impact Echo time and frequency response. The left shows the time history of 8 response signals that caused by 4 impact and the right shows the corresponding amplitude spectrum. Each chart contains the 2 sensor response with respect to the same impact

and facilitates corrosive processes in bridge decks. By measuring the electrical resistivity, the corrosion rate of reinforcing rebars can be estimated . To ensure good coupling between the electrodes of the resistivity probe and concrete, water is lightly sprayed on the electrodes.

There are four ER probes of Wenner type attached to the front end of acoustic arrays. The Resipod probes manufactured by Proceq have four electrodes with a 50 mm (2 in.) spacing between them. The spacing between the probes is about 45 cm (22 in.). As illustrated in 2.11a, electrical current is induced through two outer electrodes and the potential of the generated electrical field measured using two inner electrodes. The two are used to calculate the electrical resistivity. To establish the electrical contact between the deck surface and probes, the probes' electrodes are being continuously moistened using a spraying system. The spraying system as shown in 2.11b sprays water on each of the electrodes using very fine copper tubes at the end of each data collection cycle.



(a) (b)

Figure 2.11: (a) Physical interpretation of electrical resistivity sensor measurement[78]; (b) The spray system used to moisture of the electrical resistivity sensor.

2.3.4 Visual detection of surface cracks

Two wide-lens Cannon cameras are used to capture the bridge deck surface images which could be used for crack detection and mapping. The two cameras are mounted on two linear actuators on the front side of the RABIT. During data collection the two actuators are fully extended to make the camera image free of occlusion of RABIT parts such as impact echo arrays. During transportation, the rod could retrieving back for easy moving and transport. The robot collects images at every 2ft (0.61cm). Each of the cameras covers an area of a size of 1.83m × 0.6m as shown in 2.12c. To enable image stitching, the images simultaneously collected by these two cameras have about 30 percent overlap area and the images captured in two sequential stops for the same camera have an overlap area of 20 percent. Each camera is equipped with a flash to remove shadow at night time and mitigate shadow at day time. Moreover, one 360 degree field of view panoramic camera is used to capture the surveyed area. The panoramic camera is mounted on retractable pneumatically actuated mast that controlled by the robot computer.

(a) (b) (c)

Figure 2.12: (a) Canon EOS Rebel T3; (b) Camera with panorama lens; (c) Deck image on the bridge joint

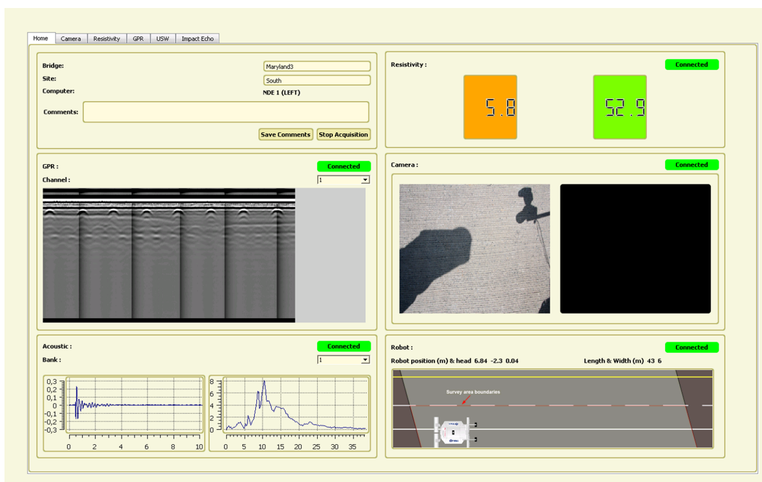


Figure 2.13: Graphical monitor interface that visualize the data for ground penetrating radar, impact echo, electrical resistivity, camera image and robot location

2.3.5 GUI and monitor station

A remote graphic user interface (GUI) is developed for the inspector to monitor the NDE data collection as shown in Figure 2.13. Through the GUI, robot operators and field engineers can remotely control the data collection in several modes: manual, semi-autonomous, and fully autonomous. All of the collected NDE data and robot navigation information the can be saved in the remote computers for data processing and analysis.

2.4 Robotic integration

2.4.1 Omni-directional mobile base

The robotic system with integrated NDE technologies is shown in 2.3. The Seekur robot from Adept Mobile Robot Inc is modified to use as the mobile base. The Seekur robot has four independent actuated wheel and each wheel has two electric motors for driving or steering. Those omni-directional wheels enable the high-agility maneuver such as parallel movement and zero radius turn which are much needed on narrow bridge decks. Moreover, the seekur robot are rated for payload up to 150 kg which is suitable to carry NDE sensor arrays. Besides that, the Seekur robot is an all-weather outdoor platform that is desirable for outdoor bridge inspection applications. The Seekur features a separate lower level controller and a higher level motion planner. The higher level motion planner supported by the user provide the 2D planar linear velocity and yaw rate as the input to the lower level controller. The real-time lower level controller is responsible for coordinating the steering and driving motors for the desired movement.

2.4.2 Navigation sensors

To achieve autonomous operation on bridge, a precise and robust localization system is required for motion planning of the mobile base. Since the NDE sensors all have fixed transformation with respect to the robot base, the registration of NDE data also requires precise and robust localization.

We design a redundant navigation system by integrating two systems. The first system is the differential global positioning system (DGPS) with real-time kinematic (RTK) correction. The DGPS consists of a fixed base-station GPS receiver and two moving GPS rovers. The base station GPS is placed on a fixed tripod during data collection, typically on the side of the bridge within a 100 m distance from the robot. Two moving GPS receivers are mounted at the front and back of the robot about 1.8 meter apart from each other. All the GPS units are manufactured by Novatel, Inc. The GPS rovers on the robot receive both the location signal from satellites and a correction signal from the base-station GPS in real time through a separate radio station from Freewave FGR series. The RTK algorithm compensates the GPS signal errors and produces a more precise positioning. The third system is wheel odometry that enables accurate distance measurement. The wheel odometry is calculated from four optical wheel encoders mounted on the wheels. The information coming from the three navigation components is fused using an extended Kalman filter (EKF). The presence of the wheel odometry is essential in the areas where GPS signal occasionally dropout.

2.4.3 Software

Three industrial standard embedded computers (from Versalogic, Inc.) are installed inside the robot. These computers can operate functionally in high temperature environment (up to 80 $\circ$C) for all-season field testing. High-speed Ethernet connections are used among these computers and each computer can also be reached individually through high-speed wireless communication by the remote computers. The NDE data and images are also transmitted in real time to the remote computers for visualization and data analysis purposes. The remote visualization and data analysis computers are located inside a full-size cargo van that is also used to transport the robotic system.

The software onboard could be divided into NDE system software running on the two Windows computer and robotic system software running on one Linux computer. The NDE system software and robotic system software communicate over Ethernet through the TCP/IP network protocols.

The NDE software is a multi-threading program that enable the simultaneously collect and monitor the data. The software is developed under Qt frameworks using C++. For each NDE computer, it runs an instance of the program responsible for the data collection on one side. The software consists the master thread providing the user interface and five slave thread that communicating with NDE sensors and robotic system. The five slaves thread are as follows:

- Robot thread: This thread is responsible for communicating to the Linux robotic computer for coordination. It receive sensor triggering command from the robotic computer and return with a acknowledgment signal through TCP/IP. It also receive the pose information from the robotic computer and the pose information is associated with collected sensor data for registration.

- Acoustic thread: This thread connects to the control board of impact echo through USB connection. The acoustic thread receives triggering command from the master thread and sending command to the control board for actual data collection. When control board receives the triggering command, it will fire the hammering in consequence and collect the response from the accelerometer. The time history data collected are saved sent back to the acoustic thread for further data processing. The time series data are then go through fast Fourier transform (FFT) and send to the master GUI thread for real-time monitoring purpose.

- Ground penetrating radar thread: This thread communicates with IDS vendor software through TCP/IP protocol. The GPR thread is responsible for start and stop command at the beginning and end of the pass and also streaming the data from the server.

- Camera thread: This thread uses the Canon SDK to trigger the shot and changing the parameters such as flash and white balance. It's also responsible for retrieving the collected image from the SD card and send back data to the GUI thread once the image collection is successful.

- Electrical Resistivity thread: This thread communicates with the Proceq Resipot through USB connection. The electrical resistivity are measured in a short period of time and averaged value of the measurement is sent back to the master thread for data logging and display.

The robotic software is running on the Linux windows computer. The software is developed using Robotic Operating System (ROS) framework. ROS organized the functional module as nodes. When launched, nodes are separate nodes and could communicate with each other through messaging. There are five nodes as follows:

- Robot node: This node connects to the low level controller of Seekur mobile robot through the RS232 serial port. It reads various robot base status and sensor information such as odometry data and bumper status etc and send out 2D velocity command and rotation speed command to the low level controller. The odometry data is shared with sensor fusion node for pose estimation.

- GPS sensor node: The GPS node connects to the Novatel GPS through RS232 serial port. The GPS receives GPS location from two GPS unit and share the data with sensor fusion node for further processing

- Sensor fusion node: The sensor fusion node receive the GPS data and odometry data from the above nodes and fuse the sensor information through Extended Kalman Filter (EKF) for accurate and robot pose estimation. The pose output is shared with other node inside the ROS.

- Motion planning node: The motion planning node takes in the boundary information of the scan area and calculate the desired route for the robot to follows. It then find out the immediate next pose for the robot to stop, typically a position where NDE sensor data need to be collected. This pose information is sent to the vehicle tracking controller to execute. Once the vehicle is stopped at the desired position, it will send out command to relay board node to press the acoustic array to the ground and then send out command to NDE computers to trigger data collection.

- Vehicle tracking controller node: This node is responsible for moving the vehicle from the current pose to the desired pose. It retrieves real-time pose information from the sensor fusion node and desired pose information from the motion planning node and send out planar velocity command and rotation speed command to robot node. When the desired point is reached, it send out confirmation to the motion planner node.

- Relay board node: The relay board node is responsible for control of various electric and pneumatic actuators on the robot. It controls the pneumatic lift and press of the acoustic and GPR array, the solenoid for controlling the water spray to moisture the tip of the electrical resistivity sensor, the linear actuator that extend the camera and the pneumatic mast that lift the panorama camera.

2.5 Robot navigation

2.5.1 Robot localization

Since the robotic system needs to cover the narrow deck surface, it is required that the localization and navigation accuracy be within a range of a few centimeters. Although high-accuracy global positioning system (GPS) with a real-time kinematic (RTK) correction can reach the requirement, it is well known that GPS signals are not always reliable and robust, especially on bridges with partial coverage, steel cables, truss elements, or other support structures. Therefore, we design a redundant pose estimation system that composed of two RTK GPS units and four wheel encoders. Similar to the approaches in [3, 133, 2], dual RTK GPS antennas are used on the developed mobile robot platform. With two sets of RTK GPS antenna mounted with enough offset, it is feasible to estimate the orientation direction of the robot by assuming the robot as a rigid body. A naive way to estimate the angle information would be calculate the angle of the line connecting the front and back GPS. But this naive method is only valid when the position variance is relative small compared to the distance between the two GPS, however this is not always the truth. Therefore, the developed navigation system also fuses the GPS measurements with the wheel odometry information through an extended Kalman filter (EKF) design [104, 134, 105]. The robotic system is equipped with four wheel encoders that able to generate 2D odometry pose data from the low level controller. The accuracy of the wheel odometry of the all-wheel steering platform is much higher than those of other types of mobile robots (e.g., car-like or skid-steered mobile robots) due to the small wheel slippages in operation [59, 134]. As robot travels the odometry error accumulated, therefore the odometry data would be better used as relative measurement in the short distance instead of the global positioning data. With the odometry-enhanced GPS navigation, the robotic system has demonstrated high-accuracy localization that meets the inspection requirements weak GPS environments.

2.5.1.1 Sensor fusion through Extended Kalman Filter (EKF)

The problem could be formulated as 2D planar pose estimation problem with input from the front and rear GPS and the wheel odometry. The GPS provide 2D position information at 10Hz in Longitude Latitude Altitude (LLA) format. The LLA measurement is converted to the initial frame with East-North-Up (ENU) orientation through the Universal Transverse Mercator (UTM) conformal projection. The origin of initial frame is set on the location where the robot is initiated. Those measurement have independent measurement error and didn't accumulate error over time. We use the dilution of precision (DOP) from the Novatel GPS as the estimated noise variance. On the other hand, the odometry provide 2D linear and angular velocity estimation at 20Hz in the body frame.

We define the robot initial frame coordinates $\mathcal{I}$ and represent the robot pose as $\mathbf{q}(k) = [x(k), y(k), \theta(k)]^{\mathbf{T}} \in \mathfrak{se}(2)$, where θ is the yaw angle. In order to design the Extended Kalman Filter (EKF), we derived the system equation and measurement equation. The state vector are set as robot pose $\mathbf{q}$ and the measurement comes from the front and rear GPS are represented as measurement vector $\mathbf{z}(k) = \left[z_x^f(k), z_y^f(k), z_x^r(k), z_y^r(k)\right]^{\mathbf{T}}$.The The discrete system dynamic equation could be formulated as

$$\mathbf{q}(k) = \mathbf{q}(k-1) + \mathbf{B}(k)\mathbf{u}(k) + \mathbf{w}(k)$$

where the input vector

$$\mathbf{u}(k) = [\Delta x_b, \Delta y_b, \Delta\theta_b(k)]^{\mathbf{T}}$$

represent the relative pose from odometry. The odometry data represent the center of the robot base. To alleviate the aggregated error, we only use the relative odometry data with respect to the last received odometry data. The odometry data is represented in the reference world frame as $\mathbf{z}_o(k) = \begin{bmatrix} x_o(k) & y_o(k) & \psi_o(k) \end{bmatrix} \in \mathfrak{se}(2)$, $\mathbf{u}(k) \in \mathfrak{se}(2)$ is corrected using relative pose

$$\mathbf{u}(k) = \mathbf{z}_o^{-1}(k) \circ \mathbf{z}_o(k-1)$$

the input matrix $\mathbf{B}(k)$ is the transformation matrix that project from the body coordinate to the initial frame as

$$\mathbf{B}(k) = \begin{bmatrix} \cos\theta(k-1) & -\sin\theta(k-1) & 0 \\ \sin\theta(k-1) & \cos\theta(k-1) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The process noise $\mathbf{w}(k)$ is assumed to be drawn from a zero mean multivariate normal distribution

$$\mathbf{w}(k) = [w_x(k), w_y(k), w_\theta(k)]^{\mathbf{T}} \sim \mathcal{N}\left(0, \Delta_d^2(k)\mathbf{Q}_k\right)$$

where $\Delta_d(k) = \sqrt{\Delta x_b^2(k) + \Delta y_b^2(k)}$. Here we assume the variance is proportional to the change of position.

$$\begin{cases} x(k) = x(k-1) + \Delta x_b(k)\cos\theta(k-1) - \Delta y_b(k)\sin\theta(k-1) + w_x(k) \\ y(k) = y(k-1) + \Delta x_b(k)\sin\theta(k-1) + \Delta y_b(k)\cos\theta(k-1) + w_y(k) \\ \theta(k) = \theta(k-1) + \Delta\theta_b(k) + w_\theta(k) \end{cases}$$

$$\Delta\mathbf{z}_b(k) = [\Delta x_b, \Delta y_b]^{\mathbf{T}}$$

$\mathbf{w}(k) = [w_x(k), w_y(k), w_\theta(k)]^{\mathbf{T}}$ is the process noise at time k which is assumed to be drawn from a zero mean multivariate normal distribution $\mathcal{N}\left(0, \Delta_d^2(k)\mathbf{Q}_k\right)$ where. The variance here reflect our estimate of the change of the data. Here we assume big variance on the position data and yaw angle since the data would most likely to change while the robot is moving. We assign small variance on roll and pitch angle because our robot is moving on a roughly flat surface so the roll and pitch angle won't change much.

The measurement equation is used to represent the front and rear GPS measurement in the initial frame $\mathbf{z}(k) = [\mathbf{z}_f, \mathbf{z}_r]^{\mathbf{T}} = \left[z_x^f(k), z_y^f(k), z_x^r(k), z_y^r(k)\right]^{\mathbf{T}}$. Here the front and rear GPS location have constant offset l from the center of the robot base and could be computed from robot pose $\mathbf{q}$. It's represented as

$$\begin{cases} z_x^f(k) = x(k) + l\cos\theta(k) + v_{z_x^f}(k) \\ z_y^f(k) = y(k) + l\sin\theta(k) + v_{z_y^f}(k) \\ z_x^r(k) = x(k) - l\cos\theta(k) + v_{z_x^r}(k) \\ z_y^r(k) = y(k) - l\sin\theta(k) + v_{z_y^r}(k) \end{cases}$$

$$\mathbf{v}_o(k) = \left[v_{z_x^f}(k), v_{z_y^f}(k), v_{z_x^r}(k), v_{z_y^r}(k)\right]^{\mathbf{T}} \sim \mathcal{N}\left(0, \Delta_t^2(k)\mathbf{Q}_k\right)$$

where $\mathbf{v}_o(k)$ is the measurement noise at time k which is assumed to be drawn from a zero mean multivariate normal distribution $\mathcal{N}\left(0, \Delta_t^2(k)\mathbf{R}_k\right)$. We introduce$\Delta_t$ to represent the time elapsed between k and $k-1$ timestamp and the variance of $\mathbf{v}_o$ will increase with time.

Another challenge for the EKF design is that sensor data come at different frequency. In order to accommodate this need, we design the Asynchronous EKF algorithm that shows in 1. The algorithm will only update the filter when at least one measurement of each sensor arrived with a timestamp later since last updated timestamp. The update timestamp t is set as the minimum of three latest timestamp for each sensor. We get each sensor measurement at the update timestamp through interpolation. Then the prediction step and update step of the EKF is executed to update the pose and uncertainty at timestamp t.

Algorithm 1 Asynchronous Extended Kalman Filter

Input : time series $\Delta\mathbf{z}_b = (\Delta x_b, \Delta y_b, \Delta\theta_b)$, $\mathbf{z}_f = \left(z_x^f, z_y^f\right)$, $\mathbf{z}_r = \left(z_x^r, z_y^r\right)$
Output : time series $\mathbf{q} = (x, y, \theta)$
while at least one measurement of each sensor $(\Delta\mathbf{z}_b(t_o), \mathbf{z}_f(t_f), \mathbf{z}_r(t_r))$ arrived since t^{old} **do**
  $t = \min(t_o, t_f, t_r)$
  $\Delta\mathbf{z}_b(t) \leftarrow \text{Interpolation}\,(\Delta\mathbf{z}_b(t_o), t)$
  $\mathbf{z}_f(t) \leftarrow \text{Interpolation}\,(\mathbf{z}_f(t_f), t)$
  $\mathbf{z}_r(t) \leftarrow \text{Interpolation}\,(\mathbf{z}_r(t_r), t)$
  $\mathbf{q} \leftarrow \text{Update}\,(OdometryModel, \mathbf{z}_c(t))$
  $\mathbf{q} \leftarrow \text{Update}\,(TwoGPSMeasurementModel, \mathbf{z}_f(t), \mathbf{z}_r(t))$
  $t^{old} = t$
end while

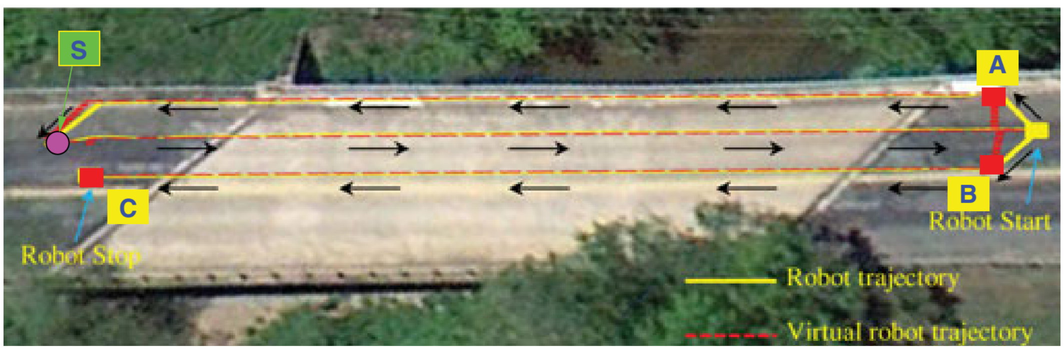


Figure 2.14: Bridge deck inspection robot coverage path planning[76]

2.5.2 Motion planning and control

The goal of the motion planning and control is to generate the desired trajectory for the robot and then to control the robot to follow the trajectory precisely. We first plan the coverage path for the bridge deck surface to generate a series of way points. We then implement a simple controller for waypoint following.

2.5.2.1 Path planning

The inspected bridge is assumed to be straight and the bridge deck area is assumed to be of a rectangular shape which are valid for most bridges. The robot motion planning is indeed a coverage planning problem [80]. A boustrophedon decomposition, also the so-called "ox plowing motion" or trapezoidal decomposition in robotics research, is used. 2.14 illustrates the robot motion on a bridge and a brief description is presented here to illustrate how to generate the robot motion trajectory. To cover the desired deck area, say the half of the bridge deck surface shown in Fig. 2.14, three GPS points at the rectangle corners noted as points A, B, and C are first obtained either through google map or in field measurement. Using the GPS coordinates of these three corners, the zigzag-shape robot motion trajectories are computed by the trapezoidal decomposition algorithm, as the arrows indicate in the figure. Waypoints are generated by interpolation with constant interval along the road direction and each waypoint represent the data collection location required by the NDE sensors.

2.5.2.2 Robot motion control

The path planning module generate a series of waypoint for the robot to follows. To complete the task from current pose to the desired pose which is the next waypoint, we implement a feed-forward and feedback framework for the trajectory controller. The controller output a speed to the low level controller provided by vendor for execution. A trapezoidal acceleration and deceleration algorithm is used to generate the desired linear and angular velocity $\mathbf{v}_{rv}$. This desired velocity is used as feed-forward signal together with the pose feedback controller which is essentially a P controller.

2.6 Experimental results

2.6.1 Localization experiment

The robot navigation system was extensively tested on Rutgers Busch campus before it was deployed on bridge decks. Fig. 8 illustrates one example of the

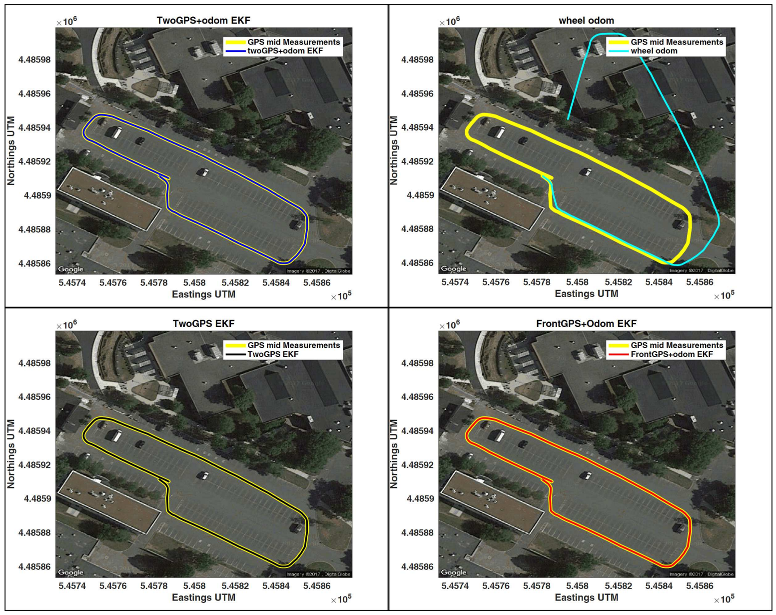


Figure 2.15: Comparison between pure odometry integration and extended kalman filter (EKF) with odometry and GPS combinations without noise in GPS. (topleft) EKF with odometry and two GPS; (topright) Pure odometry integration; (bottomleft) EKF with both front and back GPS; (bottomright) EKF with odometry and front GPS.

comparison of the results of the navigation systems based on the EKF design discussed in Section IV-A. To show the redundancy effect, we evaluate performance of navigation system on odometry only, EKF based on two GPS, EKF based on one GPS and odometry and EKF based on two GPS and odometry. We recorded actual data on the campus parking lot for simulation and the average of the front and rear GPS position are used as the ground truth. First, It is clearly shown in the top right plot in 2.15 that the odometry only method accumulate significant error along the whole trace while that the EKF-based navigation system based on combination of GPS and odometry shows a close agreement with ground truth. We then add simulated noise to the front GPS signal in order to evaluate the robustness of the navigation system. We could see from 2.16 and 2.17 the EKF based on two GPS and odometry shows the closest agreement with ground truth while others are affected if the filter include corrupted front GPS measurement.

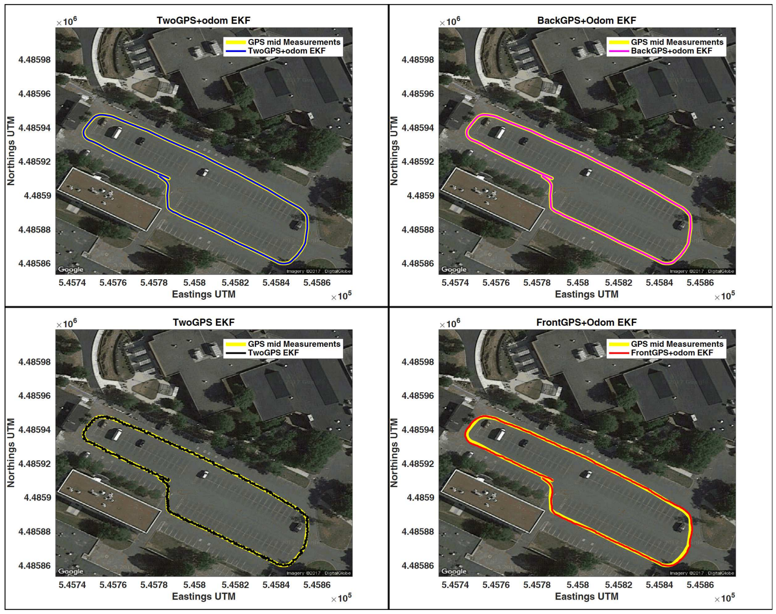


Figure 2.16: Comparison between extended kalman filter (EKF) with odometry and GPS combinations with noisy front GPS. (topleft) EKF with odometry and two GPS; (topright) EKF with odometry and back GPS; (bottomleft) EKF with both front and back GPS; (bottomright) EKF with odometry and front GPS.

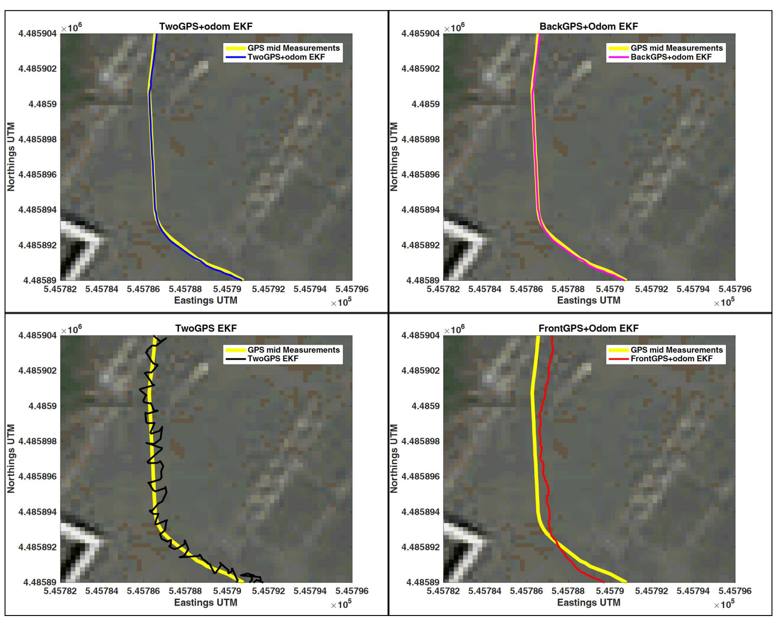


Figure 2.17: Closeup comparison between extended kalman filter (EKF) with odometry and GPS combinations with noisy front GPS: (topleft) EKF with odomety and two GPS; (topright) EKF with odometry and back GPS; (bottomleft) EKF with both front and back GPS; (bottomright) EKF with odometry and front GPS.

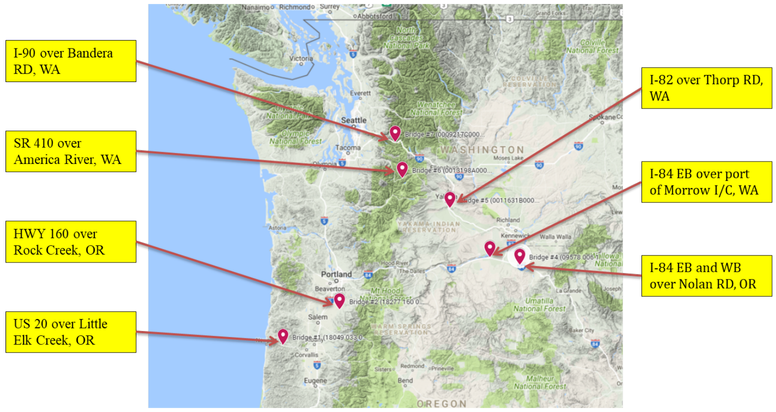


Figure 2.18: Google Map Location of bridge surveyed during trip to Washington state and Oregon state

2.6.2 Field Deployment

The RABIT has been deployed to more than 50 bridges across United States. 2.18 shows the inspected bridge during a trip to Washington State and Oregon state where we inspected 8 bridges in two weeks.

One of the example NDE survey results for the bridge are shown in 2.19. The top plot shows the resistivity map from the electrical resistivity sensor that indicating the corrosion rate. The middle plot shows the GPR condition essentially the reflection wave attenuation from the top rebar level which is an indication about corrossive environment and possible delamination. The bottom plot shows the impact echo condition map that correspond to the delamination condition. The numbers shown in these maps are calculated by using the NDE sensing data. The plotting colors are based on these calculated numbers to indicate the different deterioration severity levels with respect to delamination, corrosion rate, and the overall condition. Hot colors (reds and yellows) are an indicator of delamination, while cold colors (greens and blues) are an indicator of likely fair or good conditions. The benefits of having the condition maps from multiple complementary NDE sensors are obvious. We could conclude from three plot the bridge is in bad condition as all three condition map show large area of red. The

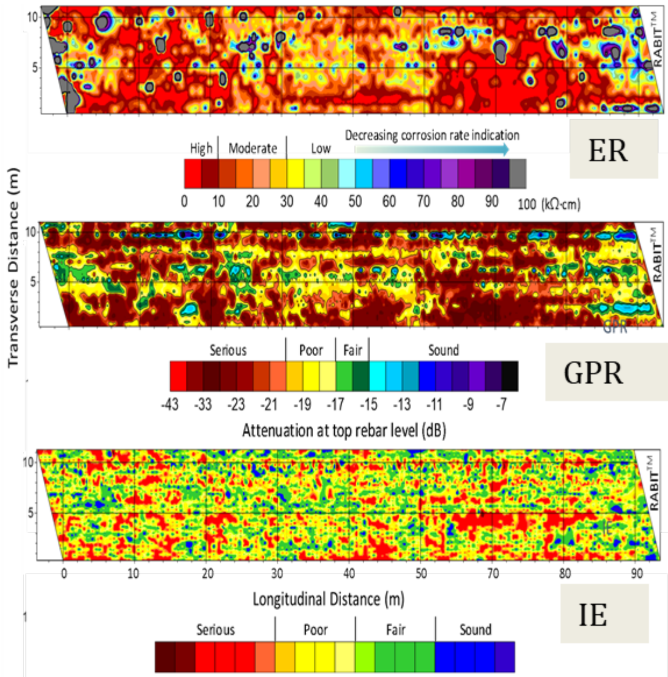


Figure 2.19: (top) Electrical resistivity map; (middle) GPR condition map; (bottom): Impact echo map[50]

bottom right of the electrical resistivity map and GPR map have dominant red areas indicating highly corrosive environment and thus probable high corrosion rate. The same area in the impact echo map shows severe delaminations. These correlations confirm that the primary cause of deterioration and delamination is the highly corrosive conditions at these locations.

2.7 Conclusion

The development an autonomous robotic system were presented for bridge deck inspection and evaluation. The main objective of the autonomous robotic NDE system is to increase the inspection efficiency, accuracy, and reduce the risk to bridge inspectors. The developed autonomous inspection system was built on a omni-directional mobile robot platform and integrated with multiple NDE technologies such as GPR, impact echo, and electrical resistivity. The robust robot localization system was built on the EKF-based fusion of the RTK GPS, and wheel odometry measurements. The robot motion control was designed through the combination of feed-forward and feedback control. The robotic system performance was validated through extensive experimental testing and field deployment.

Chapter 3

Automatic Image Mosaicing For Bridge Deck Surface Reconstruction

3.1 Introduction

The bridge scanning robot integrates multiple non-destructive evaluation (NDE) sensors including impact echo, ground penetrating radar, resistivity probe and high definition camera. Among all the techniques, a detailed panorama image of the bridge provides the cornerstones for the civil engineers to assess the bridge condition. A well formed panorama image can help researchers and engineers to identify the area with deterioration and cross-validate with other NDE sensor measurement. There are numerous researches conducted in the past decades in the field of image stitching or, in the planar scene case, image mosaicing [44], but the main focus is on generating a visually appealing panorama. In our application, our goal is to generate a geo-referenced panorama that serves as the reference for other NDE data source. Therefore, unlike traditional image stitching, we focus more on accuracy of the image registration. Also there is limited overlap between images since the image data collection need to be coordinated with other NDE sensors. This poses a challenge on the robustness of the image mosaicing algorithm and aggravating the error accumulation. Another challenge comes from the concrete bridge deck appearance which don't have many reliable features preventing traditional method to perform reliably.

In this chapter, we present a new image mosaicing system for bridge deck surface reconstruction. A framework of the image mosaicing system is proposed consisting function modules, such as frame registration, local map optimization, global optimization, and panorama creation. SURF blob detector and binary feature descriptors BRIEF are chosen to extract the corresponding features between neighboring images. Navigation data introduced in the feature-based image registration gives a huge performance boost in terms of feature matching accuracy and speed. By combining the navigation data and feature-based image registration in the graph optimization framework, our proposed approach inherits the drift-less nature from GPS while still maintains local accuracy of feature-based image registration. We validate the accuracy of the system through physical experiment on real bridges. This work has been published in [89] and the author contribute to the majority of research and writing.

The rest of the chapter is organized as follows. We first review the related work in the field of image stitching and bridge deck scanning. We then introduce our bridge scanning robot setup in section 3.3. We start presenting our mosaicing algorithm in the overview section 3.4, followed by registration of individual frame in section 3.5, local map optimization in section 3.6 and global optimization in section 3.7. Image blending process is described in section 3.8. Validation of our method on real bridge test data is presented in section 3.9 before the conclusive summary in section 3.10.

3.2 Related works

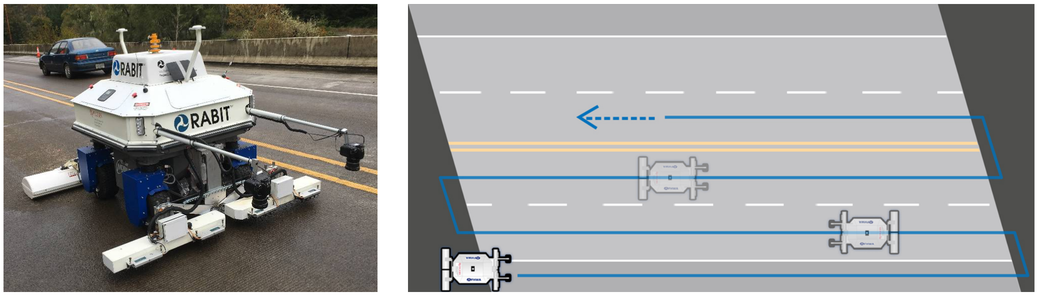
During the past two decades, various robots were developed for bridge inspection and maintenance. In [92], a robotic system was developed for removing the paint of bridge bottom. In [90], a similar system was used for imaging the bottom surface of bridge. Those systems usually are the integration of industrial robot with the peeper crane which still require human intervention. In [77][76], the Robotics Assisted Bridge Inspection Tool (RABIT) was developed for autonomous nondestructive bridge deck inspection. Up to date, the RABIT system have been used to scan more than fifty bridges across the states.

Cameras are widely used in bridge deck or road scanning [18]. The scan is done using either line scan camera [91] or area camera [99]. Line scan cameras are known for its short exposure time and fast data acquisition rate, thus often integrated onto specialized vehicles to scan the road surface [91]. The line scan cameras are often triggered by encoder pulses and require synchronization between the the camera and registration data. One drawback for line scan cameras is that it is difficult to attain perfect match due to no overlap between line scans. In contrast with line cameras, area cameras, which are widely available, take discrete snapshot of the ground plane. The images taken sequentially usually have overlapped area. An image registration algorithm is later performed to assemble multiple images into a panorama image. This process is referred as image stitching, or in the planar case image mosaicing.

Image mosaicing could be categorized as a special case of scene reconstruction where images are related by planar homography only. In [44], an in-depth review is provided for the existing image mosaicing algorithms. Mosaicing involves two major step in processing: registration and blending. Registration refers to the process of align multiple images to the reference coordinate based on the calculated geometric transformation. Though attempts have been made to overcome the registration errors by utilizing blending, the significance of accurate registration in image mosaicing still remains unquestionable [44]. There are two major ways for frame to frame image registration: area-based registration and feature-based registration [123]. Area-based method, such as [93], computes the pixel similarity between windows of two images. It is sometimes preferred because it makes the optimal use of information available since it measures the contribution of every pixel. But this also make it vulnerable to the illumination change. It also has a limited range of convergence comparing with feature-based method. Feature-based method, such as [12], relies on the extracted salient feature from the images to compute the geometric transformation between images. It is considered more robust than the area-based method in case of illumination variation and doesn't have the convergence range problem. But in real application scene, it is necessary to choose between different blob-like features, e.g. SURF [9] and corner-like features, e.g. ORB [118] to achieve good result.

Navigation information alone is not accurate enough for image stitching in most case. It is sometimes used jointly with image registration, such as in aerial imaging [131] or seabed imaging [36]. In [131], position is only used heuristically to determine the potential overlapped image. In [36], navigation information is used as the initial guess for the optimization to align the images.

Visual SLAM has received significant advancement which is closely related to the image mosaicing field. In [101], a SLAM system using ORB feature was proposed and represents the most reliable SLAM system up to date. A graph optimization framework g^2o [74] was used to carry the local and global bundle adjustment. SLAM usually rely on video input which assumes relative small frame to frame movement. While it's not directly applicable to our setup, our system builds on the graph optimization framework and the concept of local map in [101].



(a) (b)

Figure 3.1: (a) Inspection robot camera in the extended position for scanning; (b) Ox-plow scan trajectory

3.3 Image mosaicing for bridge scanning robot

3.3.1 Bridge scanning using camera

The bridge inspection robot is equipped with two high definition Nikon DSLR camera to capture the scene with the capture area overlapped about 30 percent. Figure 3.1a shows the camera in the inspection mode where two linear actuators extend the camera rods to the scanning position. The scanning pattern is shown in Figure 3.1b. The robot follows a coverage trajectory and will change heading direction when shifting scanning lanes. The robot stops roughly every 2 feet to capture the sensor data including taking photos, while at the same time GPS location of robot is stored and associated with the sensor data. In practice, the accuracy of GPS varies between 5cm to 30cm. The stop pattern results in 20 percent of overlap between sequential images. There is also about 25 percent overlap in the images between neighboring lanes.

There are several coordinate system involved here. GPS data could be converted to the Cartesian UTM coordinate with respect to the UTM zone. Since the surveyed bridge is usually far from the UTM origin, large number in coordinates will lead to poor numerical stability in the later optimization stage. For this reason, we need to convert UTM coordinate to a close world coordinate. The world coordinate is setup on the ground with origin at the robot initial position as shown in Figure 3.1b, with x axis pointing towards traffic direction and y axis pointing across the bridge. The $x - y$ plane in the world coordinate is also used as the composting plane for output of the final panorama. The robot body frame is setup in the center of the robot where the GPS measures. The camera frame is the frame with the origin in the camera optical center and $\Delta \in \mathbb{R}^3$ represent the constant offset of the camera in the robot body frame. This offset is measured manually and used as a constant parameter in our setup.

3.3.2 Problem statement

We consider the image mosaicing problem for the bridge scanning robot. The following assumption is made: 1. The robot make stops roughly every 2ft; 2. GPS data is logged together with the captured images when stops; 3. The robot make parallel runs and patterns are known. Our goal is to accurately stitch the images and generate a geo-referenced panorama image for the whole bridge.

3.4 Method overview

3.4.1 Feature detector and descriptor choice

There are numerous feature detector and descriptors invented in the last decades for finding the correspondence point between images. Feature detectors are looking for salient point or regions in the image and could be roughly grouped as corner detectors and blob detector. Given that we have a high definition image of the bridge deck, blob detector could detect more features since the concrete surface is tend to have more blobs than corners. Moreover our test shows that SURF extracts a more evenly distributed feature points set than the corner feature detector such as ORB. For those features detected, we need to extract descriptors that reflecting the local region of the feature. Here instead of using the original SURF descriptor, we use the binary feature descriptors BRIEF to generate the descriptor. We find out that BRIEF descriptor not only outperforms SURF descriptor in speed and accuracy [15], but it is also more robust to illumination changes which is suitable for our the outdoor environment with plenty of shade and illumination change. We validate the feature performance in our test scene in the experiment section.

In order to improve the robustness and speed of feature matching, we take advantage the special pattern of our motion planning to use the orientation independent feature descriptor. This is done by pre-aligning images in orientation with the compositing plane. The robot has roughly the same heading direction within one scanning lane and change heading direction only when shifting lanes. This enables us to discard the orientation information in the descriptor and receive a performance boost in terms of feature matching accuracy and speed.

3.4.2 Map representation: data structure for map points and frames

The map is represented as a graph of frames and map points. For each image captured using left or right camera, a frame $\mathcal{F}_i$ is created to store:

- The camera rigid body pose $\mathbf{T}_{wi} \in \mathbf{SE}(3)$.
- Map points set $\boldsymbol{\mathcal{M}}_i$ detected in the frame.
- The navigation data including 2D position and heading direction associated with the frame when captured.
- Image $\mathcal{I}_i$, which is undistorted and pre-aligned roughly in orientation with respect to the compositing plane.

- The camera intrinsic matrix $\mathbf{K}_i$ for image $\mathcal{I}_i$ including focal length and principal point. $\mathbf{K}_i$ is transformed from original camera intrinsic matrix $\mathbf{K}$ since $\mathcal{I}_i$ is pre-aligned in orientation with respect to the compositing plane. $\mathbf{K}$ is obtained through camera calibration for both left and right camera.

- A plane $\mathcal{P}_i$ that represent local planar surface.

- Neighboring frames and relative transformations.

A map point m_j is created by triangulating corresponding feature points in different images. A map point m_j stores:

- 3D position $X_{m_j} \in \mathbb{R}^3$ in the world coordinate system.

- The image frames that observe this map point and its corresponding feature location and octave in that image. We avoid storing store every SURF feature and its descriptor, instead we save the "index" of the SURF point.

- The reference frame that the map point is first triangulated. We rely on the reference frame to infer the point after the pose graph optimization.

3.4.3 System overview

Our system, as show in Figure 3.2, includes five main modules: individual image registration, local map optimization, global optimization including pose graph optimization and global map optimization, and final stitching. When a new image is retrieved from the database, it goes through the image registration for initial pose estimation. The new image is first prepossessed to create the frame $\mathcal{F}_i$ with proper data structure association and pre-aligned with compositing plane. We then search in the frame database for the closest frame with $\mathcal{F}_i$. Here frame database includes all the frames that have been successfully registered. The initial

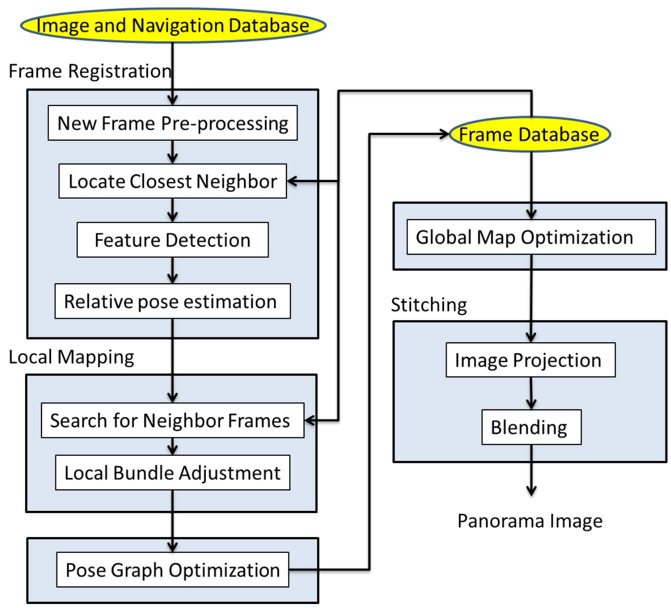


Figure 3.2: Image mosaicing system Overview

pose of $\mathcal{F}_i$ and its associated map points $\mathcal{M}_i$ are obtained through the feature-based alignment with the closet frame. The initial pose of $\mathcal{F}_i$ is used for searching the potential overlap frames and uses feature based method to find the relationships between $\mathcal{F}_i$ and its neighbors. A local map optimization module is used to refine the $\mathcal{F}_i$ and the local map. The final step of the registration is using the pose graph optimization to further refine all the frames in the database. Before creating the final panorama, an optional global map optimization could be carried out to fine tune the map points and frame poses.

3.5 Image registration for individual frame

When a new image is retrieved from the data set, we need to align the image with the images already registered. We use a two step approach to register the image. We first search for the closest frame in the frame database based on navigation data. The closest frame is used to estimate the initial pose of the new frame. We then estimate the relative transformation between the new frame and its potential neighbors. This constructs a more densely connected graph and increases the robustness in the optimization stage. Map points are triangulated based on the relative pose estimation. In the second step, we use a local bundle adjustment to further refine the pose of the new frame and new map points. Relative pose is stored between neighboring frame and later used for the pose graph optimization.

3.5.1 Corresponding feature extraction

For a pair of potential neighbor frames, feature extraction serves as a foundation for the pose estimation. To estimate the pose robustly, the corresponding feature points extracted need to have low false correspondence rate and well spread across the overlapping area. We choose SURF blob detector to detect the features and the orientation independent BRIEF descriptor to extract the features as described above. By incorporating the navigation data, we could further improve the performance by reducing the search region in the image.

To make sure the feature detected is well spread even in case of presence of shade and illumination variation, we set a low Hessian threshold for the SURF blob detector and generate a dense set of feature candidates. In the matching stage, Hamming distance is used to match the BRIEF descriptor of SURF point. Ratio test and symmetry test are applied to suppress the false correspondence, providing cleaner data for the pose estimation. The above procedure extract plenty of corresponding feature points across the overlapping area but may with variable density. This variability could cause overweighting in the correspondence rich region of the images. Also excessive number of map points will increase the burden of optimization procedure with map points. So we use non-maximal suppression to downsize the detected correspondences with uniform density while keeping the more salient ones.

3.5.2 Relative camera pose estimation

Since we assume a locally planar scene between overlapping images, we could estimate the homography from the corresponding feature points. Homography matrix $\mathbf{H}_{ri}$ represents the relationship between two images of the planar scene

$$\mathbf{H}_{ri}\begin{bmatrix} p_i \\ 1 \end{bmatrix} = \begin{bmatrix} p_r \\ 1 \end{bmatrix}$$

where p_i and p_r are the corresponding feature point in the new image $\mathcal{F}_i$ and reference image $\mathcal{F}_r$. Here we estimate $\mathbf{H}_{ri}$ through 4-point RANSAC algorithm [38] from the corresponding features. The homography matrix alone defines relationship between images and so it is used to stitch the images [123]. This approach only works for a few images but will fail in case of large image set if the accumulated error is not addressed. So in our approach, we further decompose the homography to estimate the transformation between images. This provide us the advantage to address the accumulated error in the optimization stage by incorporating navigation data. $\mathbf{H}_{ri}$ is related to the transformation elements $\mathbf{R}_{ri}$ and $\mathbf{t}_{ri}$ and to the normal of the plane $\mathbf{n}_{ri}$ through

$$\lambda\mathbf{H}_{ri} = \mathbf{K}_r(\mathbf{R}_{ri} + \mathbf{t}_{ri}\mathbf{n}_{ri}^T)\mathbf{K}_i^{-1}$$

where $\mathbf{K}_i$ and $\mathbf{K}_r$ are the camera calibration matrix for the aligned image. Here $\mathbf{K}_i$ and $\mathbf{K}_r$ are transformed from the original camera intrinsic matrix $\mathbf{K}$ because of images went through clockwise or counter-clockwise rotation in the initial alignment. Here camera pose $\mathbf{R}$ and $\mathbf{t}$ could be recovered up to scale through SVD decomposition.

A number of verifications are then performed to ensure that this homography corresponds to a valid camera motion:

1. Check that the relative pose does not include improper rotations.

2. Check the normal of the local planar scene are within certain range.
3. The number of SURF correspondences between the two images must be larger than a certain threshold.

3.5.3 New map point creation

We then triangulate point according to the feature point correspondence. If the feature point is already associated with a map point, then that map point is recaptured with new observation otherwise we create the new map point by triangulation. The triangulated point should be roughly on one plane, so any point far from the plane will be discarded. Since the pose estimation only recovers the pose up to scale, we re-scale the cameras and feature points by adjust the camera to plane distance to a measured value. This provides a good initial guess for the optimization which will further refine the map point position and camera pose.

3.6 Local map optimization

Local map optimization is often referred as local bundle adjustment. A bundle adjustment procedure will optimize the map points positions and camera poses by minimizing a cost function. Local bundle adjustment is used to make sure the new camera position and map points are coherent with the existing camera positions and map points. Using the neighboring frames and points only, we avoid optimizing on too many edges and vertexes in the graph optimization which takes a much longer time to process.

3.6.1 Neighbor frames search

Before constructing a local map, we need to get the relationship between the new frame and its neighbors. To get the potential neighbors, we search in the frame database for the candidates within radius r of the new frame based on both the navigation data and estimate pose. We iterate the process in section 3.5 to find out the relationship between neighboring frames and co-visible map points. Those findings add more constraints in the optimization stage to form a strong connection between frames.

3.6.2 Local map construction

One of the key steps for the local bundle adjustment is to construct a local map. The local map should strike a balance between complexity and completeness. Our local map contains the currently processed frame, all the neighboring frames, and all the map points observed by those frames. All other frames that see the map points but are not neighbors to the currently processed frame are included in the optimization but remain fixed [101].

3.6.3 Constraints

Constraints represent the relationships between frames and map points. The constraints defined here are not only used for local map optimization, but also for the pose graph optimization and global map optimization.

3.6.3.1 Map point observation constraints

For every map point included in the optimization, it corresponds to several observations that made from neighboring frames. Each observation forms a constraint. We define the world coordinate of map point m_j as $X_{m_j} = (x_{m_j}, y_{m_j}, z_{m_j})$ and ith frame poses as $\mathbf{T}_{iw} \in \mathbf{SE}(3)$, where w denotes the world reference. $\mathbf{u}_{i,j} \in \mathbb{R}^2$ is the observation of the map point j in the image frame i. The error term for this observation is

$$\mathbf{e}_{i,j} = \mathbf{u}_{i,j} - \pi(\mathbf{T}_{iw}, X_{m_j})$$

where π_i is the projection function

$$\pi(\mathbf{T}_{iw}, X_{m_j}) = \begin{bmatrix} \frac{u_{i,j}}{w_{i,j}} \\ \frac{v_{i,j}}{w_{i,j}} \end{bmatrix}$$

$$\begin{bmatrix} u_{i,j} \\ v_{i,j} \\ w_{i,j} \end{bmatrix} = \mathbf{K}_i(\mathbf{R}_{iw} X_{m_j} + \mathbf{t}_{iw})$$

where $\mathbf{R}_{iw} \in \mathbf{SO}(3)$ and $\mathbf{t}_{iw} \in \mathbb{R}^3$ are the rotation and translation parts of $\mathbf{T}_{iw}$, and $\mathbf{K}_i$ is the camera intrinsic parameter matrix associated with frame $\mathcal{F}_i$. The cost function to be minimized given the map point set $\boldsymbol{\mathcal{M}}$ is

$$\mathbf{E}_{\boldsymbol{\mathcal{M}}} = \sum_{m_j \in \boldsymbol{\mathcal{M}}} \sum_{\mathcal{F}_i \in \mathscr{F}_j} \left\| \mathbf{e}_{i,j}^T \boldsymbol{\Omega}_{i,j}^{-1} \mathbf{e}_{i,j} \right\|_\delta$$

where $\boldsymbol{\Omega}_{i,j}$ is the covariance matrix associated with the octave at which the keypoint was detected, $\mathscr{F}_j$ is the frame set that seen the map point m_j and $\|.\|_\delta$ is the Huber norm

$$\left\| r^2 \right\|_\delta = \begin{cases} \frac{r^2}{2\delta} & \text{if } |r| \leqslant \delta \\ |r| - \frac{\delta}{2} & \text{otherwise} \end{cases}$$

3.6.3.2 Plane constraints

A plane $\mathcal{P}_i$ is associated with each frame $\mathcal{F}_i$ representing the planar surface observed. The plane constraint is added to ensure all the map points in the same image stay roughly on one plane during optimization. This constraint is necessary whenever we add the map point constraints. Different from highly overlapped frame used in SLAM problem, the frames here have little overlap in between causing the scale to drift from frame to frame. The plane constraint helps to maintain the scale between two frames by forcing the map points observed in one image stay in one plane. For each map point m_j, we add the point to plane constraint that restrain the point to lie on the plane. The error term is

$$\mathrm{e}_{i,j} = \mathbf{n}_i^T X_{w,j} + d_i$$

where $\mathbf{n}_i$ is the unit normal of the plane and d_i is distance from the world origin to the plane. The cost function is

$$\mathbf{E}_{\boldsymbol{\mathcal{P}}} = \sum_{\mathcal{P}_i \in \boldsymbol{\mathcal{P}}} \sum_{m_j \in \mathcal{M}_i} \left\| \lambda \mathrm{e}_{i,j}^2 \right\|_\delta$$

where λ is a scalar representing the confidence in the local flatness of the observed scene. Huber function is used again to mitigate the influence of the outlier. Since $\mathrm{e}_{i,j}$ is represented in meters and we assume the map point lies within $1cm$ from the plane, we set the $\delta = 0.01$ and $\lambda = 100$ empirically.

3.6.3.3 Navigation data constraints

The navigation information especially the position is coming from the GPS. Although accuracy varies, the GPS measurements are not subject to the accumulated error. Our frame to frame estimate from the image is quite accurate thanks to the high definition camera but even small errors could build up causing drift in the long run. Those two source of pose estimation are mutually complement, so we try to combine the best of both. Here we add the navigation constraints to reduce the drift of the image registration. For each frame $\mathcal{F}_i$ we use the position from the navigation data and forming the error function as

$$\mathbf{e}_i = \mathbf{t}_{wi} - (\mathbf{R}_w^N \Delta + \mathbf{t}_w^N)$$

here $\mathbf{R}_w^N$ and $\mathbf{t}_w^N$ represent navigation data in the world coordinate and $\Delta \in \mathbb{R}^3$ represent the constant offset of the camera in the robot body frame. The cost function for the given set of frames $\mathscr{F}$ is

$$\mathbf{E}_{\mathcal{N}} = \sum_{\mathcal{F}_i \in \mathscr{F}} \mathbf{e}_i^T \Sigma_i^{-1} \mathbf{e}_i$$

here Σ_i represent the covariance of the navigation data. For the isolated frames which don't have overlap with any other frames, the navigation constraint serves as the only constraint to the frame.

3.6.3.4 Co-visible frame pose constraints

For two neighboring frames $\mathcal{F}_i$ and $\mathcal{F}_j$, we could define the error as

$$\mathbf{e}_{i,j} = \log_{\mathbf{SE}(3)}(\mathbf{T}_{ij}\mathbf{T}_{jw}\mathbf{T}_{iw}^{-1})$$

where $\mathbf{T}_{ij}$ is the relative transformation computed from the local bundle adjustment, $\mathbf{T}_{iw}$ and $\mathbf{T}_{jw}$ are the world pose of frame $\mathcal{F}_i$ and $\mathcal{F}_j$, and $\log_{\mathbf{SE}(3)}$ transforms the error to the tangent space so that the error $\mathbf{e}_{i,j} \in \mathbb{R}^6$. The cost function to be optimized is

$$\mathbf{E}_{\mathcal{G}} = \sum \mathbf{e}_{i,j}^T \theta_{i,j} \mathbf{I} \mathbf{e}_{i,j}$$

where $\theta_{i,j}$ is the weight of the edge and could be set as the number of common map points. In the map point triangulation step, we perform a non-maximal suppression to keep the map point evenly distributed. The number of retained corresponding feature points is based on the area of bounding rectangle of all the corresponding feature points in the frame. This procedure is try to maintain a constant map point density across the map. Therefore the weight $\theta_{i,j}$ of the edge is roughly proportional to the overlapped area.

3.6.4 Cost function for local map optimization

Given the local map consisting local map points $\boldsymbol{\mathcal{M}}_L$ and local frame set $\mathscr{F}_L$, the cost function of the local bundle adjustment could represent as the combination of map point observation constraints, plane constraints and navigation constraints

$$\mathbf{E} = \mathbf{E}_{\boldsymbol{\mathcal{M}}_L} + \mathbf{E}_{\boldsymbol{\mathcal{P}}_L} + \mathbf{E}_{\mathcal{N}_L}$$

where $\boldsymbol{\mathcal{P}}_L$ and $\mathcal{N}_L$ are the planes constraints set and navigation constraints set associated with the local frame set $\mathscr{F}_L$ respectively.

3.7 Global optimization

After performing the local map optimization, we use global optimization to refine the estimation of map. We implement a fast routine optimization using the pose graph and navigation data and an optional global map optimization procedure that include all the frames and map points.

3.7.1 Pose graph optimization

An undirected weighted graph is used to represent the pose graph that connect all the neighboring frames. Each node is a frame and an edge between two frames exists if they share enough observations of the same map points. The edge contains the relative pose information that obtained from local bundle adjustment. Performing a pose graph optimization would distribute the error along the graph. The pose graph optimization includes the navigation data constraints to eliminate drift. So the cost function for the pose graph optimization would be

$$\mathbf{E} = \mathbf{E}_{\mathcal{G}} + \mathbf{E}_{\mathcal{N}}$$

where $\mathbf{E}_{\mathcal{G}}$ is the graph constraints set for the whole pose graph and $\mathbf{E}_{\mathcal{N}}$ is the navigation constraints set for all the nodes.

3.7.2 Global map optimization

After all the frames are registered, we could do an optional map optimization to further refine the estimate of map points. The whole map optimization include all the frames and map points and take a longer time to process. The cost function of the global map optimization includes the map point observation constraints, local plane constraints and navigation constraints

$$\mathbf{E} = \mathbf{E}_{\boldsymbol{\mathcal{M}}} + \mathbf{E}_{\boldsymbol{\mathcal{P}}} + \mathbf{E}_{\mathcal{N}}$$

where $\boldsymbol{\mathcal{P}}$ and $\mathcal{N}$ denote the plane constraints set and navigation constraints set associated with the all the frames, and $\boldsymbol{\mathcal{M}}$ denotes all the map points.

3.8 Panorama image creation

In order to obtain a panorama image, we need to project the images onto a planar surface. The $x-y$ plane in the world reference frame is chosen as the compositing surface. So we need to estimate the homography for each frame $\mathcal{F}_i$ that serves the projection

$$\mathbf{H}_{wi} \begin{bmatrix} u_{i,j} \\ v_{i,j} \\ 1 \end{bmatrix} = \begin{bmatrix} x_{m_j} \\ y_{m_j} \\ 1 \end{bmatrix}, \; for \; m_j \in \mathcal{M}_i$$

where $\mathbf{u}_{i,j} = (u_{i,j}, v_{i,j})$ is the image coordinates of the map point m_j observed in frame $\mathcal{F}_i$. Again, we could use the RANSAC method to estimate $\mathbf{H}_{wi}$ from the map point set $\mathcal{M}_i$ associated with each frame $\mathcal{F}_i$.

We then use simple feathering to blend the projected images. Feathering is a blending technique to do a weighted average with a distance map. It weights pixels near the center of the image more heavily and down-weights pixels near the edges. Therefore, we could mitigate the difference caused by small mis-alignment and illumination change.

3.9 Results

3.9.1 Feature detector and descriptor comparison

We compare the blob feature detector SURF and corner detector ORB in Figure 3.3. We choose a example scene including plenty of illumination change which reflect the typical outdoor experiment. We adjust the parameters so that SURF and ORB detected roughly the same amount of feature points. From Figure 3.3, we could see ORB corners are concentrated in the region with high texture but didn't generate enough point in low texture area. Especially in the shading area where the contrast is low compared the non-shaded area, there are only a few points generated. In comparison, the SURF detector generally well spreads the points with no short-coming in the shaded area.

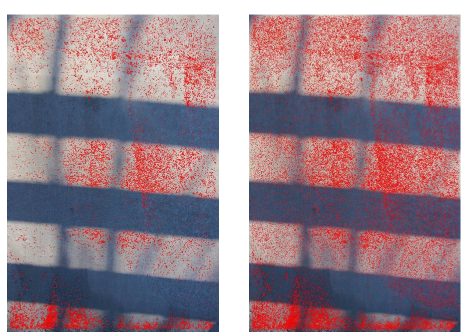
Table 3.1: Comparison of different feature descriptors in point correspondence detection

	SURF	U-SURF	BRIEF	BRIEF Guided
Corr. Correspond.	915	1763	4435	**4628**
Corr. Ratio	77%	90%	95%	**96**%
Comp. time (sec)	21.1	14.6	14.2	**7.2**

We compare our navigation-guided corresponding point extraction with SURF, Upright SURF (U-SURF) and BRIEF descriptor in Table 3.1. The comparison uses brutal force matcher to match the points detected by SURF detector. We use L1-norm to match the descriptor in the SURF and U-SURF case and Hamming distance in the BRIEF case. From Table 3.1, we can see that orientation independent descriptor U-SURF and BRIEF, enabled by initial alignment, perform better and faster than the original SURF. Meanwhile, more than two times point correspondences are extracted using BRIEF than U-SURF. Our BRIEF method guided by navigation performs the best which outperform the original BRIEF in processing time and also extract more correspondences.

3.9.2 Validation of image registration accuracy

In order to assess the accuracy of image registration, we paint an array of dots manually on the bridge that are evenly spaced with 2 feet. Here we assumed the painted dots are accurate and used as benchmark to evaluate the final panorama. The test scene is a $110\text{ft} \times 10\text{ft}$ area of a concrete bridge scene with plenty of shade and illumination change. A zoomed-in portion in the panorama is in shown in Fig 3.4 including 13×4 images. The grid is added later on to show how the dot deviated from its nominal position. The white dots in the whole panorama are then extracted from the image and compared with a grid in Figure 3.5. We can see from both Figure 3.4 and Figure 3.5 that dots are correctly aligned with the grid. The error term representing the distance from nominal positions is distributed as normal distribution $\mathcal{N}(1.51\text{in}, 0.54\text{in}^2)$. The results indicate our proposed algorithm could generate an accurate mosaicing image even in presence



(a) (b)

Figure 3.3: Feature detector comparison shows blob detectors are more evenly spread than corner detector. (a) ORB corner detector; (b) SURF blob detector

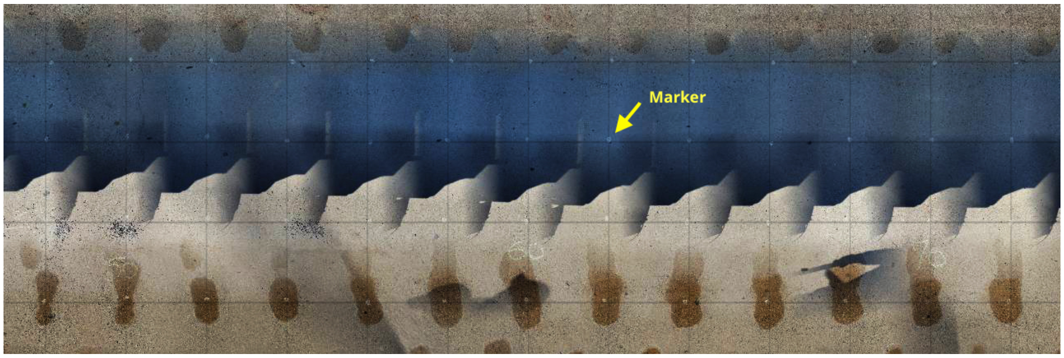


Figure 3.4: Validation experiment based on painted point arrays

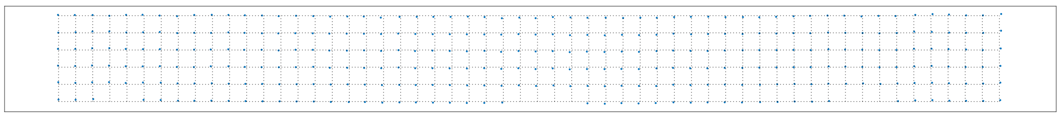


Figure 3.5: Dots extracted from the panorama compared with grid for its nominal positions

of shade and illumination change.

3.9.3 Stitched panorama

Here we also provide a complete stitched panorama for the north bound of OR 213 (HWY 160) over Rock Creek, as shown in Figure 3.6. The test was performed on a overcast day so there is little shadow in the image set. The bridge surface measured as 68ft $\times$ 20ft including 240 images. Each image has the resolution of 3456×2306 pixels. We could see the structure lines including bridge joints remain straight with little distortion which is an indication of good registration.

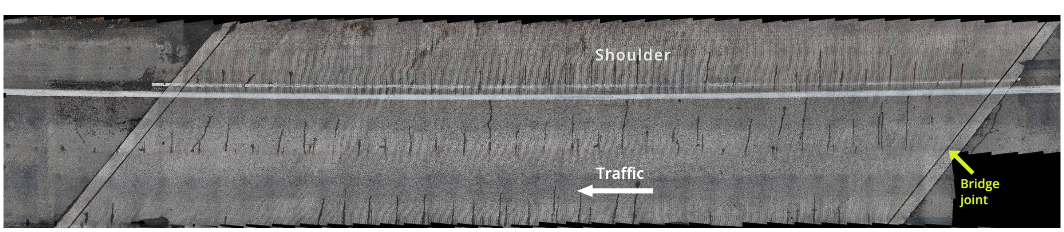


Figure 3.6: Produced panorama image for one side of the bridge deck

3.10 Conclusion

In this chapter, we present a new image mosaicing system for the bridge deck surface reconstruction. By fusing the navigation data with feature-based image registration in the graph optimization framework, our proposed approach inherits the drift-less nature from GPS while still maintaining local accuracy of feature-based image registration. We evaluate the accuracy through quantitative test on real bridges and show our system is robust to interference in the outdoor environment such as illumination variation. Given the accuracy achieved, our system could not only be used to generate geo-referenced panorama but also serve as the reference for other NDE sensor data registration.

Chapter 4

Bridge Rehabilitation Robot

4.1 Introduction

Defects such as delaminations are one of the biggest problems in deterioration of bridge decks. They are basically horizontal cracks in the concrete that occur mainly due to rusting of steel rebars. These delamination reduce the strength of the concrete significantly. Figure 4.1 below shows a core from a bridge that shows the delamination. Since they normally do not have any surface openings before the late stage such as spalling, they are very difficult to detect and repair in the early stage.

The state of the art practices of repairing delamination are extremely high cost and labor intensive. Figure 4.2 shows an example of the current practice to repair a block of damaged bridge deck. The repair process is completely manual and involves removing and then replacing the entire patch of damaged bridge deck. This process is time consuming leading to extended lane closure, low efficiency

Figure 4.1: Cored samples from a bridge deck showing delamination

Figure 4.2: Current late stage defect repair practices

since it also remove some portion that still in good condition, and posing safety risk to the field engineer and construction workers.

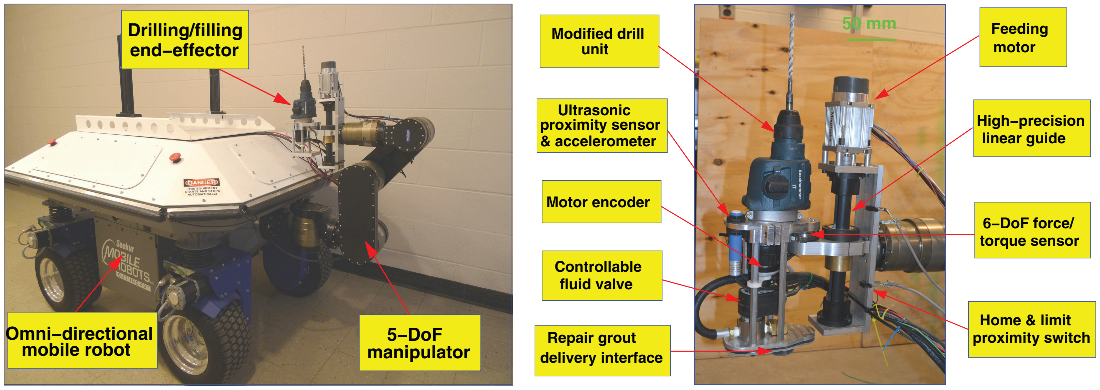
The Automated Non-destructive Evaluation and Rehabilitation System (ANDERS) project supported by National Institute of Standards and Technology (NIST) aims to provide a uniquely comprehensive tool that will transform the manner in which bridge decks are assessed and rehabilitated. While the Non-Destructive Evaluation (NDE) robot RABIT introduced in the first chapter provided a solution for early stage defect detection, the non-destructive rehabilitation (NDR) system is aiming at delivering a non-destructive, rapid, cost effective rehabilitation at an early stage of deterioration. The NDR system uses robotics and automation for precision and rapid delivery of novel inorganic composite of alkali alumina-silicate matrices reinforced with nano/micro fibers for rapid, non-destructive repair and rehabilitation of bridge decks.

The rest of the chapter is organized as follows. In the next section, we give an overview of the NDR system developed for the ANDERS project. In section 3, we introduced the end-effector design of the drilling and filling system. In section 4, we introduce the mobile manipulator system and its planning algorithm. In section 5, we present the various experimental validation result for the NDR system.

4.2 Overview

In order to achieve precise minimal invasive rehabilitation on the bridge deck, we design the NDR system composed of three main parts the mobile robot base, the 5 degree of freedom manipulator and customized designed end-effector for robotic drilling and grout filling. We use the same Seekur robot from Adept Mobile Robots Inc. as the RABIT inspection robot introduced in Chapter II. The seekur robot have zero turning radius and high payload which is suitable executing the heavy duty repairing on the narrow bridge surface. To further improve the precision in rehabilitation, we mount a 5DoF manipulator from Schunk on mobile base to position the end-effector at the desired pose. The manipulator helps to offset the positioning error from the mobile base and could reach the precision of 5cm. Moreover, we design the end-effector to fulfill the requirement of minimal invasive rehabilitation which involves drilling and filling module. The end-effector requires a compact design that satisfy the payload and power constraint of the mobile manipulator design.

The operating procedure of NDR system start once received the targeted defect location in GPS coordinate. The on-board motion planning algorithm plan the motion path and control the mobile base to pose that ensure the targeted defect location is within the workspace of the manipulator. This relieve the precision control requirement for the mobile base which subject to heavy load and slower response. At the defect locations, the robotic manipulation will first use the drilling unit to drill several small holes with a pre-specified geometry (e.g., triangle) at certain depths to reach the crack locations inside the bridge deck. The feeding of the drill unit is provided by a separate motion control unit, rather



(a) (b)

Figure 4.3: (a) Mobile manipulator-based autonomous rehabilitation platform; (b) Closed-up view of the customized end-effector

than by the manipulator. The same robot manipulator will rotate its end-effector such that through the drilled holes, the material filling mechanism mounted on the same end-effector will facilitate delivery repair materials into small hairline crevasses and form strong and durable bonds with the parent material of the bridge decks.

4.3 End-effector design

The early-stage deterioration of bridge decks starts beneath the surface. To repair the delamination inside concrete deck, we drill 3/8 inch holes on the concrete and high-strength concrete grout is injected and filled these cracks. The end-effector is designed to complete the drilling and filling that rehabilitate the defect area. The end-effector is composed of the drilling module, filling module and the linear stage that provide the progression of the drilling unit and also apply suitable force to the seal of the filling unit. The weight of end-effector is constrained by payload capacity of the manipulator and hence we want to reduce the weight on the end-effector and make it as compact as possible.

4.3.1 Drilling module

4.3.1.1 Drill selection

To drill on concrete or rocks, hammer drill should be used. Besides the drill bits are different in geometry, the difference between a hammer drill and a regular drill for metal is that a hammer drill creates vibratory motion of the drill bit while it rotates. There are two types of hammer drills, one with a simple mechanical component to generate the vibratory motion of the drill bit and the rotary hammer drill which uses a pneumatic piston to create a powerful impact to drive the drill bit. Fig. 4.4 shows a drilling time comparison in experiments to drill a 6 mm (diameter) hole on concrete mortar with a depth of 76 mm using a regular hammer drill and a rotary hammer drill sets. Various thrust forces are used in the experiments. From the results shown in the figure, it is clearly observed that: (a) it takes much less time for the rotary hammer drill than that for the regular hammer drill set for drilling the same hole; and (b) the drilling time for the rotary hammer drill is independent with the applied thrust force, while for the regular hammer drilling, a larger thrust force results in a shorter drilling time. The impact-induced drilling process such as the one generated by rotary hammer drill is also called percussive drilling. In chapter 5, we present the models to capture the percussive drilling and also use these modeling development to interpret and optimize the drilling performance.

From the experiments shown in Fig. 4.4, it is obvious that the rotary hammer drill is the choice for a faster drilling process than the regular hammer drill though the former is more expensive than the latter.

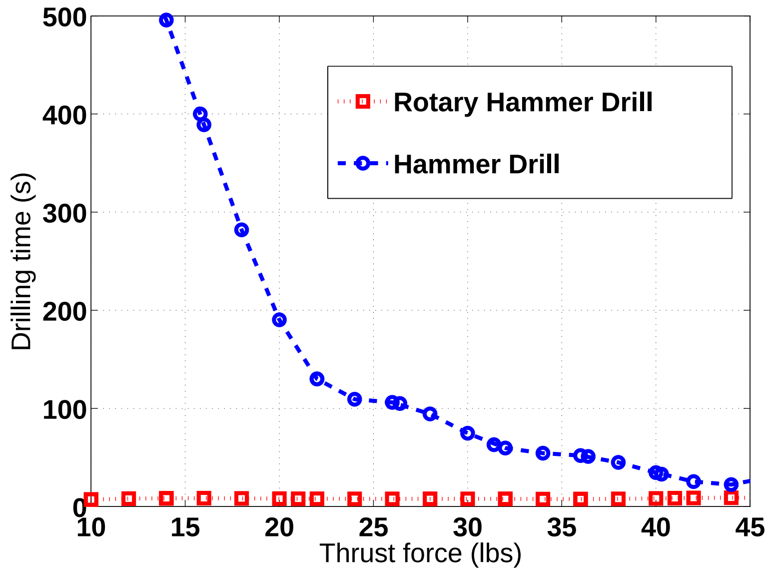


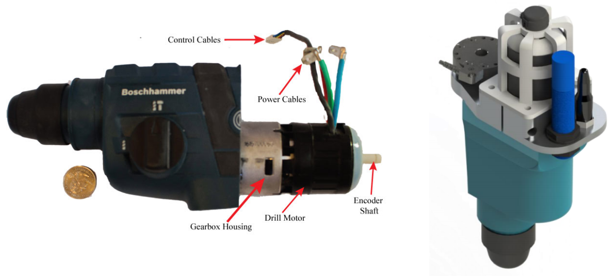
Figure 4.4: Comparison of drilling time for drilling a hole on concrete mortar with a depth of around 76 mm.

4.3.1.2 Drill customization

The rotary hammer drill is a powerful tool and usually has a bigger size than the regular hammer drill. Considerable effort is made to reduce the bulky size while adding additional sensor.

The Bosch drill is stripped out the battery and controller part as shown in Figure 4.5a. We only keep the essential mechanical component and the driving motor of the rotary hammer. The manual trigger is replaced by the digital and analog output from the onboard computer. The battery is moved to the inside of the mobile robot to reduce the payload of the drill. The drill is clamped around the gearbox housing by two piece of customized mounting plate as shown in Figure 4.5b. Those two mounting plate also serves as the mounting point for additional sensors.

In order to monitor the drill motor speed, we need to add an encoder to the drill. The encoders with friction wheel is too big to fit with the drill, we therefore decided to measure the drill speed from the back through co-axis mounting. We



(a) (b)

Figure 4.5: (a) Bosch rotary hammer drill exposed for modification and control; (b) Custom drilling unit assembled with sensors such as encoder, force torque sensor, ultrasonic distance sensor and accelerometer.

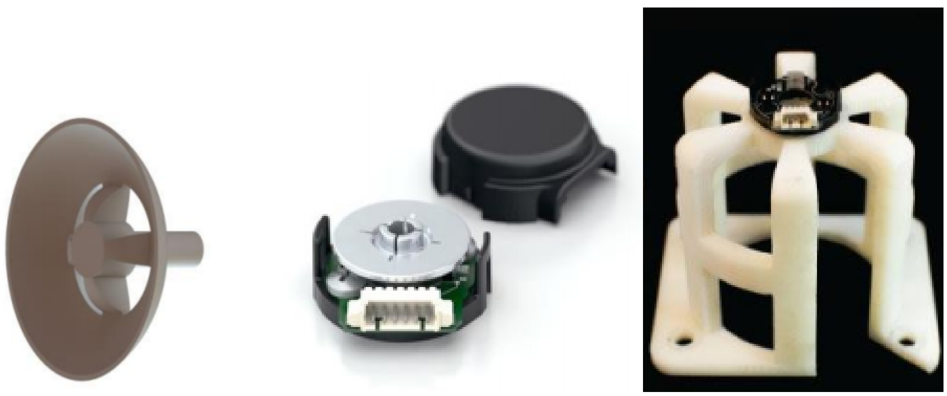
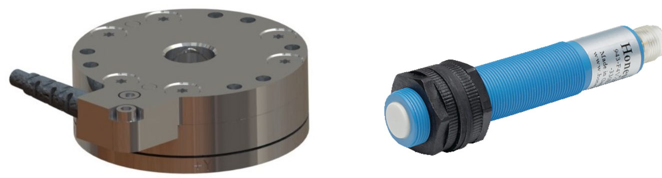


Figure 4.6: Custom drill speed measurement. (left) 3D printed cap for connection of the motor back shaft and encoder; (middle) Encoder from US Digital that could measure up to 25000 rpm; (right) 3D printed fixture that attach the encoder with the drilling unit.

first measure the shaft speed of the drill chuck through friction wheel encoder and determined as 6300-6500 rpm. Since the gear ratio between the motor and the chuck is 42:13, therefore the maximum rotation speed of the motor is determined as around 21000 rpm. In order to measure the high rpm, we select the digital encoder from US Digital E4P-125-250-N-S-D-T as shown in the middle of Figure 4.6. We 3D print the shaft cap according to the contour of the motor back as show on the left side of Figure 4.6 and glued to the back of the motor. We also 3D print the mounted structure of the rotary encoder as shown on the right side of Figure 4.6 and the shaft is tightly fitted to the encoder rotary disk.



(a) (b)

Figure 4.7: (a) ATI force-torque sensor that measure the drilling and filling force; (b) Honeywell ultrasonic sensor that measure the distance from the end-effector to ground

4.3.1.3 Force-torque Sensor

In order to measure and control the force applied to the drilling and filling unit, we add a force-torque sensor from ATI Industrial Automation (9105-TW-MINI45-ERA-2.5) as shown in Figure 4.7a. The ATI force-torque sensor is selected for its compact size and wide force range in z direction with a maximum allowable over load of ±2300 lbf. The sensor is mounted between the clamping mounting plate and the linear stage as shown in Figure 4.5b and Figure 4.3b with its z-axis parallel to the drilling axis.

4.3.1.4 Ultrasonic sensor

Although we could estimate the height of end-effector through forward kinematics of the mobile manipulator, that estimation is subject to deflection of mobile robot suspension and the resulting accuracy varies. We therefore add an ultrasonic sensor to the end-effector for determine the distance of the end-effector to the ground. The Honeywell's industrial ultrasonic sensor 943-F4Y-2D-1C0-300E as shown in Figure 4.7b is chosen because it's small, dust-proof and waterproof. It has a measure range of 3.9 inch to 31.4 inch which is suitable for the application. It needed to be mounted facing down while drilling and hence it was mounted on the drill mount as shown in Figure 4.5b.

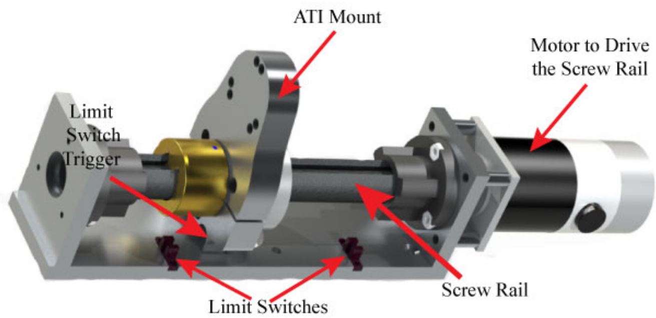


Figure 4.8: Custom made linear Stage that provide the feeding of the drilling and filling procedures

4.3.1.5 Linear stage

The use of the hammer drill for effectively and efficiently drilling of concrete materials will bring a large vibratory motion, which could potentially damage the Harmonic Drive gearbox inside the manipulator. So we decide to use a linear stage to provide the progressive motion while keeping the manipulator in the magnetically locked position.

We designed our custom linear stage to move the drilling unit and filling unit as shown in Figure 4.8. We choose a high torque but low speed brushed motor from Maxtor and a screw rail from Haydon-Kerk for its light weight and compact design. The screw rail has a stroke length of 4 inch which could fulfill the requirement of drilling a 3 inch depth hole. A custom made c-clamp is used to connect the linear guide with the drilling module through the ATI force torque sensor. The c-clamp is also doubled as the trigger of the limit switches that added to avoid collision between the moving stage and screw rail end. We use aircraft grade Aluminum Alloy 7075 to construct all the mounting structure to reduce weight and provide higher resistance to bending. Finally, the linear stage is mounted to the manipulator using an adapter plate.

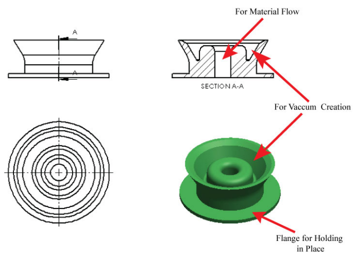
4.3.1.6 Material delivery unit

Concrete repair usually use polymer resins as repair material. The polymer resins is not mechanically compatible with concrete substrate and therefore the repairs quality is questionable. Pumping these material into hairline cracks is very difficult due to high viscosity. The ANDERS material development team identified that the nano/micro inorganic composites for rapid, non-destructive repair of thin delaminations was successful using alkali alumina-silicate matrices reinforced with nano/micro fibers [71]. They demonstrated that certain mixes of these gave a material with good mechanical properties and had desired flow for the hair-line cracks as thin as 0.03 inches [70].

Following the manual delivery result in [71], we design our automated material delivery unit that could fulfill the requirement to inject the matrices into the 3/8 inch hole that drilled by the rotary hammer. There are several constraints need to fulfilled in this design. First the unit need to be small and light for robot mounting. Second, the system need to provide 75 psi maximum pressure which need for injecting the matrices into hairline crack. Third, the system should be resistant to the highly alkaline alumina-silicate matrix.

We use the back-end of the drilling unit as the delivery side as shown in Figure 4.3b, so the manipulator only need to flip the orientation the end-effector to do filling instead of drilling. To minimize the weight, the functional filling part on the end-effector is essentially a custom made seal and a solenoid valve to control the fluid. The solenoid valve is connected through the 1/4 tube with the pump and then the tank storing the matrices. The pump and tank are mounted remotely on the mobile base.

The hole had to be properly sealed when the matrices was pumped into the hole, therefore we custom designed our seal since the mechanical fixture presented in [71] is infeasible due to the size. The seal is designed based on the concept



(a)

(b)

Figure 4.9: (a) Seal design three side view; (b) Diaphragm pump that used for filling unit

of vacuum cups as shown in Figure 4.9a. and . The seal was 3D printed using silicone shore A40 by Stratsys μPrint.

We choose KNF's micro diaphragm liquid pump to provide the pumping as shown in Figure 4.9b. This is a positive displacement type pump so any back-flow is essentially prevented. The pump delivers maximum 300 ml/min and can operate at max 87 psi pressure. It has self-priming suction of about 8.86" Hg and can run dry and is also extremely chemical resistant.

4.4 Mobile manipulator

The mobile manipulator combine the complement advantages of the mobile robot and the manipulator. The mobile robot provides mobility to the manipulator essentially extend the workspace of the manipulator to infinity. The manipulator provides extra degree of freedom to the mobile robot that enables more dexterous task.

In the NDR task, the mobile manipulator is required to put the end-effector at designated position and orientation. The position is required by the GPS location of the defect and the specific orientation is required for either the drilling or filling task.

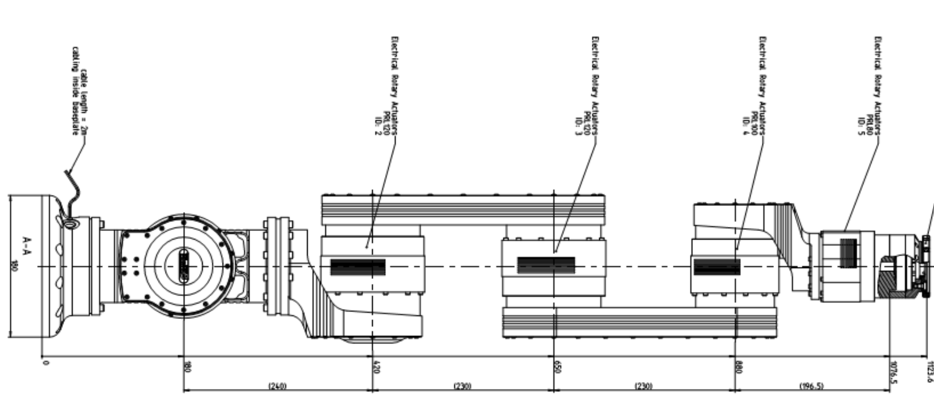


Figure 4.10: 5 degree of freedom arm configuration from Schunk

4.4.1 Mobile base

The mobile base shares similar design as the RABIT inspection robot. A omni-directional Seekur robot from Adept is used as mobile base for its high payload and tight turning radius. The robotic system is equipped with two RTK GPS units and four wheel encoders. Similar as RABIT in Chapter 2, the GPS data are fused with wheel encoder measurements through an EKF design and could reach accuracy of 2cm.

4.4.2 Manipulator

The manipulator provide more flexibility to the system by adding extra degree of freedom allowing for more delicate task. To achieve full 3D position and orientation of the end-effector, we need a manipulator with at least 6 joints. But since our task such as drilling and filling are only concerning the 3D position and direction, we could use a 5 joints manipulator to fulfill our task requirement. We choose the Schunk powercube arm as our manipulator for its high payload-weight ratio and modular design. The configuration of our arm is shown in Figure 4.10. Forward kinematics and inverse kinematics are solved for the configuration while considering the collision constraint.

4.5 Mobile manipulator planning

The mobile manipulator composed of the mobile base and manipulator provide more flexibility for the task. The goal for mobile manipulator planning is to enable efficient planning to achieve desired accuracy of 2cm. Although the localization system on the robot provide accuracy about 2cm, the movement accuracy of mobile base is around 10-15 cm due to the large payload and limit in the low level controller. The manipulator however could provide millimeter grade accuracy once the inverse kinematics are solved because of high precision joint motor and rigid connection between joints. We design our mobile manipulator planning algorithm as two decoupled steps for mobile base and manipulator separately. First step is move the mobile base such that the desired defect location is within the workspace of the manipulator. This step is guided by the mobile base motion planner and EKF-based localization system. Second step is to move the manipulator to the position for the rehabilitation based on the current mobile base position and orientation. This step is executed by the inverse kinematic module and joint motion planner.

As the first step to plan the motion of the mobile robot, we first define three region in the 2D plane in the world as shown in Figure 4.11a. The area I is the manipulator end-effector workspace on the ground given the current mobile robot position and orientation. This end-effector workspace on the ground describes the ground area that could be reached by the tip of the drill bit in the uppermost position. The area I is generate through simulated inverse kinematic calculation and it's actually an symmetric area between two co-concentric arcs. The workspace is constrained by the geometry of the manipulator and also the collision between links and mobile base. If the target hole position is within the workspace, the robot will only command the manipulator to the desired position as shown in

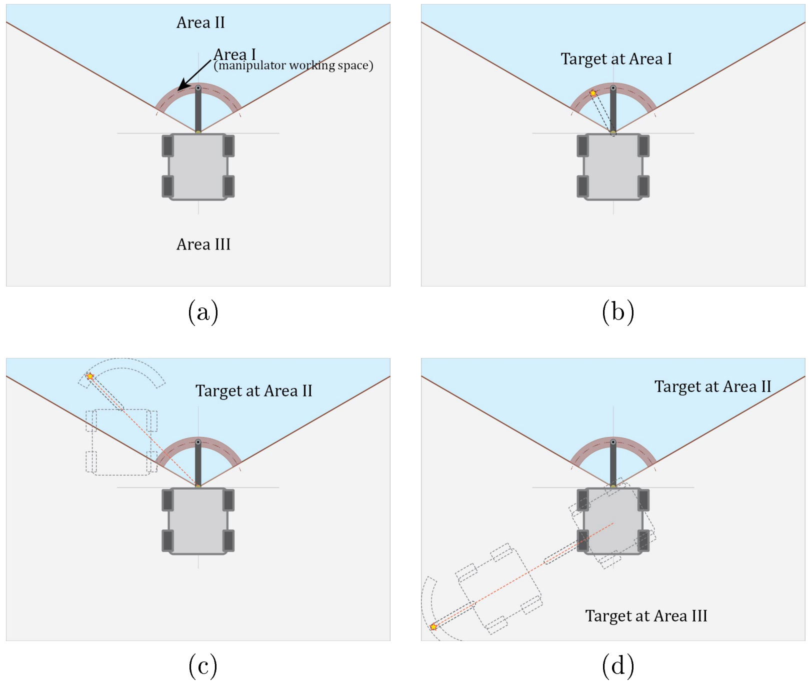


Figure 4.11: Mobile Manipulator Planning: (a) Workspace decomposition; (b) Target in area I; (c) Target in Area II; (d) Target in Area III

Figure 4.11b. The area II is defined as the region within the central angle of the workspace arcs but not belongs to area I. As shown in Figure 4.11c, when the target falls in the area II, the robot will first command the mobile base move parallely to the location where the target fall in the new end-effector workspace area. The robot then command the manipulator to move to desired position for the drilling. The area III is defined as the rest of the 2D planar space. As shown in Figure 4.11d, the robot first orient the base to the target defect position and then move forward to the pose that the target is within the end-effector workspace.

Once the mobile robot moves to the position where the target falls in the end-effector workspace, we don't move the tip of the drill directly on the ground surface because the tip might be hitting the ground if the ground is not perfectly flat. Therefore, we first move the manipulator to about 50cm above the desired drilling position and then use the Honeywell ultrasonic sensor to measure the distance to the ground. The robot then moved the end-effector to drill the target hole according to drilling procedure shown in Algorithm 3. Then the robot drill three release holes around the center hole so that once the cracks are filled up the matrices will come out of the release holes. Once finish the drilling, the robot start the filling procedure by using the filling unit on the other end of the end-effector as shown Algorithm 4.

Algorithm 2 GPS-guided drilling and filling

Input: 2D GPS location in UTM Cartesian coordinate $\mathbf{x}$, desired hole depth d
repeat
  OrientRobotTowardTarget($\mathbf{x}$);
  MoveRobotTowardTarget($\mathbf{x}$);
until $\mathbf{x}$ in the workspace of manipulator
$h \leftarrow$ MeasureDistToGround($\mathbf{x}$);
DrillHole($\mathbf{x}, h, d$);
for i=0;i<3 **do**
  $\Delta\mathbf{x} \leftarrow \left[r\cos\frac{2i\pi}{3}, r\sin\frac{2i\pi}{3}\right]$;
  DrillHole($\mathbf{x} + \Delta\mathbf{x}, h, d$);
end for
FillHole($\mathbf{x}, h, d$)

4.5.1 Drilling procedure

The drilling procedure assumes the target is within the reach of the manipulator and takes in the target location as UTM coordinate, the ground to drill tip distance measured from the ultrasonic sensor and the desire hole depth. The arm moves the drill tip to be 1cm above the desired drilling position, the linear stage then start to push the drill toward the ground. We will trigger the rotary hammer once the static force applied measured from ATI force torque sensor is higher than the threshold. This is necessary to avoid the drill bit wobbling on the concrete surface causing awkward situations such as bending. The feeding motor will push the drill until the drilling depth is above the required 3 inches and then pulled the drill up to the upper limit switch position.

Algorithm 3 Drilling procedure for DrillHole($\mathbf{x}, h, d$)

```
Input: 2D coordinate in UTM coordinate x, ground to arm distance h, desired hole depth d
MoveArmOverDrill(x, h);
repeat
    FeedingMotorFwd(P_1);
until F > threshold;
TurnOnDrill(ω);
repeat
    FeedingMotorFwd(P_2);
until depth > d
repeat
    FeedingMotorBwd(P_3);
until back micro switch triggered
TurnOffDrill();
```

4.5.2 Filling procedure

The task for the filling procedure as shown in Algorithm 4 is to grantee a smooth matrices delivery by precisely aligning the nozzle with drilled hole and forming proper seal while deliver the matrices. The filling procedure takes in the distance to the ground measured from the Honeywell ultrasonic sensor and the target filling position which is the center hole drilled in the drilling procedure. The robot moves the filling unit to the position where the seal is 5cm above the drilled hole and aligned in the vertical direction. Since the fill is at its back-most position and the feeding motor start to push the seal towards the hole. Once the force measured from the ATI force torque sensor is above the empirical threshold that we could ensure the nozzle is properly sealed with hole, the feeding motor stops the progression and the pump and solenoid is turned on to deliver the matrices to the drilled hole. We used a fixed empirical time interval to determine the filling time and once the filling stops, the feeding motor move the filling unit to its back-most position which defined by the limit switch.

Algorithm 4 Filling procedure for FillHole($\mathbf{x}, h$)

Input: 2D coordinate in UTM Cartesian coordinate $\mathbf{x}$, the distance to ground h
MoveArmOverFill($\mathbf{x}, h$);
repeat
    FeedingMotorBwd(P_1);
until $F >$ threshold
TurnOnFill(t);
repeat
    FeedingMotorFwd(P_3);
until front micro switch triggered

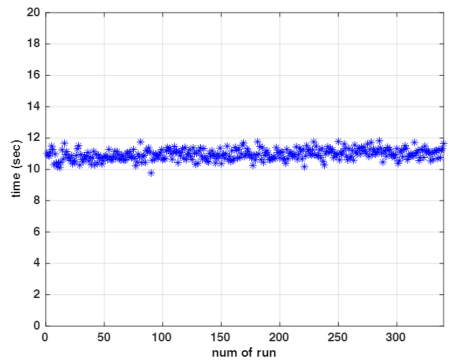
4.6 Experimental results

4.6.1 Drilling reliability test

Since the manipulator drilling system is subject to strong vibration, we test the drilling system through extensive testing. We run marathon indoor testing to verify the drilling performance due to wearing of the drill bit and also the reliability of the manipulator drilling. We drilled a total of 360 holes on 10 concrete blocks without stop, the drill result is shown in Figure 4.12a. We also plot the drilling time for each run in Figure 4.12b and we could see the drilling time doesn't change with respect to the wearing. This could be contribute to the percussive motion of the drill bit plays a major role in drill progression.

4.6.2 Material delivery test

In order to test the material delivery to the delaminations, we make our artificial crack by gluing two bricks together using silicone and have a thin 3mm aluminum plate as spacer as shown in Figure 4.13a. We drilled two holes on the concrete block, one as filling hole, the other as release hole. Figure 4.13b shows a successful test where the matrices comes out of the release hole.



(a) (b)

Figure 4.12: Marathon reliability test: (a) Drilled block after 360 drilling; (b) Drilling time for 3 inches holes over 360 runs

(a) (b)

Figure 4.13: Material delivery test. (a) Artificial crack created by gluing two concrete blocks; (b) Successful material delivery where matrix comes out of the left release hole.

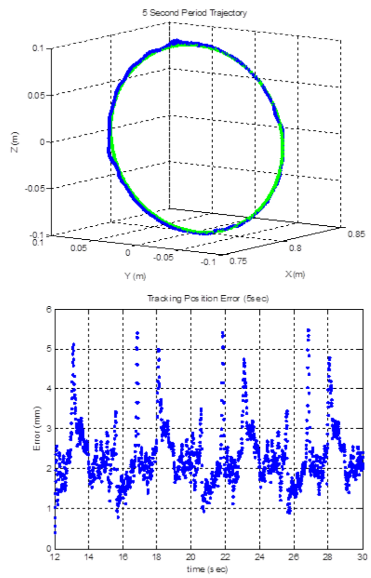


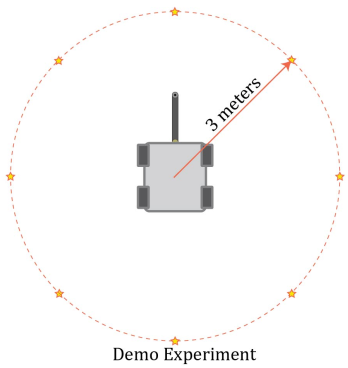
Figure 4.14: Manipulator trajectory tracking performance

4.6.3 Manipulator trajectory tracking test

We test the trajectory tracking of our Schunk 5 degree of freedom manipulator through following the 3D circle while maintaining the constant orientation of the end-effector. We test on the same 3D circle but with different period as 5 second and 10 second. The result shows in 4.14 as the left is for the 5 second period circle and the right is for the 10 second period circle. The bottom shows the tracking error separate for two cases with varying speed reference trajectory.

4.6.4 Mobile manipulator rehabilitation precision test

We validate the precision of the mobile manipulator rehabilitation on campus as shown in Figure 4.15b. We set up the 8 rehabilitation points along a circle with 3 meter radius as shown in Figure 4.15a. We measure the GPS locations of all marked points and send to the robot through WiFi. The robot performs rehabilitation procedures at these 8 points in sequence. The drilling/filling accuracy is illustrated in the Figure 4.16 where we measure the accuracy of rehabilitation



(a)

(b)

Figure 4.15: On-campus navigated rehabilitation test: (a) Navigated rehabilitation precision test setup (b) Test scene

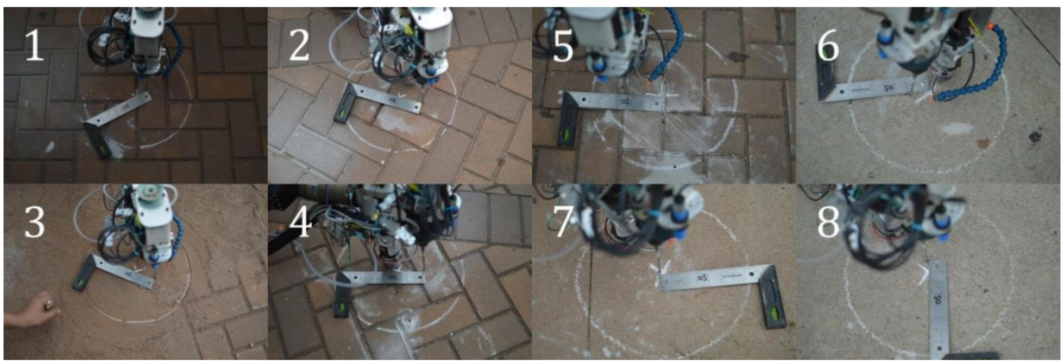


Figure 4.16: Measuring navigate rehabilitation accuracy on campus

points. The mean and variance of the rehabilitation position error are 5.23 cm and 2.46 cm, respectively.

4.6.5 On bridge field testing

4.6.5.1 Pohatcong bridge

We conduct our field rehabilitation test on the Pohatcong Creek bridge deck in November of 2013 as shown in Figure 4.17a. The NDR system performed smoothly to conduct the drilling and filling tasks. The rehabilitation sites are within 3-4 cm range of the targeted locations. Figure 4.17b shows one of the three drilled/filled holes and the red cross is the targeted location.

(a) (b)

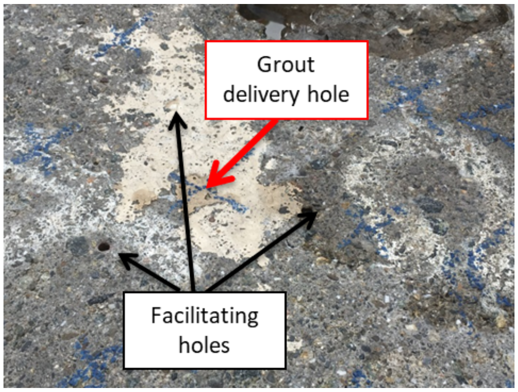
Figure 4.17: Pahatcong Creek bridge test: (a) Rehabilitation in process; (b) One of three rehabilitation sites where the red cross mark is the required position and the hole filled with matrix is the actually rehabilitation position.

Figure 4.18: The NDR robot on the testing bridge on September 12, 2016.

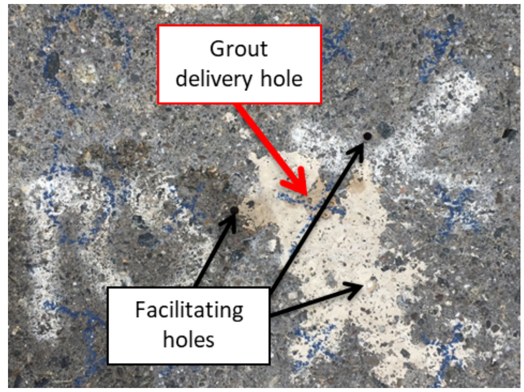
4.6.5.2 Bridge No. 1618152

Following the suggestions of the NDE testing results, we selected Bridge No. 1618152 as the NDR field testing site. The field test was conducted on September 12, 2016. As shown in Figure 4.18, the NDR robot was transferred to the bridge. Two delamination sites #4 and #6 were selected to conduct the NDR tasks. The locations of these two sites are presented in the NDE section.

Figure 4.19 shows the top view of the NDR repaired deck surface for two delamination sites #4 and #6. The drilling process was run smoothly. However, the grout delivery was not successful for either site. The main problem seems



(a)



(b)

Figure 4.19: Top view of the NDR field testing results from the deck surface. (a) Delamination site #4. (b) Delamination site #6.

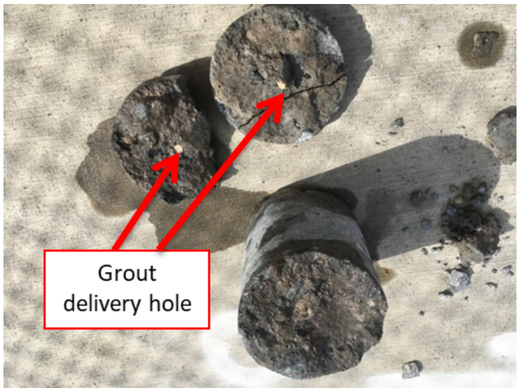
to be that either the delamination thicknesses are not large enough to let the grout flow through, or the debris or other materials block the delivery of the grout. During the grout delivery process, we have applied significantly large back pressure and tried to pump the grout into the crack zone. But eventually it was not successful to inject the grout materials into the bridge decks. As we can see from the figures, the grout materials were spread out from the nozzle on the deck surface. During the pumping process, we did not see any grout coming out of any of the three facilitating holes, in contrast to that we observed in the on-campus testing. We also tried to apply vacuum at the facilitating holes to possibly help create low pressure inside the crack zone for easily grout flow. The results were not improved. To test the connectivity of the crack zone among the four drilled locations, we used water (low viscosity) rather than repairing grout on site #4 and it was found that water cannot be pumped into the bridge decks. Because of unsuccessfully delivery of repairing grout at these two sites, the robotic NDR testing did not continue on any other detected delamination sites.

4.6.5.3 Post rehabilitation assessment

The post-rehabilitation assessment of the NDR rehabilitated sections involved assessment of why the repairing grout materials cannot be delivered to the delamination sites. It would also provide valuable diagnosis and also possible further suggestions to improve the robotic NDR systems and process design. We considered to take cores at the rehabilitated sections as a direct, effective way for the NDR performance evaluation and possibly trouble-shooting and diagnosis of the NDR process.

We conducted coring of the same bridge on October 3, 2016 and one core was taken at site #4. Figure 4.20(a) shows the core extracted from site #4 and Figure 4.20(b) shows the top view of the deck surface after the core extracted. From the core shown in Figure 4.20(a), we clearly observed that injected grout was only stayed inside the delivery hole and did not spread out on any part of the crack zone. Indeed, two layers of cracks existed at this site and the grout failed to spread out into either one of these two crack zones. The main reason of this failure of grout delivery is primarily clogged interface between the drilled hole and the crack zones. One possible cause of the clogged crack zone is highly likely the small-size debris generated by the drilling process. The debris cannot be removed and cleaned through blowing high-pressure air after these holes are created.

Since we also tried to use water (instead of the repairing grouts) in the filling process and the water still cannot go through the crack zone, we believe that the viscosity of the grout is not the primary reason that causes the delivery failure. We also observed that the crack zone surfaces were not smooth and there were many small debris on the crack zone surfaces. It is not clear whether enough gap spaces existed inside the crack zones and this could be another possible reason for the failure to deliver any fluids into the delamination areas.



(a)

(b)

Figure 4.20: A sample extracted core at site #4. (a) The extracted core (broken due to two delamination layers). (b) Top view of the deck surface after the core extraction.

4.7 Conclusion

The development and demonstration of an autonomous Non-Destructive Rehabilitation robot was presented for provide minimal invasive rehabilitation for bridge deck delamination. The main objective of the autonomous robotic system is to improve the efficiency and accuracy of bridge deck rehabilitation and reduce the risk to bridge workers. The developed NDR system integrate an omni-directional mobile robot platform, a 5 degree of freedom manipulator and a custom made end-effector for drilling and filling procedures. In this chapter, the mechatronic design to integrate the various sensors and actuators with the mobile robot platform and motion planning algorithm were mainly presented for enabling precise bridge deck rehabilitation. The robotic system performance was validated through extensive experimental testing and field deployment.

Chapter 5

Modeling Of Percussive Drilling For Bridge Deck Rehabilitation

5.1 Introduction

The bridge rehabilitation robot is targeted to deliver a nondestructive, rapid, cost effective rehabilitation at early stage of deterioration. The early-stage deterioration of bridge decks starts beneath the surface that cannot be reached directly as shown in the cored sample of Figure 4.1. Therefore a minimally invasive robotic drilling into bridge decks is a necessary step to effectively reach the locations where defects initiate[125].

Concrete drilling is different with metal drilling process that has been studied extensively in the past (e.g., [106]). While metal drilling uses thrust forces or torques for cutting, concrete drilling uses impact as the main source to crush the bristle concrete materials which commonly known as hammer drill. Because of this fundamental difference, understanding the mechanisms in concrete drilling is a challenging task due to the complicated energy transfer and complex bit-concrete interactions during impact. For high-quality robotic drilling on concrete, modeling of the drilling process is a critical step to design the robotic control systems. The goal of this chapter is to present a concrete drilling model and a mechatronic design for autonomous, highly-efficient robotic bridge decks rehabilitation. This work has been published in [88] and the author contribute to the majority of research and writing.

5.1.1 Related works

In [62, 65, 27, 39], empirical percussive drilling models are proposed to capture the drill bit impact interactions with rocks. A hysteresis relationship between the drill bit penetration and applied force is commonly assumed known in these models. Computational approach, such as finite element method or other impact energy-based simulation, are also used to study the percussive drilling in [109, 107, 22, 21]. Several analytical models are proposed to capture the impact energy as wave transmission between the drill bit and the rock [95, 94]. In these models, both the penetration-force relationship and the impact wave form are needed to completely solve the percussive drilling problem. In [108, 29], computational approach is used to calculate the energy and impact interactions between various components in hammer drills used in practice.

For viewpoint of control system design of hammer drill bit-concrete interactions, all of the above mentioned percussive drilling models are not desirable. The empirical model cannot give the physical interpretation and connection with drilling process parameters in practice, the computational models are too complicated for control design purposes, while the impact wave propagation models are too simplified for capturing the actual hammer drill systems. Instead, we propose a dry friction-based percussive drilling model that is inspired and extended from the model in [73] and the work in [109]. The proposed model is compact in mathematical representation and therefore, is desirable for use of designing control systems for drill bit-concrete interactions. Moreover, the model captures the penetration-force relationship through the dry friction characteristics and can readily be used to interpret the rock crush/chipping phenomena [109]. Compared to the results with only analysis and simulation in [73], we here present a more comprehensive model with extensive experimental validation. We also provide physical interpretation for the model parameters and estimation of these parameters can be easily obtained through commonly used mechanics experiments. The main contribution of the work presented in this chapter lies in the new percussive drilling model that is attractive for designing and optimizing the control of drilling process in concrete or rocks. The model also provides a means to further design, optimize and enhance the drilling performance for applications such as robotic bridge deck rehabilitation.

The rest of the chapter is organized as follows. In Section 5.2, we briefly describe the mechanical mechanism of the rotary hammer drill. The hammer drill kinematics and the minimum force deduction are presented in Section 5.3. We then present the percussive drilling model in Section 5.4. Experimental results are discussed in Section 5.5 and we finally summarize in Section 5.6.

5.2 Rotary hammer drill mechanism

We use a compact rotary hammer drill (model RHH180-01 from Bosch Group) for the experiment. To illustrate how the rotary hammer drill generates the high-energy impact, Fig. 5.1 shows the internal mechanism of the drill unit used in experiments. A high-speed brushless motor is used to drive both the rotational and vibratory motions for a compact design. A pair of gears is used to directly engage the motor output rotation to drill bit. A cam-crank mechanism is employed to simultaneously convert the rotational motion into the linear motion and then drive the piston-anvil pair for vibratory impact motion for the drill bit. There is a switch to dis-engage the cam-crank mechanism so that the vibratory motion of the drill bit is disabled and the drill behaves the same as a regular drill. The impact-induced drilling process such as the one generated by rotary hammer drill is also called percussive drilling. In the following sections, we present the models

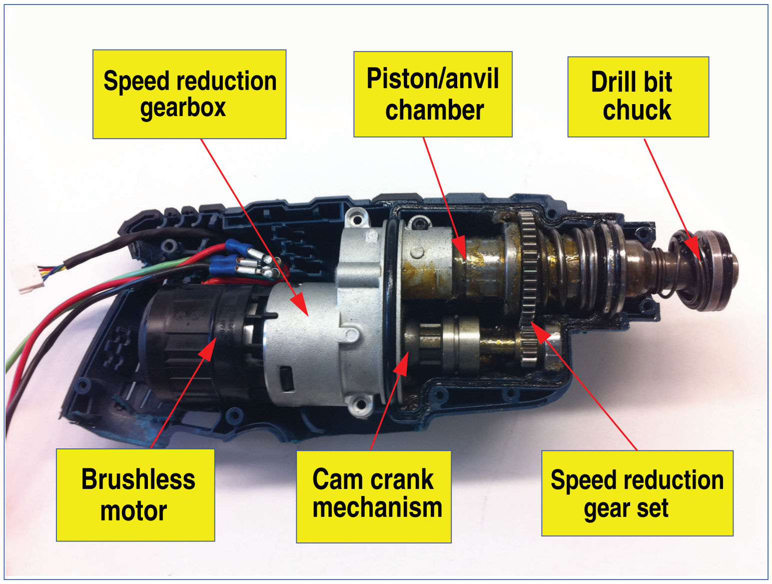


Figure 5.1: Mechanism of a commercial rotary hammer drill.

to capture the percussive drilling and also use these modeling development to interpret and optimize the drilling performance.

5.3 Hammer drilling kinematics and minimal thrust force

Figure 5.2 illustrates the mechanical structure of the hammer drill unit and also the percussive drilling process. The impact motion is produced by the piston-anvil pair and driven by the cam-crank mechanism. The piston is first accelerated and hits the anvil. The anvil then hits the drill bit at high speed to generate the impact on concrete. Once the drill bit bounces back from the concrete after impact, it pushes back the anvil and then the piston back to its original position for the next impact cycle. Two limit (rubber) blocks are used to restrict the anvil motion.

For analysis convenience and explicitly clear, we only consider the steady-state cyclic drilling process. Let m_p, m_b, and M_d denote the mass for the piston, the drill bit and the entire drill unit, respectively. The mass of the anvil is relatively small and we assume a perfect energy and momentum transfer during its impact

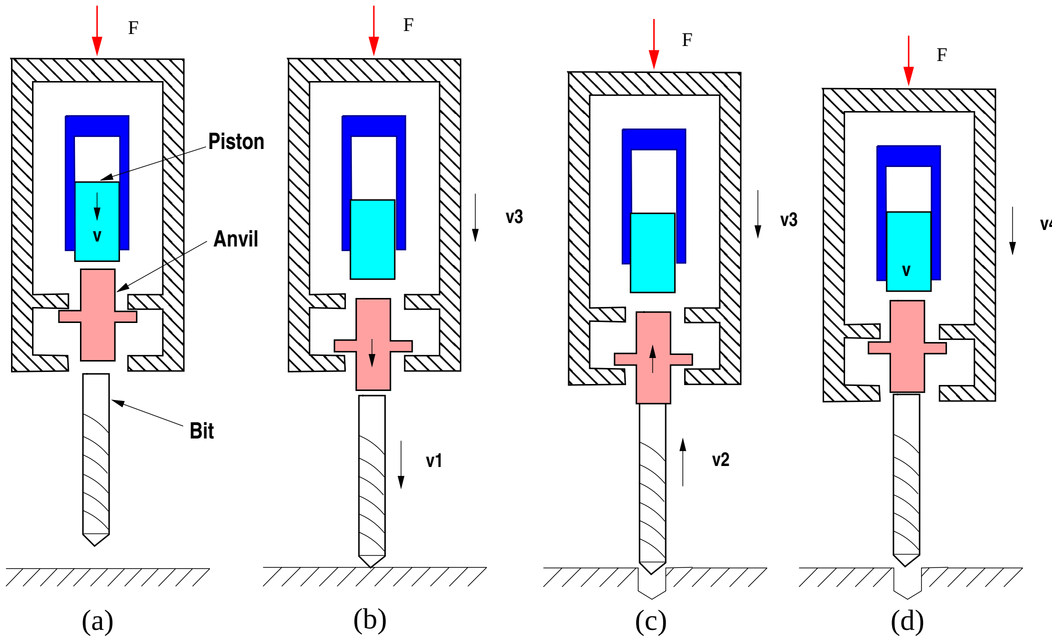


Figure 5.2: A snapshot of percussive drilling process for the rotary hammer drill.

with the drill bit. We also denote the applied thrust force on the drill unit is F_t and the period of cyclic impact is T. During the period T, the bit-concrete impact happens within a short period T_I and we also denote $T_{NI} = T - T_I$. To analyze the piston-anvil-bit motion and their kinematics, we assume that during each impact cycle, the piston is accelerated to the same velocity v_{p0} relative to the drill housing before it hits the anvil. We denote the drill bit velocities before and after bit-concrete impact as v_{b0} and v_{b1}, respectively. The velocity of the drill housing just before the bit-rock impact is denoted as v_{M0} and its velocity after the bit bounces back and hits the anvil and the housing is denoted as v_{M1}.

Assuming that the conservation momentum during the piston-anvil-bit interactions, we have

$$m_b v_{b0} = m_a v_{a0} = m_p(v_{M0} + v_{p0}), \tag{5.1}$$

where m_a and and v_{a0} are the anvil's mass and velocity after piston-anvil impact. After anvil-bit impact, the drill bit keeps its velocity v_{b0} until it hit the concrete. After penetrating into the concrete for distance d_0, it bounces back to hit the anvil with velocity v_{b1}. The kinetic energy of the drill bit is partly dissipated in the bit-concrete impact and thus, we have

$$\frac{1}{2}m_b v_{b0}^2 > \frac{1}{2}m_b v_{b1}^2 \Longrightarrow |v_{b0}| > |v_{b1}|. \tag{5.2}$$

During the bit-concrete impact, the impulse momentum can be calculated as

$$I_I = \left| \int_0^{T_I} F_I(t)dt \right| = m_b \left(v_{b0} - v_{b1}\right)$$

where $F_I(t)$ is the bit-concrete interaction force during the impact. In the above calculation, we define the positive velocity as downward. Using (5.1) and above calculation, it is straightforward to obtain

$$m_b v_{b0} < I_I < 2m_b v_{b0}. \tag{5.3}$$

After the drill bit bounces back and hits the anvil, we assume that the drill bit travels along with the drilling unit at velocity v_{M1} and therefore, we have the momentum balance equation

$$m_b v_{b1} + M_d v_{M0} = (m_b + M_d)v_{M1} \approx M_d v_{M1},$$

where in the last step, we use the fact $m_b \ll M_d$. After the impact time period T_I, the entire drilling unit, including the bit, travels under the thrust force F_t and gravitational force $M_d g$, g is the gravitational constant, that is, under acceleration $a_d = \frac{F_t}{M_d} + g$. Under such acceleration, the entire drill unit moves downward and such motion depends on thrust force F_t. If we define the threshold thrust force F_{th} under which the entire drill unit travels exactly the same as the bit penetration distance d_0 within time period T_{NI}, then we have

$$d_0 = \frac{1}{2}(v_{M1} + v_{M0})T_{NI}, \tag{5.4}$$

where we use the kinematic relationship of the drill unit under constant acceleration a_d with initial and final velocity v_{M0} and v_{M1}, respectively. On the other hand, under F_{th}, considering the drilling unit and the drill bit as one body, v_{M1} and v_{M0} can be obtained from the momentum balancing equation during time periods T_I and T as follows.

$$M_d v_{M1} = (F_{th} + M_d g)\, T_I - I_I, \tag{5.5}$$

$$M_d v_{M0} = (F_{th} + M_d g)\, T - I_I. \tag{5.6}$$

Since $T_I \ll T$, we therefore approximate $T_{NI} \approx T$ and thus, F_{th} is obtained by solving (5.4)-(5.6) as

$$\begin{aligned} F_{th} &= \frac{2d_0 M_d}{T_{NI}(T_I + T)} + \frac{2I_I}{T_I + T} - M_d g \\ &< \frac{2d_0 M_d}{T^2} + \frac{2m_p v_{p0}}{T} - M_d g \end{aligned}$$

Using inequality (5.3), we obtain the range of F_{th} as

$$\frac{2d_0 M_d}{T^2} + \frac{m_p v_{p0}}{T} - M_d g < F_{th} < \frac{2d_0 M_d}{T^2} + \frac{2m_p v_{p0}}{T} - M_d g.$$

From the above discussion, it is straightforward to know that if the thrust force is larger than the threshold value, i.e., $F_t > F_{th}$, the drill unit is pushed down even before the bit bounces back and therefore, the bit will be in contact all time. If the thrust force is small or negative (i.e., pulling the drill unit upwards), for each steady-state, we have the momentum balancing equation

$$(F_t + M_d g)T - I_I = 0.$$

Therefore, the minimum thrust force $F_{t\,\min}$ to maintain a steady-state percussive drilling needs to satisfy the following equality.

$$F_{t\,\min} + M_d g = \frac{I_I}{T}$$

Using (5.3), from the above equation we obtain

$$\frac{m_p v_{p0}}{T} - M_d g < F_{t\,\min} < \frac{2m_p v_{p0}}{T} - M_d g. \tag{5.7}$$

Experimental results will be used to validate the above calculation in Section 5.5 and it is validated that for the portable rotary hammer drill unit, it is required a small thrust force to perform efficient drilling.

5.4 Percussive drilling model

5.4.1 Modified dry friction-based drilling model

Figure 5.3 illustrates the dry friction-based percussive drilling model for the bit-concrete interactions. The drill bit is modeled as a mass m_b with initial velocity v_{b0}. It hits the mass-less concrete surface S_1 and then moves together downwards. The actual penetration into the concrete is captured by the displacement of mass less surface S_2. The displacements of S_1 and S_2 are denoted as x_1 and x_2, respectively. The spring with constant k_1 is used to model the elastic effect of the concrete materials. The output forces of dry friction elements P_1 and P_2 have the property with their displacements x_2 and x_3 as

$$P_1(x_2) = \begin{cases} F_{k_1} & \text{if } F_{k_1} \leq P_{10} \\ P_{10} & \text{otherwise} \end{cases}$$

and

$$P_2(x_3) = \begin{cases} F_{k_2} & \text{if } F_{k_2} \leq P_{20} \\ P_{20} - k_3 x_3 & \text{if } F_{k_2} > P_{20} \text{ and } x_3 \leq \frac{P_{20}}{k_3} \\ 0 & x_3 > \frac{P_{20}}{k_3} \end{cases} \tag{5.8}$$

where P_{10} and P_{20} are constants, $k_3 > 0$ is a constant, and F_{k_1} and F_{k_2} are the forces for the springs with constants k_1 and k_2, respectively. By the above definitions, the nonlinear output forces of dry friction elements P_1 and P_2 are dependent on the applied forces (i.e., spring forces) and for P_2, its output force also depends on the displacement x_3 of the mass less surface S_3 while P_1 does not depend on x_2.

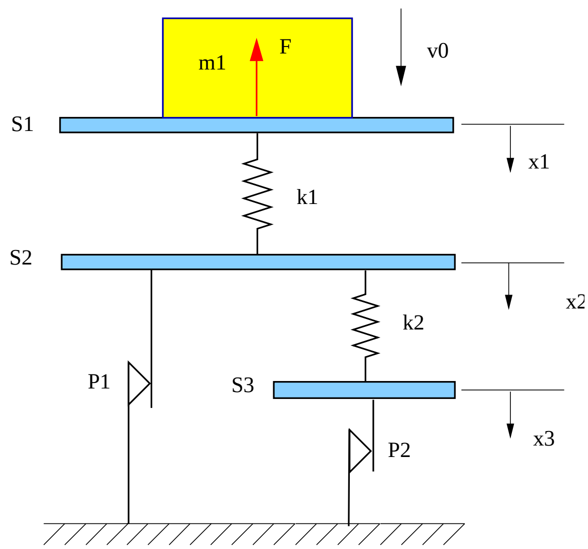


Figure 5.3: Modified dry friction-based percussive drilling model for bit-concrete interactions.

The purpose of introducing two dry friction elements P_1 and P_2 is to capture the crushing and chipping effects, respectively, in percussive drilling process[109]. P_{10} represents the threshold for the crushing force on concrete, P_{20} indicates the threshold for the chipping force on the concrete, and k_1 is the elastic constant of concrete materials. The spring constants k_2 and k_3 are used to capture the crushing and chipping effects and we will discuss later. The attractive property of using the nonlinear dry friction model is that the force-penetration relationship shows the similar commonly reported hysteresis characteristic [62, 27, 39]. in the following, we discuss detailed analysis and present the physical interpretation for the model parameters.

Due to the discontinuous forces introduced by P_1 and P_2, we consider the force-displacement relationship into five stages depending on the initial kinetic energy of the drill bit, that is, v_{b0}.

Stage I: Elastic deformation. In this case, neither of P_1 and P_2 is in motion and

v_{b0} is not large enough to generate impact force $F_I > P_{10}$, namely,

$$F_I = k_1 x_1 \le P_{10},\ x_3 < \frac{P_{20}}{k_3}.$$

The motion equations for the bit-concrete interactions are

$$m_b \ddot{x}_1 + k_1 x_1 = 0,\ x_2 = 0,\ x_3 = 0.$$

A linear relationship exists between F_I and x_1 and its slope is $\frac{dF_I}{dx_1} = k_1$.

Stage II: Crushing process. In this case, surface S_2 moves but surface S_3 does not move. The two springs k_1 and k_2 act in series and therefore,

$$P_{10} + P_{20} \ge F_I = k_1(x_1 - x_2) > P_{10},\ x_3 < \frac{P_{20}}{k_3}.$$

The motion equations are given as

$$m_b \ddot{x}_1 + k_1(x_1 - x_2) = 0,\ x_3 = 0$$

and $k_1(x_1 - x_2) = P_{10} + k_2 x_2$. Using the above equation, we obtain impact force $F_I = \frac{k_1 k_2 x_1 + k_1 P_{10}}{k_1 + k_2}$ and the slope of the F_I-x_1 curve

$$\frac{dF_I}{dx_1} = \frac{k_1 k_2}{k_1 + k_2} =: k_2'.$$

Stage III: Chipping process 1. In this case, all S_1, S_2, and S_3 move and $x_3 \le P_{20}/k_3$. By (5.8), we have

$$F_I = k_1(x_1 - x_2) \ge P_{10} + P_{20} =: P_{20}'.$$

The motion equations are

$$m_b \ddot{x}_1 + k_1(x_1 - x_2) = 0,\ k_2(x_2 - x_3) = P_{20} - k_3 x_3,$$

and $k_1(x_1 - x_2) = P_{10} + k_2(x_2 - x_3)$. Similar to the previous case, we obtain the slope of the F_I-x_1 curve as

$$\frac{dF_I}{dx_1} = \frac{k_3 k_2'}{k_3 - k_2'} =: k_3' < 0$$

by the choice of $k_3 < k_2'$.

Stage IV: Chipping process 2. Similar to Stage III, the condition for this stage is

$$P_{20}' > F_I = k_1(x_1 - x_2) \geq P_{10},\ x_3 > \frac{P_{20}}{k_3}$$

and the motion equations are

$$m_b\ddot{x}_1 + k_1(x_1 - x_2) = 0,\ k_1(x_1 - x_2) = P_{10},\ x_3 = x_2.$$

The slope of the F_I-x_1 curve is $\frac{dF_I}{dx_1} = 0$.

Stage V: Elastic relaxation. In this case, both S_2 and S_3 do not move and only S_1 re-bounces back and this represents the concrete's elasticity. Thus, $F_I = k_1x_1 < P_{10}$ and the motion equations are

$$m_b\ddot{x}_1 + k_1(x_1 - x_2) = 0,\ \dot{x}_2 = 0,\ x_3 = x_2.$$

The slope of the F_I-x_1 curve is $\frac{dF_I}{dx_1} = k_1$.

5.4.2 Drilling performance with the dry friction-based model

Depending on the initial kinetic energy of the drill bit, i.e., v_{b0}, the bit penetration follows the motion calculation as described in Stages I-V in the previous section. In this section, we explicitly present the penetration analysis and calculation using the dry friction-based model in Section 5.4.1.

Using the results provided by the dry friction-based percussive drilling model, Figure 5.4 illustrates the relationship between the drill bit-concrete impact force F_I and bit traveling displacement x_1 for Stages I-V (i.e., from elastic deformation to elastic relaxation) discussed in the dry friction-based drilling model. Corresponding to Stages I (elastic deformation) to V (elastic relaxation) in sequence, lines $\overline{OA}$, $\overline{AB}$, $\overline{BC}$, $\overline{CD}$, and $\overline{DE}$ indicate their F_I-x_1 relationships. Although each piecewise line segment indicates one stage, for a given bit impact energy

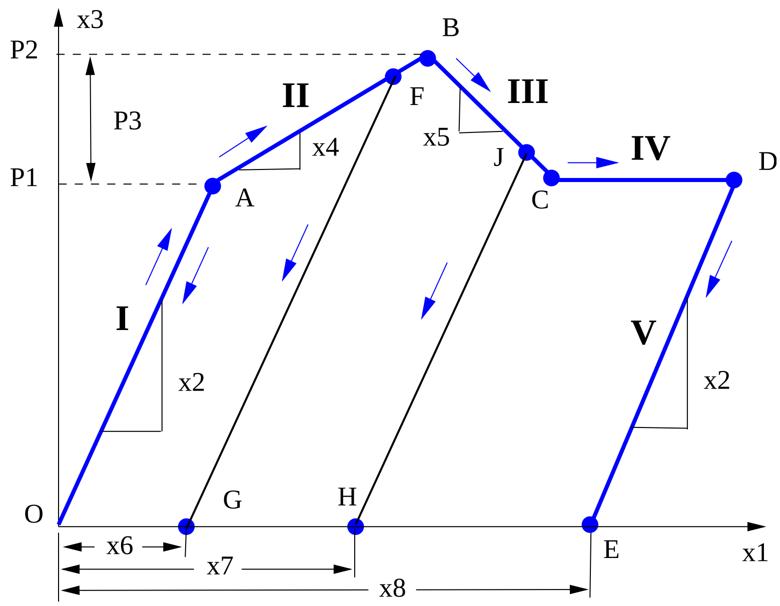


Figure 5.4: Schematic of the bit-concrete impact force F_I and the deformation x_1. The actual concrete penetration distance is x_2 and is related to x_1.

the changes of the F_I-x_1 relationship follows the progressive process from Stage I to V, that is, following the ordered curve $\overrightarrow{OABCDE}$. If the bit impact energy cannot sustain for all five stages, say only reaching Stage II at point F in the figure, then the concrete relaxation will follow the linear elastic relationship as that in Stage I, that is, following line $\overline{FG}$. In this case, the F_I-x_1 relationship will follow the ordered line segment $\overrightarrow{OAFG}$. When the drill bit is re-loaded for the next impact cycle, it will follow the same elastic slope, such as that of $\overrightarrow{GF}$. Similar discussions and interpretation are also reported in early literature about percussive drilling, such as those in [62, 109].

The drill bit penetration x is indeed modeled as the displacement x_2 of surface S_2 in the dry friction-based model. Depending on which stage a particular drilling process reaches, the penetration distance x is different. For a drilling process only in Stage I, the bit-concrete interaction only involves the elastic deformation and no actual penetration is achieved, that is,

$$x_p^{\mathrm{I}}(v_{b0}) = 0, \tag{5.9}$$

where x_p^i is denoted the drilling penetration x for the ith Stage, i =I, II, $\cdots$, V. In this case, the initial kinetic energy of the drill bit is less than the potential energy threshold at which the crushing process begins, namely,

$$\frac{1}{2}m_b v_{b0}^2 \leq \frac{1}{2}\frac{P_{10}^2}{k_1}, \Longrightarrow v_{b0} \leq v_{b0}^{\mathrm{I}} := P_{10}\sqrt{\frac{1}{m_b k_1}}. \tag{5.10}$$

If only the crushing process happens in percussive drilling, that is, Stage II, the F_I-x_1 follows $\overrightarrow{OAFG}$ in Figure 5.4. In this case, the initial kinetic energy of the drill bit satisfies

$$\frac{1}{2}\frac{P_{10}^2}{k_1} < \frac{1}{2}m_b v_{b0}^2 \leq \frac{1}{2}\frac{P_{10}^2}{k_1} + \frac{1}{2}\frac{P'_{20}P_{20}}{k'_2}$$

and thus

$$v_{b0}^{\mathrm{I}} < v_{b0} \leq v_{b0}^{\mathrm{II}} := \sqrt{{v_{b0}^{\mathrm{I}}}^2 + \frac{P'_{20}P_{20}}{m_b k'_2}}. \tag{5.11}$$

The penetration distance is

$$x_p^{\mathrm{II}}(v_{b0}) = \frac{P_{10}}{k_2}\left[\sqrt{\frac{k_1}{k_1+k_2}\left(1 + \frac{v_{b0}^2}{{v_{b0}^{\mathrm{I}}}^2}\frac{k_2}{k_1}\right)} - 1\right]. \tag{5.12}$$

When a drilling process both crushes and chips the concrete, i.e., in Stage III, the initial kinetic energy for the drill bit must meet the following requirements.

$$\frac{1}{2}m_b {v_{b0}^{\mathrm{II}}}^2 < \frac{1}{2}m_b v_{b0}^2 \leq \frac{1}{2}\frac{P_{10}^2}{k_1} + \frac{P_{10}+P'_{20}}{2}\left(\frac{P_{20}}{k'_2} - \frac{P_{20}}{k'_3}\right).$$

Thus, we have

$$v_{b0}^{\mathrm{II}} < v_{b0} \leq v_{b0}^{\mathrm{III}} := \sqrt{{v_{b0}^{\mathrm{II}}}^2 + \frac{P_{20}\left[P'_{20}(k'_2 - k_3) + P_1 k_3\right]}{m_b k_3^2}} \tag{5.13}$$

and the penetration distance is

$$\begin{aligned} x_p^{\mathrm{III}}(v_{b0}) &= \frac{P'_{20}}{k_3} - \frac{P_{10}}{k_2} + \frac{k_3 - k_2}{k_2 k_3} \\ &\quad \sqrt{m_b\left[k_1 {v_{b0}^{\mathrm{I}}}^2 - k'_3\left({v_{b0}^{\mathrm{III}}}^2 - v_{b0}^2\right)\right]}. \end{aligned} \tag{5.14}$$

Note that $k'_3 < 0$ and also $k_3 < k'_2 < k_2$ by the previous discussions.

If the drilling process in Stage IV, that implies a large initial bit velocity v_{b0}, i.e., $v_{b0} > v_{b0}^{\mathrm{III}}$, and the penetration distance is

$$x_p^{\mathrm{IV}}(v_{b0}) = \frac{m_b v_{b0}^2}{2P_{10}} - \frac{P_{10}}{2k_1} - \frac{P_{20}^2}{2k_3 P_{10}}. \tag{5.15}$$

The detailed derivations and calculations of penetration functions (5.12) and (5.14) are omitted here due to the page limit.

The value of the velocity bound v_{b0}^{III} is typically large such that Stage IV unlikely happens in percussive drilling process. Indeed, most literature such as [62, 109, 22, 21] do not discuss and report this observation. As show in Section 5.5, in the portable rotary hammer drilling experiments, only first two stages are observed and most the existing work only report the first three stages.

Summarizing the above discussion, Figure 5.5 shows the relationship between the drill bit penetration per impact x_p and its initial impact velocity v_{b0}. Note that the penetration per impact depends not only the bit initial impact velocity v_{b0} but also on the properties of the bit-concrete interaction, such as stiffness and strength coefficients that are reflected by model parameters k_1, k_2, P_{10}, etc. In the next section, we discuss how to obtain the values of these parameters.

5.4.3 Estimation of the model parameters

To use the dry friction-based percussive drilling model, several model parameters need to be identified and estimated through experiments. In this section, we briefly discuss the physical meaning or interpretation of these parameters and then present methods to estimate them.

Spring constant k_1 represents the elastic deformation constant of the drill bit-concrete interactions. The elastic deformation and relaxation happen in the first and the last stage of the previously discussed drilling model. Because of this understanding and assumption, the value of parameter k_1 can be estimated

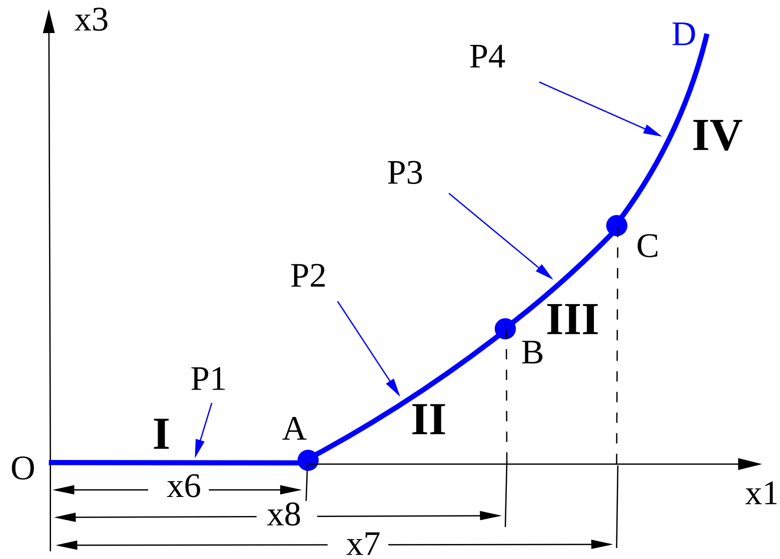


Figure 5.5: Schematic of the relationship between the bit penetration x_p and the bit initial impact velocity v_{b0}.

through the static loading testing when the drill bit starts to penetrate a flat surface of the concrete material.

The estimates of parameters P_{10} and k_2' can be obtained through the obtained experimental penetration per impact vs. bit initial velocity relationship. Assuming the experimental data fit the analytical predictions in Stages I and II (curve $\widehat{OAB}$) as shown in Figure 5.5, with known k_1 value, values of P_1 and k_2 can be obtained through (5.12) with a nonlinear least square fitting algorithm. A similar approach can be used to estimate the values of parameters P_{20} and k_3 with the prediction function (5.14) in Stage III. However, in the experiments using the portable rotary hammer drill set, the initial bit velocity is not large enough to produce such experimental testing data.

5.5 Experimental results

In this section, we present the results from various experiments to validate the pure percussive models and analysis discussed in previously sections.

5.5.1 Drilling testing setup

Besides the robotic drilling end-effector, an indoor drilling testbed is built show in Figure 5.6. The main reason of using this testbed for validating the percussive models and analysis is the rigid support for the drill unit rather than the flexible support of the robotic manipulator. We use the same modified rotary hammer drill unit as that shown in Figure 4.5a. To measure the drilling forces and torques, a 6-DoF force/torque sensor (model Mini45 from ATI Inc.) is used in the testbed. A cable potentiometer (model SP1-25 from Celesco Inc.) is used to measure the feeding distance and also the penetration distance. An optical encoder is used to monitor the drilling speed. The entire drilling fixture is mounted on a vertical linear guide that is rigidly attached to the wall. A set of counter-weights are used to adjust the drilling thrust force. We use the real-time xPC system (from Mathworks Inc.) for data acquisition and drilling speed control with a data acquisition card (model PCI-6221 from National Instruments Inc.) A drill bit with a 6.35-mm diameter (i.e., 1/4 inch) is used in all experiments. Two different concrete materials are used in the experiments: homogeneous concrete mortar samples and aggregated, heterogeneous concrete samples.

We disengage the cam-crank switch on the hammer drill so that only pure percussive drilling tests are conducted in experiments. The values of the physical parameters of the drill unit are listed in Table 5.1. The maximum impact frequency for the hammer drill used in the lab is around is around 75 Hz and thus the minimum percussive period $T_{\min} = 1/75 = 13.3$ ms as listed in Table 5.1. Similar to the approach in [108], the maximum initial bit impact velocity is measured as $v_{b0}^{\max} = 3.5$ m/s. The initial bit impact velocity can be controlled by the computer between 0 and $v_{b0}^{\max}$.

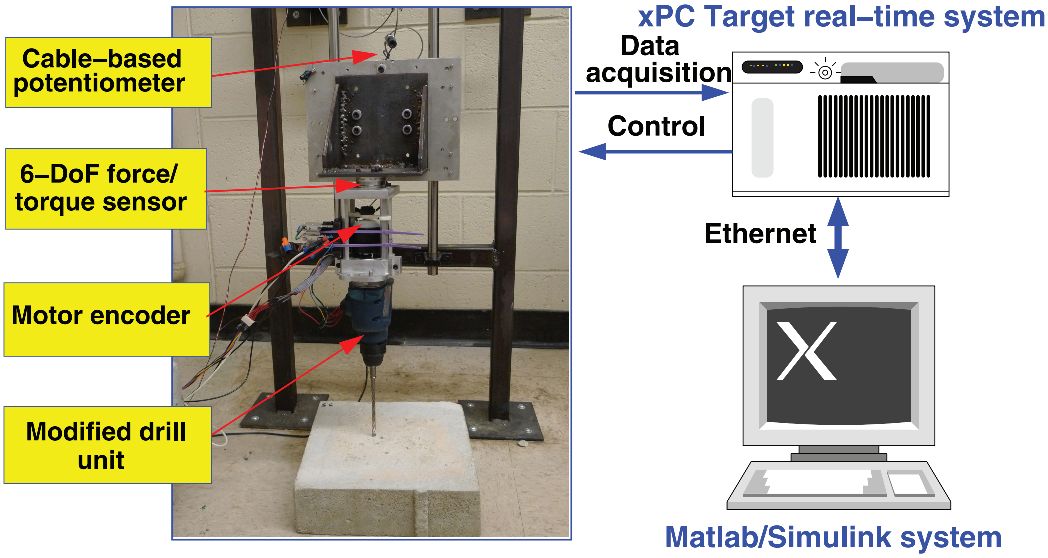


Figure 5.6: The percussive drilling experimental setup.

Table 5.1: The values of the physical parameters of the drill unit

m_p (kg)	m_b (kg)	M_d (kg)	$v_{b0}^{\max}$ (m/s)	$T_{\min}$ (s)	
0.05	0.05	3.0	3.5	0.0133	

5.5.2 Pure percussive drilling results

5.5.2.1 Minimum thrust force validation

The analysis and results of the minimum thrust force presented in Section 5.3 are validated through a set of pure percussive drilling experiments on mortar samples. Figure 5.7 shows the average and standard derivation of the penetration per impact (i.e., x_p) under changing total thrust force $F_t + M_d g$ and impact frequency of 75 Hz. It is clear from the results shown in the figure that all tests with the total thrust force less than 31.1 N fail to penetrate into the concrete, while all tests with the total thrust forces greater than that value perform almost the same penetration per impact. The estimate of the minimum thrust force is then $F_{t\min} = 1.7$ N, which is consistent with analytical result given in (5.7). The penetration per impact x_p under large thrust forces are slightly smaller than those under low thrust forces. This could be due to the accumulated drilling debris in the experiments under large thrust forces.

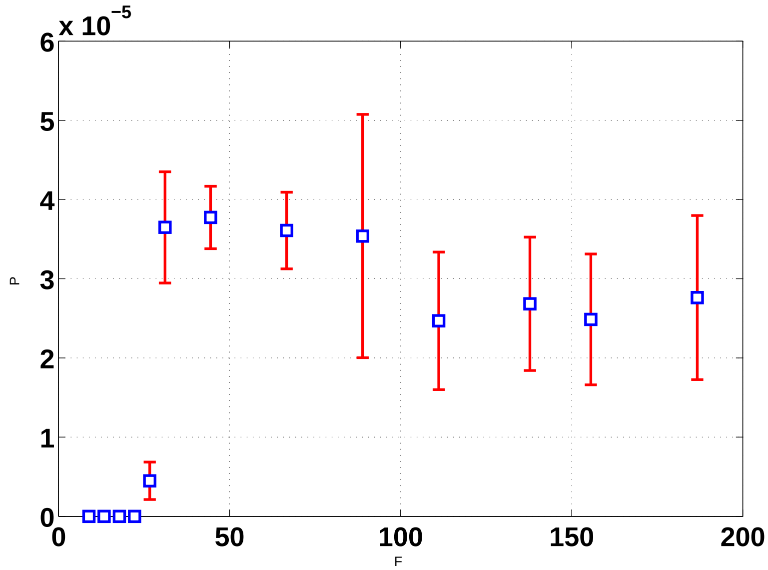


Figure 5.7: Penetration per impact on a mortar sample with various total thrust forces $F_t + M_d g$.

5.5.2.2 Model parameter estimation

As we discussed early, a part of the model parameters are identified based on directly experimental measurements by their physical interpretation and the values for the other parameters are estimated using the analytical model to fit the experimental data.

The value of the spring constant k_1 can be obtained directly measuring the slope of the force-penetration curves of the static indentation test. Figure 5.8 shows the indentation testing of the bit-mortar interaction. The results under multiple loading/unloading cycles demonstrate a consistent, constant slope value in the elastic region. Using these results, we estimate $k_1 = 3 \times 10^6$ N/m for the mortar material. We also conduct the similar testing for the concrete sample and the estimates of the parameter k_1 are listed in Table 5.2.

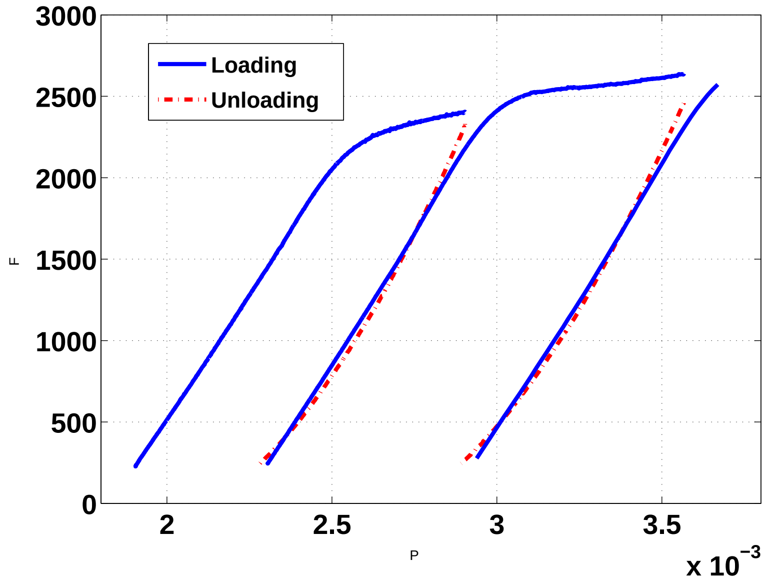


Figure 5.8: Experimental curve of the indentation testing using the drill bit on a flat surface of the mortar sample.

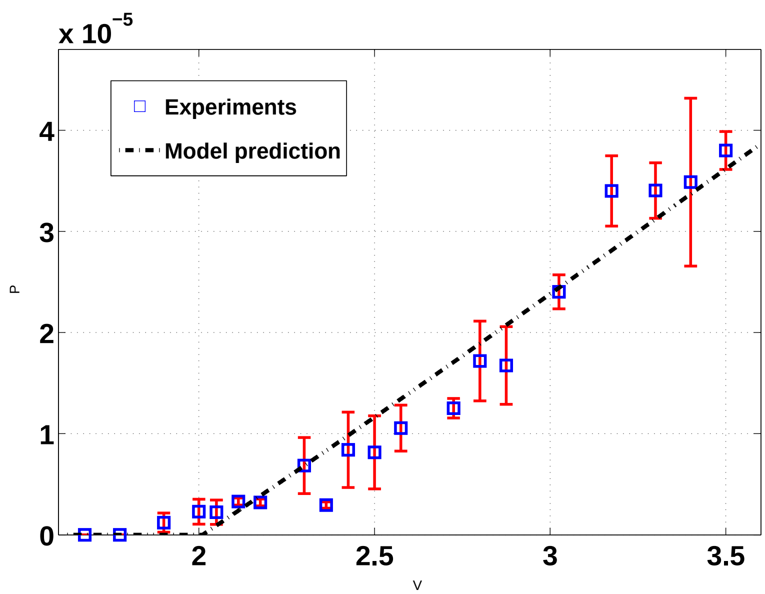


Figure 5.9: The experimental results of the penetration per impact x_p vs. the initial bit velocity v_{b0} for the mortar sample

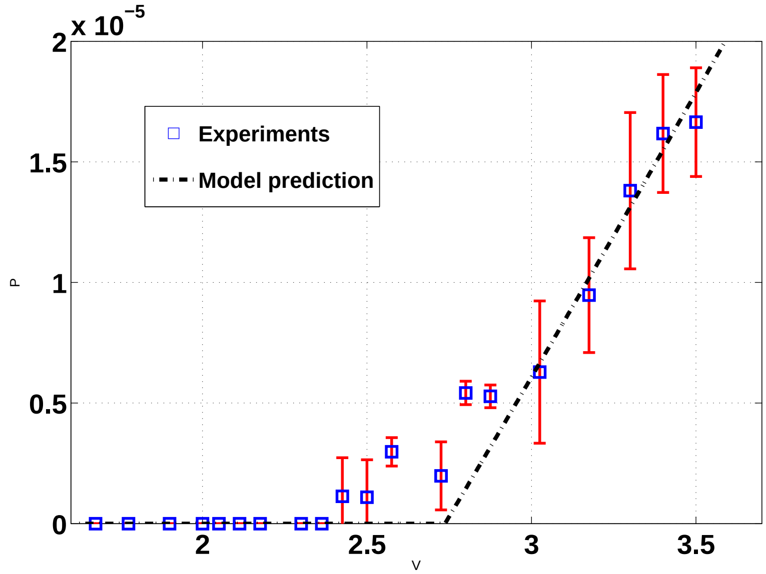


Figure 5.10: The experimental results of the penetration per impact x_p vs. the initial bit velocity v_{b0} for the concrete sample

Table 5.2: The estimated values of the dry friction-based drilling model parameters

Sample type	k_1 (N/m)	k_2 (N/m)	P_{10} (N)
Mortar	3×10^6	1.36×10^7	779
Concrete	4×10^6	1.55×10^7	1224

To estimate the values for parameters P_{10} and k_2, we conduct a series of extensive pure percussive drilling tests on both the mortar and the concrete samples. In these tests, the bit's initial velocity v_{b0} is varying for each run and then we record the penetration distance per impact x_p. Figs. 5.9 and 5.10 show the experimental results. Under a fixed v_{b0}, the drilling experiment is repeated four times and the calculated mean and standard deviation of x_p are plotted in these figures. As we discussed in Section 5.4.3, assuming the drilling process is within Stages I and II, we can use these experimental data sets to estimate the values of P_1 and k_2. The actual estimated values of P_1 and k_2 for both the mortar and the concrete samples are listed in Table 5.2. Since the physical interpretation of P_1 is the crushing threshold of the pure percussive drilling material, its value can also be estimated by compressive strength testing results, that is, $f_c = 19.6$ MPa for the high-strength mortar sample and $f_c = 41.4$ MPa for the normal concrete sample, then we estimate the total forces $P_{10}^m = f_c \times \pi 4d^2 = 630$ N for the mortar sample and $P_{10}^c = 1224$ N for the concrete sample, where $d = 6.35$ mm for the bit diameter. These values are consistent with the results in Table 5.2 that are estimated by drilling experiments.

In the above value estimation for parameters P_1 and k_2, we assume that the penetration-velocity characteristic happens within Stages I and II, that is, only elastic and crushing processes. To validate such a treatment, we conduct a static penetration test and demonstrate that no chipping happens with applied force around 5000 N, which is consistent with the result reported in [109]. For the given maximum bit initial velocity $v_{b0}^{\max} = 3.5$ m/s, we estimate the maximum impact force $F_I^{\max} \approx \sqrt{m_b k_1 \left(v_{b0}^{\max}\right)^2} = 1356$ N, which is far less than the chipping threshold. Therefore, using this drill unit, the values of other model parameters such as P_{20} and k_3 cannot be estimated and obtained.

Using the estimated model parameters, Figure 5.11 demonstrates a comparison between the experimental results and the model prediction of the penetration distance for a pure percussive drilling run on the mortar sample. The highly agreement results shown in this figure clearly demonstrate that the dry friction-based pure percussive drilling model captures the actual drilling penetration distance.

5.5.3 Discussion

From the results shown in Figs. 5.9 and 5.10, we can estimate the first critical bit initial velocities for drilling on the mortar and the concrete samples are around $v_{b0}^{\mathrm{I}} = 2.01$ m/s and 2.74 m/s, respectively. From the model prediction in (5.10), $v_{b0}^{\mathrm{I}} \propto \frac{P_{10}}{\sqrt{k_1}}$ and therefore, the critical penetration velocity is proportional to the drilling material's compressive strength. This observation is validated by the

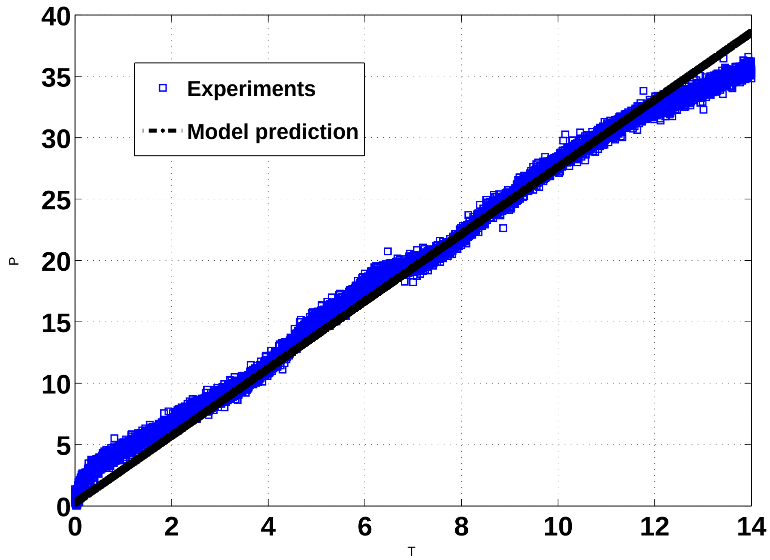


Figure 5.11: Comparison of the experiments and the model prediction for a penetration distance during one pure percussive drilling at v_{b0}^{max}.

parameter values shown in Table 5.2 with the above estimated $v_{b0}^{\rm I}$s for the mortar and the concrete samples. Moreover, from (5.12), we obtain the slope of the $x_p^{\rm II}$-v_{b0} curve at $v_{b0}^{\rm I}$ is

$$\left.\frac{dx_p^{\rm II}}{dv_{b0}}\right|_{v_{b0}^{\rm I}} = \frac{P_{10}}{(k_1+k_2)v_{b0}^{\rm I}} = \frac{\sqrt{m_b k_1}}{k_1+k_2} \propto \frac{\sqrt{k_1}}{k_1+k_2}, \tag{5.16}$$

where Eq. (5.10) is used in the second step of the above equation. From (5.16), the slope of the $x_p^{\rm II}$-v_{b0} curve only depends on the values of parameters k_1 and k_2. From Table 5.1, the values of k_2 are much larger than k_1s for both the mortar and the concrete samples. Thus, the ratio of the slopes between these samples is 1.02. This calculation implies that though their material properties have significant difference, by increasing a unit of the initial bit velocity v_{b0}, the gain of the penetration per impact is almost the same. This interesting observation has been confirmed by comparing the results shown in Figs. 5.9 and 5.10.

Although the previously discussed model and its prediction results fit well with the experiments, it only captures and explains the pure percussive drilling process. In practice, the rotational motion of the drill bit plays a critical role for highly-efficient and highly-effective drilling on concrete or rocks. Unfortunately, the model presented in this chapter does not consider and contain the influence of the bit rotation on the drilling performance. Understanding and modeling the entire rotary-percussive drilling process remains as one ongoing research direction.

5.6 Conclusion

The use of recent advances in robotics and automation technologies in bridge deck rehabilitation has motivated this study of autonomous percussive drilling on concrete. In this chapter, we presented a mathematical model of a pure percussive drilling process. A modified dry friction-based drilling model was presented to capture three major phenomenon in the drilling process: the elastic deformations, crushing and chipping of the penetrated material. We analyzed the drilling model and presented a set of analytical formulation for the critical drill bit kinetic energy and the penetration rate per impact. The model parameters were physically interpreted with the experimental testing and the values of these parameters were estimated experimentally. Finally, we validated the model prediction with experiments through extensive drilling tests. One of the natural extension of the proposed work is to understand and include the drill bit rotational contribution into the drilling model.

Chapter 6

Simultaneous Bridge Deck Inspection And Rehabilitation

6.1 Introduction

The progressive nature of the bridge deck deterioration requires early detection and intervention. A successful bridge deck health management system should be able to diagnose the problem and adopt appropriate approach to mitigate the problem before it goes beyond repairable. Therefore effective solution for bridge health management requires simultaneously identifying and fixing the problems. Having problems exposed and fixed at the same time could also substantially reduce the interruption of traffic and labor cost.

The inspection and rehabilitation task are very different in nature, therefore instead of having a monolithic system solve everything, we could decouple the inspection and rehabilitation problem to be addressed by a distributed system. This distributed system include two agents one is responsible to evaluate the bridge deck while the other is responsible to make the decision to choose the defect to fix. While the two agents make their own decision on their own, they could communicate wirelessly to share information for cooperation.

In previous chapter, we have introduced the RABIT inspection robot and ANDERS rehabilitation robot. In order for the two robots to work cooperatively as a complete solution for bridge deck inspection and rehabilitation, there are a few challenges. First, the ANDERS robot must rely on the defect information provided by the RABIT robot therefore requires RABIT generate an accurate defect map online. The online defect map requires the robot to render precised mapping in a reasonable time scale. In the previous chapter, RABIT is using a fixed step to inspect the bridge and generate an defect map offline. While we could clean the data offline, those techniques are generally not available when generating online map. Also since we need to perform operation on the bridge, we require the online defect map to have enough accuracy for the rehabilitation task. One way to boost precision is to reduce the step size which serves as sampling frequency. But a step size too small would lead to excessive inspection time which further blocking the traffic. If the step size is too large, then precision and confidence of the online map are not guaranteed. After the defect map generation, the rehabilitation robot need to determine the appropriate spots from the highly non-convex shaped heat map area.

The rest of the chapter organize as follows. In the next section, we review the related work including coverage planning and robot coordination. In section 2, we discuss the problem formulation and its assumptions and form two sub-problems as mapping problems and planning problems. In section 3, we provide an overview of the method. We then present how we use online Gaussian process to provide both global and local delamination map in section 5. Later in section 6, we use the local delamination map to adaptively adjust the step size of the inspection robot so we could accelerate the inspection process. In section 7, we describe the planning algorithm of the rehabilitation robot based on the global rehabilitation map. We provide simulation evaluation of the algorithms based on real bridge data in section 8. The whole chapter is summarized in the last section.

6.1.1 Related work

The problem for the inspection robot to determine the next sample position are closely related to the problem of adaptive sampling [58] which the goal is to choose observation locations that maximize the information gain and minimizing prediction uncertainty. Early work such as tackling the next-best-view problem [25] focus on the geometry approach for searching for the informative views. More recent approach has been adopt the probabilistic modeling, such as information gain [138] and [96].

Gaussian Processes have been used to model temperatures and other spatial phenomena[26]. It becomes a popular approach in the robotics society as it provide posterior estimation with uncertainty. [57][58] use Gaussian process to model the uncertainty of the ship hull and plan the optimal path for the underwater inspection vehicle. [96] present an informative planning algorithm with Gaussian process to enable an autonomous marine vehicle to perform persistent ocean monitoring. [119] propose an algorithm for exploration with Gaussian process in unknown environments.

The coordination between the inspection and rehabilitation is related to field of multi-robot systems. As pointed out in [132], multi-robot systems could divide into homogeneous or heterogeneous based on robot capacity and competitive or cooperative based on environment. Our system obviously falls in the heterogeneous and cooperative category. Ground and aerial robot are used together for searching the target in [14][121]. Despite the difference, the central problem lies in the multi-robot system is the resource conflict [132]. We implement an approach similar to lead-follower scheme to avoid resource conflict such as collision between robots.

6.2 Background

6.2.1 Gaussian process

Gaussian process is a non-parametric non-linear estimator that perform learning and inference from data. One of biggest advantage of Gaussian process is that it predicts both mean value and covariance which facilitate probabilistic planning and control. We briefly introduce the Gaussian process in the context of 2D spatial sensing in this section. Please refer to [130] for detailed introduction.

6.2.1.1 Gaussian process regression

A Gaussian process is a collection of random variables that any combination of the variables forms a joint Gaussian distribution. A 2D spatial Gaussian process $f(\boldsymbol{x}) : \mathbb{R}^2 \to \mathbb{R}^1$ is fully characterized by its mean value function $\mu(\boldsymbol{x})$ and covariance function $k(\boldsymbol{x}, \boldsymbol{x}')$ for $\boldsymbol{x} \in \mathbb{R}^2$

$$\mu(\boldsymbol{x}) = \mathbf{E}\left[f(\boldsymbol{x})\right],$$

$$k(\boldsymbol{x}, \boldsymbol{x}') = \mathbf{E}\left[\left(f(\boldsymbol{x}) - \mu(\boldsymbol{x})\right)\left(f(\boldsymbol{x}') - \mu(\boldsymbol{x}')\right)\right]$$

The mean value function and covariance function is determined through model and hyper-parameters.

Suppose the training data set includes N input output data pairs $\mathcal{D} = \{\boldsymbol{x}_i, y_i\}_{i=1}^N$. The output $y_i \in \mathbb{R}$ is a noisy observation of the underlying function value with zero mean Gaussian noise ω represented as

$$y_i = f(\boldsymbol{x}_i) + \omega,\ \omega \sim \mathcal{N}\left(0, \sigma^2\right)$$

We define the N-observation input design matrix as

$$\boldsymbol{X} = \left[\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\right]^T$$

where $\boldsymbol{x}_i \in \mathbb{R}^2$ as the 2-dimensional spatial position. Give all the spatial position in the N-observation input space, the corresponding observation vector is

$$\boldsymbol{y} = [y_1, y_2, \ldots, y_N]^T$$

At a testing point $\boldsymbol{x}_* \in \mathbb{R}^2$, we want to predict the output value $f_* = f(\boldsymbol{x}_*)$ according to the training dataset $\mathcal{D}$. If y_i is a Gaussian process, then given N training observations $\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N$ and testing position $\boldsymbol{x}_*$, the joint distribution of the random variables $\boldsymbol{y}$ and f_* is Gaussian could represent as

$$\begin{bmatrix} \boldsymbol{y} \\ f_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \boldsymbol{K} + \sigma^2 \boldsymbol{I}_N & \boldsymbol{K}_* \\ \boldsymbol{K}_*^T & \boldsymbol{K}_{**} \end{bmatrix}\right)$$

where $\boldsymbol{I}_N$ is $N \times N$ identity matrix and the matrix $\boldsymbol{K}$is the $N \times N$ kernel matrix represent the covariance matrix between the training samples, $\boldsymbol{K}_*$ is $N \times 1$ vector represent the cross variance between testing point and training dataset, $\boldsymbol{K}_{**}$ is covariance of testing point. The covariance element $\boldsymbol{K}$, $\boldsymbol{K}_*$ and $\boldsymbol{K}_{**}$ has the element of the following form

$$\begin{aligned} \boldsymbol{K} &= [k(\boldsymbol{x}_i, \boldsymbol{x}_j)]_{i,j=1,\ldots,N}, \\ \boldsymbol{K}_* &= [k(\boldsymbol{x}_*, \boldsymbol{x}_i)]_{i=1,\ldots,N}, \\ \boldsymbol{K}_{**} &= k(\boldsymbol{x}_*, \boldsymbol{x}_*). \end{aligned}$$

Gaussian process is often used as priors in the Bayesian setting, so the probabilistic prediction of f_* is represented as conditional distribution

$$f_* | \boldsymbol{x}_*, \mathcal{D} \sim \mathcal{N}(\mu(\boldsymbol{x}_*), \Sigma(\boldsymbol{x}_*))$$

where the mean value $\mu(\boldsymbol{x}_*)$ and the variance vector $\Sigma(\boldsymbol{x}_*)$ of the posterior distribution are given by

$$\begin{aligned} \mu(\boldsymbol{x}_*) &= \boldsymbol{K}_*^T \left[\boldsymbol{K} + \sigma^2 \boldsymbol{I}_N\right]^{-1} \boldsymbol{y} \\ \Sigma(\boldsymbol{x}_*) &= \boldsymbol{K}_{**} - \boldsymbol{K}_*^T \left[\boldsymbol{K} + \sigma^2 \boldsymbol{I}_N\right]^{-1} \boldsymbol{K}_* \end{aligned} \tag{6.1}$$

6.2.1.2 Model selection and hyper-parameter learning

The Gaussian process is determined by the kernel function which also called covariance function. The definition of the covariance function assumes the notion of similarity, which means that we expect that closer points are more likely to be similar. We focus our interest in stationary and isotropic covariance functions. In regression problem, a kernel function corresponds to a set of basis feature functions. The model selection choose the the kernel function that describes the data relation between input space. The most widely used covariance functions in machine learning are the radial basis covariance functions which represent as

$$k\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) = \sigma_f^2 \exp\left(-\frac{1}{2l}\left(\boldsymbol{x}_i - \boldsymbol{x}_j\right)^T\left(\boldsymbol{x}_i - \boldsymbol{x}_j\right)\right) + \sigma_{noise}^2 \delta_{ij} \tag{6.2}$$

The hyper-parameters $\boldsymbol{\theta} = \{l, \sigma_f, \sigma_{noise}\}$ are a set of parameters serves as the tuning knob of the covariance function and $\delta_{ij} = 1$ when $\boldsymbol{x}_i = \boldsymbol{x}_j$, otherwise $\delta_{ij} = 0$. The radial basis function is stationary and isotropic because it only depends on the distance between two data points but ignores the location of the points. The radial basis function decreases as the distance between two points increases where the speed and direction it decreases is governed by σ_f^2 and l. This also indicate point close in the input space tend to have similar values. σ_{noise} represent the noise level estimation also serves a regularization term to prevent over-fitting.

The hyper-parameters of the kernel are optimized during training by maximizing the log-marginal-likelihood which represent as

$$\log\left(\boldsymbol{y}|\boldsymbol{X}\right) = -\frac{1}{2}\boldsymbol{y}^T\left(\boldsymbol{K} + \sigma^2\boldsymbol{I}_N\right)^{-1}\boldsymbol{y} - \frac{1}{2}\log\left(\boldsymbol{K} + \sigma^2\boldsymbol{I}_N\right) - \frac{N}{2}\log\left(2\pi\right)$$

An example 1D Gaussian process prediction is shown in Figure 6.1, where the red dotted line is the function to be approximated as $f\left(x\right) = x\sin x$, the red dot are observations serves as the training set, the blue solid line is the prediction and

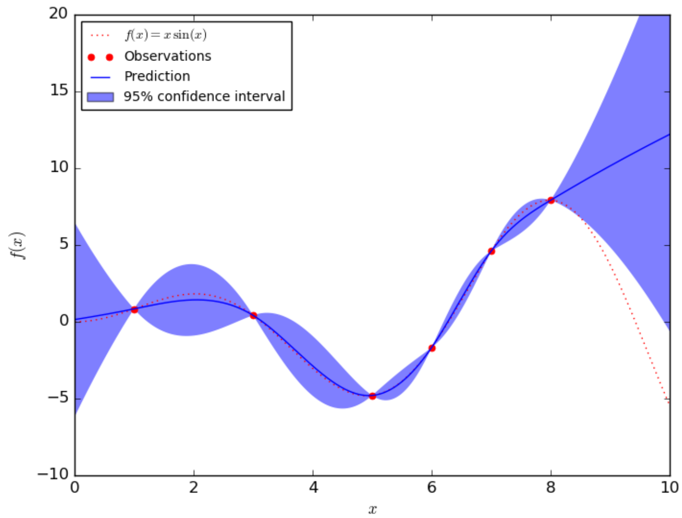


Figure 6.1: Gaussian process 1D example.

the purple area is within the 95% confidence interval. We could see around the observation point, the variance is crushed. For the extrapolation, variance start to increase as distance to the observation points increases.

6.2.2 Delamination condition map

The inspection robot is equipped with three kinds of NDE sensors GPR, resistivity probe and impact echo. The resistivity probe measures the resistance between contact points on the ground and the resistance ohm value is a indication of excessive moisture. A higher moisture are normally correlated with higher corrosion rate which requires attention for monitoring purpose but no further active intervention needed in this case. The GPR measures the corrosion condition of the rebar which might originate small crack from rebar. Impact echo sensor is measuring the frequency of the returned sound wave which is a direct indication of delamination underneath. These delaminations need to be addressed by our rehabilitation robot before it goes beyond repairable. While the resistivity probe and ground penetrating radar information related to the origination of the

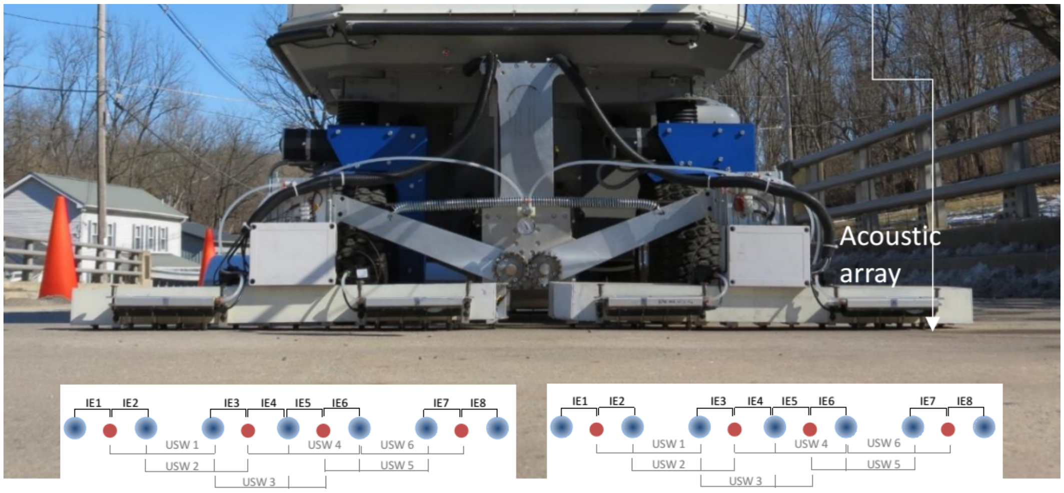


Figure 6.2: impact echo sensor in contact with the bridge deck surface at the data collection position

delamination, the impact echo sensor data is a more direct indicator of the delamination presence. So here we use only the impact echo sensor data to build the delamination map which provide guidance for the rehabilitation operation.

The impact echo sensor is designed to measure through direct contact with the bridge deck. A hammering action against the ground need to be triggered to generating the reflection wave that contains delamination information. The inspection robot need to stop every few feet to take the measurement. For each stop, the robot collect 14 points with 5.5 inch interval. The recordings are essentially discrete points $\{\boldsymbol{x}, y\}$ and therefore need to be interpolated to generate a delamination map.

6.2.3 Problem statement

We have two heterogeneous robot perform inspection and rehabilitation simultaneously on the bridge deck area. Both robot equipped with same computation resources and therefore computational load could be distributed between two robots. Both inspection and rehabilitation robot are omni-directional and therefore easy to maneuver even in tight bridge space. We make the following assumptions for the work site scenario:

1. The bridge surface is assumed to be flat for simplicity. It could also be easily extended to curved bridge surface through projection.

2. We assume the delamination depth is within the reach of the drill bit and therefore all the delamination area with various depth are projected to the bridge surface which is a 2D euclidean space.

3. A low latency network communication is well established during the task. In practice we have a WiFi network setup and if the bridge is more than the single wireless router coverage area, a repeater could always be used to extend the WiFi range.

4. We have no previous knowledge about the health condition of the bridge. The inspection and rehabilitation robots have to rely on the data collected for in-situ decision making.

We're aiming to deploy inspection and rehabilitation robot simultaneously and finish the tasks as quickly as possible. In order to do that we need to solve three sub-problems.

Problem 1 (Online mapping problem): Given the history of the discrete measurement $\boldsymbol{y}_{1:i}$, we need to provide a delamination condition map along with the prediction uncertainty map in an online manner. Both the delamination map and confidence map would be used for the planning of inspection and rehabilitation robots.

Problem 2 (Inspection robot planning problem): The inspection robot follows a fixed ox-plow route that pre-computed for the bridge deck. We need to design an planning algorithm to change the stop interval adaptively to reduce the number of stops but still keep high precision in prediction.

Problem 3 (Rehabilitation robot planning problem): Given the updated delamination condition map, plan for next rehabilitation point to maximize repair efficiency and minimize resource conflict with the inspection robot.

6.3 System overview

In order to complete the inspection and rehabilitation tasks simultaneously, there are several subsystem that work closely with each other. We first provide an overview of the system and we will provide details in the next sections.

The inspection robot first start to inspect the bridge following the pre-defined ox-plow pattern with the initial step size. The rehabilitation robot follow the leader inspection robot to minimize the resource conflict such as collision. In each stop, the inspection robot collect impact echo data at 14 contact points. Once data is collected, both the local prediction module and global prediction module started the training procedure. The local prediction module is used to predict the delamination condition in the area ahead of the inspection robot along the pre-defined path. The prediction, particular the uncertainty, is used to determine the step size of the inspection robot. In area of high variance in measurement, the robot start to reduce the step size and vice versa. In order to predict the local uncertainty and decide the step size, the local prediction module need to complete the model training and prediction in real time. This is possible since the local prediction module only use the data within a certain range for training, therefore the training and prediction are fast and scalable.

In addition to the local prediction module, we also implement a global prediction module on the rehabilitation robot which is used for determine the rehabilitation point for the rehabilitation robot. The global prediction module used all the data accumulated during scanning and generate a prediction map for the delamination condition for the whole bridge. The raw impact echo data collected by the inspection robot is passed through network to the rehabilitation robot. Those data packets is small enough so there is no significant delay to impact the system performance. Since the global prediction map takes in all the data collected, the training time is much longer than the local prediction map. While the training is in process for all the inspection data, the local prediction map is responsible for updating the global map where the data is already collected but not included in the current session of global training.

Once the global map is generated, we need to find out the rehabilitation point. The goal is to find the severe delamination area and large delamination. We focus on large delamination to improve the efficiency by reducing the number of operations. We first apply a empirical threshold to the global prediction map and extract all the areas [122] in the map that have delamination index higher than the threshold. It result in binary map with the patches of various size and shape. We want to focus on the larger patches for efficiency, therefore we calculate the area of each blob and filter out smaller blobs by a threshold. Inside each remaining blob, we need to find a point that close to the center that could be best facilitate the material delivery. The centroid is not suitable for our scenario since there exist non-convex shape and the centroid might even falls out of the blob area. We therefore fix the rehabilitation point as the point furthest from the edge. This approach is showed to be generate robust rehabilitation point.

The rehabilitation points feed to the rehabilitation robot task queue in the sequence that it has been found. The rehabilitation robot perform the minimal invasive procedure on the rehabilitation point queue in sequence.When there is no rehabilitation point in the queue, the rehabilitation robot act as the simple follower to the leader which is the inspection robot. Since the inspection robot route has no overlap between passes and the rehabilitation robot is always in the

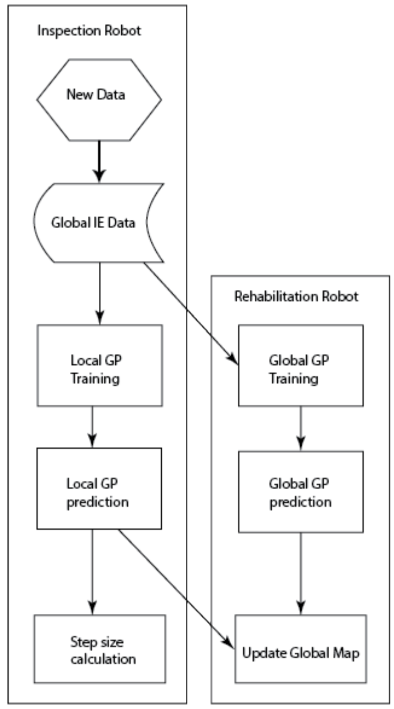


Figure 6.3: Gaussian process based delamination map prediction

inspected area, so there is no collision concerns between two robots.

6.4 Gaussian process regression for delamination map

For each impact echo measurement, the robot acquires a normalized assessment value $Z_i \in [-1.0, 1.0]$ in the measurement location $\boldsymbol{x}_i$, where the lower the assessment value represent the worse delamination condition.

The Gaussian process is used to process the impact echo data to generate the delamination map online. The generated map is used by the inspection and rehabilitation robot for planning purpose. The computation time of Gaussian process regression will generally increase cubicly with number of training samples. While we scanning the a 30m×10m bridge, it could take up to 10 secs for the onboard computer to train the model. This couldn't provide the real-time computation time that required for planning such as determine the next stop of inspection robot.

We keep two sets of Gaussian process model in our system. Those two models are trained in separate threads where a high priority Gaussian process training running on the inspection robot providing real-time local delamination uncertainty map and a lower priority global Gaussian process training running on the rehabilitation robot providing global delamination map. Once receive new data, the local delamination uncertainty map could be updated in real-time with training and prediction in less than 100ms. This facilitate the inspection robot to determine the next step size in real time. In contrast, the global map is mainly updated by the global Gaussian process model when it finished the training. In areas where global model haven't been able to incorporate newly collected, the local Gaussian process model would be responsible for updating. This makes the global map generation more scalable to larger inspection area.

6.4.1 Local delamination uncertainty map

The local uncertainty map is used to determine the next stop along the inspection route. The variance is a good representation of the uncertainty with regard the prediction and we could get from the Bayesian Gaussian process prediction.

The local prediction window is a rolling rectangular area that along the path of the inspection robot. As shown in Figure 6.4, the area start from the the current location of the impact echo arrays and extend along the robot moving direction. The area has the same width as the array and the length is set as the maximum step size of the inspection robot. The local training window is another rolling rectangular windows that centered in the prediction area. It is used to include the data points collected in the past inspection route for the local training dataset. This rectangular essentially enlarge the prediction area to include more local samples that are relevant to the prediction. The local Gaussian process model is trained on the local training dataset to predict the delamination

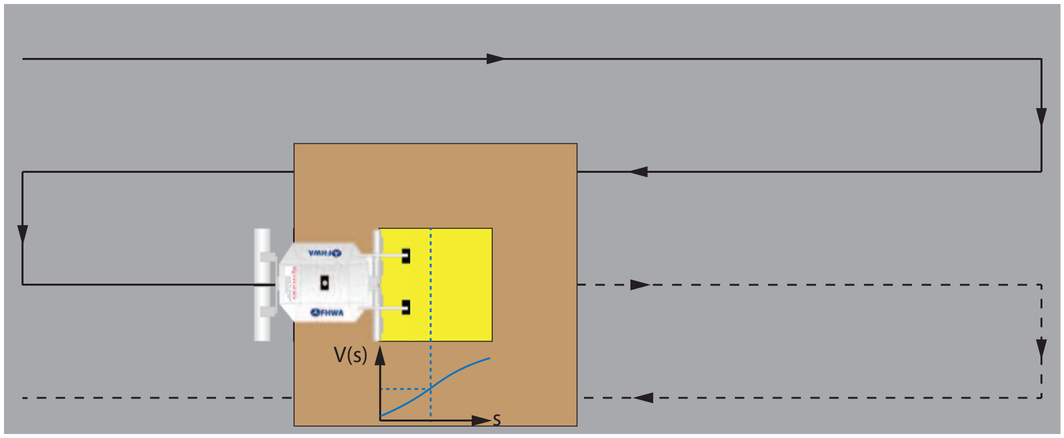


Figure 6.4: Local rolling window for delamination and uncertainty training and prediction. The yellow rectangular area represent the prediction area. The brown area represent area where the training sample are included. The solid black line is the route that has completed from inspection. The dashed black line is the future inspection route. The plot describes the average variance increase w.r.t the distance from current impact echo array position

and uncertainty in front of the inspection robot. This approach effectively reduce the training time of the Gaussian process model to below 100ms since the amount of training data point are constrained. It provides a good approximation to local delamination condition with highly relevant local data. The uncertainty map generated is used for determine the next step size of the inspection robot in Section 6.5.

6.4.2 Global delamination map

The global delamination map is the prediction of the delamination condition for area that has been surveyed by the inspection robot. It is used by the rehabilitation robot to determine the next rehabilitation point. The global delamination map is mainly updated by the global Gaussian process model. The global Gaussian process model uses all the data that collected at that time and the training time will increase cubicly as the training sample increased. It could take up to 10 secs for the training for a bridge deck of size 30m×10m. During training, new data will still be collected from the inspection robot. Therefore the local Gaussian process model is responsible for updating the map where the prediction from the global Gaussian process haven't been able to update. In situation that there is overlapping area between update from local Gaussian process at different steps, only the prediction with lower variance will be updated.

6.5 Adaptive step length planning for inspection robot

In order to collect delamination data, the inspection robot has to come to a full stop and press the impact echo array in close contact with the bridge deck. This is the most time consuming part of the inspection procedure.

The step size of the inspection robot essentially determined the density of the sample points collected and the denser sample points could potentially lead to a more precise delamination map. While we want build a precise delamination map which is essential for the rehabilitation operation, we need to do trade off between the data collection time and accuracy. If the robot could increase the step size in deck area with low variation in delamination condition, the robot could potentially stop less frequently. This would greatly improve the data collection efficiency without sacrificing precision.

The variance quantify the spatial dependency between the existing measurements and predicted points. From the variance estimation 6.1 and kernel equation 6.2, we could see once the kernel function is fixed, the variance monotonically decrease with respect to the distance of the predicted points and measurement. The hyper-parameter of the kernel function is optimized in each iteration. Particularly the length scale l will be larger causing higher confidence if the measurement shows agreement with trend. The property is particular useful where the local delamination map shows little to no variation which could potentially encourage larger step size and reduce the amount of samples required. In area with high variation of measurement, the higher variance suggesting the robot should slow down and increase sampling rate to measure more frequently. In all, the variance allows us to adjust the sampling frequency, essentially the stop interval, to reach desired accuracy of delamination map.

The variance in the Gaussian process regression is used to adaptively adjust the step while scanning. The inspection robot still follows the ox-plow route precomputed as in the previous chapter, but with a variable step size. Right after collecting the data, the robot generate a prediction for the area in front of the inspection robot as shown Figure 6.4. This area have the width of impact echo array width and the length as the maximum step size set as 0.6 meter. We divide the prediction area into grid of 1cm and we compute the grid-wise prediction variance of the future area according to the optimized kernel parameter. To simplify the problem, we use the average variance value $V(s)$ to represent uncertainty, which is calculated among all the lateral grid cells for distance s along the moving axis. Therefore, we have a curve representing the mono-increasing function between moving distance and average variance along lateral axis. We take the step size which is equal to the pre-determined threshold value of the variance. If the variance at the maximum step size is still lower than the threshold, we take the maximum step. The step size could represent as

$$s_i = \max\left(V_i^{-1}(\alpha), s_{max}\right)$$

where $V_i(s)$ is the variance function respect to the distance s_i from the ith data collection location, α is the threshold value for the acceptable prediction variance and s_{max} is maximum step size.

6.6 Target planning for rehabilitation robot

The job of the rehabilitation robot is to fix the delamination area which detected through inspection. The rehabilitation robot first start to follow the inspection robot with a safety distance. Once a rehabilitation target is available, the rehabilitation travels to the point to perform the minimal invasive rehabilitation procedures. Once the job finished, the rehabilitation robot check the global delamination map again to see if there is any other rehabilitation point which haven't been treated. When there is no job remains, the rehabilitation robot start to following the inspection robot again.

6.6.1 Rehabilitation point determination from global delamination map

Once a global delamination map is available, the rehabilitation robot queries the geo-referenced delamination map to get the next untreated delamination region for the next movement. The next rehabilitation point are then passed into the motion planner to execute the procedures as shown in 2.

To determine the next delamination point for repairing, we first perform a threshold operation to choose the region with medium to severe delamination. This results in multiple patches with variable size. We apply an erode morphological operation to remove the thin connection such as dumbbell shape as shown in Figure 6.5a, because the thin connection could provide additional resistance to the disperse of the matrix. We then filter out the smaller areas which might due to false positive or just isolated points. If the isolated points or area is on the edge of the known inspected area, that area could later been incorporated into a larger delamination area when neighboring points is available in the later pass. This reduces the total number of the rehabilitation procedures that required. We then threshold the global variance map and use that as mask to choose rehabilitation areas with high confidence. As a last step, we also mask out the previously treated region. If the treated region is later discovered to have a larger span, we will still perform operation on the newly discovered area.

Once we have mapped out the area to perform the rehabilitation procedures, we need to determine the point to drill and fill inside the area. One possible solution is to use the centroid which calculated by the momentum of heat map area. However the individual region might be highly non-convex, a resulting centroid might be outside the delamination region as shown in Figure 6.5b. Here we propose to choose the furthest interior point from the region boundary. In order to do that, we perform a distance transform to get the distance to the boundary for every point. If multiple furthest points exists with the same distance to the boundary, we choose the one closest to the centroid. This approach could guarantee the drilling and filling point is inside the boundary and as close as possible to the center, and make sure the epoxy material disperse evenly to the rest of delamination region. In practice, to reduce the computation to query the distance to boundary for each point, we only perform calculation on the locations where we collected data points. Those points have the highest confidence and could make sure we perform the rehabilitation operation on the right spot. We then select the rehabilitation point with the earliest timestamp that have been collected by the inspection robot as the next target.

6.6.2 Collision avoidance

In order for the rehabilitation robot to work together with inspection robot, we need to design a coordination algorithm for the two. Since the inspection robot follows an ox-plow route to perform coverage planning which is time consuming, we tend to keep the motion of inspection robot unperturbed and make it as the

(a) (b)

Figure 6.5: (a) Dumbbell shape that has thing connection between larger patches; (b) Highly non-convex area where the centroid is out of the area

master in the coordination. The rehabilitation robot will try to avoid motion plans that conflict with the inspection robot. There are in general two modes for the rehabilitation robot to follow: lead-follower mode and active rehabilitation mode. The lead-follower mode is used when no targeted rehabilitation point is available and the rehabilitation robot is just following the inspection robot in the back so that when a delamination region is available it could quickly react to that. The active rehabilitation mode is used when one or more delamination area needed to be repaired. When travel to the target, the rehabilitation robot will stay inside the inspected area and the inspection robot doesn't travel back to the inspected area, so there is no risk of collision between two robots.

6.7 Experiment validation

6.7.1 Simulation environment

To validate the proposed algorithms, we build a simulation system in ROS Gazebo to verify the performance. A real bridge delamination map is used as the ground truth for the testing scene where we could closely mimic the delamination distribution on the bridge deck in the real-life scenario. Through the UI interface, the user could specify the inspection area by define a polygon. Inside simulation system, the inspection and rehabilitation robot are modeled as a collection

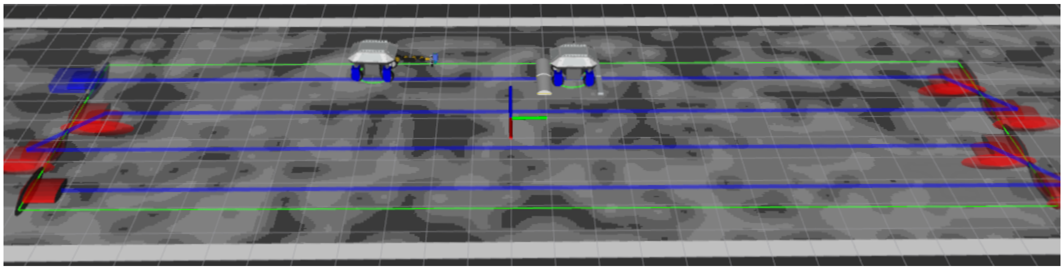


Figure 6.6: Simulation environment in ROS gazebo

of chained rigid bodies. The moving mechanism such as omni-directional wheel for the robot base movement and kinematics of the manipulator are modeled as kinematic chains in the ROS Gazebo. For collision check and visualization, the CAD model of inspection and rehabilitation robot are converted to meshes and matched with each moving rigid body.

Through the user interface, the user could selected the area to inspect as showed in the green lines as shown in Figure 6.6. Then the coverage path planner will plan the inspection route through Boustrophedon decomposition [24] shown as blue lines for straight path and red arrows for pure rotation point in Figure 6.6.

6.7.2 Variable step size validation

We implement adaptive step size algorithm for the inspection robot in the simulator. The result is shown in Figure 6.7. The colored map is the ground truth delamination map where warmer red color indicate more severe delamination and cold blue color indicate no delamination. The dots arrays are the impact echo data collection position. For each dotted straight line, it composed of 14 dots representing 14 impact echo measurements in on stop. From the figure, we could see the bottom part of the image shows more variation in the delamination condition and the robot stops more frequently producing sensor dots. However, in the top right corner where there mostly no delamination or low delamination, robot perform a larger stride. This shows our algorithm could adaptively decrease the step size for more complex areas and increase the step size for less complex area.

6.7.3 Rehabilitation points determination

We show the rehabilitation points determination in Figure 6.8. Figure **??** shows the ground truth delamination condition of the bridge deck and Figure 6.8a shows the severe patches through thresholding. Figure 6.8b shows the path to repair after the erode morphological operation and path size filtering. The red crosses are the rehabilitation points which could deal with highly non-convex shapes and dumbbell shapes.

6.8 Conclusion

In this chapter, we addressed several problem in the simultaneous deployment of the inspection and rehabilitation robots. To enable the prediction of the delamination condition while collecting data, we first introduce two Gaussian process prediction module based on local or global data. The variance output of local prediction module is used to adaptively determine the next step size of the inspection robot. The global delamination map goes through multiple image processing procedures to generate the rehabilitation point for the rehabilitation robot. In order to validate the concept, we design the simulation system that simulate the coordination of two robots on real impact echo data. The system is validated through the customized simulation system that could take in the real bridge data.

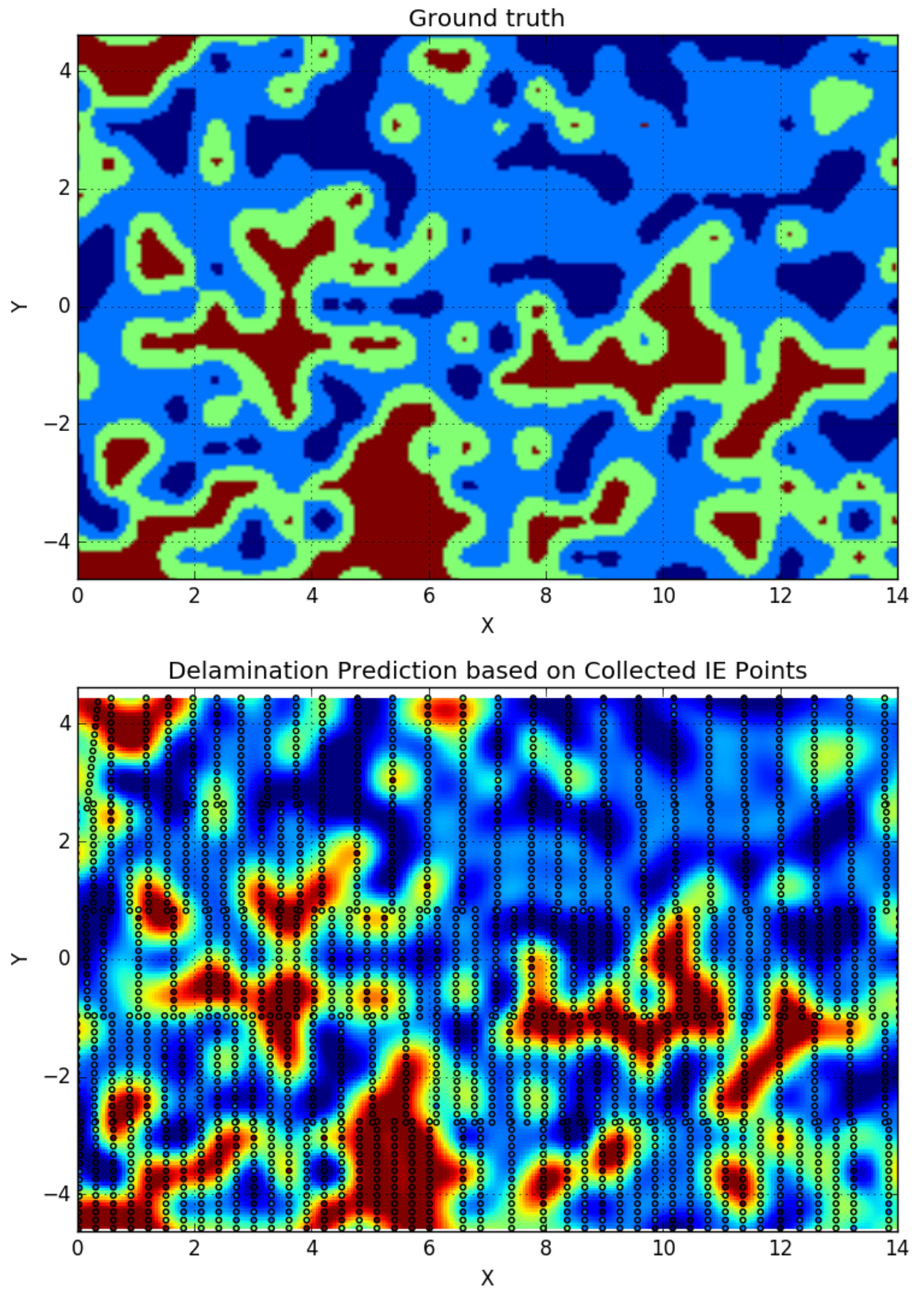


Figure 6.7: Delamination map generated through inspection with variable step size

(a)

(b)

Figure 6.8: Rehabilitation points determination. (a) Ground truth delamination map; (b) Thresholded delamination map; (c) Delamination map after erosion and size filtering and the determined rehabilitation points

Chapter 7

Conclusion And Future Work

7.1 Conclusions

In this dissertation. we focused on the development of the autonomous robotic systems for the bridge deck inspection and early delamination rehabilitation. The bridge deck is the most weathered part of the bridge due to its direct exposure to running traffic and environmental stress. Defects such as delamination could develop long before it could be inspected visually at late stage where rehabilitation cost is very high. Therefore a more economical way for bridge management should go through frequent bridge deck inspection and early defect rehabilitation that prevent the further defect development. For bridge deck inspection, non-destructive evaluation has been emerged as a promising approach [52] however the bottleneck lies in the low efficiency manual deployment. On the other hand, there is no available solution for early defect mitigation such as minimal invasive rehabilitation. In this dissertation we try to bring the robotics technology to the inspection and rehabilitation of the bridge deck which could enable massive deployment.

We first developed the bridge inspection robot that equipped various non-destructive evaluation technology that could provide a comprehensive view of the bridge deck. We solved two challenging problems. One is the hardware and software integration of various non-destruction evaluation sensors and the robotic system. The other is that we achieve robust pose estimation in presence of noise which is safety critical and essential for the autonomous inspection. We demonstrated the performance of the inspection robot through extensive field deployment.

Another critical aspect for the bridge inspection is the automated data processing process. We developed a new image mosaicing system to automatically processing the massive image data collected from inspection. The system combined the drift-less GPS information and feature-based frame to frame registration in the graph optimization framework. We showed the accuracy of the proposed algorithm through quantitative testing on the bridge decks.

To mitigate the early defect of bridge deck such as delamination, we developed the autonomous rehabilitation robot that performs minimal invasive rehabilitation. We integrated a 5 degree of freedom manipulator with mobile base that could precisely deliver the procedures to the required rehabilitation point. We designed a custom-made end-effector that could perform rehabilitation procedures including rotary percussive drilling and material delivery. The robotic system was validated through extensive testing and demonstrated the minimal invasive rehabilitation on real bridges.

One of the critical part of the rehabilitation procedure is rotary percussive drilling which produce large vibration. We studied the percussive drilling to identify the appropriate threshold force that maximize the drilling efficiency. We also proposed a dry friction-based model to capture the percussive drilling process. We presented the analytical formulation of bit concrete energy transfer and the analytical solution of the penetration rate. We validate the model prediction through indoor testing on a custom test platform.

To further improve the efficiency, we studied the problem to simultaneously deploy the inspection and rehabilitation robot. We used Gaussian process to predict the delamination condition based on the spatial discrete impact echo measurement. In order to solve the training speed issue of the Gaussian process, we implemented a multi-threading scheme to create two training sessions for local and global data separately. The prediction uncertainty local map enabled us to adaptive change the step size of the inspection robot in real-time therefore effectively shortening the inspection time to obtain a map precised enough for rehabilitation. Also, we developed an algorithm to choose the next target for the rehabilitation robot based on the global delamination map. This work was validated through a custom made simulation that could take in real bridge data.

7.2 Future works

There are a few research directions that follow this dissertation work.

In chapter 2, we developed a bridge inspection robot that perform non-destructive evaluation through contact measurement. During each stop, the impact echo array need to be pressed against the ground to enable the hammering operation. This procedure is the most time consuming part of the inspection which could be potentially replaced by contact-less sensors. Also, the array itself could right now tolerate slight curved ground by the spring mechanism, but the contact requirement might not be fulfilled when moving at the edge of bridge deck where more curved road surface is designed for drainage. Again, a contact-less sensor again might be able to help in those situations. [69] has proposed an contact-less air-coupled system that fused the impact echo with surface wave measurement. Getting this technology integrated will allow the inspection robot to inspect the bridge without stop and avoiding the associated acceleration and deceleration. This will dramatically increase the inspection speed.

In chapter 3, we developed a image mosaicing system to reconstruct the bridge surface. Our future work will include shadow removal from the panorama which is now impacting the visual appearance. It would also be interesting to adapt our system for online panorama generation. Another possible direction is to integrate with the crack detection algorithm to generate a crack map which will be very useful in practice.

In chapter 4, we developed a bridge rehabilitation robot that use a mobile manipulator to perform rehabilitation procedures essentially drilling small holes and filling with fluid material. In order to facilitate the material delivery, the filling procedures need to be more thoroughly studied. In one of the field test, the material filling are not successful even with manual filling. It's possible that debris result from percussive drilling could clogged the delivery opening as pointed out in [34]. Preliminary approach such as a internal vacuum drill bit proposed in [34] could be a solution to the potential problem but need further development and evaluation. On the other hand, we only rely on open-loop filling time that proportionally to the estimated delamination size. However, this procedure should be replaced by using computer vision techniques to monitor if any material comes out of the release hole or the filling seal.

In chapter 5, we proposed a dry friction model to capture the pure percussive drilling during rehabilitation. This model should be extended to account the rotary effect of rotary percussive drilling which helps to remove the pulverized material after impact. Moreover, the model should be integrated with the robotic manipulator control to improve the penetration rate or suppress the undesired vibration.

In chapter 6, we proposed the online delamination prediction based on Gaussian process that facilitate the simultaneous deployment of the inspection and rehabilitation robot. Although the approach was validated through simulation, however physical test should be conducted to confirm the performance. Also, we employ a probabilistic framework for the delamination mapping problem where we didn't assume any prior knowledge. Incorporating the previous inspection result in the probabilistic framework could be helpful for identifying the area prone to delamination and accelerate the inspection. Moreover, although impact echo measurement is most related to the delamination, we could also include the relevant measurement such as ground penetrating radar and electric resistivity in the probabilistic framework to further improve the delamination prediction quality. Lastly, the inspection robot currently followed a fixed route with variable step size essentially a 1D problem, however with more prior information the robot should be able to determine the next target without need to following the pre-defined inspection route. An informative planning problem need be further studied by factor in the information gain and effort for the movement.

References

[1] Ikhlas Abdel-Qader, Osama Abudayyeh, and Michael E Kelly. Analysis of edge-detection techniques for crack identification in bridges. *Journal of Computing in Civil Engineering*, 17(4):255–263, 2003.

[2] Farhad Aghili and Alessio Salerno. Driftless 3-d attitude determination and positioning of mobile robots by integration of imu with two rtk gpss. *IEEE/ASME Trans. Mechatronics*, 18(1):21–31, 2013.

[3] Aaron Arsenault, Steven A Velinsky, and Ty A Lasky. A low-cost sensor array and test platform for automated roadside mowing. *IEEE/ASME Transactions on Mechatronics*, 16(3):592–597, 2011.

[4] ASCE. 2009 report card for america's infrastructure. Technical report, American Society of Civil Engineers (ASCE), 2009.

[5] ASCE. 2017 infrastructure report card. Technical report, American Society of Civil Engineers (ASCE), 2017.

[6] Pejman Azarsa and Rishi Gupta. Electrical resistivity of concrete for durability evaluation: a review. *Advances in Materials Science and Engineering*, 2017, 2017.

[7] Christopher L Barnes and Jean-François Trottier. Ground-penetrating radar for network-level concrete deck repair management. *Journal of Transportation Engineering*, 126(3):257–262, 2000.

[8] AD Batako, VI Babitsky, and NA Halliwell. Modelling of vibro-impact penetration of self-exciting percussive-rotary drill bit. *Journal of sound and vibration*, 271(1):209–225, 2004.

[9] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359, 2008.

[10] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006.

[11] B. Blazejczyk-Okolewska and T. Kapitaniak. Dynamics of impact oscillator with dry friction. *Chaos, Solitons & Fractals*, 7(9):1455–1459, 1996.

[12] Matthew Brown and David G Lowe. Automatic panoramic image stitching using invariant features. *International journal of computer vision*, 74(1):59–73, 2007.

[13] Roger D Browne. Mechanisms of corrosion of steel in concrete in relation to design, inspection, and repair of offshore and coastal structures. *Performance of concrete in marine environments*, 65:169–204, 1980.

[14] Jonathan Butzkey, Andrew Dornbushy, and Maxim Likhachevy. 3-d exploration with an air-ground robotic system. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3241–3248. IEEE, 2015.

[15] Michael Calonder, Vincent Lepetit, Mustafa Ozuysal, Tomasz Trzcinski, Christoph Strecha, and Pascal Fua. Brief: Computing a local binary descriptor very fast. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7):1281–1298, 2012.

[16] Nicholas J Carino. The impact-echo method: an overview. In *Structures 2001: A Structural Engineering Odyssey*, pages 1–18. 2001.

[17] Denis A Chamberlain and Ernesto Gambao. A robotic system for concrete repair preparation. *IEEE robotics & automation magazine*, 9(1):36–44, 2002.

[18] Sylvie Chambon and Jean-Marc Moliard. Automatic road pavement assessment with image processing: Review and comparison. *International Journal of Geophysics*, 2011, 2011.

[19] Peter C Chang, Alison Flatau, and SC Liu. Health monitoring of civil infrastructure. *Structural health monitoring*, 2(3):257–267, 2003.

[20] Bernardino Chiaia. Fracture mechanisms induced in a brittle material by a hard cutting indenter. *International Journal of Solids and structures*, 38(44):7747–7768, 2001.

[21] Luciano E Chiang. Dynamic force-penetration curves in rock by matching theoretical to experimental wave propagation response. *Experimental mechanics*, 44(2):167–175, 2004.

[22] Luciano E Chiang and DA Elias. Modeling impact in down-the-hole rock drilling. *International Journal of Rock Mechanics and Mining Sciences*, 37(4):599–613, 2000.

[23] Howie Choset. Coverage for robotics - a survey of recent results. *Annals of Mathematics and Artificial Intelligence*, 31(1-4):113–126, 2001.

[24] Howie Choset and Philippe Pignon. Coverage path planning: The boustrophedon cellular decomposition. In *Field and service robotics*, pages 203–209. Springer, 1998.

[25] Cl Connolly. The determination of next best views. In *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, volume 2, pages 432–435. IEEE, 1985.

[26] Noel Cressie. Statistics for spatial data. *Terra Nova*, 4(5):613–617, 1992.

[27] Emmanuel Detournay and Paul Defourny. A phenomenological model for the drilling action of drag bits. *International journal of rock mechanics and mining sciences & geomechanics abstracts*, 29(1):13–23, 1992.

[28] Kien Dinh, Nenad Gucunski, and Trung H Duong. Migration-based automated rebar picking for condition assessment of concrete bridge decks with ground penetrating radar. *NDT & E International*, 98:45–54, 2018.

[29] Dante A Elias and Luciano E Chiang. Dynamic analysis of impact tools by using a method based on stress wave propagation and impulse-momentum principle. *Journal of Mechanical Design*, 125(1):131–142, 2003.

[30] Brendan Englot and Franz Hover. Planning complex inspection tasks using redundant roadmaps. In *Robotics Research*, pages 327–343. Springer, 2017.

[31] Brendan Englot and Franz S Hover. Three-dimensional coverage planning for an underwater inspection robot. *The International Journal of Robotics Research*, 32(9-10):1048–1073, 2013.

[32] Brendan J Englot. *Sampling-based coverage path planning for complex 3D structures.* PhD thesis, Massachusetts Institute of Technology, 2012.

[33] Brendan J Englot and Franz S Hover. Sampling-based coverage path planning for inspection of complex structures. In *Twenty-Second International Conference on Automated Planning and Scheduling*, 2012.

[34] Moiz Abbasbhai Ezzy. Design and testing of equipment for non-destructive rehabilitation of bridge deck delaminations. Master's thesis, Rutgers University-Graduate School-New Brunswick, 2015.

[35] Jan Faigl and Geoffrey A Hollinger. Unifying multi-goal path planning for autonomous data collection. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2937–2942. IEEE, 2014.

[36] J Ferrer, A Elibol, O Delaunoy, N Gracias, and R Garcia. Large-area photo-mosaics using global alignment and navigation data. In *MTS/IEEE OCEANS Conference, Vancouver, Canada*, pages 1–9, 2007.

[37] FHWA. Status of the nations highways, bridges, and transit: Conditionsand performance, report to congress. Technical report, Federal Highway Administration (FHWA), 2008.

[38] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[39] Luiz FP Franca. A bit–rock interaction model for rotary–percussive drilling. *International Journal of Rock Mechanics and Mining Sciences*, 48(5):827–835, 2011.

[40] Enric Galceran, Ricard Campos, Narcís Palomeras, Marc Carreras, and Pere Ridao. Coverage path planning with realtime replanning for inspection of 3d underwater structures. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6586–6591. IEEE, 2014.

[41] Enric Galceran, Ricard Campos, Narcís Palomeras, David Ribas, Marc Carreras, and Pere Ridao. Coverage path planning with real-time replanning and surface reconstruction for inspection of three-dimensional underwater structures using autonomous underwater vehicles. *Journal of Field Robotics*, 32(7):952–983, 2015.

[42] Enric Galceran and Marc Carreras. Planning coverage paths on bathymetric maps for in-detail inspection of the ocean floor. In *2013 IEEE International Conference on Robotics and Automation*, pages 4159–4164. IEEE, 2013.

[43] Enric Galceran and Marc Carreras. A survey on coverage path planning for robotics. *Robotics and Autonomous Systems*, 61(12):1258–1276, 2013.

[44] Debabrata Ghosh and Naima Kaabouch. A survey on image mosaicing techniques. *Journal of Visual Communication and Image Representation*, 34:1–11, 2016.

[45] Alexander Gibson and John S Popovics. Lamb wave basis for impact-echo method analysis. *Journal of Engineering mechanics*, 131(4):438–443, 2005.

[46] K Gowers and S Millard. Measurement of concrete resistivity for assessment of corrosion. *ACI Materials Journal*, 96(5), 1999.

[47] Nenad Gucunski, Basily Basily, Jinyoung Kim, Jingang Yi, Trung Duong, Kien Dinh, Seong-Hoon Kee, and Ali Maher. Rabit: implementation, performance validation and integration with other robotic platforms for improved management of bridge decks. *International Journal of Intelligent Robotics and Applications*, 1(3):271–286, 2017.

[48] Nenad Gucunski, Basily Basily, Ali Maher, Jinyoung Kim, and Duong Trung. Use of robotics in automated and comprehensive nde of rc structures. In *Proceedings of the 1st International Conference on Construction Materials for Sustainable Future*, pages 49–55, 01 2017.

[49] Nenad Gucunski, Gary R Consolazio, and Ali Maher. Concrete bridge deck delamination detection by integrated ultrasonic methods. *International Journal of Materials and Product Technology*, 26(1-2):19–34, 2006.

[50] Nenad Gucunski, Jinyoung Kim, Kien Dinh, Jie Gong, Fei Liu, , Seong-Hoon Kee, and Basily Basily. Innovative ways in condition assessment of concrete bridge decks: data collection using robotics, and advanced data interpretation and visualization. In *SynerCrete International Conference on Interdisciplinary Approaches for Cement-based Materials and Structural Concrete*, 2018.

[51] Nenad Gucunski, Ali Maher, Basily Basily, Hung La, Ronny Lim, Hooman Parvardeh, and Seong-Hoon Kee. Robotic platform rabit for condition assessment of concrete bridge decks using multiple nde technologies. *HDKBR INFO Magazin*, 3(4):5–12, 2013.

[52] Nenad Gucunski, Brian Pailes, Jinyoung Kim, Hoda Azari, and Kien Dinh. Capture and quantification of deterioration progression in concrete bridge decks through periodical nde surveys. *Journal of Infrastructure Systems*, 23(1):B4016005, 2016.

[53] Nenad Gucunski, Francisco A Romero, P Shokouhi, and J Makresias. Complementary impact echo and ground penetrating radar evaluation of bridge decks on i-84 interchange in connecticut. In *Earthquake Engineering and Soil Dynamics*, pages 1–10, 2005.

[54] Nenad Gucunski, Greg Slabaugh, Zhe Wang, Tong Fang, and Ali Maher. Impact echo data from bridge deck testing: Visualization and interpretation. *Transportation Research Record*, 2050(1):111–121, 2008.

[55] Nenad Gucunski, Jingang Yi, Basily Basily, Trung Duong, Jinyoung Kim, Perumalsamy Balaguru, Hooman Parvardeh, Ali Maher, and Husam Najm. Concrete bridge deck early problem detection and mitigation using robotics. In *SPIE Smart Structures and Materials+ Nondestructive Evaluation and Health Monitoring*, pages 94370P–94370P. International Society for Optics and Photonics, 2015.

[56] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[57] Geoffrey A Hollinger, Brendan Englot, Franz Hover, Urbashi Mitra, and Gaurav S Sukhatme. Uncertainty-driven view planning for underwater inspection. In *2012 IEEE International Conference on Robotics and Automation*, pages 4884–4891. IEEE, 2012.

[58] Geoffrey A Hollinger and Gaurav S Sukhatme. Sampling-based robotic information gathering algorithms. *The International Journal of Robotics Research*, 33(9):1271–1287, 2014.

[59] Daehie Hong, Steven A Velinsky, and Xin Feng. Verification of a wheeled mobile robot dynamic model and control ramifications. *Journal of dynamic systems, measurement, and control*, 121(1):58–63, 1999.

[60] Eva Hörster and Rainer Lienhart. Calibrating and optimizing poses of visual sensors in distributed platforms. *Multimedia Systems*, 12(3):195–210, 2006.

[61] Eva Hörster and Rainer Lienhart. On the optimal placement of multiple visual sensors. In *Proceedings of the 4th ACM international workshop on Video surveillance and sensor networks*, pages 111–120. ACM, 2006.

[62] W. A. Hustrulid and C. Fairhurst. A theoretical and experimental study of the percussive drilling of rock, Part I - Theory of percussive drilling. *International Journal of Rock Mechanics and Mining Sciences & Geomechanics Abstracts*, 8:311–333, 1971.

[63] W. A. Hustrulid and C. Fairhurst. A theoretical and experimental study of the percussive drilling of rock, Part II - Force-penetration and specific energy determinations. *International Journal of Rock Mechanics and Mining Sciences & Geomechanics Abstracts*, 8:335–356, 1971.

[64] W. A. Hustrulid and C. Fairhurst. A theoretical and experimental study of the percussive drilling of rock, Part III - Experimental verification of the mathematical theory. *International Journal of Rock Mechanics and Mining Sciences & Geomechanics Abstracts*, 9:417–429, 1972.

[65] W. A. Hustrulid and C. Fairhurst. A theoretical and experimental study of the percussive drilling of rock, Part IV - Application of the model to actual percussive drilling. *International Journal of Rock Mechanics and Mining Sciences & Geomechanics Abstracts*, 9:431–449, 1972.

[66] Petr Janoušek and Jan Faigl. Speeding up coverage queries in 3d multi-goal path planning. In *2013 IEEE International Conference on Robotics and Automation*, pages 5082–5087. IEEE, 2013.

[67] Lars G Karlsson, Bengt Lundberg, and Karl-Gustaf Sundin. Experimental study of a percussive process for rock fragmentation. In *International Journal of Rock Mechanics and Mining Sciences & Geomechanics Abstracts*, volume 26, pages 45–50. Elsevier, 1989.

[68] Giorgos D Kazazakis and Antonis A Argyros. Fast positioning of limited-visibility guards for the inspection of 2d workspaces. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 2843–2848. IEEE, 2002.

[69] Seong-Hoon Kee and Nenad Gucunski. Air-coupled ultrasonic system for fusion of impact-echo tests and surface wave measurements (presentation video). In *Nondestructive Characterization for Composite Materials, Aerospace Engineering, Civil Infrastructure, and Homeland Security 2014*, volume 9063, page 90630M. International Society for Optics and Photonics, 2014.

[70] Matthew Klein, Giri Venkiteela, Husam Najm, and Perumalsamy Balaguru. Nanoscale materials for non-destructive repair of transportation infrastructures. In *Nondestructive Characterization for Composite Materials, Aerospace Engineering, Civil Infrastructure, and Homeland Security 2011*, volume 7983, page 798308. International Society for Optics and Photonics, 2011.

[71] Matthew J Klein. Nondestructive repair and rehabilitation of structural elements using high strength inorganic polymer composites. Master's thesis, Rutgers University-Graduate School-New Brunswick, 2013.

[72] Andreas Krause and Carlos Guestrin. Submodularity and its applications in optimized information gathering. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(4):32, 2011.

[73] Anton M Krivtsov and Marian Wiercigroch. Dry friction model of percussive drilling. *Meccanica*, 34(6):425–434, 1999.

[74] Rainer Kümmerle, Giorgio Grisetti, Hauke Strasdat, Kurt Konolige, and Wolfram Burgard. g2o: A general framework for graph optimization. In *2011 IEEE International Conference on Robotics and Automation*, pages 3607–3613. IEEE, 2011.

[75] Hung M La, Nenad Gucunski, Kristin Dana, and Seong-Hoon Kee. Development of an autonomous bridge deck inspection robotic system. *Journal of Field Robotics*, 2017.

[76] Hung M La, Ronny S Lim, Basily Basily, Nenad Gucunski, Jingang Yi, Ali Maher, Francisco A Romero, and Hooman Parvardeh. Autonomous robotic system for high-efficiency non-destructive bridge deck inspection and evaluation. In *Automation Science and Engineering (CASE), 2013 IEEE International Conference on*, pages 1053–1058. IEEE, 2013.

[77] Hung M La, Ronny Salim Lim, Basily B Basily, Nenad Gucunski, Jingang Yi, Ali Maher, Francisco A Romero, and Hooman Parvardeh. Mechatronic systems design for an autonomous robotic system for high-efficiency bridge deck inspection and evaluation. *IEEE/ASME Transactions on Mechatronics*, 18(6):1655–1664, 2013.

[78] Hung Manh La, Nenad Gucunski, Seong-Hoon Kee, and Luan Van Nguyen. Data analysis and visualization for the bridge deck inspection and evaluation robotic system. *Visualization in Engineering*, 3(1):6, 2015.

[79] Hung Manh La, Nenad Gucunski, Seong-Hoon Kee, Jingang Yi, Turgay Senlet, and Luan Nguyen. Autonomous robotic system for bridge deck data collection and analysis. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 1950–1955. IEEE, 2014.

[80] Steven M LaValle. *Planning algorithms*. Cambridge university press, 2006.

[81] Tuan Le, Spencer Gibb, Nhan Pham, Hung Manh La, Logan Falk, and Tony Berendsen. Autonomous robotic system using non-destructive evaluation methods for bridge deck inspection. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3672–3677. IEEE, 2017.

[82] Yanbo Li, Zakary Littlefield, and Kostas E Bekris. Sparse methods for efficient asymptotically optimal kinodynamic planning. In *Algorithmic foundations of robotics XI*, pages 263–282. Springer, 2015.

[83] Yanbo Li, Zakary Littlefield, and Kostas E Bekris. Asymptotically optimal sampling-based kinodynamic planning. *The International Journal of Robotics Research*, 35(5):528–564, 2016.

[84] Ronny Salim Lim, Hung Manh La, Zeyong Shan, and Weihua Sheng. Developing a crack inspection robot for bridge maintenance. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 6288–6293. IEEE, 2011.

[85] Ronny Salim Lim, Hung Manh La, and Weihua Sheng. A robotic crack inspection and mapping system for bridge deck maintenance. *Automation Science and Engineering, IEEE Transactions on*, 11(2):367–378, 2014.

[86] Jiunn-ming Lin, Mary Sansalone, and William B Streett. Procedure for determining p-wave speed in concrete for use in impact-echo testing using a p-wave speed measurement technique. *Materials Journal*, 94(6):531–539, 1997.

[87] Zakary Littlefield, Yanbo Li, and Kostas E Bekris. Efficient sampling-based motion planning with asymptotic near-optimality guarantees for systems with dynamics. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1779–1785. IEEE, 2013.

[88] Fei Liu, Mitja Trkov, Jingang Yi, and Nenad Gucunski. Modeling of pure percussive drilling for autonomous robotic bridge decks rehabilitation. In *2013 IEEE International Conference on Automation Science and Engineering (CASE)*, pages 1063–1068. IEEE, 2013.

[89] Fei Liu, Jingang Yi, and Nenad Gucunski. Accurate image mosaicing for bridge deck using graph optimization with gps data. In *2017 13th IEEE Conference on Automation Science and Engineering (CASE)*, pages 1090–1095. IEEE, 2017.

[90] Yahui Liu, Jian Yao, Kang Liu, Xiaohu Lu, and Menghan Xia. Optimal image stitching for concrete bridge bottom surfaces aided by 3d structure lines. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, 41, 2016.

[91] Gil Lopes, A Fernando Ribeiro, Neftalí Sillero, Luís Gonçalves-Seco, Cristiano Silva, Marc Franch, and Paulo Trigueiros. High resolution trichromatic road surface scanning with a line scan camera and light emitting diode lighting for road-kill detection. *Sensors*, 16(4):558, 2016.

[92] Steven J Lorenc, Brian E Handlon, and Leonhard E Bernold. Development of a robotic bridge maintenance system. *Automation in Construction*, 9(3):251–258, 2000.

[93] Bruce D Lucas, Takeo Kanade, et al. An iterative image registration technique with an application to stereo vision. In *Seventh International Joint Conference on Artificial Intelligence*. Vancouver, BC, Canada, 1981.

[94] Bengt Lundberg and P Collet. Optimal wave with respect to efficiency in percussive drilling with integral drill steel. *International Journal of Impact Engineering*, 37(8):901–906, 2010.

[95] Bengt Lundberg and M Okrouhlik. Efficiency of a percussive rock drilling process with consideration of wave energy radiation into the rock. *International Journal of Impact Engineering*, 32(10):1573–1583, 2006.

[96] Kai-Chieh Ma, Lantao Liu, and Gaurav S Sukhatme. Informative planning and online learning with sparse gaussian processes. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4292–4298. IEEE, 2017.

[97] DB Marshall, BR Lawn, and AG Evans. Elastic/plastic indentation damage in ceramics: The lateral crack system. *Journal of the American Ceramic Society*, 65(11):561–566, 1982.

[98] Kenneth R Maser and Alan Rawson. Network bridge deck surveys using high-speed radar: Case studies of 44 decks (abridgment). *Transportation Research Record*, (1347), 1992.

[99] A Miraliakbari, M Hahn, and HG Maas. Development of a multi-sensor system for road condition mapping. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 40(1):265, 2014.

[100] L. L. Mishnaevsky. Physical mechanisms of hard rock fragmentation under mechanical loading: A review. volume 32, page 763, 1995.

[101] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.

[102] Soheil Nazarian, Mark Richard Baker, and Kevin Crain. Development and testing of a seismic pavement analyzer. Technical report, National Research Council, 1993.

[103] Erling Nordlund. The effect of thrust on the performance of percussive rock drills. In *International Journal of Rock Mechanics and Mining Sciences & Geomechanics Abstracts*, volume 26, pages 51–59. Elsevier, 1989.

[104] Eric North, Jacques Georgy, Umar Iqbal, Mohammed Tarbochi, and Aboelmagd Noureldin. Improved inertial/odometry/gps positioning of wheeled robots even in gps-denied environments. In *Global Navigation Satellite Systems: Signal, Theory and Applications*. InTech, 2012.

[105] Kazunori Ohno, Takashi Tsubouchi, Bunji Shigematsu, and Shin'ichi Yuta. Differential gps and odometry-based outdoor navigation of a mobile robot. *Advanced Robotics*, 18(6):611–635, 2004.

[106] Tomas Olsson, Mathias Haage, Henrik Kihlman, Rolf Johansson, Klas Nilsson, Anders Robertsson, Mats Björkman, Robert Isaksson, Gilbert Ossbahr, and Torgny Brogårdh. Cost-efficient drilling using industrial robots with high-bandwidth force feedback. *Robotics and Computer-Integrated Manufacturing*, 26(1):24–38, 2010.

[107] S. S. Pang and W. Goldsmith. Momentum and energy processes during jackhammer operation. *Rock mechanics and rock engineering*, 22:205–229, 1989.

[108] S. S. Pang and W. Goldsmith. A model of a pneumatic jackhammer system. *Rock mechanics and rock engineering*, 25:49–61, 1992.

[109] S. S. Pang, W. Goldsmith, and M. Hood. A force-indentation model for brittle rocks. *Rock mechanics and rock engineering*, 22:127–148, 1989.

[110] Georgios Papadopoulos. *Asymptotically optimal path planning and surface reconstruction for inspection*. PhD thesis, Massachusetts Institute of Technology, 2014.

[111] Georgios Papadopoulos, Hanna Kurniawati, and Nicholas M Patrikalakis. Asymptotically optimal inspection planning using systems with differential constraints. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 4126–4133. IEEE.

[112] Georgios Papadopoulos, Hanna Kurniawati, and Nicholas M Patrikalakis. Analysis of asymptotically optimal sampling-based motion planning algorithms for lipschitz continuous dynamical systems. *arXiv preprint arXiv:1405.2872*, 2014.

[113] Robert Parrillo, Roger Roberts, and A Haggan. Bridge deck condition assessment using ground penetrating radar. In *ECNDT Conference Proceeding, Berlin, Germany*, pages 25–29. Citeseer, 2006.

[114] Ekaterina Pavlovskaia and Marian Wiercigroch. Modelling of vibro-impact system driven by beat frequency. *International Journal of Mechanical Sciences*, 45(4):623–641, 2003.

[115] Ekaterina Pavlovskaia, Marian Wiercigroch, and Celso Grebogi. Modeling of an impact system with a drift. *Physical Review E*, 64(5):056224, 2001.

[116] Prateek Prasanna, Kristin J Dana, Nenad Gucunski, Basily B Basily, Hung M La, Ronny Salim Lim, and Hooman Parvardeh. Automated crack detection on concrete bridges. *IEEE Transactions on Automation Science and Engineering*, 13(2):591–599, 2016.

[117] Michael Raupach, Kenji Reichling, Herbert Wiggenhauser, Markus Stoppel, Gerd Dobmann, and Jochen Kurz. Betoscan–an instrumented mobile robot system for the diagnosis of reinforced concrete floors. In *Concrete Repair, Rehabilitation and Retrofitting II*, 2009.

[118] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2564–2571. IEEE, 2011.

[119] Alberto Viseras Ruiz and Calin Olariu. A general algorithm for exploration with gaussian processes in complex, unknown environments. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3388–3393. IEEE, 2015.

[120] Mary Sansalone. Impact-echo: The complete story. *ACI Structural Journal*, 94:777–786, 11 1997.

[121] Changsheng Shen, Yuanzhao Zhang, Zimo Li, Fei Gao, and Shaojie Shen. Collaborative air-ground target searching in complex environments. In *2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*, pages 230–237. IEEE, 2017.

[122] Satoshi Suzuki et al. Topological structural analysis of digitized binary images by border following. *Computer vision, graphics, and image processing*, 30(1):32–46, 1985.

[123] Richard Szeliski. Image alignment and stitching: A tutorial. *Foundations and Trends® in Computer Graphics and Vision*, 2(1):1–104, 2006.

[124] TRB. Nondestructive testing to identify concrete bridge deck deterioration. Technical report, Transportion Research Board (TRB), National Academies of Sciences, Engineering, and Medicine,, 2010.

[125] Mitja Trkov, Fei Liu, Jingang Yi, and Haim Baruh. Study of concrete drilling for automated non-destructive evaluation and rehabilitation system for bridge decks. In *Nondestructive Characterization for Composite Materials, Aerospace Engineering, Civil Infrastructure, and Homeland Security 2011*, volume 7983, page 798307. International Society for Optics and Photonics, 2011.

[126] Steven A Velinsky. Technical report/new innovations. heavy vehicle system for automated pavement crack sealing. *International Journal of Heavy Vehicle Systems*, 1(1):114–128, 1993.

[127] Zhe Wendy Wang, Mengchu Zhou, Gregory G Slabaugh, Jiefu Zhai, and Tong Fang. Automatic detection of bridge deck condition from ground penetrating radar images. *IEEE transactions on automation science and engineering*, 8(3):633–640, 2011.

[128] Herbert Wiggenhauser. Advanced ndt methods for the assessment of concrete structures. In *Concrete Repair, Rehabilitation and Retrofitting II*, pages 33–44. CRC Press, 2008.

[129] Herbert Wiggenhauser and Ernst Niederleithinger. Innovative ultrasonic techniques for inspection and monitoring of large concrete structures. In *EPJ Web of Conferences*, volume 56, page 04004. EDP Sciences, 2013.

[130] Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT Press Cambridge, MA, 2006.

[131] Saeed Yahyanejad, Daniel Wischounig-Strucl, Markus Quaritsch, and Bernhard Rinner. Incremental mosaicking of images from autonomous, small-scale uavs. In *Advanced Video and Signal Based Surveillance (AVSS), 2010 Seventh IEEE International Conference on*, pages 329–336. IEEE, 2010.

[132] Zhi Yan, Nicolas Jouandeau, and Arab Ali Cherif. A survey and analysis of multi-robot coordination. *International Journal of Advanced Robotic Systems*, 10(12):399, 2013.

[133] Liman Yang, Zhongwei Guo, Yunhua Li, and Chao Li. Posture measurement and coordinated control of twin hoisting-girder transporters based on hybrid network and rtk-gps. *IEEE/ASME Transactions on Mechatronics*, 14(2):141–150, 2009.

[134] Jingang Yi, Hongpeng Wang, Junjie Zhang, Dezhen Song, Suhada Jayasuriya, and Jingtai Liu. Kinematic modeling and analysis of skid-steered

mobile robots with applications to low-cost inertial-measurement-unit-based motion estimation. *IEEE transactions on robotics*, 25(5):1087–1097, 2009.

[135] Chen Yu, Jianan Wang, Yan Ding, Jiayuan Shan, and Ming Xin. Feedback-control-aided image stitching using multi-uav platform. In *Intelligent Control and Automation (WCICA), 2016 12th World Congress on*, pages 2420–2425. IEEE, 2016.

[136] Seung-Nam Yu, Jae-Ho Jang, and Chang-Soo Han. Auto inspection system using a mobile robot for detecting concrete cracks in a tunnel. *Automation in Construction*, 16(3):255–261, 2007.

[137] Deren Yuan, Soheil Nazarian, Dar-Hao Chen, and Fred Hugo. Use of seismic pavement analyzer to monitor degradation of flexible pavements under texas mobile load simulator. *Transportation Research Record: Journal of the Transportation Research Board*, (1615):3–10, 1998.

[138] Bin Zhang and Gaurav S Sukhatme. Adaptive sampling for estimating a scalar field using a robotic boat and a sensor network. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 3673–3680. IEEE, 2007.

[139] Hang Zhou, Dongxiang Zhou, Keju Peng, Ruibin Guo, and Yunhui Liu. Seamless stitching of large area uav images using modified camera matrix. In *Real-time Computing and Robotics (RCAR), IEEE International Conference on*, pages 561–566. IEEE, 2016.