Numerical Modelling of Structural Behaviour of Continuously Reinforced Concrete Pavement

A Thesis
Submitted in Partial Fulfilment of the Requirements
For the Degree of
Bachelor of Engineering
In
Civil Engineering (Structural)
By

Tin Lai
430381847

Supervisor: Dr. Daniel Dias-da-Costa

School of Civil Engineering
University of Sydney, NSW 2006
Australia

October 2017

Disclaimers

Student Disclaimer

The work comprising this thesis is substantially my own, and to the extent that any part of this work is not my own I have indicated that it is not my own by acknowledging the source of that part or those parts of the work. I have read and understood the *University of Sydney Student Plagiarism: Coursework Policy and Procedure*. I understand that failure to comply with the *University of Sydney Student Plagiarism: Coursework Policy and Procedure* can lead to the University commencing proceedings against me for potential student misconduct under chapter 8 of the *University of Sydney By-Law 1999* (as amended).

Departmental Disclaimer

This thesis was prepared for the School of Civil Engineering at the University of Sydney, Australia, and describes a numerical study on performance of CRCP. The opinions, conclusions and recommendations presented herein are those of the author and do not necessarily reflect those of the University of Sydney or any of the sponsoring parties to this project.

ABSTRACT

Numerical Modelling of Structural Behaviour of Continuously Reinforced Concrete Pavement

by

Tin Lai

Continuously reinforced concrete pavement (CRCP) rapidly becomes a common standard in most highway pavement design due to its high performance. Early-age cracks play a crucial role in characterising CRCP's long-term performance, hence crack formations and patterns are of interest for most highway departments. The design parameters of a CRCP are studied in this work as a parametric study on efficient design configuration. Numerical experiments were performed to produce a stress profile induced by crack initiation and long-term traffic load. This work contributes a probabilistic study on crack pattern formation from typical stress profile, including environmental and shrinkage stress, as an insight into predicting the likelihood of crack patterns initiation and propagation along the depth of the concrete slab. Studies also showed transverse steel bar plays a crucial role in determining the location of the crack formation, but a minimal effect in controlling crack width and deflection. A study on the probabilistic model helps pavement design to produce a desirable crack pattern, to perform controlled-cracks that enhance the vertical load transferability of a cracked CRCP.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

1.1.1 Continuously Reinforced Concrete Pavement

Continuously reinforced concrete pavement (CRCP) is a type of highly reinforced rigid pavement that requires no preventive measure such as transverse expansion or contraction joints. It is by design that transverse cracks are expected to propagate throughout the slab, which is held together by embedding enough reinforcing steel to limit cracks' formation and effects. Due to the increased quantities of steel, it has a relatively higher cost. However, compare to other pavement types it can achieve superior long-term performance and cost-effectiveness when compared to other pavement type (Lee et al. 2014), with the ability to have service life for over 50 years under severe weather and heavy traffic conditions (Yeon 2015). The most common usage of CRCP design is in an area where load repetitions in the order of tens of millions are expected, such as in corridors where there are dense urban traffic corridors over the service life of the pavement.

Concrete pavements usually contain three layers: concrete pavement, base, and subbase. It is the concrete pavement layer that is of concern about, as cracks are the crucial factors that determine its performance. The structural design of CRCP includes the amount of steel bar reinforcement to control the length and formations of transverse cracks, the location of the steel bars, and the slab thickness to resist traffic loads (Nishizawa et al. 1998). Procedures had long been established regarding how to predict the width and spacing of transverse crack (Sato, Hachiya, and Kawakami 1989), which are often incorporated into the specification of national standards for concrete pavement design. One of the main problems with the performance of CRCP is it is heavily dependent on the early-age cracking caused by changes in temperature and drying shrinkage (Kim, M. C. Won, and McCullough 2000). The improvement of CRCP design in the past years had been focused on minimising the early-age cracking and controlling the propagation of cracks to prevent structural punchout. Numerous studies had been performed on examining how CRCP behave under different environmental conditions to refine its construction practices on crack control.

1.1.2 Long Term Performance of CRCP

CRCP had become one the most widely used rigid pavement type in highway design due to its durable nature and minimum maintenance cost. Its advantages over other rigid pavement include the lack of the needs for any transverse contraction joints or any maintenance for transverse joint sealant. In light of the long-term performance of CRCP, the failure model has been distinguished by Yoder and Matthew W Witczak (1975) into two main type - structural failure and functional failure.

Measurable structural failure in pavement is considered as a progressive phenomenon occurring with time or with an increase in loading and is associated with as the breakdown of one or more of the pavement components. It happens when a pavement reaches an established unacceptable level of distress, such as spalling or punchouts. Whereas functional failure is referring to the characteristic of functionality in the pavement can no longer be carried out, which is the safety and comfort for the user. Regarding serviceability in CRCP, structural failures lead to functional failures. Therefore, the prime contributory factor that is of primary concern in this study is structural failures. In particular, the structural failure that is of interest in this study is the early-age cracks. It is a cracking mechanism that is of dominant to CRCP within the first few weeks or months, and most of them are formed before the pavement is open for traffic. Environmental thermal load and autogenous shrinkage will be the prime contributor. Performance of CRCP after cracks are fully developed will also be investigated to find a favourable configuration of CRCP that enhance long-term performance.

1.2 Research Contributions

The contributions of this thesis is as follow:

i. Parametric studies of the structural behaviour of CRCP under the various configuration of steel bars and environmental loading.

ii. Proposed a basis of a probabilistic model of CRCP early-age crack initiation and propagation pattern, for crack prediction.

iii. Conducted studies of crack width development in a cracked CRCP section.

iv. Performed studies of vertical load transfer from traffic loading in a cracked CRCP section.

1.3 Thesis Organisation

This thesis is organised as follows:

- *Chapter 2*: This chapter presents a survey of previous studies on CRCP. In particular, the focus had been on the formation of early-age cracks, and the impact of environmental loading on such.
- *Chapter 3*: Methodology are derived in this chapter on the three main stages of analysis that are conducted in this research.
- *Chapter 4*: This chapter presents the result and discussion arise from the numerical modelling of different stages of CRCP.
- *Chapter 5*: A summary of the thesis contents and its contributions are given in the final chapter. Recommendation for future works is given as well.

Chapter 2

Literature Review

CRCP is a type of pavement where cracks are not to be avoided, but it was designed to be allowed to relieve stress concentrated throughout the continuous slab. Those cracks break the continuity of concrete; thus it will significantly influence the performance of the pavement. The shrinkage of concrete forms most cracks and lead to numerous transverse crack along the transverse direction. Majority of cracks happens before the pavement is open to traffic (Plei and Tayabji 2012), and continue to propagate afterwards due to weathering and fatigue traffic loading.

Design specification of transport department in various countries usually adopts three main criteria in designing CRCP, including (1) stress of steel bar, (2) width of crack and (3) space between cracks (McCullough, Noble, and Ma 1979). All criteria mentioned are related to shrinkage stress of CRCP which are contributed mostly by early age cracking. It has long been theorised that the characteristics of transverse cracks (crack spacing and width) have substantial effects on the development of CRCP punchout which is a result of a localised structural distress. Many previous types of research had been focusing on the predictive mechanistic models that could simulate CRCP behaviour to study on the characteristic of transverse crack spacing and width (M. Witczak, Andrei, and Houston 2004), as a predictive model will provide excellent insight for CRCP designing and to analyse its cracking behaviour.

2.1 Early-Age Cracking

The growth of crack width within concrete slabs is a direct result of internal stress concentration. When such internal stresses exceed the concrete strength, a crack may occur in such a location. Shrinkage of concrete includes drying shrinkage and thermal shrinkage. The former will reaches its limit within the first two or three years and remain changeless afterwards (Ju 2010); thus thermal shrinkage will govern as the contributory factor of shrinkage stress in the later stage. Drying shrinkage and temperature drop in concrete will cause volumetric changes in all direction of the pavement, which would induce contractions in transverse, longitudinal, and also vertical directions. The volume changes when subjected to environmental loadings such as temperature and moisture variations are severely restrained by base friction and longitudinal steel, resulting in formations of transverse cracks.

2.1.1 Autogenous Shrinkage

When subject to thermal loading, concrete is the only material that is affected by autogenous shrinkage. Autogenous shrinkage refers to the critical phenomenon in young concrete. It occurs in the hydration process in the early age after concrete was poured, where most water contents were rapidly drawn into the hydration process (Choi, Ha, and M. C. Won 2015). This phenomenon of demands for water leads to the formation of tensile stress and can lead to cracking. Conventionally curing is for this purpose which seals the surface of the concrete to prevent such evaporation of water. Therefore, the curing process after concrete was poured is of essential to minimise the effect shrinkage. Autogenous shrinkage is of importance as this only affects concrete, leads to deviation in the volumetric change in CRCP, and induce tensile stress from surrounding constraints.

2.1.2 Thermal Effect in Concrete

During the early age of CRCP, environmental loading has a direct and significant effect on it and will impact its performance. Cracks will be formed due to the environmental condition that it is exposed. Conditions such as temperature and moisture content are the critical factors, which will induce stress within the CRCP. The concrete is restrained by conditions such as its reinforcement and sub-layer friction. However, the change in temperature and moisture content will cause the concrete to expand or contract; hence the formation of cracks of CRCP during early age. The most influential factor is due to the change in temperature gradient along the depth of CRCP. Al-Qadi and Elseifi (2006) criticise that older literature often assumed a linear profile of temperature gradient along the depth of concrete, but the precise temperature profile follows a non-linear gradient. Drying shrinkage is then induced by the loss of moisture content from the temperature, which in turn induces stress within the CRCP due to the restrained movement.

The secondary phase cracking comprises differential shrinkage and pavement curling induced cracks, commonly known as pavement warping, which occurs when the rate of shrinkage differs between the pavement surface and the pavement bottom (Gerber 2011). The temperature profile will also vary between day and night. Figure 2.1 demonstrates how the curling effect will induce different internal stress and cracks in the concrete slab. This effect is caused by the change of temperature gradient along the depth of concrete, which are the characteristics of pavement warping. These effects were not considered in many older studies, where a constant temperature profile was used about time, which limits the accuracy of the model and will underestimate its effects has on the long-term performance of CRCP.

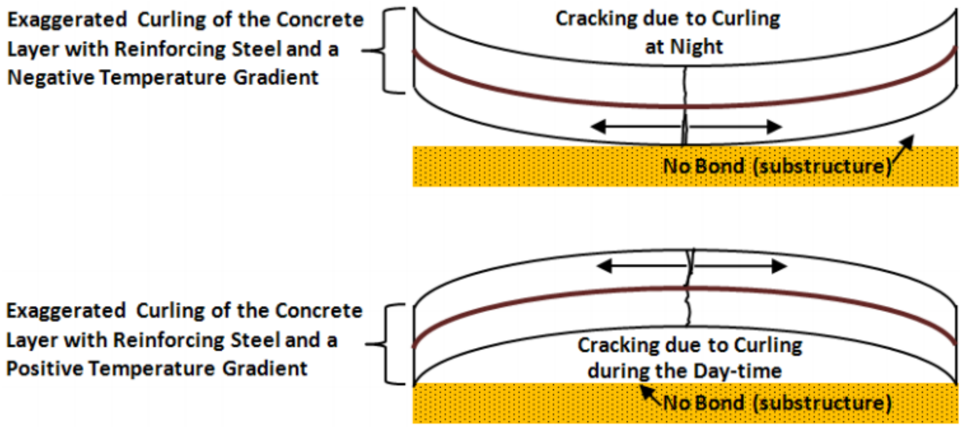


Figure 2.1: Depictions of slab curling during the day and at night with positive and negative temperature gradients respectively (Gerber 2011)

2.1.3 Zero-Stress Temperature

Thermal stress development in CRCP depends on thermal properties (such as heat of hydration and coefficient of thermal expansion), conditions at placement, environmental conditions, and the zero-stress point (Nam 2005). These are several designs and construction variables considered to have the most significant effects on CRCP behaviour and performance. A further change in temperature will always induce stresses within the concrete structure.

Thermal properties and zero-stress point are dependent on material properties and the dimension of CRCP respectively. The former is controllable via altering concrete contents, and the latter is dependent on temperature when concrete was poured. Setting temperature is the temperature at which concrete begins to resist thermal load, and it defines the zero-stress temperature of the CRCP (Yeon 2015). Cast-in-place concrete structures experience a large volume expansion shortly after concrete placement due to the heat of hydration of cement, and compressive stress build-up within the structure due to the concrete structure being restrained by external environments such as subbase friction or reinforcement steel bars. Developed compressive stress will eventually equilibrium with the external environment until it is completely relieved, reaching the point where longitudinal residual stress in concrete element becomes zero. Temperature at this point it is referred to as zero-stress temperature, as demonstrated in Figure 2.2 (Yeon, Choi, and M. C. Won 2013).

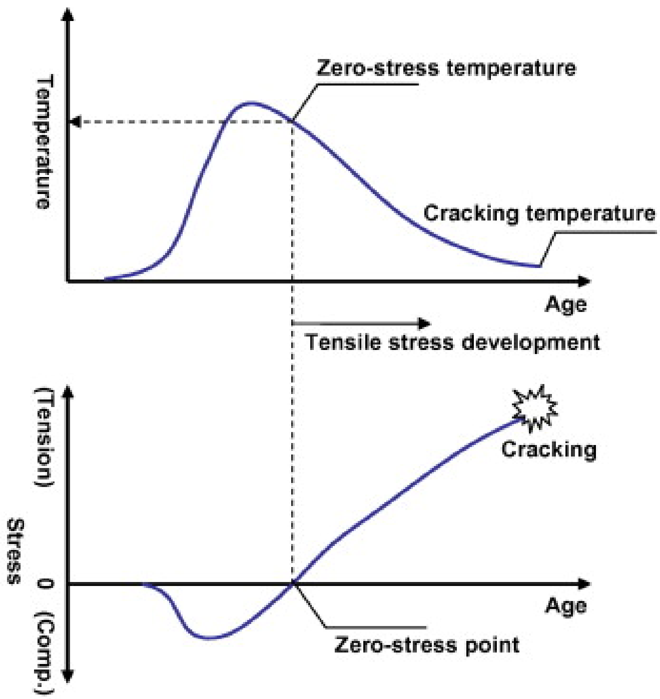


Figure 2.2: Zero stress formation in typical restrained concrete structure (Yeon, Choi, and M. C. Won 2013)

2.1.4 Thermal Load Pattern

Thermal load pattern is a particularly important factor in determining CRCP early age cracking. The thermal profile along the slab will induce expansion or contraction within the concrete slab, which will produce stresses in concrete structures of which in the same order of magnitude as the dead or live loads in some cases (El-Tayeb et al. 2017). A field study on temperature along the depth of CRCP by Kim and Nam (2010) is shown in Figure 2.3, demonstrating the temperature changes of CRCP due to the day-night cycle. Stresses are produced when there are volumetric changes in the concrete, and there's restrains by subbase friction and steel bars. The thermal expansion of concrete and steel governs the magnitude of stress, and the effect of temperature variation is significant (F. Vecchio, Agostino, and Angelakos 1993). The variations in temperature, moisture content, non-linear thermal gradients or non-uniformly cracked members are important factors that should be included in finite element modelling as they yield radically different results (El-Tayeb et al. 2017). However, many previous studies resorted to overly simplistic single section analysis in their numerical modelling and ignored the overall structural response (Frank J Vecchio 1987; Priestley et al. 1972), resulting in an inadequate numerical result that is far from reality. It is important to note that values of temperature change and temperature gradient vary from country to country. Therefore, when performing a numerical model for a particular place, it is preferable to choose a gradient of temperature changes that are similar to the local suburb temperature profile.

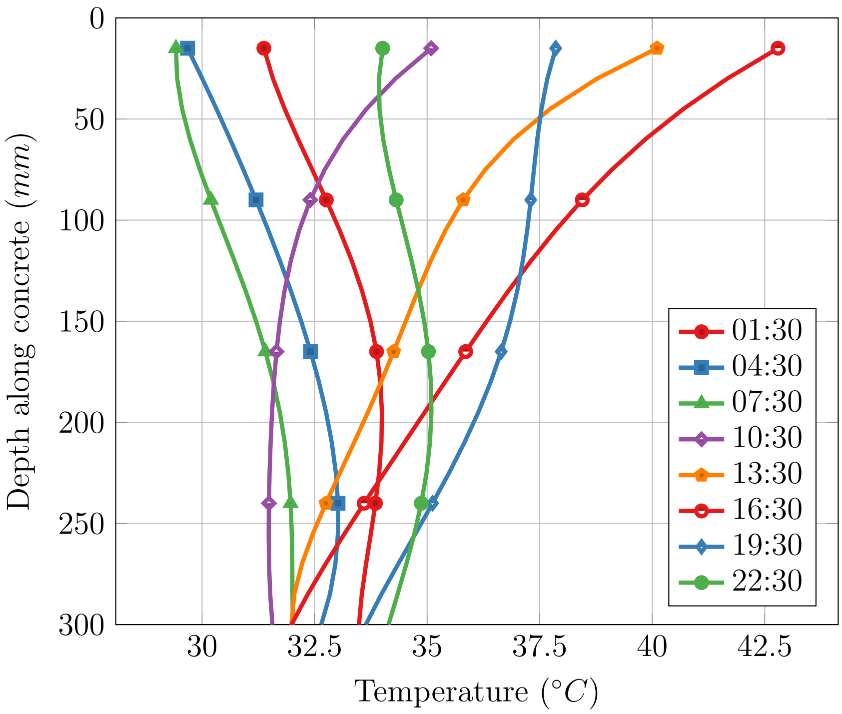


Figure 2.3: Daily cycle of concrete temperature profile (Kim and Nam 2010)

Studies conducted by El-Tayeb et al. (2017) concluded the response of concrete are sensitive to the temperature gradient of being linear or non-linear and they deviate significantly. It is more realistic to model the temperature profile gradient as non-linear as it is closer to experimental data from realistic concrete frames.

2.2 Numerical Modelling CRCP Behaviour

2.2.1 Bond-Slip Behaviour

Similar to other composite structures, the importance of the fibre-matrix bond in fibre reinforced concrete is well recognised. In cement-based composite under tension or flexure, the fibres will transmit stresses across the bonded structure across the section. However, the stress developed in the early-age crack of CRCP largely depends on the bond-slip characteristics between concrete and steel bar (Kim, M. C. Won, and McCullough 2000). When simulating thermal behaviour of CRCP in FEA, an accurate bond-slip model are needed for an accurate modelling of the mechanistic model. When concrete expands or contracts, it will induce frictional stresses in the tangential direction at the interface between the concrete layer and steel bar. The frictional bond-slip between the two the bonded layers can be modelled using spring elements. The relationship between the bond stress and the slip had been simulated in many previous studies as a finite amount number of spring elements that connect two instances together (Kim, M. Won, and McCullough 2001). A realistic bond-slip as shown in Figure 2.4, is one with a non-linear gradient with an ultimate slip, following with a point where stiffness reaches zero when the bonding breaks. The accuracy of the bond-slip behaviour is dependent on the number of elements used to simulate such bond-slip, and will ultimately approach a realistic behaviour bond-slip model as the number of spring elements used increases.

2.2.2 Winkler Model

The interaction between slab and foundation are of essential as the frictional stress determines the magnitude of stress that concrete slab experience. Non-linear temperature gradients often produce slab expansion and contraction, which leads to frictional traction between slabs and foundation. Prediction and simulation on these friction tractions are complicated by the curling of the slabs that cause some portions of the slabs to lose contact with the foundation. Therefore, a more realistic approach would be modelling the frictional traction as a tension-less stiffness in between concrete slab and subbase.

Winkler model is one of the numerical approaches on simulating an elastic foundation.

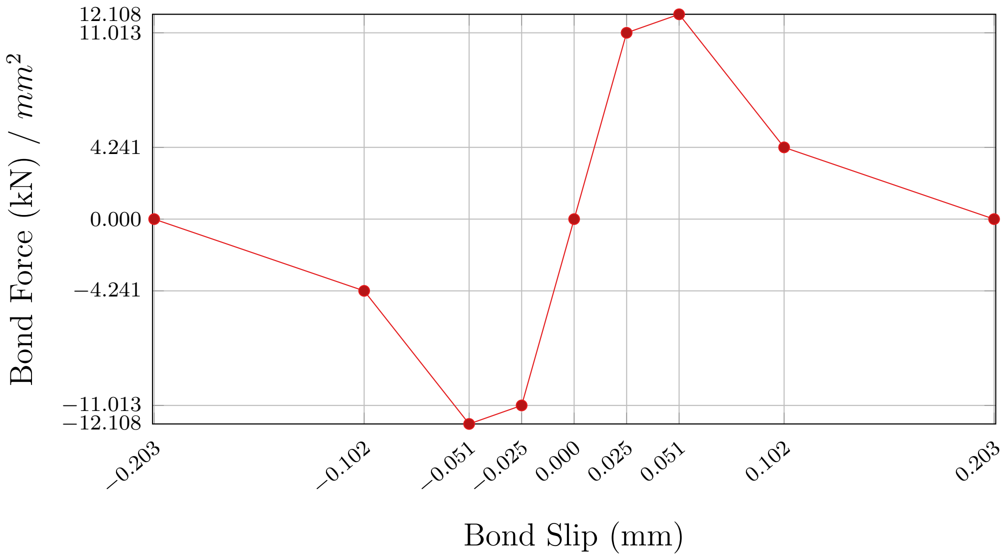


Figure 2.4: Bond-slip model between concrete and steel bar

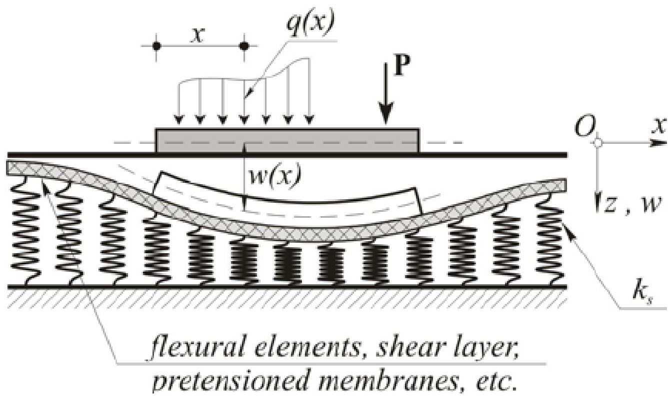


Figure 2.5: Winkler model for simulating soil interaction between the concrete slab and foundation (Teodoru 2009)

It simulates the soil-structure interaction problem as a discretised finite amount of spring elements that exceed a stiffness between each layer. Figure 2.5 shows the relations in between concrete slab and the foundation layers. There are a finite amount of stiffness members that connect each layer while enabling a different degree of stiffness to be defined by researchers. Modelling of bonds stiffness is highly desirable in numerical analysis that involves interaction between concrete and soil-like foundation.

2.3 Transverse Crack Development

Transverse crack spacing and crack widths in CRCP has been cited as one of the most critical pavement structural responses that determine its long-term performance. Those two had found to have substantial effects on the development of structural distress in CRCP, which is punchout. Most cracks occur directly near transverse steel along the transverse direction, and a high number of over $60 - 80\%$ of cracks in field observation were found to occur at or near transverse steel (Ryu et al. 2013). The high cracking likelihood indicates a substantial interaction between transverse steel and other factors causing transverse cracks. The higher probability is primarily due to the interactions between concrete volume contraction vertically and transverse steel cause more considerable concrete tensile stresses near transverse steel than other areas.

Mechanism of cracks development is of interest as a predictive model to simulate crack behaviour. If characteristics of transverse cracks have substantial effects of structural distress of CRCP, transverse crack characteristics could be controlled by adjusting transverse steel designs in the current design methods. Most current designs are based on placing just enough transverse steel to support longitudinal steel during concrete placement.

2.3.1 Crack With and Crack Spacing

Transverse crack spacing and crack width plays a crucial role in distress development and long-term performance of CRCP and governs the progressive damage that leads to pavement failure (Kohler and Roesler 2006). Concrete fatigue damage process that defines the structural failure in CRCP is mostly due to excessive crack width development in between each cracked section, which leads to deterioration of the aggregate interlock mechanism at the transverse cracks. The vertical load transfers between each cracked section are mostly contributed by longitudinal steel bars and aggregate interlock, and the failure in the latter mechanism leads to the reduction in load transfer capacity across transverse cracks. A wider crack will increase the tensile bending stress at the surface of the concrete slab, which leads to various pavement distresses, including spalling, punchouts, and steel rupture.

As suggested by experimental performed by Suh and McCullough (1994), construction season, coarse aggregated type, amount of steel and slab temperature at the time of placement will significantly affect crack width development. Warmer weather at the time of installation produces much wider cracks than in colder weather. Using a more considerable amount of longitudinal steel will create a narrower crack width. The effect of crack spacing produces no significant effect on crack width. Steel placement appears to have the most critical effect of crack formation alongside with environmental loading at the time of installation.

Chapter 3

Methodology

3.1 Analysis Stage and Properties

The analysis of this research is focused on the cracking mechanism of CRCP and its implication on CRCP in long-term performance. In particular, the area of interest is on (a) crack initiation within a few days after concrete is poured, and (b) performance of a cracked CRCP section after early age cracks are fully developed. The former will be modelled as a long un-cracked pavement with applied temperature profile and early age shrinkage. The latter will have similar properties but with a cracked boundary condition that is free to deform. The analysis will focus on thermal loads, autogenous shrinkage, and traffic loads. Material properties used in this study are listed in Table 3.1. The effects of tension stiffening after cracks had also been taken into consideration.

Table 3.1: Material properties used in CRCP modelling

CRCP Material Properties	
Concrete Strength	Concrete Grade 45
Density	2400 kg/m^3
Poisson's Ratio	0.15
Elastic Modulus (Before Crack)	14800 MPa
Elastic Modulus (After Crack)	23000 MPa
Concrete COTE	$7.2 \times 10^{-}6/^{\circ}C$
Steel COTE	$1.08 \times 10^{-}5/^{\circ}C$

The material properties used are standard practice in Australian Road Design under AS5100. Parametric study on material properties are not included in this study; instead, this study focused on the effects of configuration and placement of the steel bars on CRCP long-term performance. Traffic loads are tested according to typical loading in AS5100 and detailed in Section 3.3.4.

The dimension of CRCP is 1500mm Width $\times$ 300mm Height $\times$ 3200mm Depth for a cracked section. Whereas for the un-cracked pavement, width is assumed to be infinite with a continuous boundary on both ends. The detailed drawings for each analysis stage are shown in Section 3.3.

3.2 Parametric Study

Parametric Study had been performed on the placement and configuration of reinforcement steel bar. In particular, the depth of steel bar reinforcements, the diameter of the transverse steel bar, thermal temperature profile and the use of double reinforcement layers will be investigated. Table 3.2 summarised the type of parameters tested in this research.

Table 3.2: Parametric study on structural behaviour of CRCP

Parametric Study		
Transverse Steel Diameter	**Steel Bar Layers Depth**	**Temperature Profile** (ref. Section 2.3)
⌀12 mm	90 mm	01:30
⌀16 mm	120 mm	07:30
⌀20 mm	150 mm	13:30
⌀24 mm	180 mm	16:30
	210 mm	

Environmental loading for temperature profile used in this research is shown in Figure 3.1, which acts as the thermal loads in crack initiation and crack width analysis. Drying shrinkage profile in Figure 3.2 refers to autogenous shrinkage that concrete experience in its early-age cracking stage and is applied alongside with thermal loads.

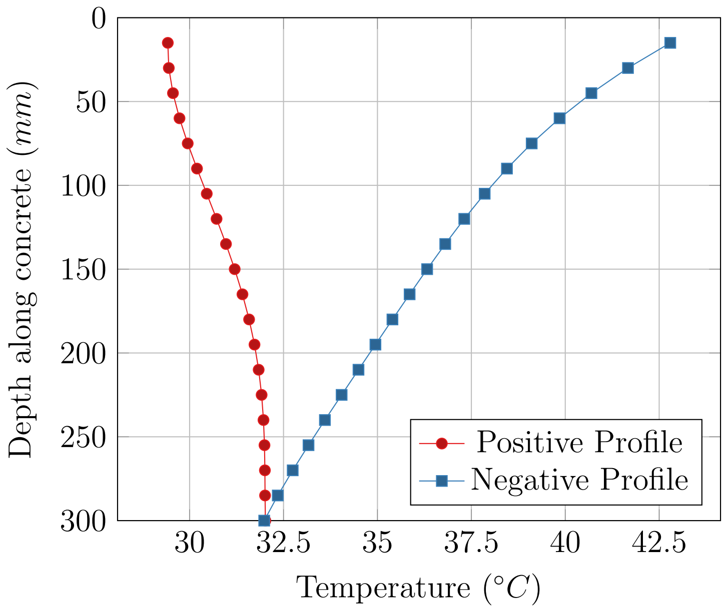


Figure 3.1: Temperature profile along depth of concrete

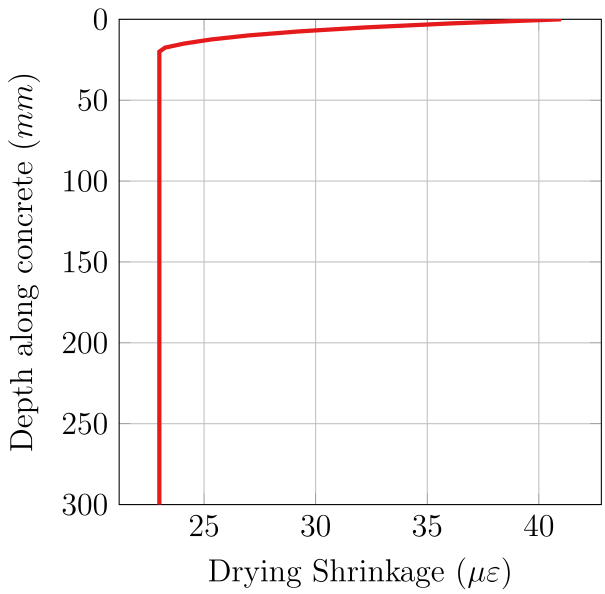


Figure 3.2: Shrinkage profile along depth of concrete

3.3 Numerical Modelling Details

The numerical modelling of this study had been divided into three main parts to study different structural behaviour of CRCP in various stage and under different loading condition – (1) *Crack Initiation* for a long un-crack pavement, (2) *Crack Width Analysis* for a cracked pavement section, and (3) *Traffic Load Performance* between each cracked section.

Table 3.3: Numerical modelling details for each stages

	Stage 1	Stage 2	Stage 3
Simulating	Crack Initiation	Crack Width	Traffic Load
Element Types	Plane Stress Element		3D Stress Element
Analysis Types	Static Thermal		Dynamic Explicit
Number of Elements	5,856	2,468	83,237
Mesh Type	Quadriateral		Hexahedral

Each analysis is aimed to simulate the same pavement with the same dimension and material properties. Numerical model settings are summarised in Table 3.3. Note that the number of elements is only a rough estimate for the analysis type as it varies depending on the parametric study settings; number shown represents the control group for the analysis stage. The numerical result is first replicated by comparing to an existing research by Kim, M. C. Won, and McCullough (2000) to verify the correctness of stress or displacement result. Further modification and analysis of parametric study are then performed with that as the basis. Abaqus scripting interface has been used to reproduce model with different dimension and configuration consistently. The Python script used is included in Appendix A as an illustration of the modelling process. The following section will be about the model details between each stage.

3.3.1 Stage 1 – Crack Initiation

Crack initiation is the first stage of the modelling of the structural behaviour of CRCP, under environmental loading. The aim is to model a realistic numerical model to simulate early-age cracking of CRCP as mentioned in section 2.1. The simulation of such shrinkage behaviour provides a basic model with concentrated stress that is of approximate to where CRCP would initiate cracks. Crucial modelling details include the non-linear thermal load's patterns that exist during afternoon time or night-time when the concrete is laid. The former will be a negative gradient of temperature change whereas the latter will be one with a positive gradient. The positive gradient temperature profile is taken from the experimental result by Kim and Nam (2010). In addition to temperature changes profile, shrinkage was also included. The effect of shrinkage has a slightly different nature compared to thermal load. The thermal load will affect both the concrete slab and the steel bars with their magnitude determined by their corresponding Coefficient of

Thermal Expansion (COTE). Whereas for shrinkage profile it will only affect concrete for its autogenous shrinkage as mentioned in section 2.1. Both temperature and shrinkage profile is defined as a *Analytical Field* function in ABAQUS, hence their effects will be continuously computed between the discretised elements by ABAQUS. The temperature profiles and shrinkage profile used in the model could be found in Figure 3.1 and 3.2 respectively.

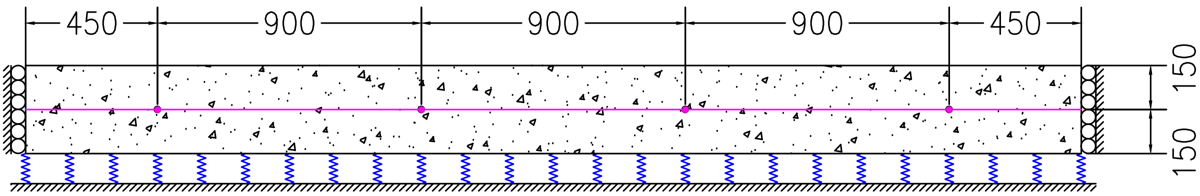


Figure 3.3: Full length un-cracked CRCP dimensions

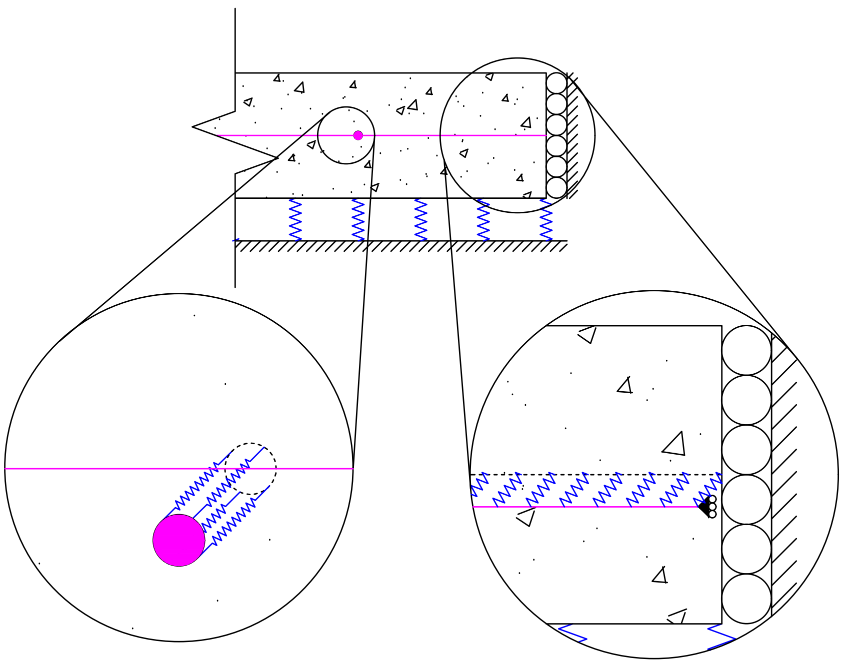
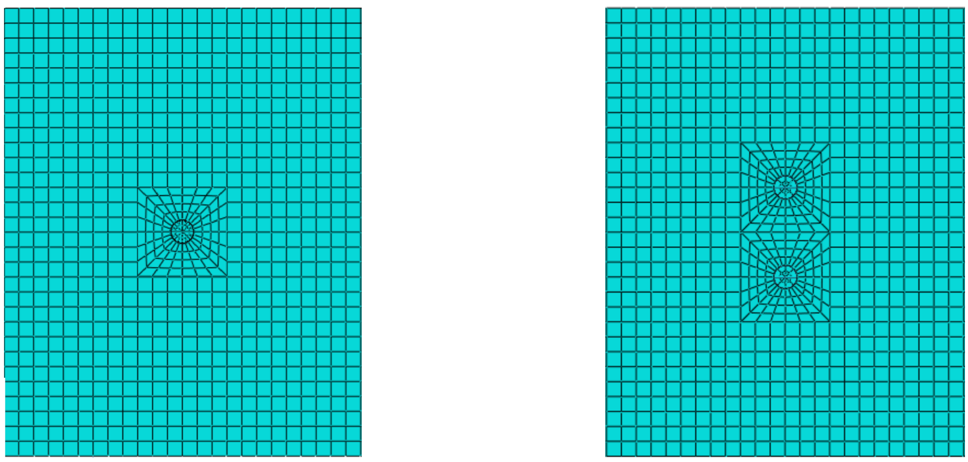


Figure 3.4: Modelling details highlighting bond-slip model
(Left) Transverse steel bar
(Right) Longitudinal steel bar

The modelling type used in this analysis is two-dimensional plane stress element to reduce computational time. The configuration of the un-crack CRCP is shown in Figure 3.3. The transverse steel bar spacing and CRCP slab dimension are designed in accordance to AS5100 (Standard 2004) and taken from the most common dimension from previous studies (Darestani et al. 2006).

Figure 3.4 shows a part of the FEA model used in this research, emphasising the bonding between concrete slab. The stress developed within the concrete slab occurs due to the shrinkage of concrete and restrains by subbase friction and steel bars. As discussed in section 2.2.1 the simulation of bond-slip behaviour is crucial for the thermal shrinkage of CRCP. Therefore, *connectors* with non-linear stiffness will be used in ABAQUS as spring element for the non-linear bond-slip behaviour. Bi-linear with ultimate slip mode will be used for the bond-slip between the concrete slab and steel bars, with stiffness profile as shown in Figure 2.4. The interaction between concrete slab and subbase are modelled using the winkle model as discussed in Section 2.2.2, which acts as a tension-less layer with a finite amount of spring elements. The concrete slab is discretised using two-dimensional plane stress brick elements. Longitudinal reinforcing steels are modelled using beam elements. Transverse reinforcing steels are modelled using a circular plan stress element. Holes were cut in the concrete element and is connected to plane stress transverse steel bar element with spring elements. The mesh around the circular hole is refined, and efforts were mainly to keep the dimension of meshes that are of close approximation to a square, as shown in Figure 3.5. The distance between each reinforcement layers in a parametric study and mesh dimension is chosen in such a way that elements around one layer of reinforcement in Figure 3.5a, or double reinforcement layers in Figure 3.5b would remain the same; to avoid stress distortion due to a distorted mesh.



(a) Single reinforcement layers **(b)** Double reinforcement layers

Figure 3.5: Mesh pattern around circular transverse steel bar

3.3.2 Probabilistic Crack Pattern

A probabilistic model of crack initiation pattern is one of the main objectives of this research. Stress profile subjected to the early-age crack mechanism (from thermal temperature loads and autogenous shrinkage) along the entire depth and width of CRCP will be collected. The stress profile is then applied to the concrete slab, under certain steel bar configuration.

Concrete strength profile will then be sampled among the entire slab. Concrete does not have a perfect uniform strength profile among every single fibre. Instead, there will always be a variation in the concrete strength due to the random nature of a concrete mixture. Common concrete mixture for pavement design has a normally distributed concrete strength of 4.5MPa with a 4.3% of variation (Choi, Ha, and M. C. Won 2015). Therefore, this nature will be investigated in this research to produce probabilistic crack patterns that are likely to occur for a CRCP under certain conditions.

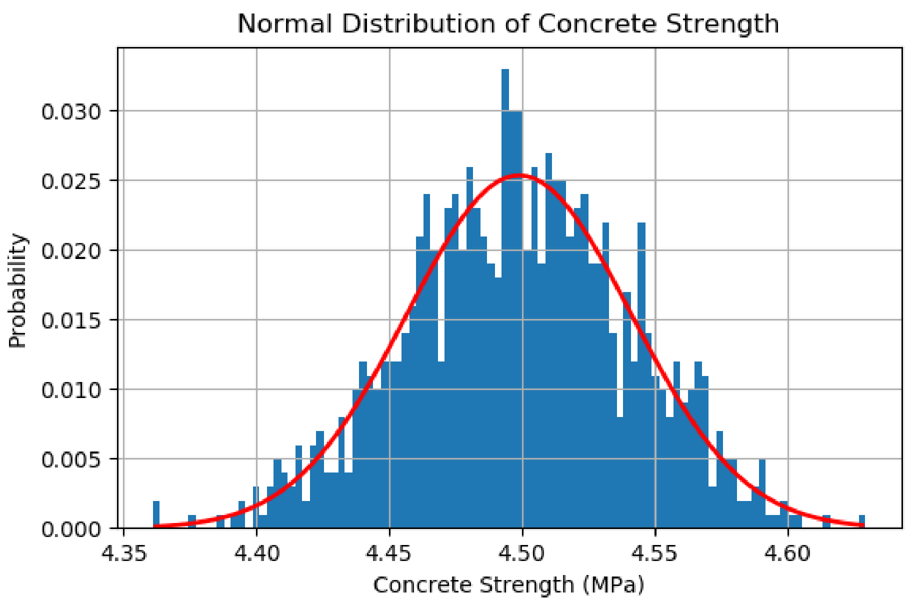


Figure 3.6: Normally distributed concrete strength

Monte Carlo Simulation will be used as a type of sampling distribution method, which is of common in pavement design field. Random sampling will be drawn from a normally distributed curve to formulate a non-uniform continuum of CRCP, to simulate the likelihood of path of crack formation. An example of concrete strength drawn from such distribution is shown in Figure 3.6. It was drawn from 1000 sampling, and this method will be used to formulate the probabilistic crack pattern under such distribution of concrete strength. Sample will be draw 1000 times along width of concrete and repeated along each depth of the concrete slab. This process enables one to informatively visualise where cracks initiate and in which direction will it propagate.

3.3.3 Stage 2 – Crack Width Analysis

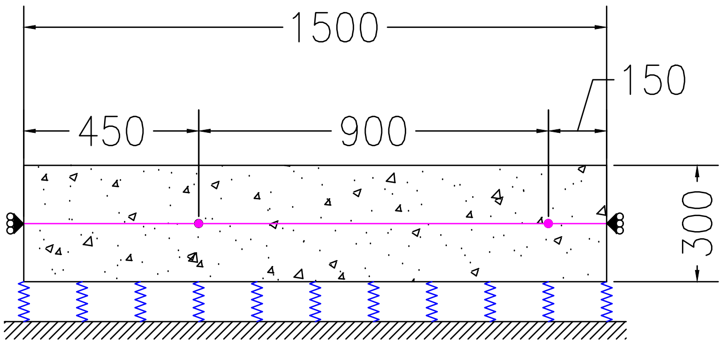


Figure 3.7: Normal Distribution of Concrete Strength

The numerical model used in crack width analysis is identical to the model for crack initiation in Section 3.3.1. Modelling details should refer back to previous sections. The main differences in this analysis are the change in elastic modulus due to tension stiffening and the dimension of the pavement. The width in this analysis will have a shorter width which reassembles the crack spacing of CRCP. Boundary condition will also differ as the concrete slab is now free to expand or contract, to simulate the deformation of a cracked CRCP section. Figure 3.7 shows the dimension of the model. The key differences in this analysis stage is that crack width will be simulated, and the displacement along the vertical path of the two ends of the cracked section will be monitored and investigated.

3.3.4 Stage 3 – Traffic Load Performance

The final stage of analysis is also to investigate the long-term performance of a cracked CRCP section, but on the performance of vertical load transfer after cracks. The same dimension as previous stages are used as shown in Figure 3.8, but in a three-dimensional scale. A 3D simulation was used instead of a 2D model, due to the limitation of plane stress analysis to something like traffic load that is not uniform in nature. A depth of 3200 mm was imposed on the cracked CRCP section because 3200 mm is the width

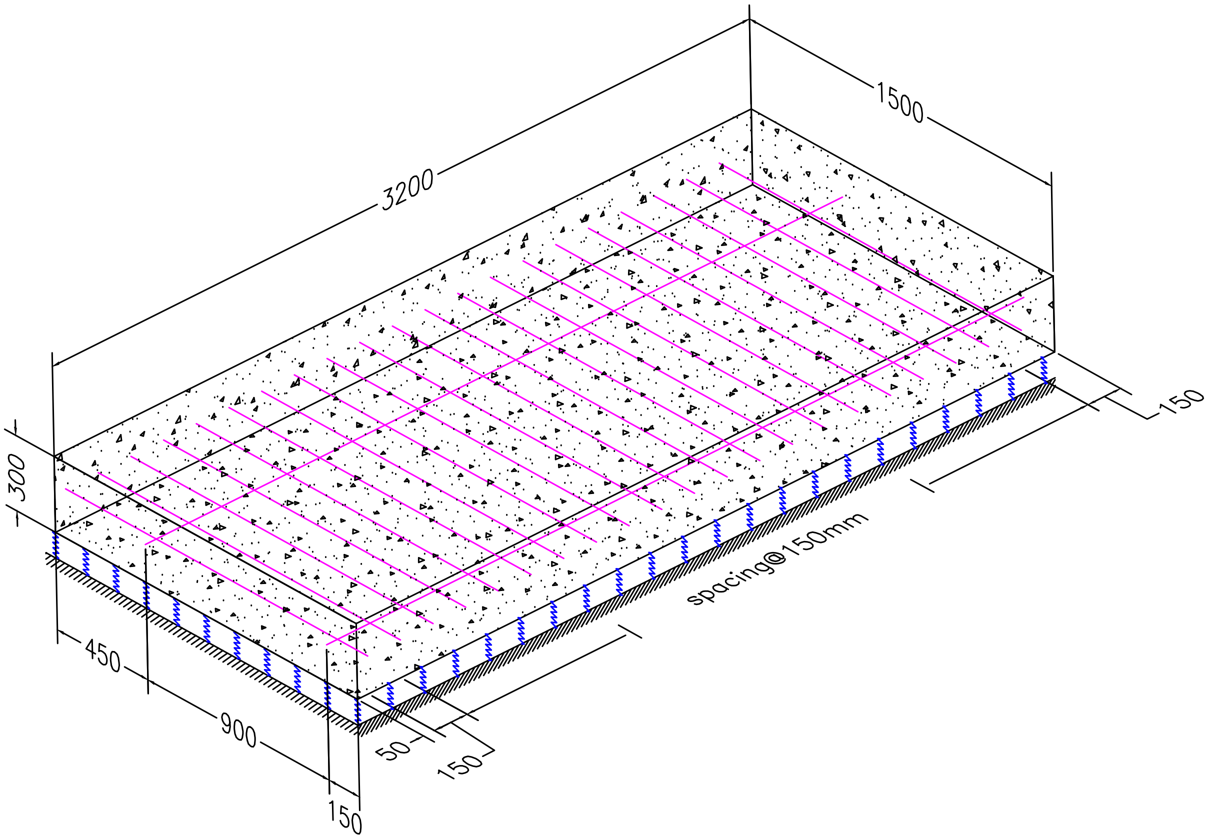


Figure 3.8: Three-dimensional cracked section of CRCP dimensions

of a standard lane according to AS5100. The longitudinal steel bars have a spacing of 150 mm, and traffic loads will be imposed on top of the surface of concrete along the longitudinal direction. An explicit dynamic simulation was used in Abaqus to model the velocity of vehicles. A speed of 80 km/h was used to simulate the average vehicle velocity on a highway. The key objective is to investigate the deflection result from vertical load transfer, to investigate the ability for a cracked section to transfer loading to adjacent cracked section. Deflection of the concrete slab will be monitored, and the maximum amount of deflection will be drawn after vehicles had passed the cracked section.

Chapter 4

Results and Discussion

4.1 Crack Initiation

The mechanism of crack initiation had long been an area of interest for many researchers within the field of CRCP enhancement. Crucial information can be obtained through understanding the mechanism, such as crack location and crack width prediction. Failure prediction model enables the properties of crack to be anticipated before the pavement is laid, which in turns allows for a better design of pavement configuration regarding design traffic load for that particular local area (Suh, Hankins, and McCullough 1992).

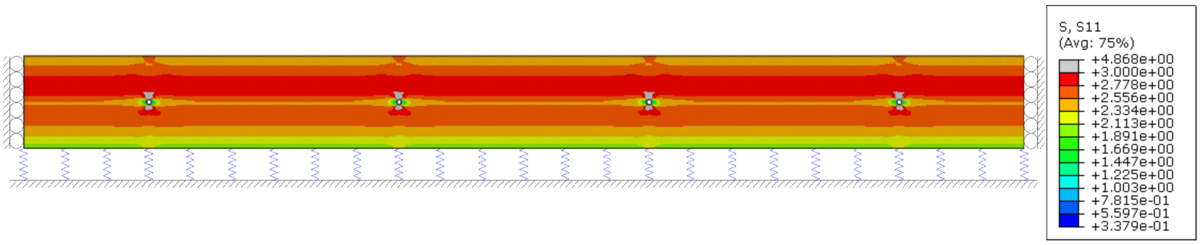


Figure 4.1: Stress profile from thermal loads and autogenous shrinkage before cracks

Stress profile for simulating early-age crack from stage 1 analysis is shown in Figure 4.2. The four spikes are from the locations around the transverse steel bar. It is clear that the location of transverse steel bar largely determines stress along the depth of concrete. The sudden drop in stress following with an immediate increase in stress is the combined effect of longitudinal steel bar (at depth 150mm). The effect is illustrated from the ABAQUS output in Figure 4.1, where the concentrated stress is on top and bottom of transverse steel bars. Whereas the left and right side of transverse steel bars experience a drop in stress due to the longitudinal steel bar restrains and the tension effect that is in-line with transverse steel bar location.

Illustrated in Figure 4.2 is the severe stress that concrete experience is at along the

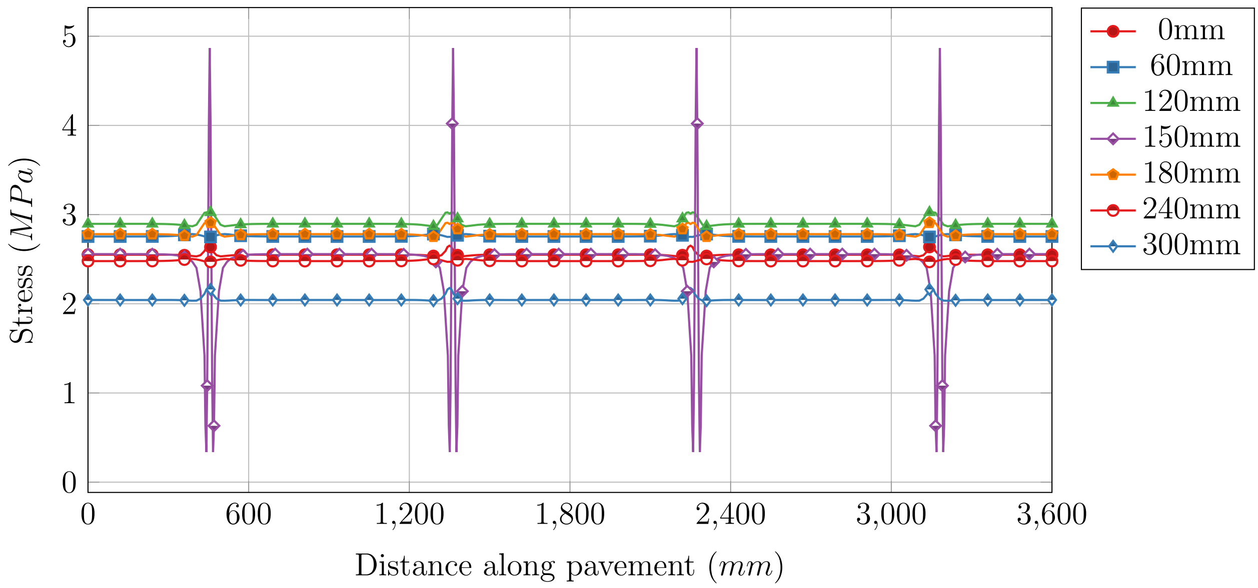


Figure 4.2: Stress profile from thermal laod and autogeneous shrinkage, along depth of concrete

depth of reinforcement steel bar (150mm). Studies also suggested crack could often propagate from the surface of the concrete pavement. Hence, the stress along the top surface (0mm) will also be investigated in this research. Parametric studies will be performed mainly via comparing the stress curve along those two critical paths. Since the stress profile is symmetric at the location of each transverse steel bars, only the stress surrounding transverse steel bar will be shown, to focus and to zoom this parametric study more closely to the stress spikes.

4.1.1 Parametric Study for Crack Initiation

Effects of varying temperature profile, diameter of transverse steel bar, location of steel bar layers, and the use of double reinforcement layers are investigated and shown in Figure 4.3, 4.4, 4.5, and 4.6 respectively. Stress along $x = 0$ mm and $x = 900$ mm are shown in each plot (symmetric portion along slab width), together with the left plot being the stress at the concrete surface and the right plot being stress along steel bar depth.

Temperature profile in Figure 4.3 are extracted from experimental data as discussed in Section 2.1.2. Stress profile along the depth of CRCP is highly sensitive to the temperature profile applied. The temperature profile is taken at different time of the day, and CRCP are subjected to a changing thermal load as a daily cycle. Therefore, for the designing purpose, the CRCP must be able to withstand the most severe temperature profile as its design parameters. Depending on the temperature profile the concrete could be subjected to a concave up stress or concave down stress, along with the different degree of stress developed in the slab. The bend-upward stress and bend-downward stress are developed due to temperature profile being positive gradient and negative gradient respectively. The two most severe case is extracted and shown in Figure 3.1.

Furthermore, out of the two most distinct profile, positive gradient temperature is the more critical case as suggested by the highest stress in both concrete surface and path along steel bar depth in Figure 4.3. The bend-downward stress has a higher difficulty to develop due to sub-base constrains underneath the concrete slab. Whereas for bend-upward stress the concrete slab is free to contract upward and hence it induces the highest stress out of all temperature profile. Therefore, temperature profile 07:30 will be used as the base case for all other parametric studies.

When comparing the effects of other parameters, it does not appear to have a significant impact on the stress developed along the concrete slab. The stress along steel bar depth almost always appears identical to each other despite there is a change in steel bar location in Figure, transverse steel bar diameter or the use of double reinforcements. The main differences are the stress concentration surrounding the transverse steel bar. The transverse steel bar is at the location of $x = 450$ mm, and intuitively as the steel bar location move closer to the concrete surface the stress concentration on the surface increase. The increase in diameter also increases the concentration on the concrete surface in Figure 4.4, despite the depth of steel bar remaining the same. This suggested the effect of transverse steel bar will propagate along the height of the slab as the diameter increases. Further investigation will be performed on using these stress data to produce a probabilistic model of a crack profile.

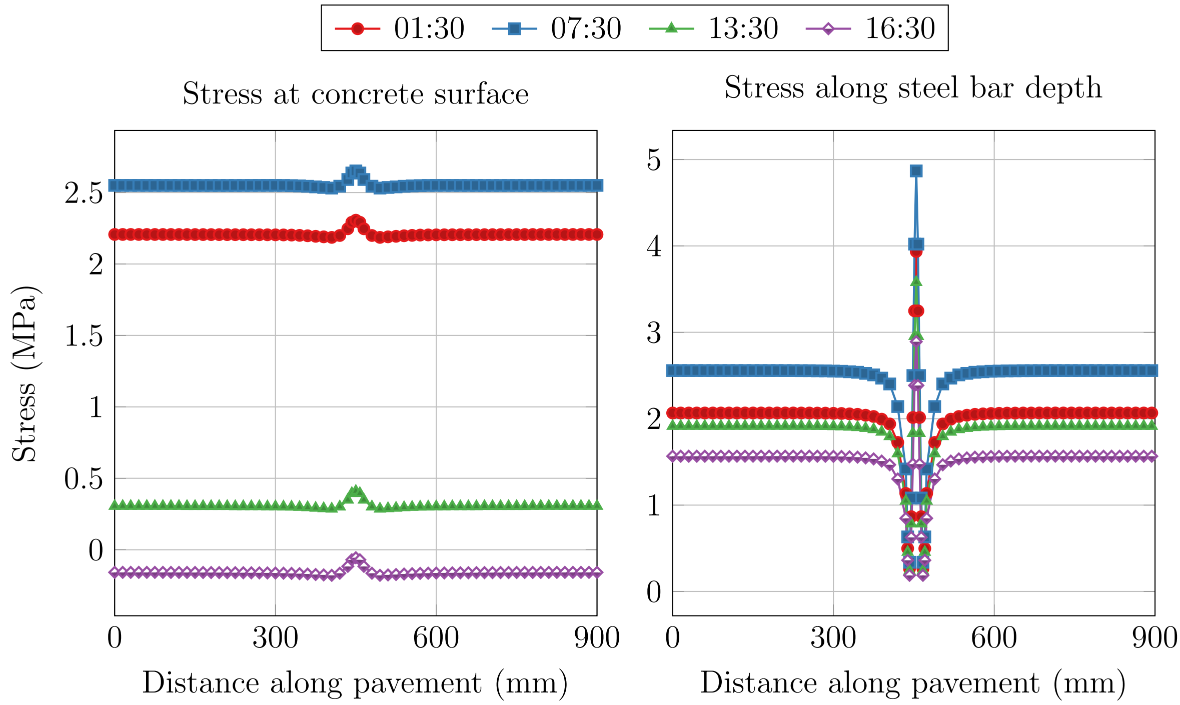


Figure 4.3: Crack Initiation Analysis – Temperature profile effect

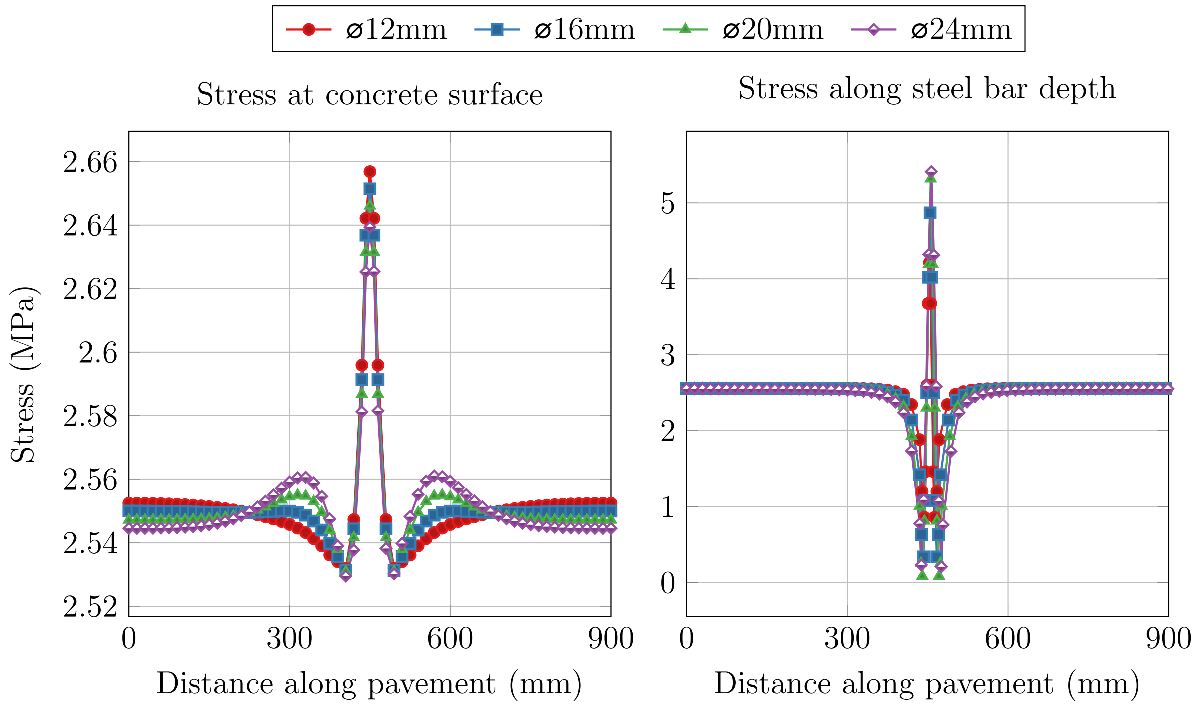


Figure 4.4: Crack Initiation Analysis – Transverse steel bar diameter effect

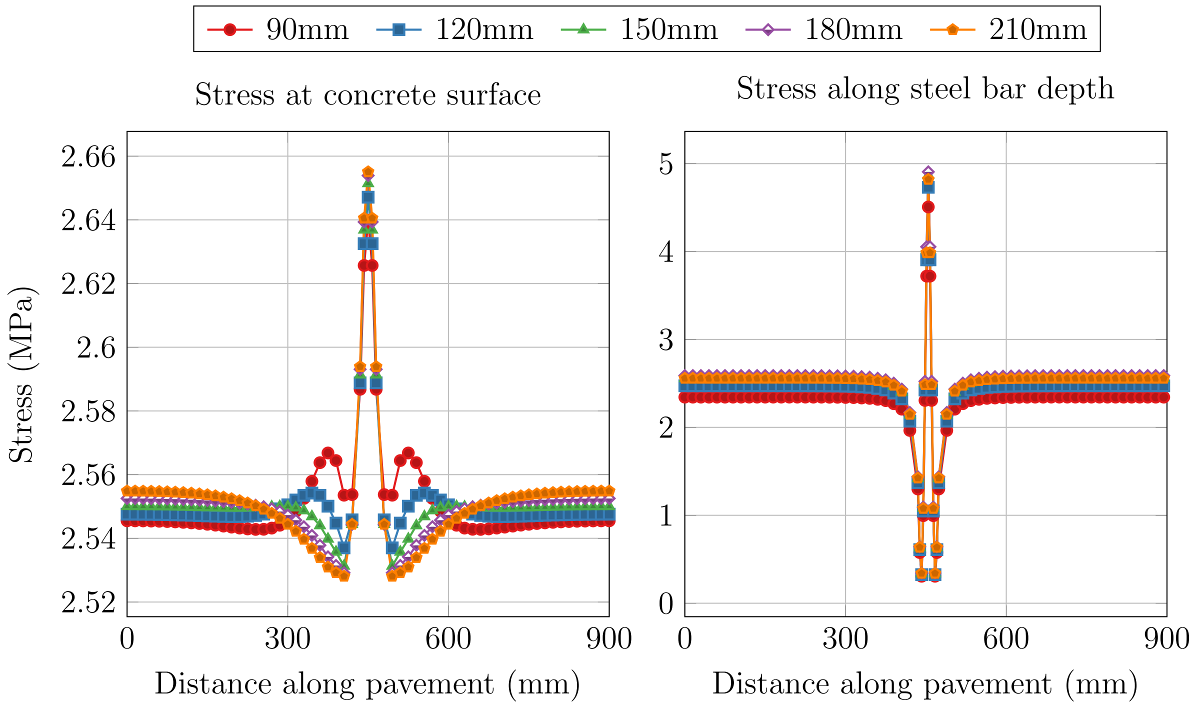


Figure 4.5: Crack Initiation Analysis – Steel bar depth effect

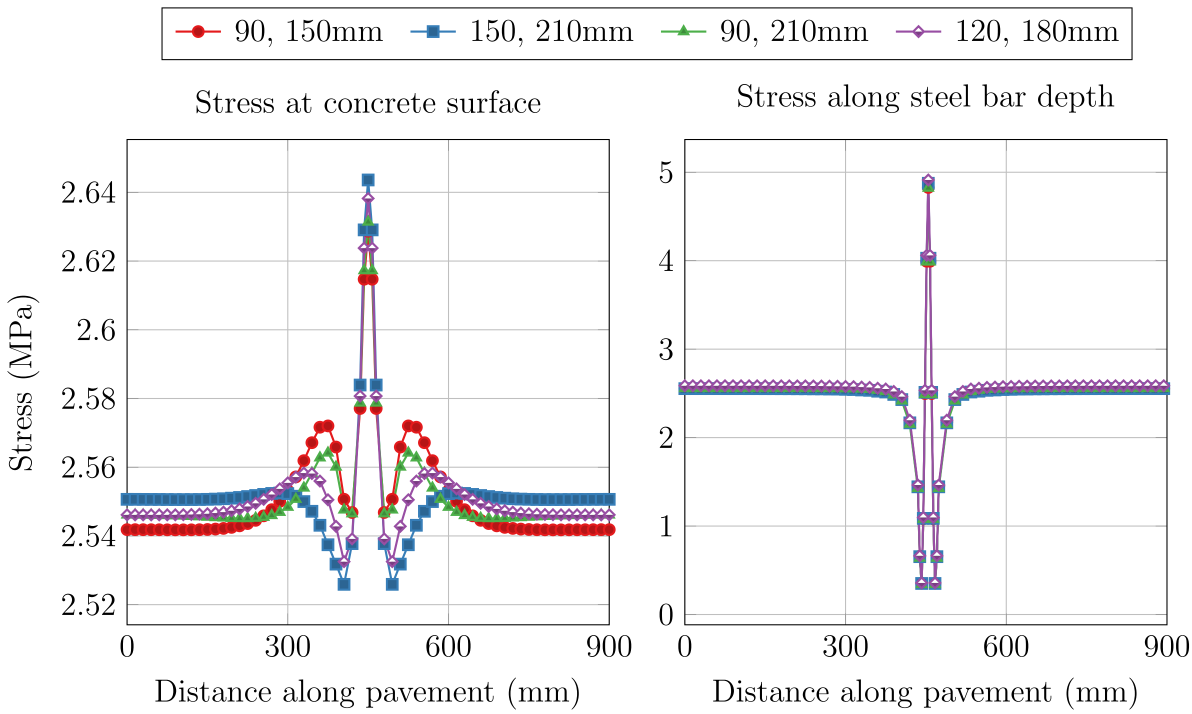


Figure 4.6: Crack Initiation Analysis – Double reinforcement layers

4.1.2 Probabilistic Model of Crack Formation

Probabilistic models of crack formation are performed using a normally distributed concrete strength of 4.5MPa with a 4.3% of a variation. The concrete stress obtained in Section 4.1.1 are used for this analysis. The details of the random sampling are discussed in Section 3.3.2. For each depth along the concrete slab, 1000 sampling was performed to obtain a crack probability. The sampling procedure is repeated for the entire depth of concrete. The result is then visualised using *Kernal Density Estimation* (KDE) as a non-parametric way to estimate the probability density function of the crack pattern in a continuous domain for the entire concrete slab (Terrell and Scott 1992). The crack pattern for the crack initiation can then be drawn from the finite data sample. The sampling procedure was needed as the result output from Abaqus are discretised in a finite amount of elements, and a direct visualise would result in a discontinuous domain pattern. The results are shown in Figure 4.7 and 4.8 for using different transverse steel bar diameter and depth of double reinforcement respectively.

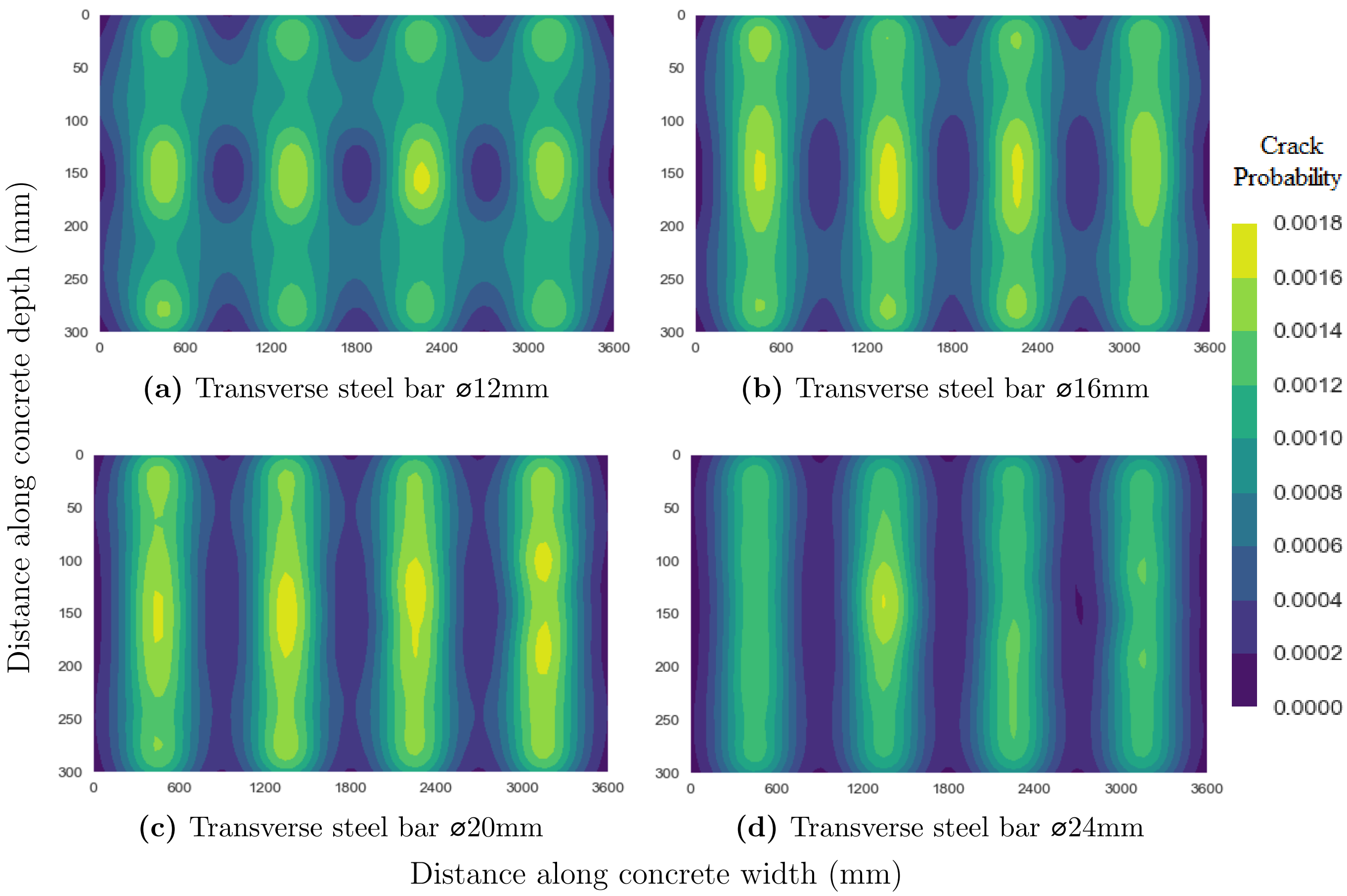
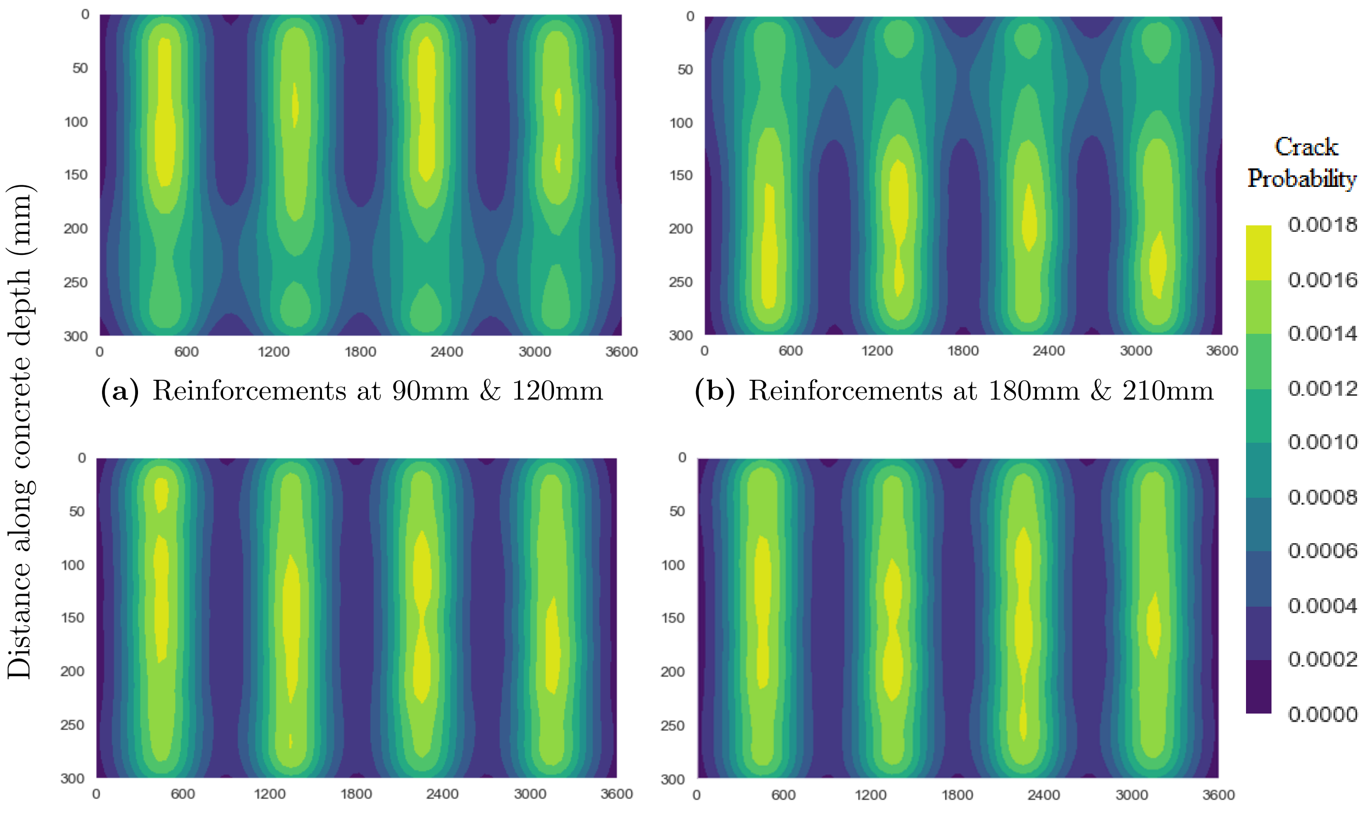


Figure 4.7: Probabilistic Crack Model – transverse steel bar diameter effect

The colour bar of crack probability for the KDE plot shall be taken as the probability of crack occurring at the given depth at that particular location. The darker the location, the less likely it is for a crack to occur at that given location. The x and y axis represent the width and depth of the concrete slab respectively which are same as before. Therefore, the colour map provides a probabilistic estimation of where the crack occurs and where it will likely to propagate. For example, Figure 4.7a represents a crack pattern for CRCP that uses a 12mm diameter of transverse steel bar. It is clear that there is four highly probable location of a concentrated area for a crack to occur along depth 150mm (those four are the location of transverse steel bars). It is intuitive as demonstrated before that stresses are concentrated around transverse steel bar location. However, as the depth of concrete move upward or downward, the probable location of crack formations starts to dilute to the surrounding area. At a depth of 50mm, there is a much higher probability for a crack to propagate side-way instead of directly on top of transverse steel bar, which would explain the reason that even though concrete stress is the highest along the depth of transverse steel bar location, only $60-70\%$ of cracks occur at that given location (Choi, Ha, and M. C. Won 2015). It is because of the non-homogeneous and slight deviation of concrete strength among the volume of the structure, where a crack would naturally propagate to the weaker part of a structure if concrete strength is not perfectly uniform.

Therefore, information that can be drawn from the KDE colour map is how likely it is for the crack initiation to initiate, as well as what path it will likely to propagate along. Al-Qadi and Elseifi (2006) has suggested the crucial role that transverse steel bar plays in crack formation. The claim is confirmed by the effect as shown in Figure 4.7. By comparing Figure 4.7a and Figure 4.7d it is clear the increase of diameter had uniformly concentrated the likelihood of crack formation at the same approximate of transverse steel bar located along the entire depth. In Figure 4.7a and 4.7b there is an apparent likelihood for cracks to form around the transverse steel bar but quickly fade away as cracks propagate upward or downward. However, in Figure 4.7c and 4.7d the likelihood for a crack to propagate is transforming into a direct vertical path and unlikely to diverse in the horizontal direction. This suggested the increase in transverse steel bar is sensitive to crack propagation, and the larger the diameter, the more likely it is for cracks to propagate vertically upward from the vicinity of transverse steel bar. If the transverse crack of CRCP is to be preferable to control (or avoid) to on top of the location of transverse steel bar, one could alter the diameter of transverse diameter to achieve this effect.

Compare to the interesting information that can be drawn from Figure 4.7, Figure 4.8 with double reinforcement layers provides information that is more or less expected. KDE plot in Figure 4.8a and 4.8b demonstrates the effect of having double reinforcement on the upper half and the lower half of the concrete slab respectively. The result confirms the hypothesis that there is a higher chance for crack formation at a location around transverse reinforcement, and this effect quickly fades away as the depth move further away. Figure 4.8c and 4.8d demonstrates for a concrete slab to have double reinforcement that are equal distance to the centerline. The effect is similar to increasing transverse diameter in Figure 4.7c and 4.7d where it will concentrate the crack likelihood into a vertical path along the depth of the transverse steel bar location. Altering steel bar diameter might be another method for the transport department to control and enforce transverse crack formation at a certain location along pavement beside altering diameter.



(a) Reinforcements at 90mm & 120mm

(b) Reinforcements at 180mm & 210mm

(c) Reinforcements at 120mm & 180mm

(d) Reinforcements at 90mm & 210mm

Distance along concrete width (mm)

Figure 4.8: Probabilistic Crack Model – reinforcements layers depth

In light of such a visual realisation of how cracks propagate along the depth of the slab, a further investigation was made to simulate and visualise the effect of a staggered pattern of double reinforcement layers. The two layer of reinforcement steel bars is at depth 90mm and 210mm with an equal distance from the centerline. The configuration and output from Abaqus are shown in Figure 4.9. Modelling details are same as before but with one layer of reinforcement layers shifted half the spacing. The result of the KDE plot is shown in Figure 4.10. The stress profile and magnitude are similar as before, but an unusual pattern of crack probability had emerged in the KDE plot.

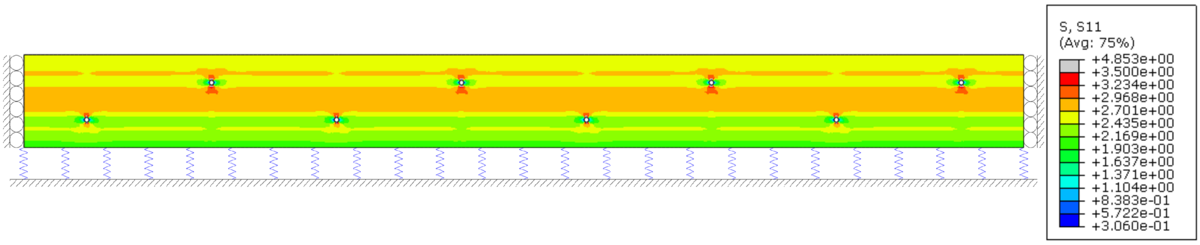


Figure 4.9: Probabilistic Crack Model – Staggered layers model

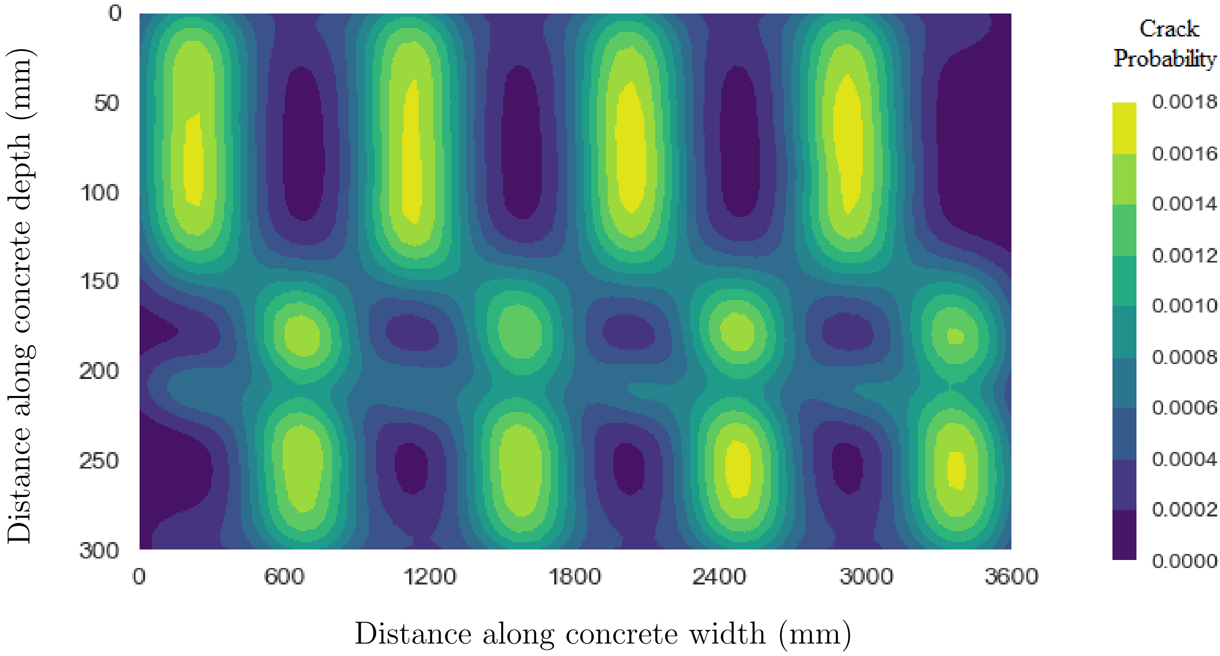


Figure 4.10: Probabilistic Crack Model – Stagger layers pattern effect

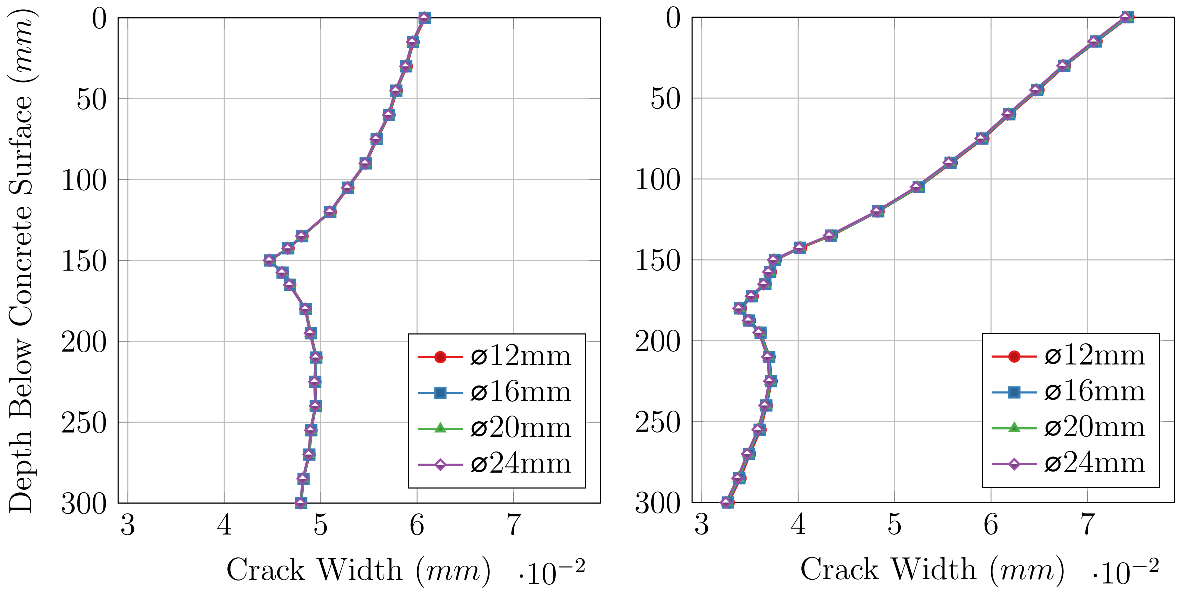
Likelihood of cracks along concrete slab is now in a staggered pattern with a high concentration at the upper half of the concrete slab. The probability of the path that a crack propagates is now not a direct vertical path, but with a horizontal deviation at the middle of the concrete slab. The important implication for this type of crack formation allows a better load transfer from vertical loading such as traffic. As previously discussed in Section 2.3 vertical load transfer for a cracked section are mostly by longitudinal steel bar and aggregate interlock between cracked section. However, if a staggered crack pattern is formed along the depth of concrete slab, there will then be a lateral surface in between each cracked CRCP section that allows for an effective way of vertical load transfer. The staggered pattern can reduce the damage from traffic fatigue due to poor load transfer, and minimise structural punch-out. This is a suggestion for a plausible way to control and utilise the path of a transverse crack formation to enhance load transfer after cracks had been formed for better long-term performance of CRCP.

4.2 Crack Width Analysis

Crack width controlling measure can be performed to minimise the effect such as rainwater regression (Larrard 2005), which will cause erosion within the reinforcing steel bar. They must be limited as much as possible, as erosion could accelerate the traffic induced damages and damage long-term performance. Sensitive test of crack width analysis had been performed with a different configuration to determine which configuration is preferable for minimising crack width. A cracked width of 1500mm was chosen along with an asymmetric dimension to investigate a possible crack configuration that both agrees to probabilistic crack model discussed in Section 4.1.2 and with an asymmetric dimension that produces non-uniform stress profile along the width of CRCP. Crack width for the cracked section produce are symmetrical on both ends but are polar opposites. Figure 4.11, 4.12 and 4.13 shows the parametric study of transverse steel bar diameter, steel bar locations, temperature profile, and double reinforcement effects respectively.

4.2.1 Parametric Study for Crack Width Analysis

Transverse steel bar diameter has been shown to plays a crucial role in crack initiation in Section 4.1.2. However, its effects to crack width development are minimal as shown in Figure 4.11a and 4.11b. Using different diameter of transverse steel bar has no effect on the crack width on the two ends of the cracked CRCP section. It is likely due to its effect only affects concrete slab vertically, and the effects quickly fade away to the two ends of the concrete slab. Note that the diameter shown only represents the change in transverse steel bar, with the diameter of longitudinal steel remaining the same in all cases.



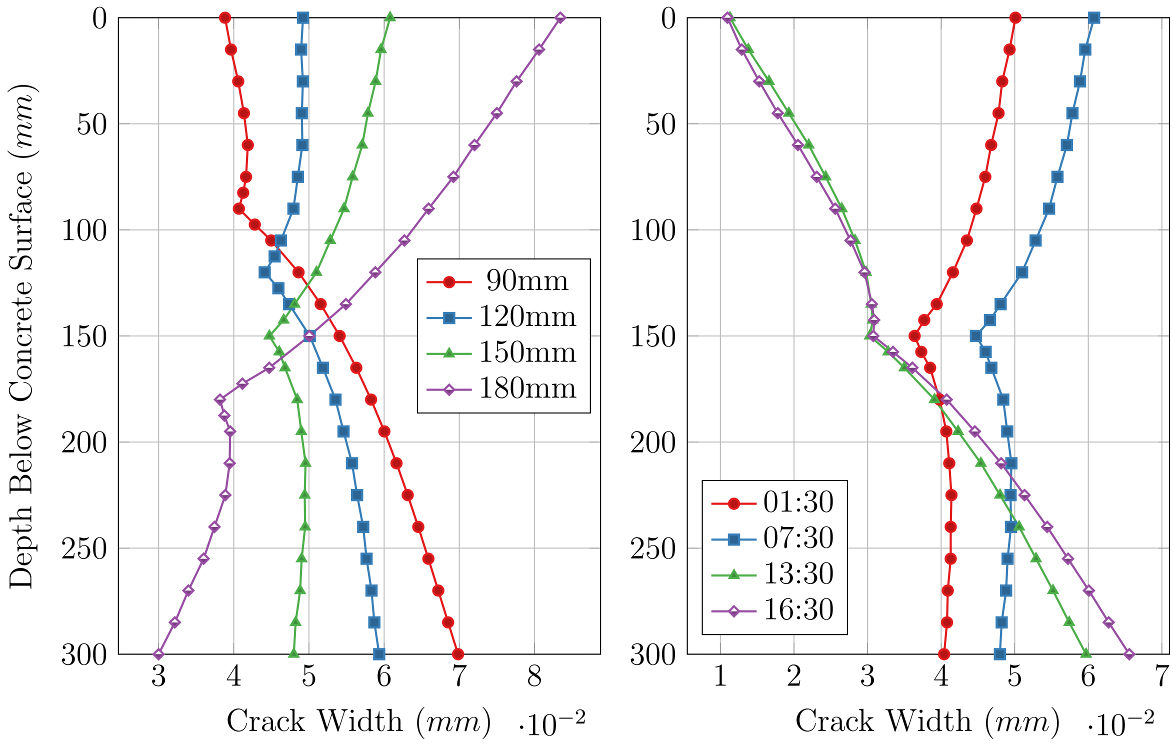
(a) Sinle reinforcement at 150mm **(b)** Double reinforcement at 150mm & 180mm

Figure 4.11: Crack width analysis – Transverse steel bar diameter effect

Similar to crack initiation, temperature profile also plays a vital role in crack width development as demonstrated with the diverging width pattern shown in Figure 4.12b. Two of the temperature profiles form a uniform crack width while the other two form an increasing pattern of crack width from the surface. The shift in crack width implies crack width will change along with the time of the day that concrete was laid, which confirms the studies by (Kohler and Roesler 2006) who concluded the critical role of temperature plays to crack width development.

Altering depth of steel bar does not appear to decrease the maximum amount of crack width, but instead, it changes the location of maximum or minimum crack width occurring along y-axis. The location where longitudinal steel bar crosses would have roughly the minimum amount of crack width. In general, if the purpose is to minimise the amount of rainwater ingression into cracks between each section, it might be preferable to place steel bars closer to the surface of the concrete slab to tighten it.

Using double reinforcement to control crack width appears to be ineffective as well as demonstrated by Figure 4.13. Crack width does indeed decrease as suggested by the



(a) Reinforcement layer location effect

(b) Temperature profile effect

Figure 4.12: Crack width analysis – transverse steel location and temperature profile effect

overall trend along concrete slab depth. However, one extra reinforcement layer does not has a significant effect on tightening the width, but it appears that extra layer only has a localised effect on tightening crack width that is within a close approximate of that layer. Tightening of cracks only affect within around ±50mm depth of the steel bar layers. Increasing longitudinal steel bar diameter as suggested by Suh and McCullough (1994) will likely to be a more efficient method for this purpose.

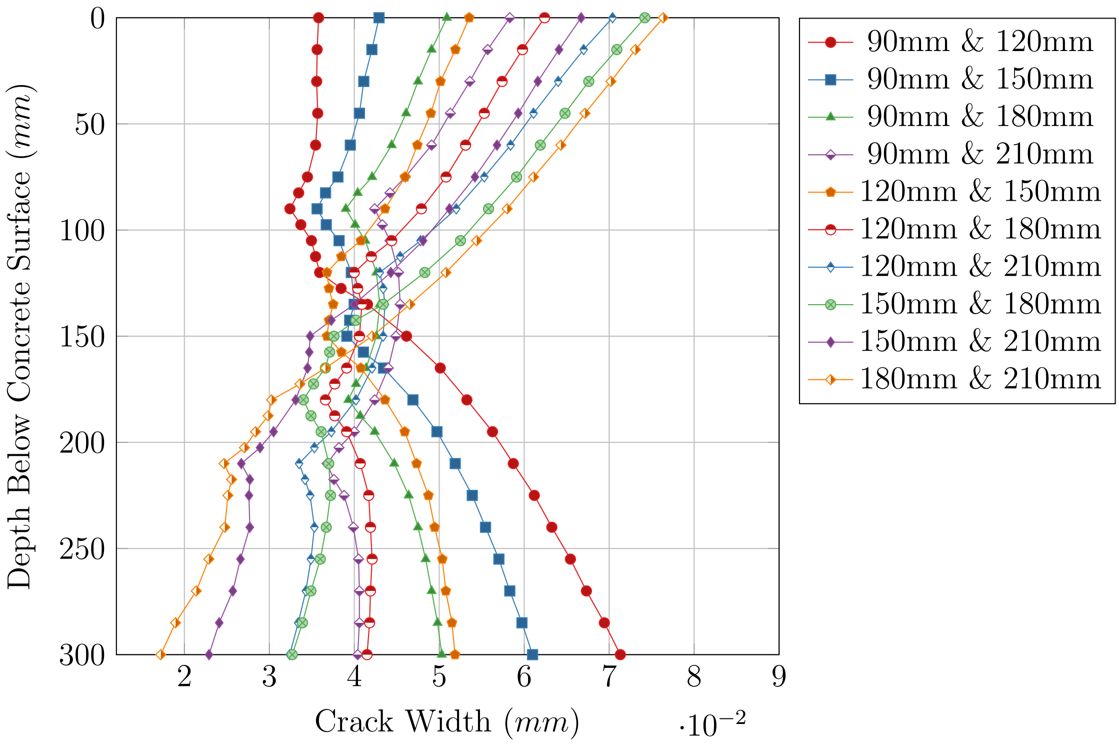


Figure 4.13: Crack width analysis – Transverse steel bar location effect, with double reinforcement layers

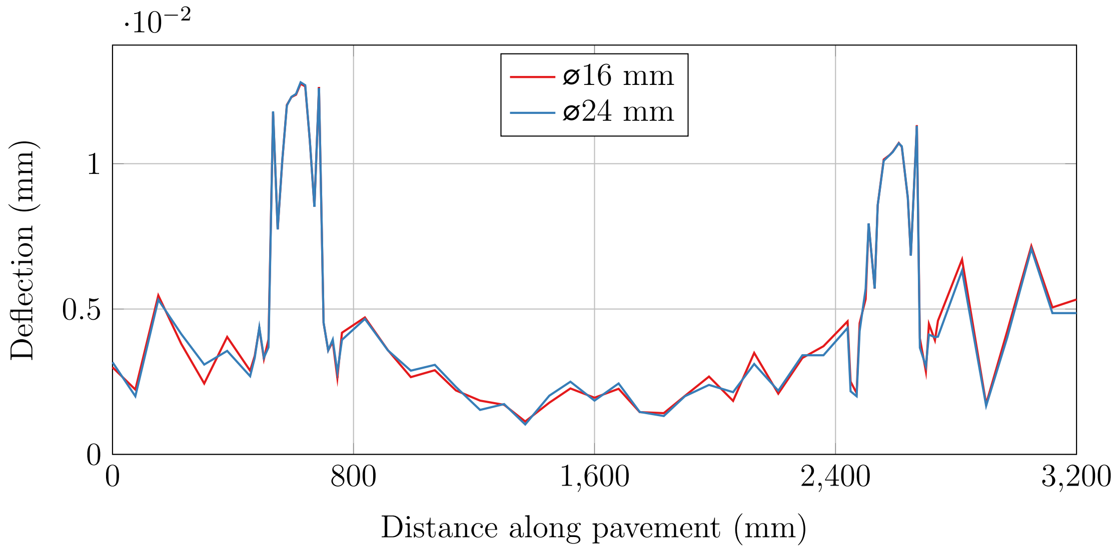
4.3 Traffic Load Performance

Load transfer performance is particularly crucial after CRCP had cracked, regarding its vertical load transfer ability along with the deflection of the slab. A significant amount of deflection within a concrete slab will accelerate concrete damage by traffic fatigue. Performance of a cracked section was performed in a three-dimensional model with dynamic velocity, and the result is shown in Figure 4.15. Sensitivity test on a similar configuration was performed to determine the preferable option to minimise deflection.

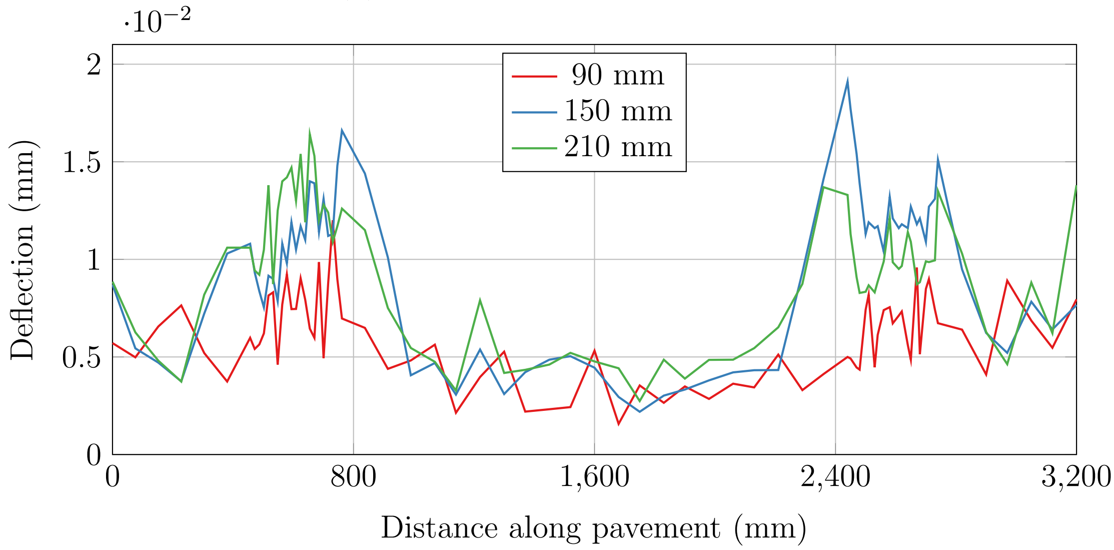
4.3.1 Parametric Study on Load Transfer Performance

Similar to result from crack width analysis, the diameter of transverse steel bar has an indistinguishable effect on affecting deflection of the concrete slab. Changing different diameter in Figure 4.14a has no effect which implies diameter has little effect after the concrete slab has cracked. Therefore, if the purpose is to minimise crack width or deflection, transverse steel bar diameter would be an ineffective solution to them.

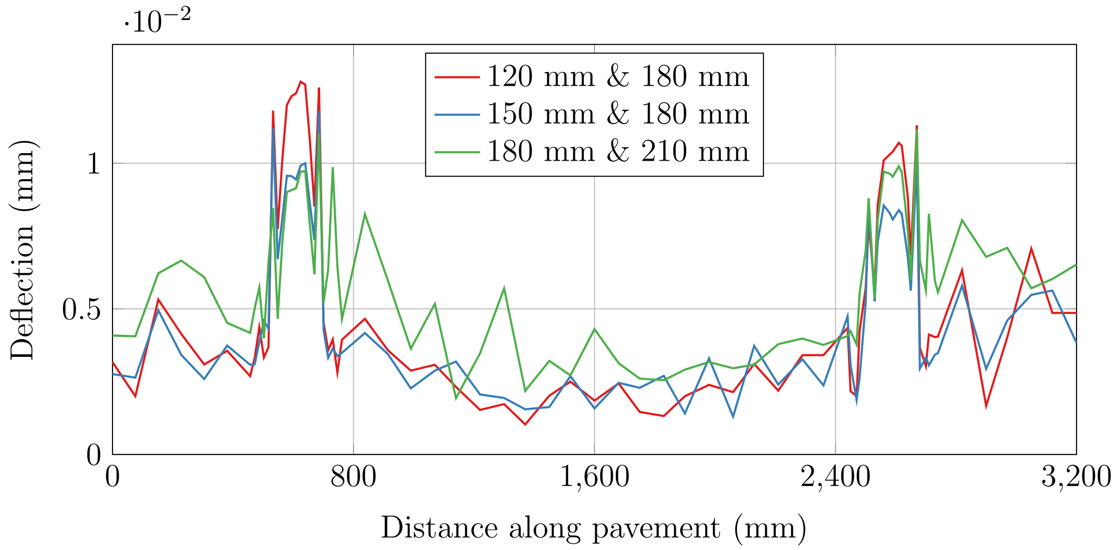
The location of steel bar reinforcement is crucial to slab deflection. The close to the surface the less deflection induce on the concrete slab, as shown in Figure 4.14b. Minimising deflection helps to reduce fatigue stress and implies better load transfer between neighbourhood CRCP section. Although using double reinforcement layers could also minimise concrete slab deflection in Figure 4.14c, the effect is not as significant as steel bar location. Traffic load performance is best controlled by placement of steel bar reinforcement and is insensitive to diameter or amount of steel ratio throughout the slab.



(a) Varing transverse steel bar diameter



(b) Varing reinforcement location



(c) Using double reinforcement layers

Figure 4.15: Load Transfer Performance – parametric study

Chapter 5

Conclusion

5.1 Crucial Findings

Crack spacing and width plays a crucial role in determining the long-term performance in the later stage of CRCP. The formation is mostly by the early-age crack which forms within a few weeks after concrete was poured. Crack spacing determines the dimension of cracked CRCP section in the latter stage, whereas crack width determines the vertical load transfer between each cracked CRCP section. Therefore, the prediction and controlling of such cracks are of essential to the designing process of CRCP.

The crack initiation process in a newly poured CRCP is concentrated at the vicinity of transverse steel bar location. The KDE plot as shown in Figures 4.7 and 4.8 demonstrates the high probability of crack formation and propagation near the location of transverse steel bar. That is a result of transverse steel restrained from the vertical contraction of concrete which induce a significant amount of concentrated stress at the top fibre of the concrete stresses near transverse steel. The effect determines the crucial role that transverse steel plays in crack development (Timm, Guzina, and Voller 2003). Crack initiation was also found be mostly dependent on the diameter of transverse steel, which is an interesting parameter to control is a more precise crack control is desirable.

The transverse steel bar was found to have an insignificant effect on controlling or minimising the crack width between each cracked section. The effect of longitudinal steel bar is the determining factor, and transverse steel bar appears to have minimal effect after the crack already been formed. A larger diameter of steel bar does not reduce deflection when subjected to traffic loads. However, the placement of steel bar layers helps to control the crack width in the y-axis direction. Placing steel bar reinforcements closer to surface helps to produce a narrower crack width closer to the concrete surface, which is desirable to minimise rainwater ingression in order to prevent erosion of steel bar. Closer to the surface also helps to minimise the deflection induced by traffic loads, and hence a better load transferring in between each cracked CRCP section.

5.2 Future Work

During the progress of this research, one of the bottlenecks encountered is from the physical performance of computer within computer lab. The initial approach of this study was to use a three-dimensional model to simulate the thermal effect in a 3D scale. However, the amount of time it takes for a numerical model to completes takes in the order of 15-20 hours. This is an enormous amount of time due to the number of spring elements it takes to simulate the bond-slip behaviour and elastic foundation of the subbase friction. The modelling of each stage also take many iterations and causes an enormous amount of time wasted. Although a two-dimensional approach might not be too far from reality, a 3D model might be preferable in the future if a more realistic simulation of CRCP is desire.

Further investigation should also be performed on the probable crack patterns as it enables researchers and pavement designer to better account for the crack pattern before the pavement was laid. It enables one to control better the behaviour of transverse crack patterns such as creating a staggered pattern for better vertical load transfer, as suggested in Section 4.1.2. Future work should be focused on more closely on the sensitivity of different configuration of CRCP for a better crack control mechanism.

References

Choi, Seongcheol, Soojun Ha, and Moon C Won (2015). "Mechanism of Transverse Crack Development in Continuously Reinforced Concrete Pavement at Early Ages". In: *Transportation Research Record: Journal of the Transportation Research Board* 2524, pp. 42–58.

Darestani, Mostafa et al. (2006). "Experimental study on structural response of rigid pavements under moving truck load." In:

Gerber, Johan Andries Kritzinger (2011). "Characterization of cracks on ultra thin continuously reinforced concrete pavements". PhD thesis. University of Stellenbosch.

Ju, Suoji (2010). "General Analytic Solution for Thermal Stress of CRCP Based on Actual Boundary Condition". In: *Measuring Technology and Mechatronics Automation (ICMTMA), 2010 International Conference on*. Vol. 2. IEEE, pp. 343–346.

Kim, Seong-Min and Jeong-Hee Nam (2010). "Measurements and experimental analysis of temperature variations in Portland cement concrete pavement systems". In: *Road Materials and Pavement Design* 11.3, pp. 745–771.

Kim, Seong-Min, Moon C Won, and B Frank McCullough (2000). *Three-dimensional nonlinear finite element analysis of continuously reinforced concrete pavements*. Tech. rep. Citeseer.

Kim, Seong-Min, Mooncheol Won, and B Frank McCullough (2001). *CRCP-9 computer program user's guide*. Tech. rep. Center for Transportation Research, Bureau of Engineering Research, The University of Texas at Austin.

Kohler, Erwin and Jeffery Roesler (2006). "Crack spacing and crack width investigation from experimental CRCP sections". In: *International Journal of Pavement Engineering* 7.4, pp. 331–340.

Larrard, François de (2005). "High-Performance Concrete Carpet: A Hydraulic Flexible Wearing Course: Part I—Design". In: *Road materials and pavement design* 6.4, pp. 533–547.

Lee, Seung-Woo et al. (2014). “Comparison of performance of overlay pavements constructed on deteriorated concrete pavements using LTPP database”. In: *KSCE Journal of Civil Engineering* 18.5, pp. 1380–1388.

McCullough, B Frank, CS Noble, and JCM Ma (1979). *Limiting Criteria for the Design of CRCP*. Tech. rep. Center for Highway Research, University of Texas at Austin.

Nam, Jeong-Hee (2005). “Early-age behavior of CRCP and its implications for long-term performance”. PhD thesis.

Nishizawa, Tatsuo et al. (1998). “Study on thermal stresses in continuously reinforced concrete pavement”. In: *Transportation Research Record: Journal of the Transportation Research Board* 1629, pp. 99–107.

Plei, Michael and Shiraz Tayabji (2012). *Continuously Reinforced Concrete Pavement Performance and Best Practices*. Tech. rep.

Priestley, MJN et al. (1972). “Thermal gradients in bridges-some design considerations”. In: *New Zealand Engineering* 27.7, p. 228.

Al-Qadi, IL and MA Elseifi (2006). “Mechanism and modeling of transverse cracking development in continuously reinforced concrete pavement”. In: *International Journal of Pavement Engineering* 7.4, pp. 341–349.

Ryu, Sung Woo et al. (2013). “Continuously reinforced bonded concrete overlay of distressed jointed plain concrete pavements”. In: *Construction and Building Materials* 40, pp. 1110–1117.

Sato, Ryoichi, Yoshitaka Hachiya, and Atsuhito Kawakami (1989). *Development of New Design Method for Control of Cracking in Continuously Reinforced Concrete Pavement. Proceedings, 4th International Conference on Concrete Pavement Design and Rehabilitation, Purdue University, April 18-20, 1989*. Tech. rep.

Standard, Australian (2004). “AS5100. 2-2004, Bridge design—Part 2: design loads”. In: *Standards Australia, Sydney (Australia).*

Suh, Young-Chan, Kenneth D Hankins, and B Frank McCullough (1992). *Early-age behavior of continuously reinforced concrete pavement and calibration of the failure prediction model in the CRCP-7 program*. Tech. rep. Center for Transportation Research, Bureau of Engineering Research, University of Texas at Austin.

Suh, Young-Chan and B Frank McCullough (1994). “Factors affecting crack width of continuously reinforced concrete pavement”. In: *Transportation Research Record* 1449.

El-Tayeb, Essam H et al. (2017). "Thermal analysis of reinforced concrete beams and frames". In: *HBRC Journal* 13.1, pp. 8–24.

Teodoru, Iancu-Bogdan (2009). "EBBEF2p-A Computer Code for Analyzing Beams on Elastic Foundations". In: 6, pp. 28–44.

Terrell, George R and David W Scott (1992). "Variable kernel density estimation". In: *The Annals of Statistics*, pp. 1236–1265.

Timm, D.H., B.B. Guzina, and V.R. Voller (2003). "Prediction of thermal crack spacing". In: *International Journal of Solids and Structures* 40.1, pp. 125–142. ISSN: 0020-7683. DOI: https://doi.org/10.1016/S0020-7683(02)00496-1. URL: http://www.sciencedirect.com/science/article/pii/S0020768302004961.

Vecchio, FJ, N Agostino, and B Angelakos (1993). "Reinforced concrete slabs subjected to thermal loads". In: *Canadian Journal of Civil Engineering* 20.5, pp. 741–753.

Vecchio, Frank J (1987). "Nonlinear analysis of reinforced concrete frames subjected to thermal and mechanical loads". In: *ACI Structural Journal* 84.6, pp. 492–501.

Witczak, MW, D Andrei, and WN Houston (2004). "Guide for mechanistic-empirical design of new and rehabilitated pavement structures". In: *Transportation Research Board of the National Research Council*, pp. 1–91.

Yeon, Jung Heum (2015). "Implications of zero-stress temperature for the long-term behavior and performance of continuously reinforced concrete pavement". In: *Construction and Building Materials* 91.Supplement C, pp. 94–101. ISSN: 0950-0618. DOI: https://doi.org/10.1016/j.conbuildmat.2015.05.043. URL: http://www.sciencedirect.com/science/article/pii/S0950061815005401.

Yeon, Jung Heum, Seongcheol Choi, and Moon C Won (2013). "Evaluation of zero-stress temperature prediction model for Portland cement concrete pavements". In: *Construction and Building Materials* 40, pp. 492–500.

Yoder, Eldon Joseph and Matthew W Witczak (1975). *Principles of pavement design.* John Wiley & Sons.

Appendices

Appendix A

Abaqus Python Script for Two-Dimensional CRCP

Listing A.1: Python script to produce 2D CRCP model in Abaqus Scripting Interface

```python

#############################################################
##                          MODEL SETTINGS                       ##
#############################################################

PARTITION_SIZE_MODIFER = 0.5

model_name = '2D_CRCP'
model_width = 3600
model_height = 300
rebar_location = model_height/2

partition_size = 30 * PARTITION_SIZE_MODIFER
mesh_size_global = partition_size
# mesh related to nearby trsbar
mesh_size_trsbar = mesh_size_global * 0.3
mesh_size_trsbar_nearby_diagonals = mesh_size_trsbar * 1.4
mesh_size_trsbar_nearby_outer_edges = mesh_size_trsbar * 1.8

rebar_heights = [model_height * 1/2]

losbar_diameter = 20
trsbar_diameter = 16

trsbar_spacing = 900
#########################################
## TEMPERATURE
TEMP_INITIAL = '-2e-4*pow(Y,2) + 6.44e-2*pow(Y,1) + 37.72'

temp_0730 = '1.65e-9*pow(Y,4)-1.27e-6*pow(Y,3)+2.92e-4*pow(Y,2)-9.46e-3*pow(Y,1)+29.5'
temp_1630 = '1.86e-9*pow(Y,4)-1.51e-6*pow(Y,3)+4.63e-4*pow(Y,2)-9.36e-2*pow(Y,1)+44.1'

conc_elastic_modulus_before_crack = 14800
conc_elastic_modulus_after_crack  = 23000
CONC_ELASTIC_MODULUS = conc_elastic_modulus_before_crack

shrinkage_as_temp_0730_before_crack = [
    (0,      23.80555556),
```

```
    (15,    26.0718195 ),
    (30,    26.25201872),
    (45,    26.3626095 ),
    (60,    26.53663622),
    (75,    26.755398  ),
    (90,    27.00219872),
    (105,   27.262347  ),
    (120,   27.52315622),
    (135,   27.7739445 ),
    (150,   28.00603472),
    (165,   28.2127545 ),
    (180,   28.38943622),
    (195,   28.533417  ),
    (210,   28.64403872),
    (225,   28.722648  ),
    (240,   28.77259622),
    (255,   28.7992395 ),
    (270,   28.80993872),
    (285,   28.8140595 ),
    (300,   28.82297222)
    ]

shrinkage_as_temp_1630_before_crack = [
    (0    ,   38.40555556),
    (15   ,   39.44739513),
    (30   ,   38.47540882),
    (45   ,   37.50157563),
    (60   ,   36.65471782),
    (75   ,   35.91216753),
    (90   ,   35.25351682),
    (105,   34.66061763),
    (120,   34.11758182),
    (135,   33.61078113),
    (150,   33.12884722),
    (165,   32.66267163),
    (180,   32.20540582),
    (195,   31.75246113),
    (210,   31.30150882),
    (225,   30.85248003),
    (240,   30.40756582),
    (255,   29.97121713),
    (270,   29.55014482),
    (285,   29.15331963),
    (300,   28.79197222)
    ]

shrinkage_as_temp_0730_after_crack = [
    (0    ,   16.02888889),
    (15   ,   21.99973617),
    (30   ,   23.78993539),
    (45   ,   24.47052617),
```

```
    (60   ,   24.64455289),
    (75   ,   24.86331467),
    (90   ,   25.11011539),
    (105, 25.37026367),
    (120, 25.63107289),
    (135, 25.88186117),
    (150, 26.11395139),
    (165, 26.32067117),
    (180, 26.49735289),
    (195, 26.64133367),
    (210, 26.75195539),
    (225, 26.83056467),
    (240, 26.88051289),
    (255, 26.90715617),
    (270, 26.91785539),
    (285, 26.92197617),
    (300, 26.93088889)
    ]

shrinkage_as_temp_1630_after_crack = [
    (0  ,   30.62888889),
    (15 ,   35.3753118),
    (30 ,   36.01332549),
    (45 ,   35.6094923),
    (60 ,   34.76263449),
    (75 ,   34.0200842),
    (90 ,   33.36143349),
    (105, 32.7685343),
    (120, 32.22549849),
    (135, 31.7186978),
    (150, 31.23676389),
    (165, 30.7705883),
    (180, 30.31332249),
    (195, 29.8603778),
    (210, 29.40942549),
    (225, 28.9603967),
    (240, 28.51548249),
    (255, 28.0791338),
    (270, 27.65806149),
    (285, 27.2612363),
    (300, 26.89988889),
    ]

TEMP_ANALYSIS_STAGE = temp_0730
SHRINKAGE_ANALYSIS_STAGE = shrinkage_as_temp_0730_before_crack

if model_name in mdb.models.keys():
    del mdb.models[model_name]
mdb.Model(modelType=STANDARD_EXPLICIT, name=model_name)

mdl = mdb.models[model_name]
```

```python

##################################################
from part import *
from material import *
from section import *
from assembly import *
from step import *
from interaction import *
from load import *
from mesh import *
from optimization import *
from job import *
from sketch import *
from visualization import *
from connectorBehavior import *
############################################################
##                      HELPER METHODS                    ##
############################################################
import itertools
FLOAT_TOLERANCE = 0.001
def eql(a, b):
    # A comparasion method with tolerance
    return abs(a - b) <= FLOAT_TOLERANCE

def edgeToArray(edge):
    # Convert an edge to edge array
    return mdl.rootAssembly.edges[edge.index:edge.index+1]

def edgeNameToEdge(edgeName):
    # given the name of an edge, return an edge repo containing
        that edge
    ## EXTREMELY SLOW but a work around for not being able to
        construct repo by demand
    for i in mdl.rootAssembly.edges:
        if i.featureName == edgeName:
            return i

def edgeNameToEdgeArrayFilter(edgeNames, filter):
    # given a list of edge names, it will return a list of edge
        array with a filter
    # It will return a tuple of two list, the first list if when
        the optional filter return true
    _a = []
    _b = []
    for i in [edgeNameToEdge(j.name) for j in edgeNames]:
        if i is None:
            continue
        if filter(i):
            _a.append(edgeToArray(i))
        else:
            _b.append(edgeToArray(i))
```

```
    return _a, _b

def conArray(array):
    # given a list of array, return a concentrated array
    _tmp = array[0]
    for i in range(1, len(array)):
        _tmp += array[i]
    return _tmp

def array_append(array, new_item):
    if new_item is None:
        return array
    if array is None:
        return new_item
    else:
        return array + new_item

def model_sbar_location_generator(dimension, spacing):
    # given the sbar spacing and the dimension of the block,
    # return the location of sbar that lies within the model.
    return (spacing/2+x*spacing for x in range(int(1+(dimension-spacing/2)/spacing)))

def losbar(height):
    # return a formatted rebar name
    return 'losbar-{0}'.format(int(height))
def trsbar(height, x):
    # return a formatted rebar name
    return 'trsbar-{0}-x{1}'.format(int(height), int(x))
######################################################

mdl.ConstrainedSketch(name='__profile__', sheetSize=3000.0)
mdl.sketches['__profile__'].rectangle(point1=(0.0, 0.0),
    point2=(model_width, model_height))
for rebar_y in rebar_heights:
    for x in model_sbar_location_generator(model_width, trsbar_spacing):
     mdl.sketches['__profile__'].CircleByCenterPerimeter(center=(
        x, rebar_y), point1=(x + trsbar_diameter/2, rebar_y))
mdl.Part(dimensionality=TWO_D_PLANAR, name=
    'concslabPart', type=DEFORMABLE_BODY)
mdl.parts['concslabPart'].BaseShell(sketch=
    mdl.sketches['__profile__'])
del mdl.sketches['__profile__']

for rebar_y in rebar_heights:
    # make a hollow hole in concslab
    mdl.ConstrainedSketch(name='__profile__', sheetSize=3000.0)
    mdl.sketches['__profile__'].Line(point1=(0.0, rebar_y), point2=(
```

```
        model_width, rebar_y))
    mdl.sketches['__profile__'].HorizontalConstraint(
        addUndoState=False, entity=
        mdl.sketches['__profile__'].geometry[2])
    mdl.Part(dimensionality=TWO_D_PLANAR, name=losbar(rebar_y), type=
        DEFORMABLE_BODY)
    mdl.parts[losbar(rebar_y)].BaseWire(sketch=
        mdl.sketches['__profile__'])
    del mdl.sketches['__profile__']

    # make a plane stress trsbar
    for x in model_sbar_location_generator(model_width, trsbar_spacing):
        mdl.ConstrainedSketch(name='__profile__', sheetSize=3000.0)
     mdl.sketches['__profile__'].CircleByCenterPerimeter(center=(
          x, rebar_y), point1=(x + trsbar_diameter/2, rebar_y))
        mdl.Part(dimensionality=TWO_D_PLANAR, name=trsbar(rebar_y, x), type=
            DEFORMABLE_BODY)
        mdl.parts[trsbar(rebar_y, x)].BaseShell(sketch=
            mdl.sketches['__profile__'])
        del mdl.sketches['__profile__']
############################################################
##                 Setup Material Property                ##
############################################################
# Concrete
mdl.Material(name='Concrete')
mdl.materials['Concrete'].Elastic(table=((CONC_ELASTIC_MODULUS, 0.15), ))
mdl.materials['Concrete'].Expansion(table=((7.2e-6, ), ))
mdl.materials['Concrete'].Density(table=((2.4e-06, ), ))
mdl.materials['Concrete'].Viscoelastic(domain=TIME, table=((
    0.45, 0.45, 48.0), ), time=PRONY)
# Steel
mdl.Material(name='Steel')
mdl.materials['Steel'].Elastic(table=((200000.0, 0.0), ))
mdl.materials['Steel'].Expansion(table=((1.08e-05, ), ))
#####################################################
### Create Instances
#####################################################
# Create Instance
mdl.rootAssembly.DatumCsysByDefault(CARTESIAN)
mdl.rootAssembly.Instance(dependent=ON, name='concslab',
    part=mdl.parts['concslabPart'])
mdl.rootAssembly.makeIndependent(instances=(
mdl.rootAssembly.instances['concslab'], ))

for rebar_y in rebar_heights:
```

```
    mdl.rootAssembly.Instance(dependent=OFF, name=losbar(rebar_y), part=
        mdl.parts[losbar(rebar_y)])
    for x in model_sbar_location_generator(model_width, trsbar_spacing):
        mdl.rootAssembly.Instance(dependent=OFF, name=trsbar(rebar_y, x), part=
            mdl.parts[trsbar(rebar_y, x)])

mdl.ConstrainedSketch(gridSpacing=1.11, name='__profile__',
        sheetSize=44.77, transform=
        mdl.rootAssembly.MakeSketchTransform(
        sketchPlane=mdl.rootAssembly.instances['concslab'].faces[0],
        sketchPlaneSide=SIDE1, sketchOrientation=RIGHT, origin=(0,0,0)))
# partition vert and Horziontal
for i in range(int(model_width/partition_size)-1):
    mdl.sketches['__profile__'].Line(point1=(partition_size * (i+1), 0),
                                           point2=(partition_size * (i+1), model_height))
for i in range(int(model_height/partition_size)-1):
    mdl.sketches['__profile__'].Line(point1=(0, partition_size * (i+1)),
                                           point2=(model_width, partition_size * (i+1)))
mdl.rootAssembly.PartitionFaceBySketch(faces=
        mdl.rootAssembly.instances['concslab'].faces,
        sketch=mdl.sketches['__profile__'])
del mdl.sketches['__profile__']

## Partioning Longitudinal and Transverse steel bar in Part
    ##Creating DatumPointBy
for rebar_y in rebar_heights:
    datums_pts = []
    for i in range(int(model_width/partition_size)-1):
        datums_pts.append(mdl.rootAssembly.DatumPointByCoordinate(coords=(
            partition_size*(i+1), rebar_y, 0.0)))
        ## Partitioning the steel bar
    i = 0
    for p in [mdl.rootAssembly.datums[d.id] for d in datums_pts]:

        mdl.rootAssembly.PartitionEdgeByPoint(edge=
        mdl.rootAssembly.instances[losbar(rebar_y)].edges[i], point=p)
        i += 1

    # partition trsbar
```

```
    for x in model_sbar_location_generator(model_width, trsbar_spacing):
        for i in [trsbar(rebar_y, x), 'concslab']:
            mdl.ConstrainedSketch(gridSpacing=1.11, name='__profile__',
            sheetSize=44.77, transform=
            mdl.rootAssembly.MakeSketchTransform(
            sketchPlane=mdl.rootAssembly.instances[i].faces[0],
            sketchPlaneSide=SIDE1, sketchOrientation=RIGHT, origin=(0,0,0)))
            # partition vert and Horziontal
            mdl.sketches['__profile__'].Line(
                point1=(x - partition_size, rebar_y),
                point2=(x + partition_size, rebar_y))
            mdl.sketches['__profile__'].Line(
                point1=(x, rebar_y - partition_size),
                point2=(x, rebar_y + partition_size))
            # partition diagonal
            mdl.sketches['__profile__'].Line(
                point1=(x - partition_size, rebar_y - partition_size),
                point2=(x + partition_size, rebar_y + partition_size))
            mdl.sketches['__profile__'].Line(
                point1=(x - partition_size, rebar_y + partition_size),
                point2=(x + partition_size, rebar_y - partition_size))
            mdl.rootAssembly.PartitionFaceBySketch(faces=
                    mdl.rootAssembly.instances[i].faces,
                    sketch=mdl.sketches['__profile__'])
            del mdl.sketches['__profile__']


## Define Connector behavior
    ## losbar-BondSp behavior of a whole element
mdl.ConnectorSection(name='losbar-BondSp interior-HORZ',
     translationalType=AXIAL)
mdl.sections['losbar-BondSp interior-HORZ'].setValues(behaviorOptions=
    (ConnectorElasticity(behavior=NONLINEAR, table=((0.0, -0.2032), (
    -4241.1416, -0.1016), (-12107.77521, -0.0508), (-11013.28706, -0.0254), (
    0.0, 0.0), (11013.28706, 0.0254), (12107.77521, 0.0508), (4241.1416,
    0.1016), (0.0, 0.2032)), independentComponents=(), components=(1, )), ))
    ## losbar-BondSp behavior of a whole element
mdl.ConnectorSection(name='losbar-BondSp corner-HORZ',
    translationalType=AXIAL)
```

```
mdl.sections['losbar-BondSp corner-HORZ'].setValues(
    behaviorOptions=(ConnectorElasticity(behavior=NONLINEAR, table=((0.0,
    -0.2032), (-2120.5708, -0.1016), (-6053.887607, -0.0508), (-5506.643529,
    -0.0254), (0.0, 0.0), (5506.643529, 0.0254), (6053.887607, 0.0508), (
    2120.5708, 0.1016), (0.0, 0.2032)), independentComponents=(), components=(
    1, )), ))
mdl.ConnectorSection(name='losbar-BondSp all-VERT',
    translationalType=CARTESIAN)
mdl.sections['losbar-BondSp all-VERT'].setValues(
    behaviorOptions=(ConnectorElasticity(table=((1e+15,  ), ),
    independentComponents=(), components=(2, )), ))
mdl.ConnectorSection(name='trsbar-ALL-RIGID',
    translationalType=CARTESIAN)
mdl.sections['trsbar-ALL-RIGID'].setValues(
    behaviorOptions=(ConnectorElasticity(table=((1e+15,1e+15  ), ),
    independentComponents=(), components=(1,2 )), ))


############################################################
##                        SHEAR LAYERS                    ##
############################################################

### shear layer connector section properties
CONC_BASE_FRICTION_STIFFNESS_VERT_FORCE = 32027.1956 * PARTITION_SIZE_MODIFER
CONC_BASE_FRICTION_STIFFNESS_VERT_DISPLACEMENT = 50.8

mdl.ConnectorSection(name='ConcBase-Friction interior-VERT',
    translationalType=CARTESIAN)
mdl.sections['ConcBase-Friction interior-VERT'].setValues(
    behaviorOptions=(ConnectorElasticity(behavior=NONLINEAR, table=((0.0, 0.0),
     (CONC_BASE_FRICTION_STIFFNESS_VERT_FORCE,
      CONC_BASE_FRICTION_STIFFNESS_VERT_DISPLACEMENT)
    ), independentComponents=(), components=(2, )), ))

mdl.ConnectorSection(name='ConcBase-Friction corner-VERT',
    translationalType=CARTESIAN)
mdl.sections['ConcBase-Friction corner-VERT'].setValues(
    behaviorOptions=(ConnectorElasticity(behavior=NONLINEAR, table=((0.0, 0.0),
     (CONC_BASE_FRICTION_STIFFNESS_VERT_FORCE / 2,
      CONC_BASE_FRICTION_STIFFNESS_VERT_DISPLACEMENT)
    ), independentComponents=(), components=(2, )), ))

```

```
CONC_BASE_FRICTION_STIFFNESS_HORZ = 236.4211 * PARTITION_SIZE_MODIFER

mdl.ConnectorSection(name='ConcBase-Friction interior-HORZ',
    translationalType=CARTESIAN)
mdl.sections['ConcBase-Friction interior-HORZ'].setValues(
    behaviorOptions=(ConnectorElasticity(table=((CONC_BASE_FRICTION_STIFFNESS_HORZ, ), ),
    independentComponents=(), components=(1, )), ))

mdl.ConnectorSection(name='ConcBase-Friction corner-HORZ',
    translationalType=CARTESIAN)
mdl.sections['ConcBase-Friction corner-HORZ'].setValues(
    behaviorOptions=(ConnectorElasticity(table=((
    CONC_BASE_FRICTION_STIFFNESS_HORZ / 2, ), ),
    independentComponents=(), components=(1, )), ))

##############################################################
##              Connect steel bar with concrete            ##
##############################################################
### For trsbar
trsbarBond = []
for stbar in [i for i in mdl.rootAssembly.instances.keys() if 'trsbar' in i]:
    vertices = mdl.rootAssembly.instances[stbar].vertices
    for stbarV in vertices:
        concV = mdl.rootAssembly.instances['concslab'].vertices.findAt(stbarV.pointOn[0])
        if concV is None:
            continue
        ## connect these two point
        _tmp = mdl.rootAssembly.WirePolyLine(mergeType=IMPRINT, meshable=OFF
            , points=((stbarV, concV), ))
        trsbarBond.append(_tmp)

### For losbar
# store the wire in lists
stbarBondVert = []
stbarBondHort = []

for stbar in [i for i in mdl.rootAssembly.instances.keys() if 'losbar' in i]:
    vertices = mdl.rootAssembly.instances[stbar].vertices
    for stbarV in vertices:
        concV = mdl.rootAssembly.instances['concslab'].vertices.findAt(stbarV.pointOn[0])
        if concV is None:
            continue
        ## connect these two point
        # Vertical wire
```

```
        _tmp = mdl.rootAssembly.WirePolyLine(mergeType=IMPRINT, meshable=OFF
            , points=((stbarV, concV), ))
        stbarBondVert.append(_tmp)
        ## Horziontal wire
        _tmp = mdl.rootAssembly.WirePolyLine(mergeType=IMPRINT, meshable=OFF
            , points=((stbarV, concV), ))
        stbarBondHort.append(_tmp)

###########################################################
##          Connect concrete slab with shear layer          ##
###########################################################

# iterate all concrete slab vertices and only apply connector for the one with constraints y=0
concBaseBondVert = []
concBaseBondHorz = []

for v in mdl.rootAssembly.instances['concslab'].vertices:
    if int(v.pointOn[0][1]) == 0: # at y=0
        # Vertical wire
        _tmp = mdl.rootAssembly.WirePolyLine(mergeType=IMPRINT, meshable=OFF
            , points=((None, v), ))
        concBaseBondVert.append(_tmp)
        # Horziontal wire
        _tmp = mdl.rootAssembly.WirePolyLine(mergeType=IMPRINT, meshable=OFF
            , points=((None, v), ))
        concBaseBondHorz.append(_tmp)

# Defining node sets
mdl.StaticStep(name='Static-thermal', previous='Initial')
mdl.steps['Static-thermal'].setValues(initialInc=4.0,
    timeIncrementationMethod=FIXED, timePeriod=12.0)

faces = None
cells = None
edges = None
vertices = None
for k in mdl.rootAssembly.instances.keys():
    faces = array_append(faces, mdl.rootAssembly.instances[k].faces)
    cells = array_append(cells, mdl.rootAssembly.instances[k].cells)
    edges = array_append(edges, mdl.rootAssembly.instances[k].edges)
    vertices = array_append(vertices, mdl.rootAssembly.instances[k].vertices)

```

```
TEMP_REGION = Region(
    faces=faces,
    cells=cells,
    edges=edges,
    vertices=vertices
)

temp_csys = mdl.rootAssembly.DatumCsysByThreePoints(
        origin=(0,model_height,0),
        point1=(model_width,model_height,0),
        point2=(0,0,0),
        name='Temp-field-csys', coordSysType=CARTESIAN)
temp_csys = mdl.rootAssembly.datums[temp_csys.id]

mdl.ExpressionField(description=
    'Zero stress temp', expression=
    TEMP_INITIAL, localCsys=temp_csys, name='Zero Stress Temp')
mdl.ExpressionField(description=
    'Afterward temp profile', expression=
    TEMP_ANALYSIS_STAGE, localCsys=temp_csys, name='Temperature profile after exposing')

mdl.Temperature(createStepName='Initial',
    crossSectionDistribution=CONSTANT_THROUGH_THICKNESS, distributionType=FIELD
    , field='Zero Stress Temp', magnitudes=(1, ), name='Zero-stress-temp',
    region=TEMP_REGION)
mdl.Temperature(createStepName='Static-thermal',
    crossSectionDistribution=CONSTANT_THROUGH_THICKNESS, distributionType=FIELD
    , field='Temperature profile after exposing', magnitudes=(1, ), name='Temp-Profile',
    region=TEMP_REGION)

##########################################################
##                    Shrinkage Stage                   ##
##########################################################
mdl.StaticStep(name='Static-Shrinkage', previous='Static-thermal')
mdl.steps['Static-Shrinkage'].setValues(
    timeIncrementationMethod=AUTOMATIC, timePeriod=1.0)
for i in range(int(model_height/partition_size)+1):
    mdl.Temperature(createStepName='Static-Shrinkage',
        crossSectionDistribution=CONSTANT_THROUGH_THICKNESS, distributionType=
        UNIFORM, magnitudes=(SHRINKAGE_ANALYSIS_STAGE[
        len(SHRINKAGE_ANALYSIS_STAGE)-1-i][1], ), name='ShrinkageLvl_'+str(i),
        region=(mdl.rootAssembly.instances['concslab'].vertices.getByBoundingBox(
```

```
        yMin=i*partition_size, yMax=i*partition_size),))

############################################################
##                  Set boundary condition                ##
############################################################

##################
##   Concslab   ##
##################
concslab_left_vertices = mdl.rootAssembly.instances['concslab'
    ].vertices.getByBoundingBox(xMin=0, xMax=0)
concslab_left_edges    = mdl.rootAssembly.instances['concslab'
    ].edges.getByBoundingBox(xMin=0, xMax=0)

concslab_right_vertices = mdl.rootAssembly.instances['concslab'
    ].vertices.getByBoundingBox(xMin=model_width, xMax=model_width)
concslab_right_edges    = mdl.rootAssembly.instances['concslab'
    ].edges.getByBoundingBox(xMin=model_width, xMax=model_width)

mdl.rootAssembly.Set(name='ConcLeft', vertices=concslab_left_vertices,
    edges=concslab_left_edges)
mdl.rootAssembly.Set(name='ConcRight', vertices=concslab_right_vertices,
    edges=concslab_right_edges)
mdl.DisplacementBC(amplitude=UNSET, createStepName='Initial'
    , distributionType=UNIFORM, fieldName='', localCsys=None, name='ConcLeft',
    region=mdl.rootAssembly.sets['ConcLeft'], u1=SET, u2=UNSET,
    ur3=SET)
mdl.DisplacementBC(amplitude=UNSET, createStepName='Initial'
    , distributionType=UNIFORM, fieldName='', localCsys=None, name='ConcRight',
    region=mdl.rootAssembly.sets['ConcRight'], u1=SET, u2=UNSET,
    ur3=SET)

###################
##   Steel Bar   ##
###################
sbar_left = None
sbar_right = None
for rebar_y in rebar_heights:
    for i in [k for k in mdl.rootAssembly.instances.keys() if 'sbar' in k]:
        sbar_left = array_append(sbar_left, mdl.rootAssembly.instances[i
            ].vertices.findAt(((0, rebar_y, 0.0),), ))
        sbar_right = array_append(sbar_right, mdl.rootAssembly.instances[i
```

```
            ].vertices.findAt(((model_width, rebar_y, 0.0),), )
               )

mdl.rootAssembly.Set(name='SBLeft', vertices=sbar_left)
mdl.rootAssembly.Set(name='SBRight', vertices=sbar_right)
mdl.DisplacementBC(amplitude=UNSET, createStepName='Initial'
    , distributionType=UNIFORM, fieldName='', localCsys=None,
       name='SbarLeft',
    region=mdl.rootAssembly.sets['SBLeft'], u1=SET, u2=UNSET,
    ur3=SET)
mdl.DisplacementBC(amplitude=UNSET, createStepName='Initial'
    , distributionType=UNIFORM, fieldName='', localCsys=None,
       name='SbarRight',
    region=mdl.rootAssembly.sets['SBRight'], u1=SET, u2=UNSET,
    ur3=SET)


############################################################
##         Assign material properties to section          ##
############################################################

## Assign concrete properties
mdl.HomogeneousSolidSection(material='Concrete', name=
    'ConcSection', thickness=model_height/2)

# Create set
f = mdl.parts['concslabPart'].faces
mdl.parts['concslabPart'].Set(name='Side', faces= f)

mdl.parts['concslabPart'].SectionAssignment(offset=0.0,
    offsetField='', offsetType=MIDDLE_SURFACE, region=
    mdl.parts['concslabPart'].sets['Side'], sectionName=
    'ConcSection', thicknessAssignment=FROM_SECTION)

# Add thickness to slab
## Assign steel properties
mdl.CircularProfile(name='Sbar_radius', r=losbar_diameter/2)
mdl.BeamSection(consistentMassMatrix=False, integration=
    DURING_ANALYSIS, material='Steel', name='losbarSection',
       poissonRatio=0.0,
    profile='Sbar_radius', temperatureVar=LINEAR)

for rebar_y in rebar_heights:
    # trsbar
    mdl.HomogeneousSolidSection(material='Steel', name=
        'trsbarSection', thickness=None)
    for x in model_sbar_location_generator(model_width,
       trsbar_spacing):
        mdl.parts[trsbar(rebar_y, x)].SectionAssignment(offset
            =0.0,
            offsetField='', offsetType=MIDDLE_SURFACE, region=
```

```
            (mdl.parts[trsbar(rebar_y, x)].faces,), sectionName=
            'trsbarSection', thicknessAssignment=FROM_SECTION)

    # losbar
    e = mdl.parts[losbar(rebar_y)].edges
    mdl.parts[losbar(rebar_y)].Set(name='LongSteel', edges= e)

    mdl.parts[losbar(rebar_y)].SectionAssignment(offset=0.0,
        offsetField='', offsetType=MIDDLE_SURFACE, region=
        mdl.parts[losbar(rebar_y)].sets['LongSteel'],
        sectionName='losbarSection', thicknessAssignment=FROM_SECTION)
        # Assign material orientation
    mdl.parts[losbar(rebar_y)].MaterialOrientation(
        additionalRotationType=ROTATION_NONE, axis=AXIS_3, fieldName='', localCsys=
        None, orientationType=GLOBAL, region=
        mdl.parts[losbar(rebar_y)].sets['LongSteel'],
        stackDirection=STACK_3)
    mdl.parts[losbar(rebar_y)].assignBeamSectionOrientation(
        method=N1_COSINES, n1=(0.0, 0.0, -1.0), region=
        mdl.parts[losbar(rebar_y)].sets['LongSteel'])


##############################################################
##            Assign connector section to wire              ##
##############################################################

if len(trsbarBond) > 0:
    ## Assigning for bond slip between stbar and concrete
    trsbarBond, _ = edgeNameToEdgeArrayFilter(trsbarBond,
                                                         lambda x: True)
    ##trsbar
    mdl.rootAssembly.Set(name='STCONC-trsbar', edges=conArray(trsbarBond))
    mdl.rootAssembly.SectionAssignment(region=
        mdl.rootAssembly.sets['STCONC-trsbar'], sectionName=
        'trsbar-ALL-RIGID')
    mdl.rootAssembly.ConnectorOrientation(localCsys1=
        mdl.rootAssembly.datums[1], region=
        mdl.rootAssembly.allSets['STCONC-trsbar'])

##losbar
##  Vertical
stbarCorner, stbarInterior = edgeNameToEdgeArrayFilter(stbarBondHort,
                                    lambda x: eql(x.pointOn[0][0], 0) or
                                              eql(x.pointOn[0][0], model_width))

```

```
mdl.rootAssembly.Set(name='STCONC-Interior', edges=conArray(
    stbarInterior))
mdl.rootAssembly.SectionAssignment(region=
     mdl.rootAssembly.sets['STCONC-Interior'], sectionName=
     'losbar-BondSp interior-HORZ')
mdl.rootAssembly.ConnectorOrientation(localCsys1=
     mdl.rootAssembly.datums[1], region=
     mdl.rootAssembly.allSets['STCONC-Interior'])

mdl.rootAssembly.Set(name='STCONC-corner', edges=conArray(
    stbarCorner))
mdl.rootAssembly.SectionAssignment(region=
     mdl.rootAssembly.sets['STCONC-corner'], sectionName=
     'losbar-BondSp corner-HORZ')
mdl.rootAssembly.ConnectorOrientation(localCsys1=
     mdl.rootAssembly.datums[1], region=
     mdl.rootAssembly.allSets['STCONC-corner'])

##  Horziontal
stbarVert, _ = edgeNameToEdgeArrayFilter(stbarBondVert,
                                         lambda x: True)

mdl.rootAssembly.Set(name='STCONC-VERT', edges=conArray(
    stbarVert))
mdl.rootAssembly.SectionAssignment(region=
     mdl.rootAssembly.sets['STCONC-VERT'], sectionName=
     'losbar-BondSp all-VERT')
mdl.rootAssembly.ConnectorOrientation(localCsys1=
     mdl.rootAssembly.datums[1], region=
     mdl.rootAssembly.allSets['STCONC-VERT'])

############################################################
##    Assign section for conc base & shear layer wires    ##
############################################################

# Horziontal

concBaseBondCorner, concBaseBondInterior =
    edgeNameToEdgeArrayFilter(
     concBaseBondHorz,
    lambda x: eql(x.pointOn[0][0], 0) or
              eql(x.pointOn[0][0], model_width))

mdl.rootAssembly.Set(name='CONCBASE-Interior-HORZ',
                          edges=conArray(concBaseBondInterior))
mdl.rootAssembly.SectionAssignment(region=
     mdl.rootAssembly.sets['CONCBASE-Interior-HORZ'],
     sectionName='ConcBase-Friction interior-HORZ')
mdl.rootAssembly.ConnectorOrientation(localCsys1=
     mdl.rootAssembly.datums[1], region=
     mdl.rootAssembly.allSets['CONCBASE-Interior-HORZ'])
```

```
mdl.rootAssembly.Set(name='CONCBASE-corner-HORZ',
                     edges=conArray(concBaseBondCorner))
mdl.rootAssembly.SectionAssignment(region=
    mdl.rootAssembly.sets['CONCBASE-corner-HORZ'], sectionName=
    'ConcBase-Friction corner-HORZ')
mdl.rootAssembly.ConnectorOrientation(localCsys1=
    mdl.rootAssembly.datums[1], region=
    mdl.rootAssembly.allSets['CONCBASE-corner-HORZ'])


# Vertical

concBaseBondCorner, concBaseBondInterior = edgeNameToEdgeArrayFilter(
        concBaseBondVert,
        lambda x: eql(x.pointOn[0][0], 0) or
                  eql(x.pointOn[0][0], model_width))

mdl.rootAssembly.Set(name='CONCBASE-Interior-VERT',
                     edges=conArray(concBaseBondInterior))
mdl.rootAssembly.SectionAssignment(region=
    mdl.rootAssembly.sets['CONCBASE-Interior-VERT'],
    sectionName='ConcBase-Friction interior-VERT')
mdl.rootAssembly.ConnectorOrientation(localCsys1=
    mdl.rootAssembly.datums[1], region=
    mdl.rootAssembly.allSets['CONCBASE-Interior-VERT'])

mdl.rootAssembly.Set(name='CONCBASE-corner-VERT',
                     edges=conArray(concBaseBondCorner))
mdl.rootAssembly.SectionAssignment(region=
    mdl.rootAssembly.sets['CONCBASE-corner-VERT'], sectionName=
    'ConcBase-Friction corner-VERT')
mdl.rootAssembly.ConnectorOrientation(localCsys1=
    mdl.rootAssembly.datums[1], region=
    mdl.rootAssembly.allSets['CONCBASE-corner-VERT'])

###########################################################
##                    Mesh the sections                  ##
###########################################################

all_instances = [mdl.rootAssembly.instances[i] for i in mdl.rootAssembly.instances.keys()]
all_instances = tuple(all_instances)

for i in mdl.rootAssembly.instances.keys():
    _instance_mesh_size = mesh_size_global
    if 'trsbar' in i:
        _instance_mesh_size = mesh_size_trsbar
    mdl.rootAssembly.seedPartInstance(deviationFactor=0.1,
```

```
        minSizeFactor=0.1, regions=(mdl.rootAssembly.instances[i],),
        size=_instance_mesh_size)

for rebar_y in rebar_heights:
    for x in model_sbar_location_generator(model_width, trsbar_spacing):

############################################################
##        seed the innear diagonals nearby of trsbar       ##
############################################################

        _edges = mdl.rootAssembly.instances['concslab'
            ].edges.getByBoundingBox(xMin=x-partition_size,
              xMax=x+partition_size,
              yMin=rebar_y-partition_size,
              yMax=rebar_y+partition_size)
        mdl.rootAssembly.seedEdgeBySize(constraint=FINER,
            deviationFactor=0.1, edges=_edges, minSizeFactor=0.1,
            size=mesh_size_trsbar_nearby_diagonals)

############################################################
##           seed the outer edge nearby of trsbar          ##
############################################################

        # Left
        _edges = mdl.rootAssembly.instances['concslab'
            ].edges.getByBoundingBox(xMin=x-partition_size,
              xMax=x-partition_size,
              yMin=rebar_y-partition_size,
              yMax=rebar_y+partition_size)
        # Right
        _edges += mdl.rootAssembly.instances['concslab'
            ].edges.getByBoundingBox(xMin=x+partition_size,
              xMax=x+partition_size,
              yMin=rebar_y-partition_size,
              yMax=rebar_y+partition_size)
        # Up
        _edges += mdl.rootAssembly.instances['concslab'
            ].edges.getByBoundingBox(xMin=x-partition_size,
              xMax=x+partition_size,
              yMin=rebar_y+partition_size,
              yMax=rebar_y+partition_size)
        # Down
        _edges += mdl.rootAssembly.instances['concslab'
            ].edges.getByBoundingBox(xMin=x-partition_size,
              xMax=x+partition_size,
              yMin=rebar_y-partition_size,
              yMax=rebar_y-partition_size)
        mdl.rootAssembly.seedEdgeBySize(constraint=FINER,
```

```
            deviationFactor=0.1, edges=_edges, minSizeFactor=0.1,
            size=mesh_size_trsbar_nearby_outer_edges)

############################################################
##              seed the parameter of the trsbar          ##
############################################################

        _edges = mdl.rootAssembly.instances['concslab'
            ].edges.getByBoundingBox(xMin=x-trsbar_diameter/2,
              xMax=x+trsbar_diameter/2,
              yMin=rebar_y-trsbar_diameter/2,
              yMax=rebar_y+trsbar_diameter/2)
        mdl.rootAssembly.seedEdgeBySize(constraint=FINER,
            deviationFactor=0.1, edges=_edges, minSizeFactor=0.1,
            size=mesh_size_trsbar)

mdl.rootAssembly.setMeshControls(elemShape=QUAD, technique=STRUCTURED,
    regions=mdl.rootAssembly.instances['concslab'].faces)
mdl.rootAssembly.generateMesh(regions=all_instances)


############################################################
############################################################


############################################################
##                    Submit job to Abaqus                ##
############################################################

NUM_OF_CPUS = 4
NUM_OF_GPUS = 0
import datetime
jobname = model_name+'_@_'+datetime.datetime.now().strftime("%d%m%y_%I-%M%p")
print('> Submitting analysis now with name "'+jobname+'"')
mdb.Job(atTime=None, contactPrint=OFF, description='', echoPrint=OFF,
    explicitPrecision=SINGLE, getMemoryFromAnalysis=True, historyPrint=OFF,
    memory=90, memoryUnits=PERCENTAGE, model=model_name,
    modelPrint=OFF, multiprocessingMode=DEFAULT, name=jobname,
    nodalOutputPrecision=SINGLE, numCpus=NUM_OF_CPUS, numDomains=4, numGPUs=NUM_OF_GPUS, queue=None
    , resultsFormat=ODB, scratch='', type=ANALYSIS, userSubroutine='',
    waitHours=0, waitMinutes=0)
mdb.jobs[jobname].submit(consistencyChecking=OFF)
```