Daniel Cantero
Terje Kanstad

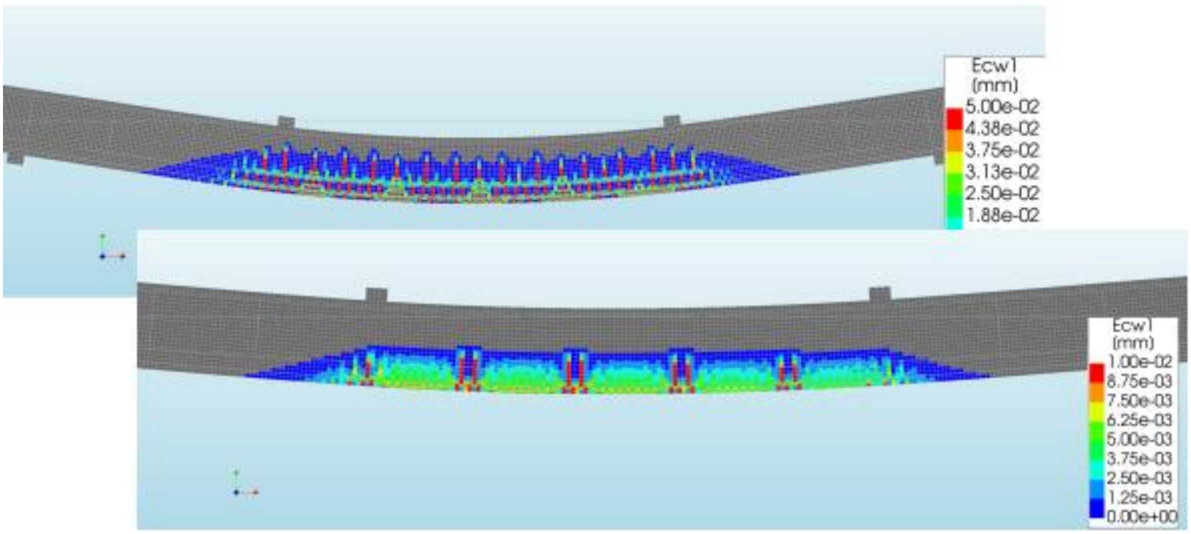
NTNU
Norwegian University of
Science and Technology

Numerical investigations of damaged post-tension systems and their structural effect on bridges

Report

Trondheim – May – 2022

NTNU
Norwegian University of
Science and Technology
Faculty of Engineering
Department of Structural Engineering



ISBN: 978-82-7482-202-3

NTNU
Norwegian University of
Science and Technology

Preface

In April 2018 Norwegian University of Science and Technology (NTNU) and the Norwegian Public Roads Administration (NPRA) started a collaboration to contribute to the Research and Innovation program *Better bridge maintenance – Project 2 Reinforcement corrosion – Part 4 Maintenance of prestressed concrete structures*. The collaboration focused only on post-tensioned bridges. It aimed at studying the failure mechanisms of such bridges and the investigation of several particular case studies of bridges in Norway. A review, summary and conclusion of these activities are presented in this report.

During this project two additional documents have been elaborated:

- Menga, T. Kanstad, D. Cantero. Corrosion-induced failures of post-tensioned bridges. NTNU report, 2021.
- G. Pinto, D. Cantero. Modelling post-tensioned structures with DIANA FEM software. NTNU report, 2021.

The main outputs and conclusions from these two documents have been included in the present report.

The authors acknowledge the support provided by the Norwegian Public Road Administration (NPRA). In particular, we would like to thank Lise Bathen and Karla Hornbostel for their guidance, facilitation of information, feedback, constructive discussions, and revision of the published reports. We would like to thank also Håvard Johansen and Magdalena Jadwiga Osmolska for their review and feedback on this final document.

Abstract

This study provides additional knowledge about the structural consequences of damaged PT systems, with particular focus on the presence of voids in the duct. The work consists of two distinct parts. The first part is a literature review on various relevant topics including, damages in PT bridges, corrosion mechanisms, NDT, and existing guidelines. In the second part, the problem is investigated numerically by FEM analysis, analysing simple beams, case studies and generic bridge typologies. Finally, this report provides a summary of the most relevant conclusions.

Indexing terms	**Stikkord**
Post-tensioned bridges	*Etterspente broer*
Numerical modelling	*Numerisk Modellering*
Damage	*Skader*

Table of contents

List of abbreviations

DMRB = Design Manual for Roads and Bridges (UK)

FEM = Finite Element Model/Modelling

NDT = Non-Destructive Testing/Techniques

NTNU = Norwegian University of Science and Technology

NPRA = Norwegian Public Roads Administration

PT = Post Tensioned

SLS = Serviceability Limit State

ULS = Ultimate Limit State

1. Introduction

Prestressed concrete allows for the introduction of large axial stresses in concrete structures. These large forces are introduced by prestressing (hence the name) high-strength steel. These axial stresses not only change the internal stress distribution, resulting in larger parts of the structure working in compression, but also depending on the relative location of the active reinforcement, they (partly or fully) compensate the external loads of the structure. Prestressing can be achieved either by pre-tensioning or post-tensioning (PT), depending on when during the construction process the prestressing force is applied to the high-strength steel. When the structure is post-tensioned, the axial force is applied after the concrete has gained sufficient strength. The steel strands used in PT systems are usually grouped into tendons that are placed inside ducts. These ducts are somewhat larger in diameter than the tendons. The free space between the duct and the tendons can be filled with cementitious material (grout) or a lubricant (grease), which result in bonded and un-bonded solutions respectively. In both cases the filling is introduced mainly to provide an additional level of protection. The mechanical behaviour of both solutions is somewhat different, and they are used generally for different applications. In addition, the location of the PT tendons can be internal or external. This report focuses on internal PT systems with full bond between steel and concrete achieved by grouted ducts.

In Norway, grouted PT systems have been used in bridge engineering and it allowed for the construction of longer bridge spans. The design of bridges in general should follow the bridge design guideline (Handbook N400 [1]) and that the prestressing system shall be according to the general specifications in Handbook R762 [2]. More in particular, NPRA published a guide (Report Nr. 668 [3]) that describes how to design PT bridges. An extended example of calculations for a PT concrete plate/beam bridge can be found in [4].

However, it has been found that PT bridges have durability problems that can seriously compromise its structural safety. There exist established recommendations on design of PT for durability [5]. But because the high-strength steel used for the active reinforcement is prone to corrosion and its fundamental role in the structural behaviour, the failure of that component has severe consequences on the structural safety. The durability depends not only on the durability of the materials but also on the correct installation of the components in the PT system. Full grouting of a duct requires skilled personnel and strict application of construction procedures [6]. If voids are present in the ducts, serious steel corrosion can develop because of the lack of the protecting grout [7]. In addition to execution mistakes, the choice of materials can also lead to inappropriate properties and conditions [8]. Notably, grout segregation and bleed water result in unfavourable environments that initiate and accelerate the corrosion of the prestressing steel.

Since the first application of the PT technology several progressive improvements have been implemented in the materials to use, execution procedures, and quality control. In Europe, a series of standards were developed in 1997, namely EN 445 [12], EN 446 [13] and EN 447 [14], to ensure quality of the grout material and injection procedure in order to avoid previously detected issues (voids, grout segregation and bleeding). Additional developments in this regard included the release of the European Technical Approval for PT systems [10] and the establishment of common levels of competence for PT systems installation [11].

In Norway this is reflected in “prestressed steel works” (*Spennarmeringsarbeider*) document NB14 [9]. In addition, stricter regulations are defined in NPRA’s R762 manual [2] regarding maximum allowable chloride content as a chemical accelerator for the concrete during cold temperatures, strength classes and concrete cover for the reinforcement. Despite these improvements, durability problems are still being reported in new PT structures. Furthermore, a large proportion of the existing bridges were constructed during the period 1950-1970, before the recent technological improvements. Thus, the concerns about the durability of PT bridges are particularly relevant for older existing constructions, which in addition to normal age-related deterioration processes they might have faulty PT installations.

Traditionally PT structures were believed to be reliable, requiring little maintenance [15]. However, after what is considered the first bridge collapse due to a damaged PT system in 1985 (*Ynys-y-Gwas*) a serious of bridge failures have followed. The direct identification of tendon damage is hindered by the fact that the tendons are inside the concrete section and complicated further by the lack of observable indications of damage. This has been clearly highlighted in several publications. In order to stress this point, it was deemed relevant to include direct citations from those documents below:

- "Safety critical defects in post-tensioned concrete are typically hidden, very difficult to detect and may result in a brittle mode of failure.". And "Visual inspections alone may not give warning of imminent collapse; on the other hand, intrusive investigations can be expensive and potentially damaging for the structure and should only be carried out if there is a clear need." [16]
- "Corrosion damage to internal tendons is virtually impossible to detect without use of invasive inspection methods." [17]
- "Although the evidence of tendon corrosion arising from a deck waterproofing defect usually shows up as simple water leakage or the presence of seepage products such as efflorescence or traces of rust, there are a number of cases where tendons have been partly corroded without any external signs or visible defects." [18]

Corroded tendon locations are directly linked to the presence of voids or poor grout conditions. These locations exhibit poor or no bond between the high-strength steel and the concrete. In theory, the cracking behaviour of bonded sections is clearly different from unbonded ones, resulting in distinct crack patterns for each situation. However, experimentally this cannot be observed so clearly. As shown in [19], the same beam shows different cracking patterns for different levels of bonding quality. From the results it is difficult to assess the level of bonding starting from a given cracking pattern.

Furthermore, a recent survey among the transportation agencies across the US [20] highlights that the quality of the maintenance highly depends on the expertise of the coordinating agency, consultants, and the contractors involved in the repair because of the lack of national guidance document.

Therefore, PT structures can still be regarded as reliable in general, and its use in future structures is still recommended when necessary. However, designers, construction engineers and infrastructure owners must be aware of possible issues of this technology that can compromise the structural safety. In order to avoid these problems, it is necessary to carefully design all details, ensure a correct execution during construction and plan and perform appropriate maintenance procedures.

1.1. Objective of the report

The scope of this project is to study the structural consequences of damaged PT bridges with fully bonded tendons. Other prestressing solutions, such pre-tensioned bars or strands, are out of the scope of this report. In particular, it focuses on the consequences of having poor grouting conditions or voids. The associated damage scenarios (tendon section loss due to corrosion and tendon breakage) are investigated in a range of case studies and bridge typologies. The study attempts to find answers to the following questions:

- How can poor grout and voids be modelled?
- Do the numerical results identify any detectable indication of damage? Is there any identifiable structural effect that would indicate an imminent structural failure? Might the cracking pattern be a good indicator for potential presence of voids?
- Based on literature, where is it more likely to have voids and corrosion problems that might lead to tendon rupture?
- In the event of tendon breakage, what would be the remaining capacity of an existing structure?

The main goal of this report is to improve the understanding of the structural consequences of damage and/or corrosion in PT systems. This requires knowledge of possible failure mechanisms. The influence of failure of certain parts of the PT system on the load bearing capacity of the structure is to be assessed by recalculation of critical cross sections and an overall assessment of the load bearing capacity for selected cases/bridges. The case study aims at developing general recommendations for the assessment of structural consequences in case of damage of the PT system.

1.2. Methodology and report structure

In order to address these goals four distinct tasks were necessary, namely, literature review, numerical modelling, case studies and conclusions. Each of these points correspond to a separate section in this report. Section 2 reports a literature review on various aspects of the problem. The reviews have as the starting point the existing available reports and guidelines published by NPRA. Section 3 discusses the numerical modelling, provides a programmatic framework, and validates the adopted modelling strategy. Section 4 performs the numerical investigations on case studies that either correspond to existing bridges in Norway or resemble typical dimensions of certain structural typology. This work was done by master students under the direct supervision of NTNU. Finally, Section 5, summarizes all the results and conclusions.

2. Review

This section presents a review of four important aspects of the project, namely, PT system damages, corrosion mechanisms, NDT for PT systems and existing guidelines for management of PT bridges. The presented reviews give a short summary of the most relevant existing literature for each topic. The intention of each subsection is to provide complementary information to the existing reports and guidelines by the NPRA. Therefore, it is acknowledged that these reviews are not comprehensive in nature but are instead additive to the agency's available reports.

2.1. Post-tension system damages

Existing literature and results regarding damages in PT can be organized according to the addressed geometric scale of the problem. In this regard, topics are organized below in ascending order starting from the material level up to the structural level.

Regarding the material properties of corroded high-strength steel, existing literature clearly establishes that its strength and ductility is reduced. Several publications investigated the material properties in laboratory conditions [21, 22, 23], showing that corroded strands feature reductions in ultimate strain that leads to a brittle failure mode of the steel, while reporting comparatively smaller changes in modulus of elasticity. The numerical modelling of corroded high-strength steel is addressed in [24, 25, 26] that provide alternative constitutive models for corroded strands, all characterized by modulus of elasticity, strength, and ductility proportional to the corrosion magnitude.

The bond between ordinary steel and concrete has been extensively studied, also including the effect of corrosion, but few models exist for the case of strands and grout, as mentioned in [25]. This publication explores different numerical strategies to model the bond in such cases and compares them to experimental results. The proposed model consists of two sets of un-coupled springs that respectively represent the radial behaviour and the bond-slip between high-strength steel and grout. The correct representation of bond is paramount when investigating the breakage and re-anchorage of tendons. [27] studies tendon failures both numerically and experimentally focusing on the correct modelling of the transmission length at the locations near the tendon breakage. The study suggests that the bond-slip relation from the CEB-FIP Model Code 90 [28] gives satisfactory results. On the other hand, the study in [29] argues that this models the bond between concrete and duct. Since the bond actually develops between the prestressing steel and the grout, the above formulation/model was removed in the latest fib Model Code 2010 [30]. This study evaluates an analytical model presented by [31] to investigate the re-anchorage length and extends it to include the non-linear material behaviour, because the stresses in the steel might exceed its tensile strength. After experimental testing, the authors refine the bond numerical modelling, concluding that the presence of partial voided areas in the duct significantly increased the re-anchorage (transmission) length. This length depends on the size of the void because the bond is strongly influenced by the amount of confining material (grout). In [32], the authors provide an improved bond-slip relationship validated by experimental results that considers the effect of the helicoidal shapes of strands and corresponding twisting effect under tension. This formulation can also be used to account for corrosion damage, which reduces the bond strength.

Poor grouting has often been reported as a main contributor to bridge damage [7] because it exposed the strands to air and moisture. The issues with grouting are mainly due to poor grout quality and incomplete filling of the ducts. And even after the general improvements on the grout quality and control adopted since the turn of the millennium, newer grouts have demonstrated the potential for grout segregation, soft grout, excessive bleed water, and high chloride and sulphate contents [20]. The newer grouts specifications should have solved bleed-water problem and corrosion, but bridge tendon failures and occurrence of deficient grout have been still reported [33]. In addition, there are concerns about the execution quality and material properties of the so-called pour-back locations [20]. These refer to anchorages and vent locations that needed additional filling after the main grout injection had been performed. The poor execution, the shrinkage properties of the filler or even the lack of this operation, allows moisture and contaminants to reach the strands at anchorages and vents.

Therefore, the presence of grout voids is an issue that concerns all existing PT bridges and must be considered when managing bridge inventories. In fact, as a representative example, over 50% of bridges presented voids in UK [34, 35], with 25% presenting large voids or ungrouted tendons, indicating that older bridges were more likely to have more voids. However, the problem of poor grouting is not limited only to the presence of voids, but it also affects the quality of the grout in the duct of existing structures. Based on forensic investigations, [36] describes four distinct grout textures/appearances:

Type 1: Segregated wet plastic (soft) grout with a clay-like consistency.
Type 2: Segregated grout with black striated layers.
Type 3: Segregated dry grout with a chalky white consistency.
Type 4: Hardened, grey, dry grout.

According to [36], in some cases, all four grout types have been reported in the same general vicinity, where only Type 4 has the desired properties for a functional PT system. Grout segregation (Types 1 to 3) occurred at high tendon elevations and resulted in free water with possible high concentrations of corrosive ions. As a result of the poor grouts, there might be subsidence and the strands become exposed to trapped air. Corrosion that occurs in such situations is likely to be enhanced by elevated chloride concentrations but will still initiate and propagate even if the concentrations are below the 0.08% percent cement limit [36].

The effect of damaged PT systems on the structural response has mostly been investigated on scaled laboratory experiments involving simple beams with one or few tendons. For instance, [24] reports testing of corroded PT beams in three-point bending tests. There is no capacity reduction when the corrosion is near the supports because it has no significant influence on the global behaviour. However, when the corrosion is at mid-span section, tendon area losses of 9.49% resulted in capacity reductions of 15.56%. In [37] different void lengths and locations scenarios are studied. Before cracking point, all of them show the same linear relation, indicating that insufficient grout has negligible effect on the linear part of the structural response. The post-cracking behaviour shows that insufficient grouting results in section stiffness, ductility, and ultimate flexural strength reductions. Also, concentration of cracks of larger magnitudes on those beam segments with voids in the PT system are reported. Similarly, [21] shows that corrosion in strands significantly degrades the local post-cracking stiffness and that corroded strands lead to a decrease of the number of cracks, increasing their spacing, and concentrating them in the ungrouted regions. The results also show that the ultimate strength of the beam reduces, and that the failure mode changes from concrete crushing to wire rupture. [38] and [21] further confirm that strand corrosion (without voids) sharply reduces the ductility of PT beams, comparatively more than the ultimate strength. [39] addressed the effect of grouting compactness, showing that the cracking load and ultimate load of the beam tended to increase with dense grouting and that they are 25% higher than that of the prestressed concrete beams without grouting. [40] focuses on the effects of voids near the supports on shear cracking and capacity, concluding that the diagonal cracking load decreases not only with the presence of voids but also with corrosion. Regarding the shear capacity, it is reduced by corrosion, but insufficient grouting has negligible effect on it. In addition, the voids decrease the number of diagonal cracks and accelerate the crack propagation. Finally, [27] highlights that the location of tendon breakage plays an important role in the reduction of the moment capacity, observing important reductions when the breakage is at a section subjected to large bending.

Experimental testing of large-scale specimens of real size elements in bridges is limited, because of practical and economic reasons. But [41] performed testing on beams from a 45-year-old bridge. The tested beams had different levels of corrosion, ranging from no damage to severe damage. This study concludes that there may be significant reductions in load bearing capacity in PT beams with corrosion damage and that the reduction is approximately linear with the degree of corrosion area loss.

Rather than by laboratory testing, a different methodology to study the effect of damaged PT systems is to review known bridge failures. This task was performed and documented in the separate report "*Corrosion-induced failures of post-tensioned bridges*" [42], which was produced during of the project. This document reports an extended literature survey of corrosion-induced failures of PT bridges with special attention to the presence of warning signs. The report provides an in-depth analysis of the most frequent causes of corrosion. The same methodology is also presented in a summarizing document in [43] as part of CACRCS 2021 conference proceedings. The study analyses 52 prestressed structures based on publicly available reports. Each of these structures is studied systematically to: identify the grout condition, list the presence of warning signs recorded during inspection, recognize the source of corrosion, describe the damage mechanism, and characterize the failure type. The main conclusions of the report [42] are summarized here.

- There is a clear link between voids and corrosion.
- Reported damage warning signs were: shear cracks and longitudinal cracks following the duct location, concrete spalling, concrete staining, anchorage corrosion, open grout vents, water leakage, grout efflorescence.
- The absence of warning signs that indicate the presence of damage does not necessarily guarantee the structural safety. Failure may occur without showing any warning signs.
- If warning signs are detected, maintenance measures should be planned and executed without delay.
- The location of damages in PT bridges is generally at joints, anchorages, and deviation points.
- The most corrosion-sensitive structures are segmental and box section bridges.
- Most failures were attributed to design and execution mistakes that facilitated the ingress of external chlorides leading to corrosion damage.

To improve bridge condition assessment, it is necessary to identify the critical locations with respect to PT system damage. The mapping of those critical locations was presented in [5] and correspond to points in the structure with potential to develop tendon damage due to corrosion. The study identifies the locations where aggressive water can most likely get to the active steel and initiate the corrosion. These locations are categorized by the type of corrosion prevention mechanism that failed, namely, failure of external barriers or failure of tendon corrosion protection system. For each category, critical damages and locations are listed in Table 1. The corresponding sketch for each of the potential damages is also reproduced in Figure 1. Alternatively, [18] and [44] approach the classification from another perspective, in terms of possible defects. In this way it is possible to categorize critical locations depending on the type of defect, as shown in Table 2.

Table 1: List of damages that can lead to corrosion initiation [5]

Category	**Damage**
Failure of external barriers	1 Defective wearing course (e.g. cracks)
	2. Missing or defective waterproofing membrane
	3. Defective drainage intakes and pipes
	4. Wrongly placed outlets for the drainage of wearing course and waterproofing
	5. Leaking expansion joints
	6. Cracked and leaking construction or element joints
	7. Inserts (e.g. for electricity)
	8. Defective concrete cover
Failure of tendon corrosion protection system	9. Partly or fully open grouting in- and outlets (vents)
	10. Leaking, damaged metallic ducts mechanically or by corrosion
	11. Cracked and porous pocket concrete
	12. Grout voids at tendon high and low points

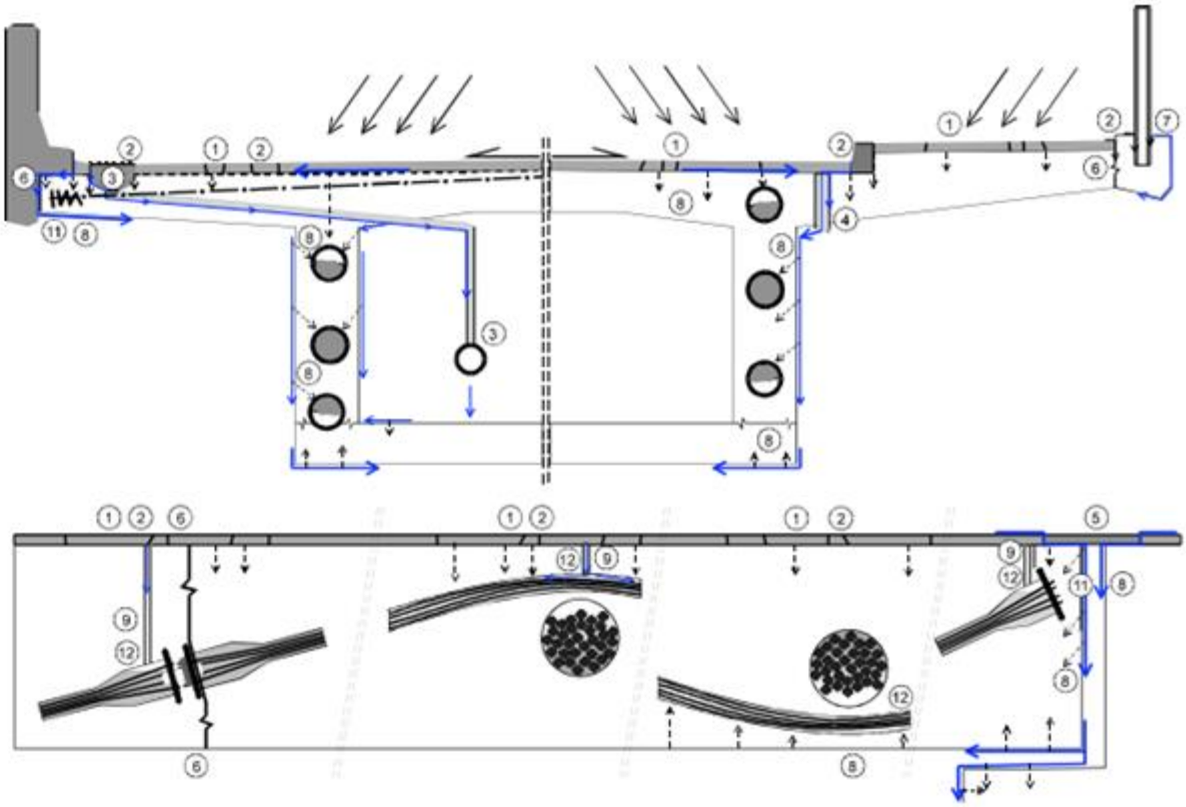


Figure 1: Hazard scenarios [5]

Table 2: Defects in PT that can lead to corrosion initiation [18, 44]

Category	Defect
Design defects	Lack of waterproofing
	Lack of sealing at anchorages
	Lack of provision of drainage
	Leaking expansion joints
	Insufficient cover
Inappropriate construction materials	Chemical composition of concrete
	Steel susceptible to stress corrosion
	Bitumen-coated Kraft paper ducts
	Lead-lined steel ducts
Construction defects	Poor waterproofing
	Incomplete grouting
Maintenance and operation defects	Failure to maintain watertightness

It is worth highlighting that one bridge typology emerges as especially problematic. As indicated in [7, 20, 42], PT segmental bridges are more susceptible to corrosion of the active steel and structural failure, which led to the moratorium on grouted PT structures in the UK. However, [7] determined that the corrosion problems can be attributed principally to poor design choices and poor-quality construction. This study concludes that segmental bridges have no intrinsic susceptibility to corrosion. Laboratory research indicates that properly designed and constructed precast segmental and PT structures are durable. In particular, the segment joints with internal tendons are corrosion resistant when properly constructed, especially when using match-cast epoxy joints with complete encapsulation of the prestressing steel in a high-quality grout [7]. Nevertheless, existing bridges of this typology require special attention, and future designs should avoid previous design mistakes. Note that in this report segmental bridges refer to segmentally constructed constructions, that feature the problematic joint between segments. On the other hand, segmentally casted bridges (like the cantilever type) do not have this problematic interface.

2.2. Corrosion mechanisms

Corrosion of steel in concrete has been extensively studied, like in the seminal book [45] that discusses in detail all possible causes of steel corrosion in concrete, together with a review of diagnosis and reparation strategies. Even though the accumulated knowledge on the topic is large, there is still a lack of understanding on the corrosion mechanism in PT systems. And as highlighted in [20], by far, the most significant problem facing PT bridges has been tendon corrosion, because strand loss can compromise structural integrity and can lead to catastrophic failure. In addition, higher tensile strength steels show an increased sensitivity to corrosion, compared to mild reinforcing steel [46].

To address this, and as part of the same research and innovation project (*Bedre Bruvedlikehold*), a study was done on the methods to identify corrosion in internal PT systems, which also provides a review of possible corrosion mechanisms [15]. This report identifies the three main relevant types of corrosion, namely, general corrosion, pitting/local corrosion and stress corrosion cracking and hydrogen embrittlement. Similar corrosion mechanisms are therefore relevant also to PT systems, as reported in [20, 47].

The most common fundamental corrosion reactions are chemical reactions between iron, water, and oxygen. Because cement grouts are strongly alkaline (pH up to 13) [5] corrosion of the steel is inhibited. However, it might be neutralized (pH falling below 9) by reaction with atmospheric carbon dioxide. Also, infiltration of chlorides will de-passivate the surface of the steel and will catalyse the onset of pitting corrosion. The two primary sources of chlorides are de-icing chemicals and seawater [20]. PT structures are not necessarily more susceptible to chloride intrusion because chlorides diffuse through these structures in the same way as in other structures. However, high-strength steel has been shown to corrode at a faster rate than mild steel due to higher level of stresses [48]. While for reinforced concrete there exist a generally accepted chloride content threshold for corrosion initiation, this is different for PT concrete. [49] mentions that there are reports of corrosion in PT bridges for lower chloride concentrations than the conservative thresholds of 0.2% by weight of cement. Furthermore, low levels of chlorides do not necessarily mean that corrosion is prevented. For example, pitting corrosion has been reported at the Herøysund bridge where low levels of chlorides were measured. Arguably pitting must be related to chlorides, but the triggering threshold might be lower for prestressed concrete. Therefore, existing threshold value on chloride concentration for corrosion initiation should be treated with caution.

Another corrosion mechanism is stress corrosion cracking and hydrogen embrittlement, that leads to more brittle steel. This phenomenon arises from a breakdown of the steel's internal structure itself [50] producing small cracks, reducing its ductility. It needs special conditions to develop including stresses above certain threshold. Thus, it is known that high-strength steel is more vulnerable to these effects. In contrast to other types of corrosion, this type of corrosion is not necessarily associated to any noticeable indication [51].

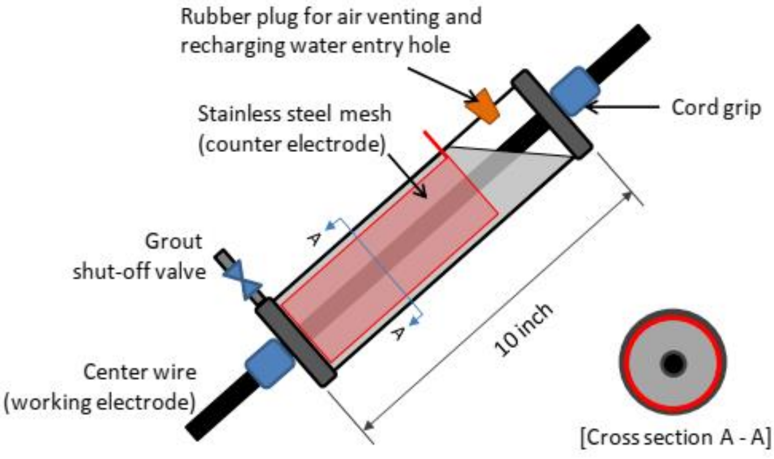
There exist reports of an additional corrosion mechanism that is particular to grouted PT systems. As briefly mentioned in [15] corrosion under low-oxygen circumstances may occur. More details about this type of corrosion can be found in [52, 53]. It is reported that corrosion has been found in areas of segregated grout, that takes a whitish phase with plastic consistence, as shown in Figure 2. It has also been found sometimes in the transition between regular grey grout and segregated grout, where a hardened light grey colour grout with small black spots was found with weak mechanical properties. These reports confirmed that these corrosion attacks took place only in the presence of the segregated grout that resulted in deep localised attacks, which resembled the form of pitting corrosion Figure 3. The investigations estimated that this phenomenon produces corrosion rates of the order of several millimetres per year. Because it is related to segregated grout, this corrosion under low-oxygen condition was usually found only in elevated parts of the tendons, near their ends.

[53] systematically tried to find the corrosion mechanism, arguing that the usual causes of steel corrosion in concrete cannot explain it. Chloride-induced corrosion is discarded because of the absence of chloride ions. Neither can this corrosion be related to carbonation, because of the alkaline nature of the segregated grout as well as the extremely localised morphology of the corrosion attacks. Also stress corrosion cracking, hydrogen embrittlement and even stray currents could be rejected as possible mechanisms in the reported experiments. Then [53] proceeds and proposes a possible mechanism to explain this corrosion phenomenon. They suggest that conditions of low availability of oxygen in the interstices among wires of the strands can be responsible for the corrosion initiation. The subsequent high rate of corrosion propagation can be attributed to macrocells that are generated at the segregated grout. The highly electrically conductive whitish paste connects the small de-passivated areas and the surrounding passive areas of the steel.

Figure 2: Example of whitish segregated grout embedding corroding strands [52]

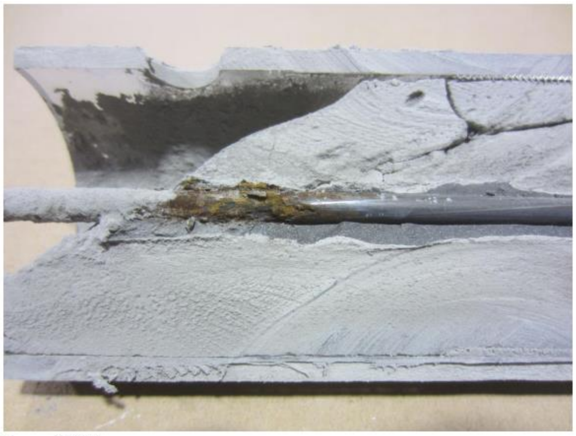
Figure 3: Example of corrosion attacks on a prestressing strand in contact with the whitish segregated grout [52]

There exist few studies on corrosion of PT, but numbers are even smaller for works that try to predict the onset of corrosion with formulation based on empirical data. This is why the efforts towards that goal reported in [54] and [33] are highlighted next. The investigation is divided in two different experimental phases. In Phase 1 [54] full-scale accelerated corrosion tests are performed with emphasis on the minimum chloride concentration needed to induce corrosion (chloride threshold), for normal grout conditions. The results concluded that there are two critical threshold values. For 0.04% of chloride concentration by weight of cement, small number of pits start to form. For threshold values 0.08% corrosion intensifies in terms of number of pits and pit depth. The report concludes that corrosion should also start below the indicated threshold values in other unfavourable conditions, such as: carbonated grout, segregated grout, duct cracks, voids filled with water or free sulphate ions in contact with the strands. In Phase 2 [33], several full-scale accelerated corrosion tests were performed for different grout materials, stress conditions, sulphate concentrations, recharging water pH and temperature. But most importantly, four conditions of grout were explored, namely, normally hardened, segregated, normally hardened admixed with sulphate and segregated grout admixed with sulphate. Figure 4 shows the schematic representation of the specimens used in the empirical test. Based on the results from the experiments, [33] proposes a methodology to forecast the onset and subsequent rate of wire, strand, and tendon fractures. The formulation factors in the extend of grout and the system deficiencies. The main inputs are wire corrosion rate and residual wire fracture strength in terms of their mean and standard deviation. The results show that corrosion rate increases in time up to a point beyond which this rate moderates. Also, that all corrosion damage was found at or near the void/grout interface, as shown in Figure 5.



Source: FHWA.

Figure 4: Schematic of test specimen containing an artificial void [33]



Source: FHWA.

Figure 5: Corroded wire that was exposed to segregated grout [33]

Therefore, corrosion will occur near voids due to the presence of water and air, or segregated grout, or both. Once corrosion has started in active steel, the corrosion rate is frequently larger than in passive reinforcement. The rate depends not only on the chemical conditions but also on the amount of stress. Corrosion will grow perpendicular to the main stress that in this case is along the cable cross-section. Corrosion results in reduction of cross-section area, which produces stress concentrations that can lead faster to plastic deformations of the steel, changing its structural behaviour and reducing its capacity to repair the oxidation film.

2.3. NDT for post-tensioning

In ordinary reinforced concrete systems, corrosion can usually be identified by visual staining, cracking, or spalling of the concrete cover. However, in PT systems these indications are rarely present. One needs to detect the damage using NDT techniques that to some extent can be used to evaluate the condition of the PT system. The NPRA older guideline from 2014 [55] suggested the use of ultrasound and x-rays. Then locations with suspected voids should be explored by direct observation with boreholes and endoscopes.

Since then, some additional research projects have been performed regarding the use of NDTs on PT structures, most notably in [56]. This comprehensive study and its complementary extended version [47] evaluate all the relevant NDT methods for PT systems. This study addresses the performance of each technique for different types of damage, which included: corrosion, section loss, breakage, grout conditions, voids, water infiltration, and tendon deterioration at the anchorage. The methods investigated in the report are listed in Table 3. The study applies these techniques to detect the different forms of damage that are broadly classified into strand defects and grout defects. The study is repeated for external and internal PT systems. In general, some of the methods show satisfactory levels of damage detection accuracy for external tendons. However, the results are worse when applied to internal tendons. None of the methods can detect defects in the strands. Whereas grout defects could be identified with low to medium accuracy using Impact Echo and Ultrasonic Echo.

Table 3: NDT methods investigated in [56]

Category by phiscal principle	**Name of method**
Electromagnetic	Ground Penetrating Radar (GPR)
	Infrared Thermography (IRT)
	Electrical Capacitance Tomography (ECT)
Magnetic	Magnetic Flux Leakage (MFL)
	Magnetic Main Flux method (MMFM)
Mechanic waves and vibrations	Impact Echo (IE)
	Ultrasonic Tomography (UST)
	Ultrasonic Echo (USE)
	Sonic/Ultrasonic Pulse Velocity (S/UPV)
	Low Frequency Ultrasound (LFUT)
	Sounding
Visual	Vistual Testing (VT)
	Borescope
Electrochemical	Electrochemical Impedance Spectroscopy (EIS)

A more recent publication by NPRA from 2021 [15] extended to Norwegian conditions the investigation to identify the most suitable NDT. This study mostly focused on methods to detect voids in the ducts and methods that indirectly can identify corrosion or tendon breakage. After reviewing all relevant NDT methods, it selected the most promising methods according to a combination of handling, cost, and accuracy criteria. Then, the study decided to proceed with further experimentation using four techniques, namely, impact echo, ultrasonic pulse echo, ground penetrating radar and cover meter. These techniques were experimentally tested on three mock-ups that consisted of a short T-beam and a small slab element with various PT conditions. These methods were then subsequently applied to four different bridges, concluding that it is possible to identify voids and define their size. However, more experience is required with additional field and laboratory tests, to improve the determination of the degree of grouting and learn optimum strategies to combine NDT with partly destructive testing. The same conclusions are reported in the inspection manual [57].

Moreover, as highlighted in [20] infrared thermography is a suitable NDT method that was not considered further in the Norwegian case studies [15, 57] This technology can detect grout deficiencies and voids within ducts by evaluating changes in the temperature distribution. This technique is relatively easy compared to others and requires only an infrared camera and the application of a thermal gradient. However, according to [56] this is only a valid approach at thin elements and not useful when ducts are deep into concrete. In addition, it is known and shown in [58, 59] that radiography can be used to detect voids in internal ducts but is limited by the health risks and safety precautions needed for its operation.

2.4. Existing guidelines and recommendations

The management of existing PT bridges, which includes inspection planning and condition assessment, is a difficult task that lacks established and practical guidelines. As an example, a study in the US [20] surveyed the state departments of transportation about their maintenance and repair practices for PT bridges. The results highlight that the quality of the maintenance highly depends on the expertise of the coordinating agency, consultants, and the contractors involved in the repair, because of the lack of a national guidance document. Therefore, this review starts listing the codes and guidelines relevant to maintenance of existing structures in general and continues reporting those documents targeted for PT bridges in different countries.

2.4.1. International documents

There exist several international standards regarding structural maintenance. ISO 13822 [60] is for the assessment of existing structures in general. It is a statement of the principals and procedures but also a guide for structural engineers. The document highlights that structural assessment is a major engineering task that requires knowledge beyond the scope of the design codes. In summary it states that "The structure shall be analysed for the ultimate limit states and serviceability limit states, using the basic variables and taking account of relevant deterioration processes.". Then, ISO 16311 [61] is specific for concrete structures. It covers all the aspects of the activities associated with the maintenance and repair of concrete structures, namely: maintenance plan, assessment process, undertaking remedial action and information recording. Part 1 provides a generic framework for structural maintenance. Part 2 focuses on the structural assessment and systematically defines the full process which includes: inspection and investigation, prediction of deterioration, evaluation of performance, decision-making activities. It indicates that time-dependent deterioration methods should be used to study the initiation of corrosion following the methods in ISO 16204 [62].

2.4.2. United States

The document [56] provides recommendations on condition assessment of PT systems from the US perspective. It divides the inspection into three tiers, each with increasing level of detailing of the inspection. Tier 1 is typical routine inspection that can only provide an indication on the existence of damage. Tier 2 is a component condition inspection (with destructive and non-destructive methods) providing indication on the localization and severity of the damage. Tier 3 consists of material inspections that result in an estimate of the remaining service life of the structure. The guideline provides then a systematic approach for choosing the optimal NDT starting from the known condition. The procedure provides a ranked list of methods. The final score for each NDT is calculated using weighted categories, which include precision, accuracy, ease of use, inspection requirements and cost. This methodology is presented in a flowchart form to facilitate the work to the decision-makers in choosing the most appropriate NDE technique for a given case.

Furthermore, a report from the Federal Highway Administration [36] suggests a guideline intended as a generalized course-of-action, acknowledging that it is flexible (can be adapted depending on the circumstances) and should be guided by engineering judgement. It defines a set of options and actions that depend on the observed chemical deficiencies (CD) and physical deficiencies (PD). Option 1 deals with situations where only chemical deficiencies are reported, whereas Option 2 is used when both, chemical and physical deficiencies are of concern. Each tendon is then assigned a grade from 1 to 10, where higher grades indicate more problematic tendon conditions. The final outcome of the proposed procedure is the action required, which are listed in Table 4. The procedure considers three main issues with grout, namely, high chloride concentration (grater that 0.08% of cement weight), voids (air pockets) and the presence of defect grout (wet and soft or segregated).

Table 4: List of actions suggested by the guideline in [36]

Action code	Action description
A1	None
A2	Expand sampling
A3	Reinspect in 5 years
A4	Reinspect in 2 years
A5	Tendon monitoring
A6	Consider repairing deficiency
A7	Structural evaluation/load rating
A8	Tendon replacement

More recently, the synthesis 562 [20], from the National Cooperative Highway Research Program, surveying current practices from each state reached the following discouraging conclusion: "Routine inspections are not sufficient in identifying issues in PT structures before they become severe". They find that in several examples no evidence of PT system damage was identified before the situation became critical. What is more, the authority in Florida specifically recommends the use of fully replaceable tendons that utilize flexible filler (unbonded with grease).

2.4.3. Japan

The Japan Highway Public Corporation investigated 120 PT concrete bridges in terms of durability. The investigation showed that 31% of the tendons had grout deficiencies (no grout, imperfect grout, and grout voids) [34, 63]. Some of those bridges were categorized as severely deteriorated. In addition, tendon corrosion, reinforcement corrosion, concrete cracks, and spalling were also found. Chloride attacks were found in bridges located in coastal areas. The analysis of the results showed that deterioration was inversely proportional to the distance to the nearest coastline. As a results of these findings, the use of grouted PT systems has been banned since 1999 [64] by the road highway authority in Japan. Therefore, most of the new prestressed constructions use external tendons [65].

2.4.4. Norway

In Norway, there exist two recent documents published by NPRA. The document [15] is based on the international standard for general principles for maintenance, repair and upgrading [61]. It states that any assessment should start first by identifying critical locations based on existing documents of the project. It divides the assessment methods into inspection and monitoring. The document identifies the two main issues relevant for Norwegian PT bridges. First there are problems in the PT systems because of lack of third-party control of the tension systems during installation, in particular regarding the duct location, integrity of tendons and applied prestressing force. The second issue is poor grouting operations and filler composition that results in probable frequent distribution of air-filled voids and trapped water.

The inspection manual [57] provides tools for planning, executing, and evaluating inspections in PT bridges with the use of NDT focusing on void location and poor grouting in the ducts. The procedure described by the guideline to perform the on-site investigations has following steps:

1) Locate reinforcement with cover meter.
2) Locate ducts using ground penetrating radar.
3) Locate voids in ducts using ultrasound pulse echo.
4) Confirmation of void with impact echo.
5) Verification of result by drilling for visual inspection and core drilling for material sample testing.

2.4.5. United Kingdom

The Design Manual for Roads and Bridges (DMRB) are a series of documents published by the road authorities from the UK. The generic principles for inspection and reporting for highway structures are described in [66]. The proposed assessment framework to decide on the required inspection frequency is based on risk analysis supported by engineering judgement. The risk score is calculated depending on several factors, namely, the likelihood of an event (defect occurring, structural failure), its potential consequences, the ability to detect a failure and the ability to control the outcome.

Then, the guideline [16] specifically addresses the management of PT concrete bridges. The intention of this guideline is to be able to categorize all PT bridges in the network using the same framework in order to define a prioritisation of interventions according to the risk of failure. This is done evaluating the risk of failure in three stages, each summarized below:

- Risk review: It is a desk study using available information that will conclude if risk assessment and management needs to be implemented. Also, during this stage the structure's critical sections (and points) must be identified. In here, critical sections are defined as those sections where there is "risk from ingress of moist air, water and contaminants that could initiate corrosion of the post-tensioning system". In addition, regions where voids are likely to occur or where yields points could form in a collapse mechanism should also be considered as critical.
- Risk assessment: This stage is used to identify the hazards relevant to the PT system, where a hazard is "anything that can adversely affect the condition of the structure's post-tensioning system(s)".
- Risk management: Based on the conclusion from previous stages it describes what activity is necessary. The framework specifies three categories for the possible activities: 1) inspections (with different levels of detail), 2) assessment (structural analysis and load assessment), 3) remediation (maintenance, repair, strengthening or replacement).

An overview of the main steps in the risk assessment is outlined below, indicating only the most relevant details for the project. For full description refer to [16].

- Review the structural form and categorise it in terms of brittle failure mode (very high to very low) using Table 5.
- Identify the vulnerable details and material hazards using Table 6.

- Determine the condition hazards using Table 7.
- For each hazard identified, the following shall be stated:
    1) The risk event (what could happen if the hazard is not dealt with)
    2) The likelihood of the risk event occurring
    3) The consequences of the risk event occurring
    4) The hazard risk level
    5) The recommended risk management measure(s)

Table 5: Bridge from and risk of brittle failure mode [16]

Bridge form	Risk of brittle failure mode
Segmental	
Beams or box girders, simply supported, non-composite, transverse joints, longitudinal post-tensioning	Very high
Beam grillage, simply supported, non-composite, longitudinal and transverse joints and post-tensioning	High
Beams or box girders, simply supported with a composite slab, transverse joints and longitudinal post-tensioning	Medium
Continuous beams, box girders or portals, non-composite, transverse joints and longitudinal post-tensioning	Medium
Continuous beams or box girders or portals with a composite slab, transverse joints and longitudinal post-tensioning	Low
Monolithic	
Beams, simply supported, non-composite, longitudinal post-tensioning	Medium
Beams, continuous, non-composite, longitudinal post-tensioning	Medium
Beam or box girder, simply supported, composite, longitudinal or longitudinal and transverse post-tensioning	Low
Beam or box girder, continuous, composite, longitudinal or longitudinal and transverse post-tensioning	Low
Beams, simply supported or continuous, non-composite, transverse post-tensioning	Very low
Solid or voided slab, simply supported or continuous, longitudinal and/or transverse post-tensioning	Very low
Post-tensioned substructure	
Bridges with ties between supports	Very high
Segmental construction, where the post-tensioning arrangement is designed to provide resistance to primary bending	Very high
Monolithic construction, where the post-tensioning arrangement is designed to provide resistance to primary bending	High
Segmental construction, where the post-tensioning arrangement is not designed to provide resistance to primary bending	Medium
Monolithic construction, where the post-tensioning arrangement is not designed to provide resistance to primary bending	Low
Tie-downs	
Continuous bridges, cantilevers and suspended spans on half-joints, with anchor spans tied down for;	
Dead load and live load	Very high
Live load only	High

Table 6: Vulnerable details and material hazards [16]

Segmental joints	In descending order of vulnerability: 1) narrow in situ mortar (typically <25mm); 2) wide in situ mortar or concrete (typically >100mm); 3) match cast glued.
Other joints	1) construction joints intersecting anchorages, couplers or tendons/ducts; 2) half-joints; 3) hinges.
Post-tensioning system	1) lack of redundancy, e.g. small number of large tendons where a severe local defect can have a serious effect on strength; 2) design where failure of a single post-tensioned element could result in structural collapse; 3) tendons located close to the upper surface of the deck where failure of deck waterproofing may lead to corrosion and loss of section; 4) use of spacers to separate post-tensioning wires making them vulnerable to crevice corrosion; 5) tendon ducts with hogging and sagging profiles with a vulnerability to void formation and water ponding; 6) unlined ducts; 7) tendons grouped together in a single duct; 8) tendons protected only by mortar or concrete; 9) unprotected anchorages; 10) anchorages concealed within joints or on upper surfaces of decks; 11) grout tubes exposed in top of deck.
Water management system	1) absent or malfunctioning drainage system; 2) absent or old deck waterproofing system; 3) absence of deck joint seals.
Materials and durability	1) low cover to reinforcement; 2) low concrete grade; 3) additives or admixtures containing chlorides used in concrete or grout; 4) insufficient longitudinal or shear reinforcement (as determined by a structural assessment); 5) reactive aggregates used in concrete.
Construction	1) grouting problems; 2) tensioning problems.

Table 7 (Part 1): Condition hazards [16]

Structure	
Cracking	1) in post-tensioned concrete elements - various locations, crack directions and causes.
Water management system	1) cracks and potholes in carriageway surfacing; 2) surface ponding on deck; 3) blocked drainage systems; 4) water trapped in boxes and other structure voids; 5) damaged or missing deck joint seals; 6) water leaks and staining on soffit and at movement joints.
Deflection	1) excessive deflection; 2) differential vertical deflection.
Concrete	1) delamination and/or spalling due to corrosion or freeze/thaw action; 2) spalling or crushing due to stress concentrations; 3) honeycombing; 4) evidence of chemical attack; 5) low cement contents.
Reinforcement corrosion	1) visual evidence; 2) evidence from corrosion testing.
Construction joints	1) leaks and stains.
Bearings	1) corrosion, deterioration and/or damage; 2) unexpected movement/rotation or failure to move/rotate as expected.
Post-tensioning system	

Table 7 (Part 2): Condition hazards [16]

Ducts	1) incorrect location; 2) corrosion; 3) perforation; 4) presence of water; 5) ungrouted.
Grout	1) cracked or shattered; 2) soft; 3) moist; 4) voids; 5) low cement contents; 6) high chloride and/or sulfate contents.
Tendons	1) incorrect type or size; 2) missing tendons; 3) corrosion resulting in loss of section; 4) fracture of wires/strands/tendons; 5) loss of prestress.
Anchorages	1) deterioration of capping material.; 2) moisture/water; 3) high chloride and/or sulfate contents in capping material; 4) corrosion; 5) missing anchorage components.

Another interesting comment within this framework concerns the voids in the duct. It provides a classification of the voids in quantitative terms, as shown in Table 8.

Table 8: Classification of voids [16]

Classification	**Description**
No void	Fully grouted
Small void	<25% of section, <50mm length
Medium void	25-50% of section, 50-300mm length
Large void	>50% of section, >300mm length
Ungrouted	No evidence of grout

Therefore, the guideline [16] provides a systematization of the assessment procedures for PT structures in terms of risk evaluation. The document also includes examples of determination of critical sections, risk assessment and prioritisation. The whole guideline is important, but arguably the most relevant information is provided in Table 5, which classifies each structural type in terms of its risk of brittle failure. It shows that the bridges with the highest risk are segmental bridges made of beams or box girders in simply supported configurations.

3. Numerical modelling

During this project several modelling options were considered. Early in the project, some case-studies were performed using NovaFrame [67]. This is a useful design tool for bridge engineers but showed some limitations in the level of detail for modelling problems in PT systems. Then Abaqus [68] was used, which is a powerful FEM software that allows modelling virtually anything. But the learning curve for the students on this software is slow and so they could not acquire sufficient skills in this software to explore the problem at hand in all its detail. Therefore, it was decided to use DIANA [69], an alternative FEM software targeted to the analysis of concrete structures. This software has multiple in-build procedures to model active and passive reinforcement in concrete and offers a wide range of material modelling options. Additionally, it allows for the generation of models via a programmatic way written in Python programming language [70].

Nevertheless, there exist a series of common strategies and recommendations, applicable to FEM in general. In particular, for the correct modelling of concrete structures, a recent guideline has been published [71] that provides specific recommendations. Some of the most relevant recommendations are reproduced next:

- Material properties: Use the values specified in the Eurocodes. For properties not defined there, use those listed in the FIB Model Code 2010 [30]. The guideline also provides examples of typical values of concrete, reinforcing steel and prestressing steel.
- Concrete modelling: Use a total strain-based rotating crack model. Represent the tensile behaviour with an exponential softening diagram. The parabolic stress strain diagram with a softening branch should be used to model the compressive behaviour.
- Steel modelling: For both, active and passive reinforcement, use an elasto-plastic material model with hardening.
- Elements: Employ elements with quadratic interpolation of the displacement field. Their shape should preferably be quadrilateral and hexahedral, for 2D and 3D models respectively.
- Solver: Achieve equilibrium between internal and external forces in each iteration using a Newton-Raphson method with an arc-length procedure.
- Convergence: Define a suitable convergence criterion for determining equilibrium. Preferably use an energy-norm together with a force-norm. Avoid using a norm based on displacements only. Suggested tolerance values are 0.01 for unbalance force and 0.001 for energy norm.

3.1. DIANA modelling and framework

The numerical models in this project have been developed to correctly model damaged PT systems, where damage is defined as voids at certain locations along the tendon and/or breakage. To represent this very particular problem, specific strategies were required that introduced various levels of complexity. In addition to the appropriate loading sequence needed to describe a PT structure, it was necessary to represent the interaction between steel and concrete (the bond) using non-linear bond-slip relationships. This section summarizes the required modelling procedures and presents a programmatic framework to build such models.

To model a PT structure using DIANA [69] it is necessary to define a load sequence that resembles the actual construction process. Different loads and material properties need to be activated and deactivated in a precise order, which is implement in the software using a phased analysis. These phases are:

- Phase 1: Apply the loads (self-weight, prestressing, and reaction forces) and assign unbonded material properties to the bond. In this initial phase, the steel and the concrete are not bonded together yet. This is achieved by defining a bond-slip relation with stiffness value close to zero. In addition, if only the prestressing load is applied to the steel there would be no direct transmission of the axial force to the concrete. Therefore, corresponding reaction forces must be applied at the anchorages.
- Phase 2: Apply tyings between tendon ends and anchorage points, change the material properties of the bond to bonded, and deactivate the prestressing and reaction forces. This phase ensures that steel and concrete are working together by defining the bond between them.
- Phase 3: Apply additional external loads

Appendix A includes all the steps necessary to generate a model of a PT beam with bond-slip reinforcement for the active steel. This step-by-step definition is not intended as a tutorial but rather as a recipe. It condenses all the relevant steps and highlights important aspects needed for the correct development of such a model. The described steps can be used to construct the model and configure the analysis directly using the graphical user interface of DIANA.

However, in order to take full advantage of the software capabilities, the modelling can also be performed in a programmatic manner by creating the correct sequence of commands using Python language [70]. Precisely this was developed during this project and is called the framework. This framework consists of a script that allows for the generation of a 3D beam with PT tendons and corresponding analysis, which can be directly used in the DIANA environment. It defines all the necessary sequence of loading phases, activation of material properties and changes in loading necessary to simulate a PT system. The framework allows for a rather flexible definition of parameters and options, permitting the implementation of a wide range of beam and prestressing configurations. Some of the most relevant modelling options are listed next:

- Dimensions and structural configuration of the beam
- Number of tendons, location, and other related properties.
- Number and type of external loads.
- Division of the tendons into smaller segments for independent assignment of bond properties. This allows for the definition of particular voids at a given tendon number and segment.
- Modelling of passive reinforcement, either longitudinal or shear links with uniform spacing between them.
- Definition of plates for anchorage, supports and loading.
- Definition of material interfaces between plates and concrete beam.
- Reduce the model by symmetry considerations, either to half-model or quarter-model.

The full script of the framework is included in the Appendix B and consists of three main parts. The user of this script has to access only the second part, where the particular project properties and options can be chosen. After that, the file can be loaded (or simply copy and pasted) into the DIANA environment. The three parts are summarized below:

1) Initial commands and definitions. These are some housekeeping commands that ensure that DIANA runs the model and that it has the appropriate Python libraries. Also, some important variables are initialized. The user does not need to modify this part.
2) Particular model definitions. This is the part of the script where the user must define the properties of the project to study. Among others, the user can specify the beam geometry, number of tendons, loads, and more. All the modelling options mentioned earlier can be selected here. This part of the script is rather user friendly with comments and indications to help the user to specifying the desired PT beam configuration.
3) Main script. The actual part of the script that generates the model and executes the analysis. It uses all the information defined earlier in the script. By means of a series of loops and conditionals, this part defines all the sequence of commands needed to generate the desired model and analysis on DIANA software. The user does not require to edit this part.

This script presented in Appendix B can be considered as the starting point for future developments. Modifications of the existing framework can be performed to obtain more tailored models with additional levels of complexity. Moreover, one can use only parts of the script to facilitate the implementation of certain aspects of the model, while the rest of the modelling and analysis is performed directly using the graphical user interface for DIANA. For further information and practical details on how to use DIANA, model PT systems and use python scripting refer to other documents generated during this project. More precisely, the report [72] offers an extended description on how to model PT systems with DIANA, with special emphasis on the bond modelling using blond-slip relationships. The thesis [73] discusses and illustrates the workflow to generate beam models in DIANA taking advantage of Python scripting. Finally, the thesis [74] is an extended explanation of the framework given in Appendix B, together with an analysis of the different modelling strategies available to study damaged PT systems.

3.2. Validation results

The presented numerical modelling strategy was validated to ensure that the numerical model delivers correct results that replicate the behaviour of real PT structures. In [8] a series of simply supported PT beams are modelled to explore all the options and parameters to correctly represent the problem. The numerical models are then used to reproduce the reported behaviour from published empirical results of real PT beams. The numerical results showed acceptable matches with laboratory data, thus confirming that the adopted modelling is correct. Subsequently, this model was used to study the difference in behaviour between bonded and unbonded prestressing. Figure 6 shows the cracks patterns in four-point bending test for the load step when the steel reaches its yield point. The bonded beam shows a narrowly distributed pattern of small crack withs, whereas the unbonded beam features a smaller number of large cracks. The modelling resulted in the well-known behavioural difference between both cases.

a)

b)

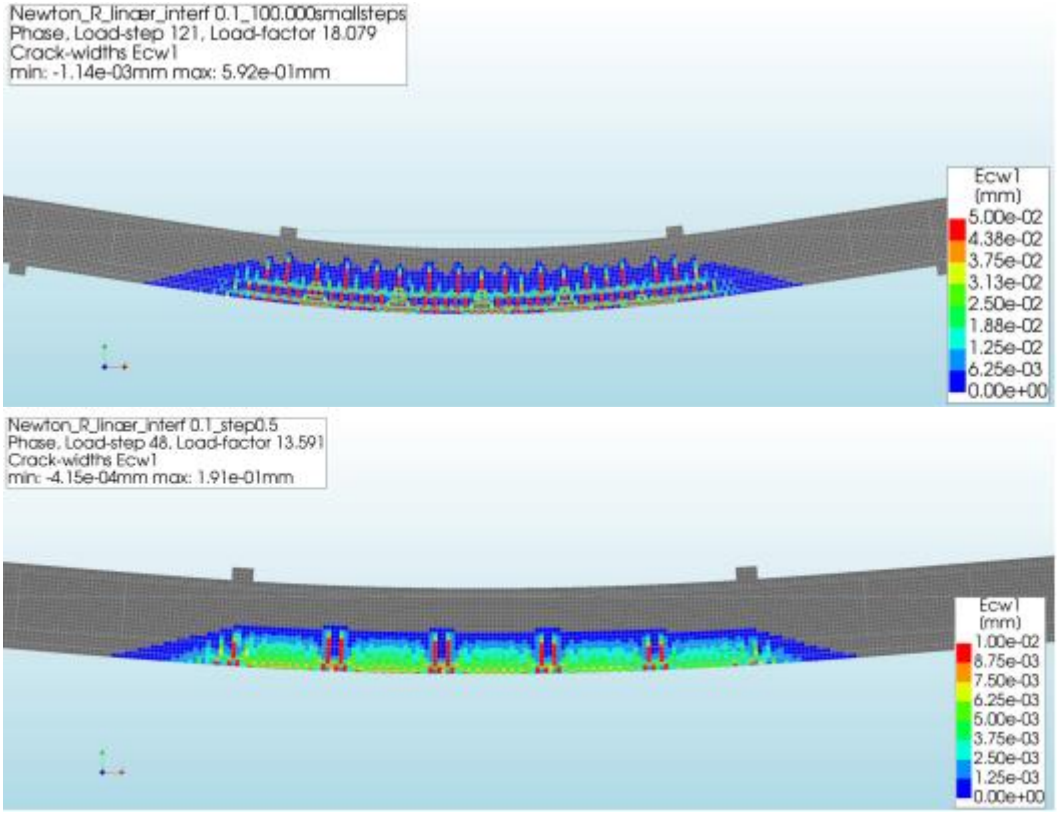


Figure 6: Crack widths at the tendon yielding points for: a) grouted beam; b) ungrouted beam

4. Thesis summaries

This section reports the works related to the case studies investigations performed during the duration of the project. These studies have been the main task for master students at NTNU that resulted in the production of various master thesis. All the case studies are of PT bridges and explored the effect of possible damages in the PT with a variety of approaches and modelling alternatives. This section provides a summary of these works. Each case study is presented in a separate subsection titled with the name of the bridge or bridge typology under investigation. Each subsection introduces the structure under investigation, describes the adopted methodology and provides a summary of the most relevant results and conclusions.

4.1. Osstrupen

Thesis title	*Analyse av FFB-bru med korrosjonsskadet spennarmering. Tilstandsvurdering og kapasitetskontroll av Osstrupen bru*
Authors	M. Aasheim and L. Hangaard
Reference	[76]

The Osstrupen bridge is a cantilever bridge (*fritt frambyggbru*) located in Flora municipality. The bridge was built in 1976 with a span of 198 m. It consists of two cantilever beams partially balanced and fixed with ballast and anchorages at the abutments. The two cantilever parts are connected at mid-span with a joint. The variable cross-section includes PT systems of the type BBRV cased in ducts of 87 mm in diameter injected with cement mortar. Each tendon is made of 56 wires of 6 mm diameter, resulting in 1583 mm^2 of steel area. The tendons are anchored with a solution that fixes each wire individually. Figure 7 provides an overview of bridge.

The bridge was designed and built considering small cover and therefore presented extended corrosion damages on the deck. The small concrete cover was selected following the construction requirements at the time, which was 25 mm according to NS 3473:1973 [75]. Compared to previous practice in Norway or other countries, the concrete cover was low and particularly low for bridge decks. According to revised calculations the cover following current valid specifications should be 50 mm inside the cross-section and 145 mm for the bridge deck. During inspections of the existing bridge, covers as low as 19 mm were observed, and some places even had exposed reinforcement.

Furthermore, the bridge presented excessive vertical deformations at mid-span due to eventual development of larger creep and shrinkage as originally assumed during its design. The bridge has undergone an important reparation in which an auxiliary steel structure was constructed inside the concrete bridge to lift the mid-span and increase the capacity of the existing structure. Note, that this study was performed with the information and conditions before this extensive reparation.

a)

b)

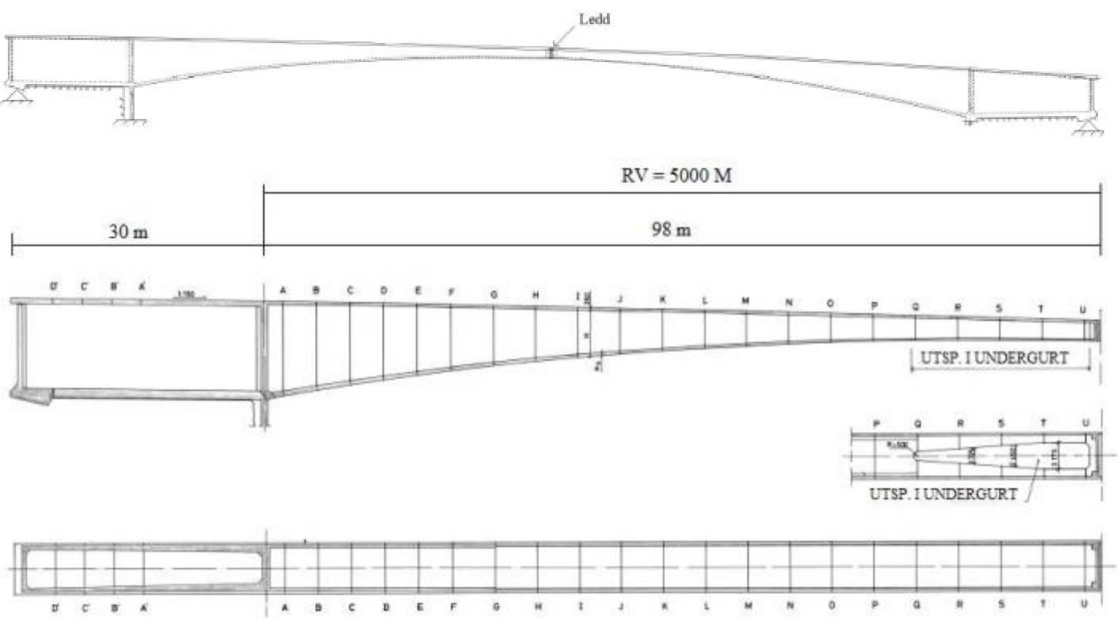


Figure 7: Osstrupen bridge; a) overview and statical system; b) drawings of one cantilever half of the bridge

The master thesis [76] that investigated the Osstrupen bridge modelled it using NovaFrame software [67], which is a frame analysis tool based on beam element theory. The work performed a capacity evaluation of the structure based on current standards. Then, it reported the current state of the structure based on available reports and a site visit. Finally, the capacity of the structure was evaluated numerically for various damage scenarios in the PT system. It evaluated the number of tendon failures needed for the structure to fail and what are the critical failure locations. Figure 8 shows an overview of the developed numerical model and the modelled tendon layout.

a) b)

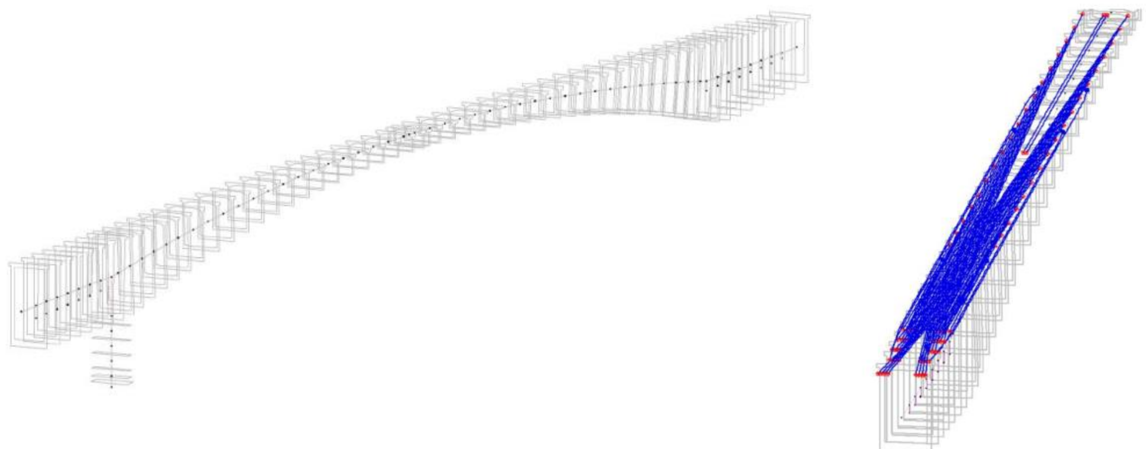


Figure 8: a) Overview of numerical model of Osstrupen in NovaFrame [67], b) Modelled tendon layout

The numerical model was verified by comparing the numerical results to hand-calculations of various limit states, namely, moment capacity and cracking moments at several cross-sections along the bridge. After satisfactory validation, the analysis showed that in its healthy state, the bridge has sufficient moment capacity with moment utilizations ranging between 87% and

94%. However, the calculations showed that cracks would appear near the abutments when loaded to the design traffic values. Elsewhere, the cracking capacity utilization decreased up to 87%.

Then the model is used to evaluate the effect of corrosion in the PT system. It conservatively assumes that corrosion leads to breakage of tendons and that one broken tendon becomes inactive along the whole structure. The study subsequently reduces the number of working tendons and keeps reducing the number until structural failure. In other words, the capacity of the structure is evaluated for different number of working tendons. This study is repeated at several sections. Analyses in the ULS indicate the maximum reduction of tendons that provides sufficient moment capacity. Studies in the SLS tensile explored the stresses in concrete with respect to cracking.

The results (Figure 9) indicate that the moment capacity is barely sufficient when three tendons are removed, and the bridge could collapse if four cables are broken. The cross-section nearest to the abutments must keep 38 of 42 tendons intact, or in other words, it could not withstand a 10% area loss of prestressing steel.

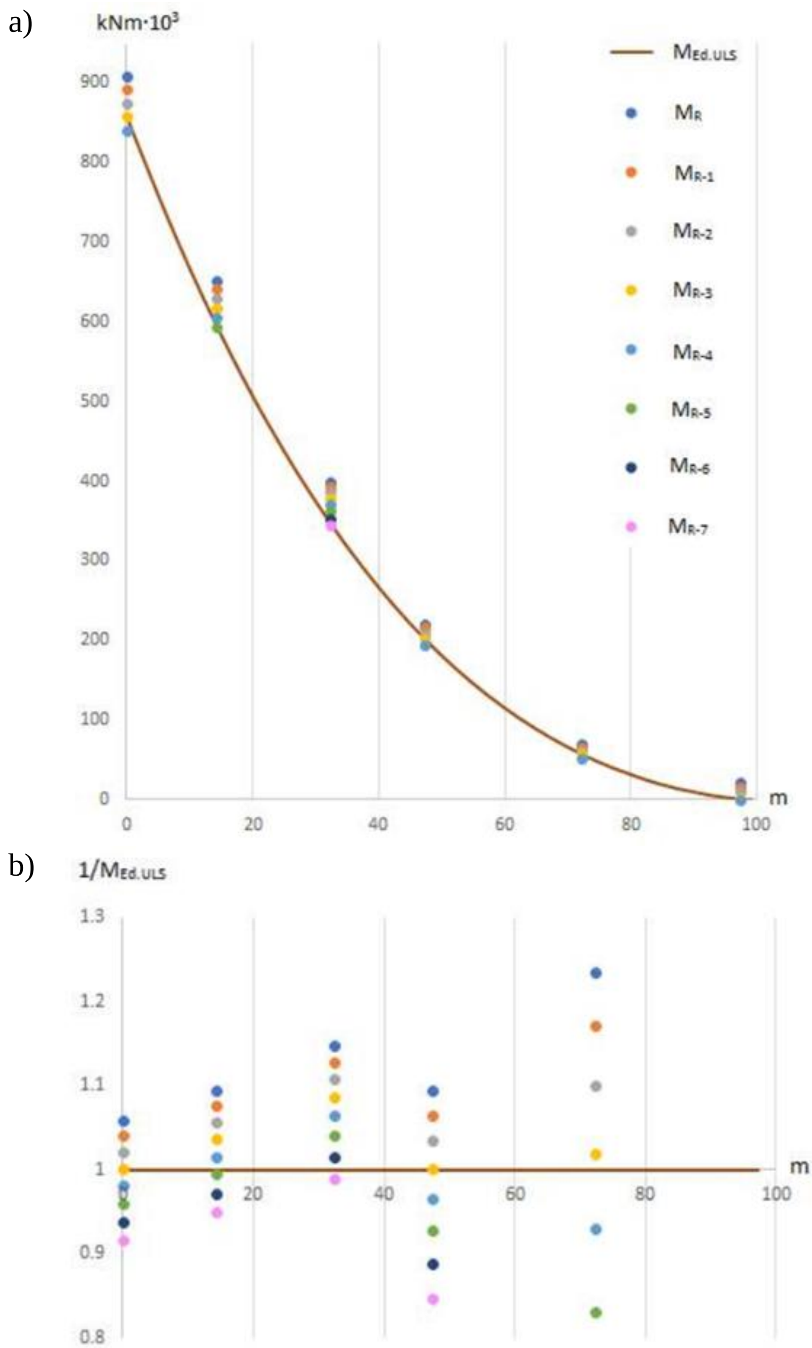


Figure 9: Moment capacity of cantilever after progressive tendon breakages. a) Calculated capacities compared to required design moment capacity; b) Capacities in relative terms

Moreover, the results from the stress analyses show that the bridge will crack for smaller number of tendon failures. This indicates that it would be possible to discover cracks before the moment capacity is exceeded. However, cracks widths of 0,3-0,6 mm are expected before surpassing the moment capacity, which would appear on deck (section's top fibre) below the road pavement. Figure 10 compares the number of tendon breakages at different sections along the bridge needed to fail certain limit state. Three different limit states are considered, namely, ULS (bending), SLS (crack width), and serviceability without traffic loading.

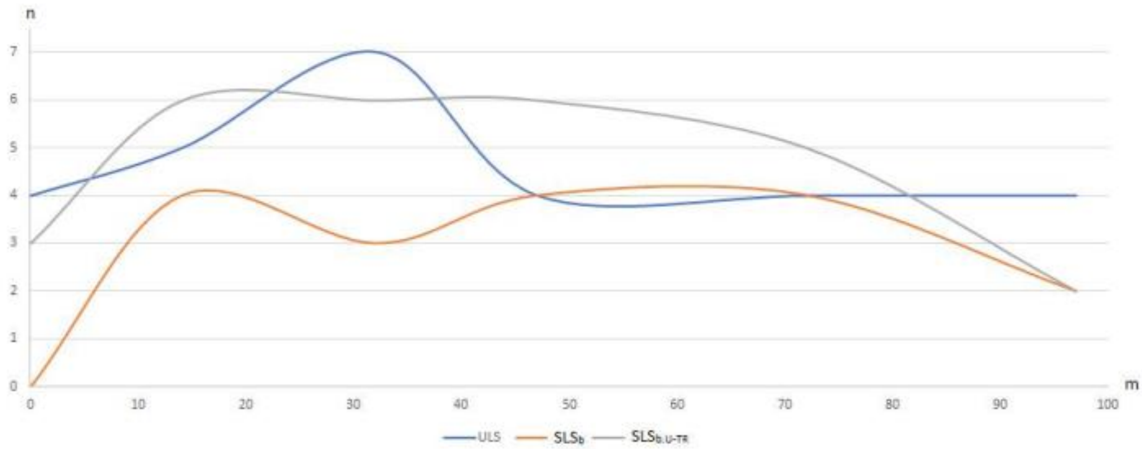


Figure 10: Number of broken tendons to reach certain limit state condition

This study concluded that the structure could tolerate at least 4 tendon failures before its structural safety is compromised. Also, that cracks will appear for fewer number of broken tendons. These cracks form at the top fibre of the cross-section, that is covered by the road pavement, and only inspections inside the bridge might be able to detect them. However, the occurrence of shear cracks or bending cracks corresponds to such a reduction in the prestressing force that the safety margin with respect to the collapse is low.

4.2. Kollstraumen

Thesis title	*Styrkeberegning og analyse av eksisterende spennarmert buebru*
Authors	H. Amiri and E. Moen
Reference	[77]

Thesis title	*Kapasitetskontroll og analyse av etteroppspent bru med redusert spennarmering*
Authors	E. Ukvitne and H. Vangdal
Reference	[78]

The Kollstraumen bridge, shown in Figure 11, is a PT arch bridge with a main span of 85 m. The bridge built in 1971 located in the Bindal municipality is the only one of its kind in Norway. The arch is below the deck and is thus under tension, which is made of concrete, reinforced and loaded by a PT system. A damage of the arch can be critical for the structural stability because of the low level of redundancy and the nature of the adopted structural solution that does not allow for significant stress redistribution. Damage of the tendons may be critical for the structure's capacity. The bridge, therefore, lacks robustness against eventual damage and is prone to collapse.

Figure 11: The Kollstraumen bridge

Figure 12 shows the cross-section of the bridge, with the deck on the top and the arch below. The arch of the bridge is made of two small rectangular concrete sections, that includes active and passive reinforcement. Each has 13 tendons of the BBRV system type and are shown in Figure 13. Two different strands dimensions are used, either 22Ø6 or 44Ø6 in ducts of 45 mm or 60 mm in diameter respectively.

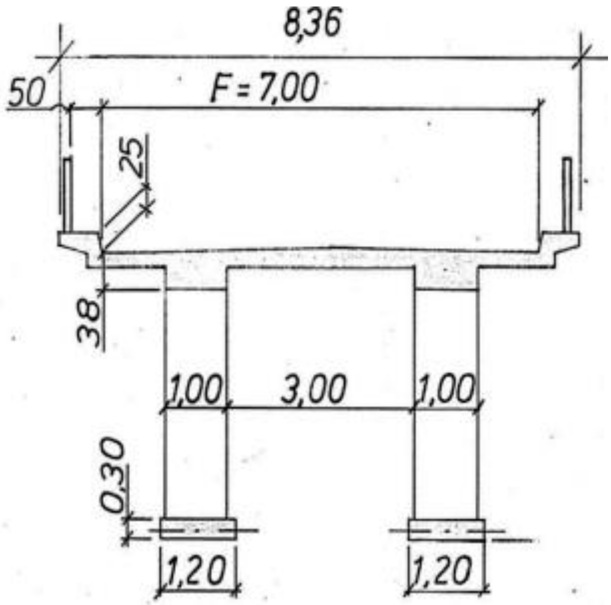


Figure 12: Bridge cross-section

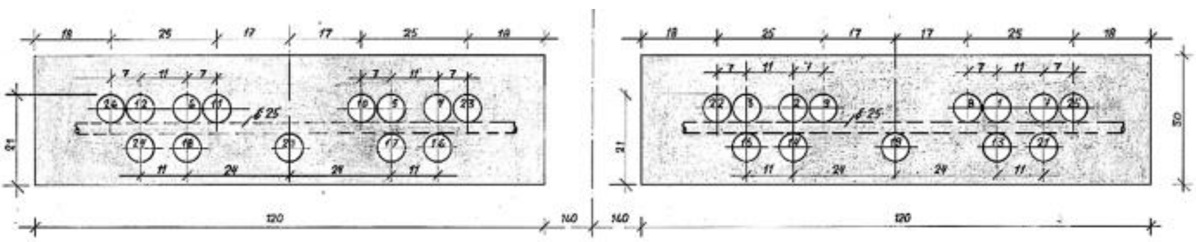


Figure 13: Cross section of post-tensioned arch elements

The tendons in the arch should have been fully grouted, which in theory should protect them from corrosion. However, there exist no records on the actual condition of the grout or the presence of voids in the ducts. Corrosion of the arch elements has been reported in recent visual inspections affecting the passive reinforcement. The design cover of the bridge was already low, but some of the measured covers are even smaller. Spalling and reinforcement corrosion have been registered from inspections in several areas on the entire bridge, as can be seen in Figure 14. In addition, a deflection of the main span of at least 250 mm and a transverse crack along the bridge deck above the pillar at the main span have been registered. These damages are considered not to be critical for the bridge's bearing capacity, assuming that the tensioning system is undamaged. It is uncertain whether this is the case, so an attempt has therefore been made to model the bridge's behaviour and capacity with reduced area of PT tendons.

Figure 14: Reported corrosion at the arch elements

This case study was investigated by two complementary master theses [77, 78], elaborated in successive years. The first study [77] focused on gathering information about the construction process of the bridge in order to develop a correct numerical model of this unconventional structural type. This was the basis to build a numerical model with NovaFrame [67], which was subsequently used to perform investigations in ULS and SLS. In [78] the bridge was modelled using Abaqus [68], which allowed to extend the analysis to include second order and non-linear effects. Figure 15 shows the final numerical model of the main span with the PT arch.

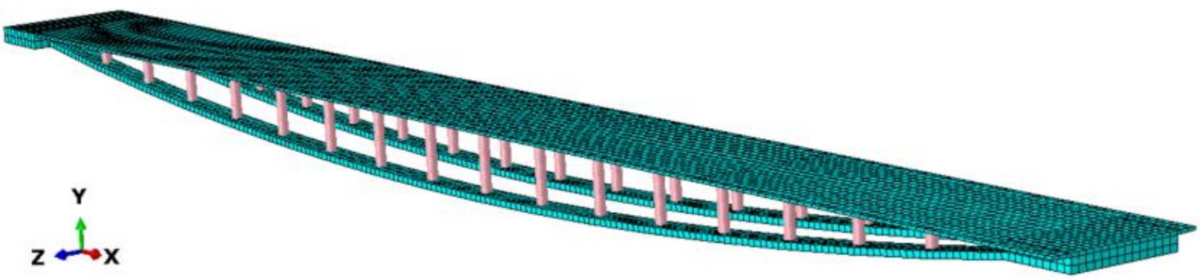


Figure 15: Numerical model of Kollstraumen bridge

The analysis in [77] concluded that, in the ULS, the moment capacity and shear capacity is adequate, while the axial utilization of the deck is exceeded, reaching 106% utilisation. However, it is acknowledged that the analysis was based on the cracked section assumption. The actual current stiffness of the structure will be bigger, resulting in smaller deformations and thus smaller moments, deeming the structure safe with respect to ULS. The SLS analysis showed that cracks will form but within the permissible width values. On the other hand, [78] an attempt was made to analyse the structure's behaviour with a reduced steel area of tendons. The document concludes that a possible failure in the construction will occur after the loss of four tendons in each of the arches. The analysis assumes that the tendons are fully grouted. The results also show that cracks will appear on the arches already after the breakage of two tendons.

4.3. Måløybru

Thesis title	*Modellering av etteroppspente betongbruer med korrosjonsskadet spennarmering*
Authors	H.R. Kvale and T.J. Opheim
Reference	[79]

The Måløy bridge, shown in Figure 16, is located in Vågsøy municipality. The construction of the bridge started in 1971 and opened to traffic in 1973. The bridge has 34 spans and a total length of 1224 m, with an average span of 30 m. However, the main spans are 125 m long which are achieved by cantilever bridge constructions. Each cantilever element is made of variable depth and has a box cross-section varying between 7 to 2 m in depth. Each of the main spans has a complex tendon layout with 23 tendons, as shown in Figure 17. Because of its location at the seaside, the bridge has shown corrosion damages.

Figure 16: Main spans of the Måløybru [80]

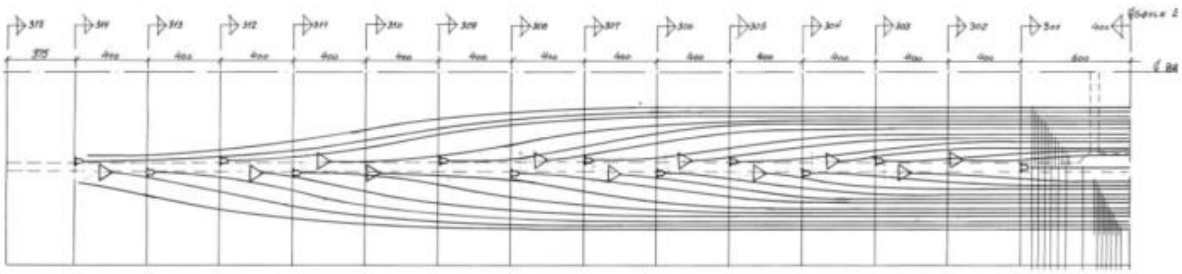


Figure 17: Top view of tendon layout of one half of the cantilever span

The work in [79] generated a numerical model of the bridge using Abaqus [68]. To adjust the computational requirements of the problem, the numerical model was reduced and considered some simplifications. More precisely, most of the analysis used a numerical model of on one half span of one of the main spans, i.e., only one cantilever beam. In order to validate this model reduction, some of the results were compared to an extended model including both sides of one of the main spans. Furthermore, to simplify the PT configuration, the number of modelled tendons was reduced while maintaining the total area of active steel present in the bridge. Figure 18 shows the reduced numerical model of the bridge with simplified tendon configuration. Passive reinforcement was included in the model (longitudinal and shear) programmatically defining an input file for Abaqus [68] generated with a Matlab [81] script. The model is then used to investigate the consequence of damaged PT. Various damage scenarios are simulated, namely, loss of bond between concrete and reinforcement, reduced tendon area and breakage of tendons.

Figure 18: PT tendons in numerical model

Severe loss in bond was modelled by removing all bonding along the whole tendons except for some region near the anchorage, effectively modelling an un-grouted system. Figure 19 shows that the differences in deformation are minimal indicating that bond loss alone has only marginal influence reducing the structural stiffness.

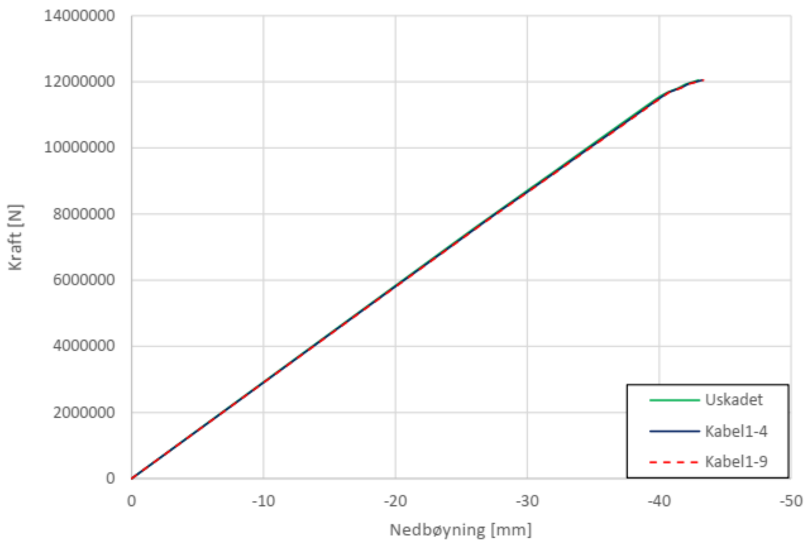


Figure 19: Effect of bond on load-displacement results

The simulation of loss of steel area was studied for a range of damage severities, locations of damage and number of affected cables. The results showed that depending on the damage scenario the local section might have a significant change in strain distribution leading to some cracking, and that the global stiffness and capacity would also be affected but to a smaller extent. Figure 20 shows the load-displacement curves for a damage scenario with uniform area loss across tendons but occurring at different sections across the bridge. The damage locations correspond to: 1) near the anchorage, 2) between 4-10 m from the anchorage, 3) between 12-17 m from the anchorage, 4) between 26-30 m from the anchorage. The results show a clear influence of the damage location on the magnitude of the cracking moment.

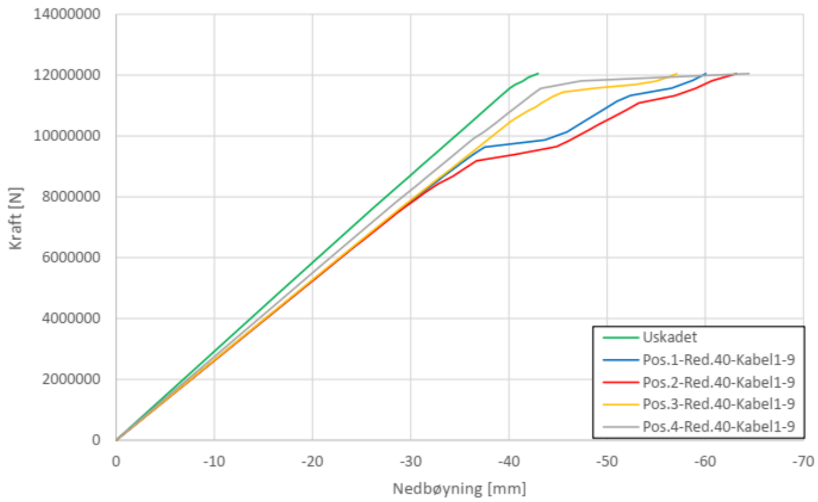


Figure 20: Load displacement result for 40% area loss at different locations for all tendons simultaneously

The general results of this work indicate that loss of bond in internal areas of the concrete has little or no effect on local and global effects. In the case of reduced tendon area and breakage of tendon, the stress-strain distribution locally changes in areas with corrosion damage, while the impact on the global behaviour is smaller. In the case of large enough damage, the concrete cracks up and gives a reduction in the stiffness of the bridge structure, due to both a shift in neutral axis and lower contribution to stiffness from concrete. Global effects are only important when the extent of the damage affects several tendons. The consequences of corrosion damage are found to be most critical in areas subjected to high bending moments, as well as areas where a damage in few tendons constitutes a large proportion of the prestress in a cross-section.

4.4. Generic beam bridge

Thesis title	*Non-linear behaviour of insufficiently grouted post-tensioned concrete members*
Authors	H. Vestad and M. Vestad
Reference	[8]

The intention in this section was to study the effect of damaged PT systems on beam bridges in general, rather than just for a particular bridge. After a thorough evaluation of the Norwegian bridge database BRUTUS, several beam bridges were gathered and compared in order to find the geometry and configuration that could be described as a typical beam bridge. Eventually, the modelled bridge was inspired by the Rossvollbru. This bridge is in Skaun municipality and is part of the E39 route. It consists of 4 spans with a maximum span length of 32 m. Figure 21 shows an overview of the bridge and the cross-section.

a)

b)

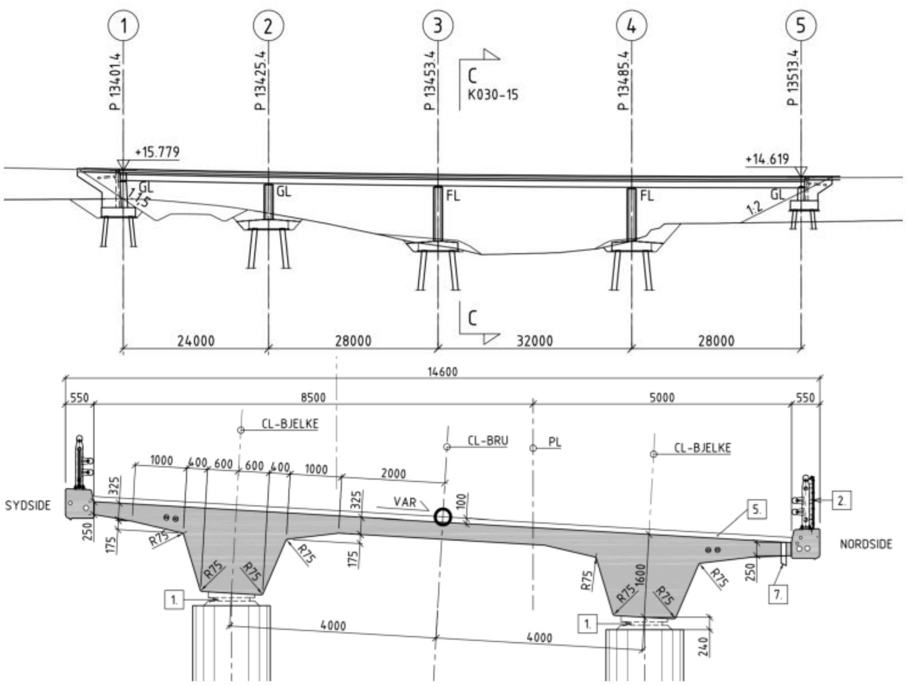


Figure 21: Rossvollbru; a) Overview of bridge; b) Cross-section

In the master thesis [8] the generic bridge was modelled using DIANA FEM program [69]. The work was divided in two parts. In the first part of this work, bending tests of simple PT beams were modelled and compared to published empirical results to ensure the validity of the modelling strategy. As mentioned already in Section 3.2, the work validated the modelling approach of PT systems using bond-slip reinforcement and the iterative solution of the non-linear problem. Subsequently, the modelling strategy and gained experience was exported to the evaluation of damaged PT bridges.

In the second part of [8], a numerical model of a generic beam bridge was developed inspired by the Rossvollbru. The final bridge model in DIANA shown in Figure 22 was adapted to simplify the modelling and gain generality. In this regard, the inclination of the section was disregarded. Also, symmetry considerations were adopted to further reduce the size of the model. The modelling parameters were again explored in parametric studies to ensure correct representation of the bond-slip behaviour of the PT system. In addition to the permanent loads, traffic loads were considered and placed in the least favourable position. Then the capacity of the bridge was evaluated by gradually increasing the traffic load. To control the results from the numerical models, analytical calculations were made. Lastly, the non-linear responses of a grouted and an ungrouted bridge model were compared.

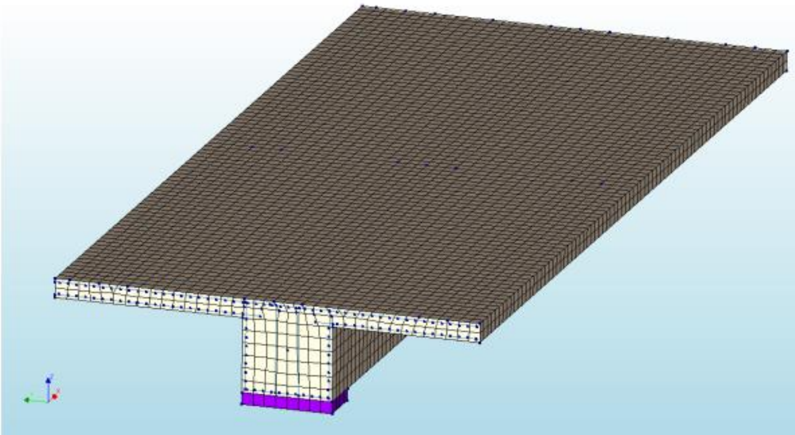


Figure 22: DIANA model of one quarter of a bridge span

The analysis of the PT bridge showed similar tendencies as reported in Section 3.2 for PT beams. The difference between the behaviour of a completely grouted tendon and a tendon without any grouting was small. Until the point of cracking, the two models had the same linear behaviour. For the non-linear part of the load-deflection curves, the ungrouted model had a slightly lower capacity, giving a 7,6 % reduction of ultimate load capacity for the beam models and a 1,8 % lower capacity for the bridge model. The crack pattern, when comparing the two cases, were very similar, but the results suggest that the cracks of an ungrouted concrete member are concentrated in groups, in contrast to the more evenly distributed cracks of a grouted concrete member.

The load displacement results for the same bridge but with different PT configurations are given in Figure 23. The results show that the difference in capacity between the grouted and the ungrouted condition are minimal, and only slightly smaller for the ungrouted case. The analysis also compared the results to a model where the active reinforcement is modelled using the embedded modelling approach. In this situation the bond between steel and concrete is perfect. Counterintuitively, the capacity for the embedded case is slightly smaller than for the other two. This indicates inherent difficulties in correct modelling of non-linear behaviour of PT systems in large structures. This was attributed to the need of further refinement in the values of the bond-slip parameters (stiffness parameters of the linear bond-slip description). However, the key is that the relative behaviour between bonded and unbonded was appropriate. In any case, the results show that the order of magnitude of all three alternatives are similar. In fact, they are very similar. There is no distinct feature that would allow us to identify in which situation we are simply by looking at one of them.

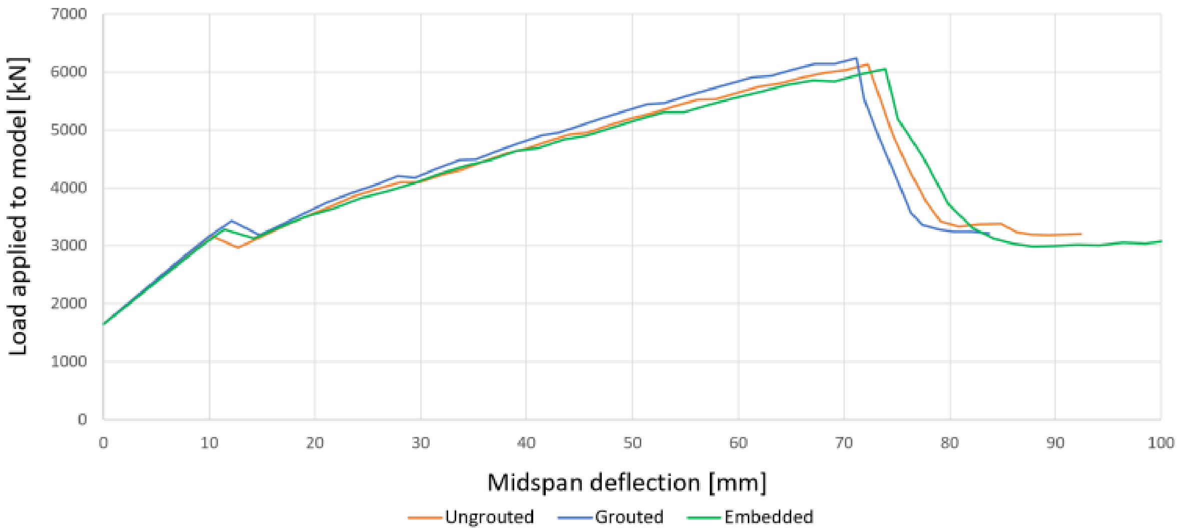
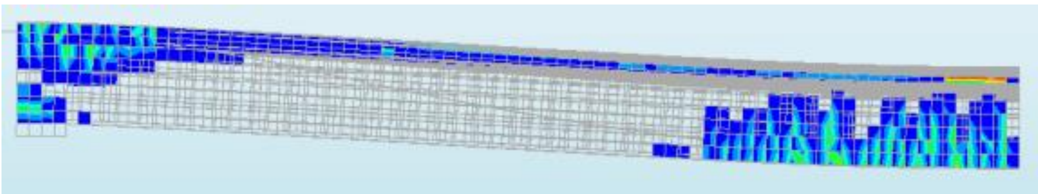


Figure 23: Load-displacement results for different modelling and grouting conditions

This is particularly evident when comparing the crack patterns of both the most extreme situations, as shown in Figure 24. The direct comparison of the crack patterns on the bridge for completely bonded compared to completely unbonded PT systems indicate some differences consistent with the results for PT beams. This is, the crack pattern for unbonded PT systems tends to have fewer cracks but with larger dimensions. However, this difference is of little use in practice for bridge inspections since there will be no reference case to compare with. Furthermore, these differences have been found assuming that all tendons are unbonded. However, the most likely real damage scenario will consist of a series of voids distributed randomly across the ducts of the bridge. It is expected then that the difference in crack pattern will be negligible compared to the fully bonded case.

a)

b)

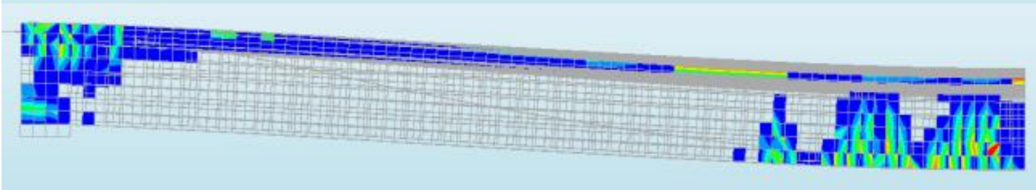



Figure 24: Cracking patters at model failure; a) Fully grouted; b) Ungrouted

Therefore, the main conclusion of [8] is that the non-linear analyses indicate that the structural effects due to lack of grouting are minimal. It was decided not to study partly grouted models, as these two extreme cases showed a very similar behaviour. Regarding the cracking behaviour of a fully grouted and an ungrouted system, the results until the crack initiation point and crack width developments were quite similar. However, the results still suggest that there is a difference in cracking pattern between the two cases. While the models with grouted tendons seem to have evenly distributed cracks, the ungrouted model had cracks that were assembled in groups. Despite this, it is unlikely that the cracking patterns for the two cases are characteristic enough to be used during a bridge inspection when the level of grouting is evaluated.

4.5. Generic box girder bridge

Thesis title	*Behavior of Post-Tensioned concrete box girder with multiple damaged tendons*
Authors	J.J. Czesak
Reference	[82]

The intention of this section was to study the consequences of damaged PT systems in generic box girder bridges. This was the main goal of the corresponding master thesis [82]. In this work a careful examination of multiple existing PT box girder bridges in Norway was done based on the information available in the BRUTUS database. Several bridges were compared in order to find the geometry and configuration that could be described as a typical box girder bridge. Eventually, it was decided that the Sykkylvsbrua (Figure 25) was deemed a representative case for this bridge typology. The bridge crosses the Sykkylvsfjorden in Sykkylven Municipality in Møre og Romsdal county. It was built in the year 2000 and has a total length of 860 m consisting of 15 spans with a maximum span of 60 m. The bridge has one traffic lane per direction and a sidewalk/bicycle path.

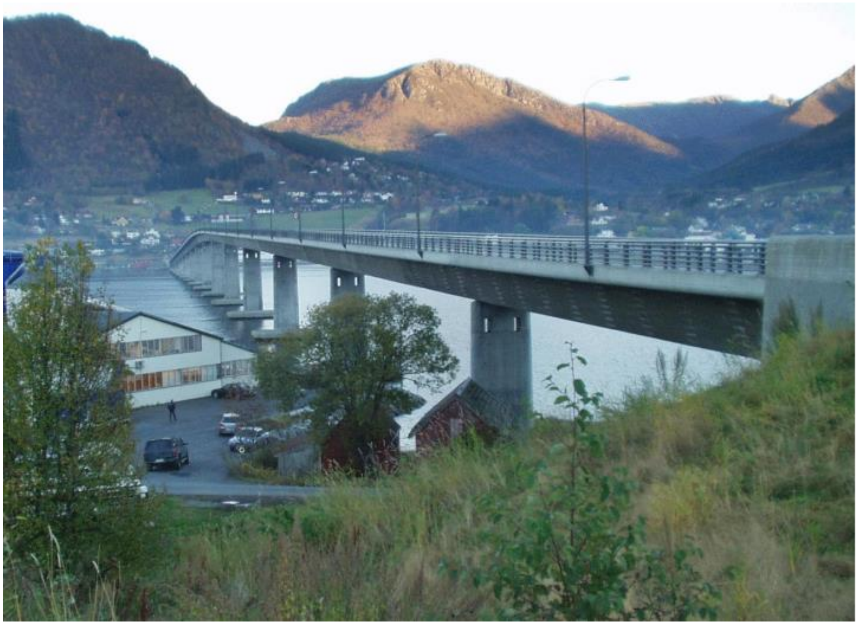


Figure 25: Overview Sykkylvsbrua

In addition to the passive reinforcement, the Sykkylvsbru has a PT system made of 8 tendons. The geometry of the active reinforcement is composed of successive parabolic shapes, with eccentricities such that the tendons are near the deck at the supports and near the soffit at the mid-span sections. Each tendon is made of multiple strands of 140 mm^2, totalling 1680 mm^2 per tendon. Figure 26 shows the cross-section of the bridge.

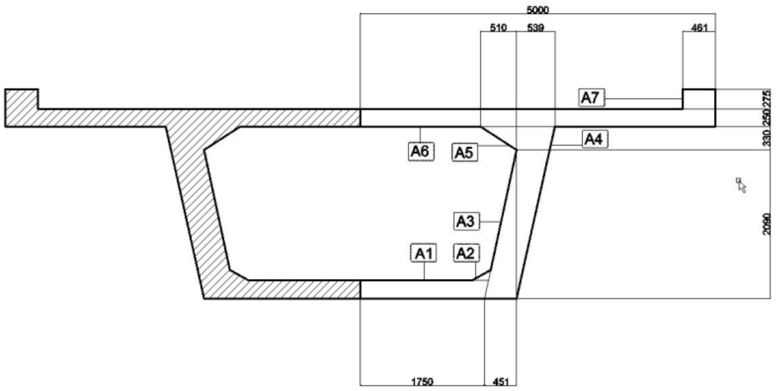


Figure 26: Sykkylvsbrua cross-section

A significant part of the work presented in this thesis [82] consists of the validation of the modelling procedures, by comparing the results from various numerical models with existing published experimental results. All PT beams are modelled using DIANA [69]. First the modelling is done for the 2D case, and subsequently extended to 3D, validating the models against results for fully and partially prestressed beams. Finally, the thesis attempts to model parts of the Sykkylysbru. More precisely, the work models half of one of the 60 m spans of the bridge. This span is modelled with fixed end supports and considering the corresponding support conditions for the symmetry considerations. Both, active and passive reinforcement, are modelled as embedded reinforcement. The nonlinear behaviour of the bridge is studied by gradually increasing the factors on the design traffic loads, which are defined according to the Eurocode 1 Part 2 [83]. Fifteen different damage scenarios of the PT system have been studied, which included various combinations on the number and location of complete tendon failures. The analysis shows a clear reduction in bridge capacity with increasing number of tendon failures (Figure 27). However, the results obtained in this study can only be considered as indicative. There are some serious doubts about the correctness of the results, because the reported capacity reduction of the bridge is only 21% (with respect to the healthy case) for the fully damaged case where all tendons are missing.

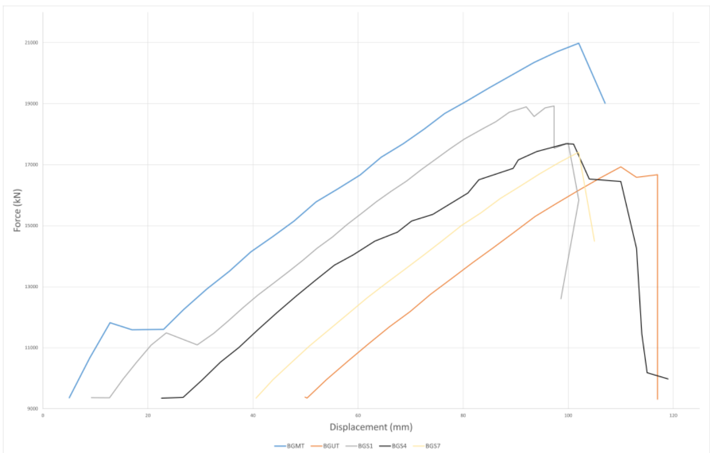


Figure 27: Load-deflection curves for a selection of scenarios

5. Summary

This study provides additional knowledge about the structural consequences of damaged PT systems, with particular focus on the presence of voids in the duct. The work consisted of two distinct parts. First, a literature review was performed on various relevant topics including, damages in PT bridges, corrosion mechanisms, NDT, and existing guidelines. In the second part, the problem was investigated numerically by FEM analysis, analysing simple beams, case studies and generic bridge typologies. The results of this study can be summarized in a series of conclusions, from which a guideline for structural inspections of PT structures can be drafted. This section also presents a list of recommendations for authorities that own bridges in general, and PT bridges in particular.

5.1. Discussion

The presented literature review has shown that much has been learned over the last decades regarding the causes, detection, and relevance of damaged PT systems. However, many difficulties and uncertainties remain, indicating that the evaluation of the condition of the active reinforcement in grouted ducts is still a challenging task. For this reason, it is paramount that infrastructure owners define maintenance strategies, which must be implemented regularly and rigorously. The review has shown that no general method exists applicable to all structures. Nevertheless, the problem is somewhat narrowed with respect to the structural typology and location on the bridge. Current UK guidelines rank the different bridge configurations in terms of its robustness (or risk of brittle failure mode), which is an idea applicable also to the Norwegian bridge network as whole. Following this criteria, segmental bridges (more precisely, segmentally mounted) has been identified as the most critical type of construction, which is a bridge typology that has rarely been used in Norway. With respect to problematic locations on the bridge and hazardous scenarios, the literature review lists the critical construction details and situations where PT damages might occur. The list is extensive, but it can be the starting point the systematization in future inspection guidelines.

5.2. Conclusions

This section lists the most relevant conclusions extracted from each of the parts in this study and grouped into categories.

<u>Bridge damages</u>

- Defects in design or execution are the likely source of corrosion deterioration in PT systems. Poor grouting has often been reported as a main contributor to bridge damage and there exists a clear link between voids and corrosion.
- Critical section locations are many and have been systematically reported in [5]. Sorting them in descending severity for structural safety, they could be summarized as: 1) Sections with maximum moment; 2) Locations where ducts intersect construction joints, 3) Near the anchorages.
- As acknowledged by several studies, it is often the case that there are no indications of damage in PT bridges. The absence of warning signs that would indicate the presence of damage does not necessarily guarantee the structural safety. Failure may occur without showing any warning signs. In those cases where warnings signs had been reported, these

were: longitudinal and shear cracks following the duct location, concrete spalling, concrete staining, anchorage corrosion, open grout vents, water leakage, grout efflorescence.

- PT segmental bridges are the bridge typology more susceptible to corrosion, but the problems can be attributed principally to poor design choices and poor-quality construction. Existing bridges of this typology require special attention, and future designs should avoid previous design mistakes.
- The robustness of a structure to damages in PT systems greatly depends on the level of redundancy, the number of tendons and the degree of static indeterminacy. Therefore, the most vulnerable typologies are statically determinate structures, namely simply supported bridges.
- In Norway, bridges are quite robust and safe against sudden failure due to problems in PT systems, because of their structural redundancy. Nevertheless, there still are certain more critical structural types like simply supported PT bridges or special structures (e.g., Kollstraumen), where their structural safety is quickly compromised under severe PT damages.
- Even if the existing PT bridge inventory is quite robust in general, it is still paramount to regularly inspect, react when necessary and maintain them accordingly.

Corrosion

- High-strength steel has been shown to corrode at a faster rate than mild steel making PT concrete more sensitive to the presence of chlorides, which probably sets the chloride content threshold for corrosion initiation to a lower value. Therefore, existing recommended threshold values on chloride concentration for corrosion initiation should be treated with caution in the case of PT bridges.
- There exist reports of an additional corrosion mechanism that is particular to grouted PT systems. It occurs under low-oxygen conditions, in areas of segregated grout with whitish phase and plastic consistency, which can be found in elevated parts of tendons near the ends. The suggested probable corrosion mechanism initiates corrosion at the interstices among wires that is then continued with the subsequent appearance of macrocells.

NDT

- None of the NDT techniques can detect defects in the strands. Whereas grout defects could be identified with low to medium accuracy using impact echo, ultrasonic echo, and ground penetrating radar. Because the condition of the strand cannot be assessed, these techniques need to be combined with partly destructive testing.
- The existing maintenance guideline in the US provides a framework to identify the best NDT for each particular case and evaluates what action should be taken.

Existing guidelines

- The intention of the existing guideline in UK is to be able to categorize all PT bridges in the network using the same framework in order to define a prioritisation of interventions according to the risk of failure. The guideline [16] provides a systematization of the assessment procedures for PT structures in terms of risk evaluation and classifies each structural type in terms of its risk of brittle failure. It stipulates that the bridges with the highest risk are segmental bridges made of beams or box girders in a simply supported configurations.

- This work provides a framework for modelling bridges with damaged PT systems. This framework gathers existing recommendations on non-linear FEM of concrete structures together with the expertise obtained during this project. The tool and resources to perform more detailed site-specific studies is available for future work.

Numerical modelling

- Even though it is possible to model the problem at hand with reasonable detail, it still is a challenging task. The investigation of the presence of voids in large structures was limited by computational power. Therefore, this required significant simplifications in the models, which resulted in conclusions that can only be considered as indicative.
- Furthermore, the problem at hand is known to be difficult. The presence of voids in PT systems has only marginal influence on the possible load effects. This is particularly true under small levels of loading such as those expected during normal operational conditions of existing bridges. The numerical results indicated that the presence of voids only showed measurable differences under ultimate loading conditions and only when the extent of the voids was large.
- It was numerically shown that there exist differences in crack patterns. As expected, bonded reinforcement leads to a more even distribution of cracks along the element. When the active reinforcement is unbonded, there is a concentration of cracks in a reduced number of locations but featuring larger widths.
- Differences in cracking patterns have been observed under laboratory conditions for various damage scenarios. However, it is not possible to directly extend these results to full-scale real structures. Actual bridges have multiple tendons and complex section geometries. The presence of voids and damages might be localized in discrete locations and single tendons. What is more, there exist no reference structures to compare to, which makes it practically impossible to assess the condition of the PT system based on the crack pattern.
- The analysis of the numerical results shows that any measurable load effect (other than cracking) is only affected marginally by the studied defects and damage scenarios. It is therefore not possible to highlight any particular indicator that would facilitate the inspection of bridges.

5.3. Guidelines

Based on the reported conclusions, it is possible to draft a guideline for the inspection of PT bridges, regarding the evaluation of the tendons' condition. The inspection could be divided in three steps.

Step 1: Search for the presence of voids using NDT at the most likely critical locations, which are:

- near the anchorages
- near couplers
- around elevated points of the PT profile with negative curvature (usually near the supports)

Step 2: Identify malfunctioning construction details that might affect the tendons. These are places that, due to poor design or execution, allow for the ingress of water in the vicinity of tendons. Possible details are: construction joints, supports, hinges, anchorages, vents, drainage systems and other.

Step 3: If any of the previous steps reported issues in a location, gain access to the duct with a borehole and evaluate the tendon condition by direct observation. If the tendon is corroded, a detailed reassessment of the structure's capacity must be requested.

The presence of voids and poor grout can be assessed with relatively good accuracy using NDT. To limit the extent of the areas to be studied, the inspection should focus only on sections where it is likely to find voids in more than one tendon. On the other hand, water ingress is directly related to a poorly designed or executed detail near the tendons. A thorough search for defects in those details should identify locations where possible water ingress has occurred. Finally, the locations that have a void or malfunctioning detail must be inspected via partially destructive methods, because none of existing NDT can assess the condition of the tendon. Careful opening of holes and direct observation has been proven to be the most effective procedure to evaluate the integrity of the PT system.

5.4. Recommendations

Specific recommendations for NPRA:

- Review the bridge database and evaluate the structural robustness of all the bridges in the road network. This applies to all types of bridges (not only PT) since the robustness is a valid safety condition to corrosion in general. Prioritize inspections on bridges with low levels of structural redundancy.
- Adopt the framework used in the UK to categorize all PT bridges in the network to define a prioritisation of interventions according to the risk of failure.

5.5. Future work

Possible future work lines to improve the assessment of existing PT bridges:

- Systematization of bridge modelling. Develop systematic methodologies for elaboration of numerical models of existing bridge. Identify best practices for modelling of damages, including: corrosion, bond deterioration, existing cracking, among others.
- Integration of inspection results. Define what information is necessary from inspection to accurately assess the structural condition and how this information should be integrated into the bridge model.

References

[1] Statens Vegvesen. Bruprosjektering Prosjektering av bruer, ferjekaier og andre bærende konstruksjoner. Håndbok N400. 2015. Link

[2] Statens Vegevesen. Prosesskode 2 Standard beskrivelse for bruer og kaier. Håndbok R762. 2018. Link

[3] H. Johansen. Beregningsveiledning for etteroppspente betongbruer. Statens vegvesen rapporter Nr 668, 2017. Link

[4] F. Andresen. Analysis and check of a prestressed concrete bridge according to European regulations. NTNU, master thesis, 2015. Link

[5] FIB. Durability of post-tensioning tendons. Bulletin 33, 2006.

[6] N. Hewson. Prestressed Concrete Bridges - Design and Construction. ICE Publishing. Thomas Telford Limited, 2012.

[7] R.W. Poston, J.P. Wouters. Durability of precast segmental bridges. National cooperative Highway Research Program, NCHRP Web document 15, 1998. Link

[8] H. Vestad, M. Vestad. Non-linear behaviour of insufficiently grouted post-tensioned concrete members, Master thesis, 2021.

[9] Norsk Betongforening. Spennarmeringsarbeider. NB14, 2016.

[10] European Organisation for Technical Approvals. 2002 ETAG 013 was released; Guideline for European Technical Approval of Post-Tensioning kits for Prestressing of Structures, ETAG 013, 2002.

[11] European Committee for Standardization. Requirements for the installation of post-tensioning kits for prestressing of structures and qualification of the specialist company and its personnel. CEN CWA 14646, 2003.

[12] European Committee for Standardization. Grout for prestressing tendons – Test methods. EN 445, 2007.

[13] European Committee for Standardization. Grout for prestressing tendons – Grouting procedures. EN 446, 2007.

[14] European Committee for Standardization. Grout for prestressing tendons – Basic requirements. EN 447, 2007.

[15] A. Karlsson, P. Jilderda, B. Täljsten. Inspeksjon av etterspent armering i betongbruer. FoU-programmet Bedre Bruvedlikehold 2017-2021. Statens vegvesen rapporter Nr. 699, 2021. Link

[16] Highways England. Design Manual for Roads and bridges – CS 465 Management of post-tensioned concrete bridges. 2020. Link

[17] M.R. Chauvin. Considerations for Development of Inspection and Remedial Grouting Contracts for Post-tensioned Bridges. Minnesota Department of Transportation, 2017.

[18] B. Godart. Pathology, appraisal, repair and management of old prestressed concrete beam and slab bridges. Structure and Infrastructure Engineering, 11:4, p. 501-518, 2015. DOI: 10.1080/15732479.2014.951865

[19] M.I. Mousa. Effect of bond loss of tension reinforcement on the flexural behaviour of reinforced concrete beams. HBRC Journal, 12, 3, p. 235-241, 2016. DOI: 10.1016/j.hbrcj.2015.01.003

[20] National Academies of Sciences, Engineering, and Medicine 2021. Repair and Maintenance of Post-Tensioned Concrete Bridges. Washington, DC: The National Academies Press. DOI: 10.17226/26172.

[21] X. Zhang, L. Wang, J. Zhang, Y. Liu. Corrosion-induced flexural behavior degradation of locally ungrouted post-tensioned concrete beams. Construction and Building Materials, 134, p. 7–17, 2017. DOI: 10.1016/j.conbuildmat.2016.12.140

[22] A. Alfailakawi, C.L. Roberts-Wollmann, M. Hebdon, I. Koutromanos. Experimental and Analytical Evaluation of Residual Capacity of Corrosion-Damaged Prestressed Concrete Bridge Girders. Virginia Tech, Department of Civil and Environmental Engineering, Report VTRC 21-R6, 2020. Link.
[23] Z.H. Lu, F. Li, Y.G. Zhao. An Investigation of Degradation of Mechanical Behaviour of Prestressing Strands Subjected to Chloride Attacking. 5th International Conference on Durability of Concrete Structures, Shenzhen, China, 2016.
[24] C.H. Jeon, C.S. Shim. Flexural Behavior of Post-Tensioned Concrete Beams with Multiple Internal Corroded Strands. Applied Science,. 10, 7994, 2020. DOI: 10.3390/app10227994
[25] B. Belletti, F. Vecchi, C. Bandini, C. Andrade, J. Sánchez Montero. Numerical evaluation of the corrosion effects in prestressed concrete beams without shear reinforcement. Structural Concrete, 21, p. 1794–1809, 2020. DOI: 10.1002/suco.201900283
[26] C.H. Jeon, J.B. Lee, S. Lon, C.S. Shim. Equivalent material model of corroded prestressing steel strand. Journal of Material Research and Technology, 8(2), p. 2450-2460, 2019. DOI: 10.1016/j.jmrt.2019.02.010
[27] D. Coronelli, A. Castel, N.A. Vu, R. François. Corroded post-tensioned beams with bonded tendons and wire failure. Engineering Structures, 31, p. 1687–1697, 2009. DOI: 10.1016/j.engstruct.2009.02.043
[28] CEB-FIP. Model Code 90.
[29] A.O. Abdelatif1, J.S. Owen, M.F.M. Hussein. Modeling and Parametric Study of the Reanchorage of Ruptured Tendons in Bonded Posttensioned Concrete. Journal of Structural Engineering, 143(12), 2017. DOI: 10.1061/(ASCE)ST.1943-541X.0001898.
[30] FIB. Model Code 2010 - Volume 1. Bulletin 65, 2012.
[31] A.O. Abdelatif, J.S. Owen, M.F.M. Hussein. Modeling the re-anchoring of a ruptured tendon in bonded post-tensioned concrete. 4th International Symposium on Bond in concrete 2012, Brescia, Italy, 2012.
[32] L. Wang, X. Zhang, J. Zhang, J. Yi, Y. Liu. Simplified Model for Corrosion-Induced Bond Degradation between Steel Strand and Concrete. ASCE Journal of Materials in Civil, 29(4), 2017. DOI: 10.1061/(ASCE)MT.1943-5533.0001784.
[33] W.H. Hartt, S.K. Lee. Corrosion Forecasting and Failure Projection of Post-Tensioned Tendons in Deficient Cementitious Grout. The Federal Highway Administration, Report FHWA-HRT-17-074, 2018.
[34] FIB. Durability of post-tensioning tendons. Bulletin 15, 2001.
[35] FIB. Structural Concrete Textbook - Volume 5. Bulletin 62, 2012.
[36] T.S. Theryo, W.H. Hartt, P. Paczkowski. Guidelines for Sampling, Assessing, and Restoring Defective Grout in Prestressed Concrete Bridge Post-Tensioning Ducts. The Federal Highway Administration, Report FHWA-HRT-13-028, 2013.
[37] L. Wang, X. Zhang, J. Zhang, Y. Ma, Y. Xiang, Y. Liu. Effect of insufficient grouting and strand corrosion on flexural behavior of PC beams. Construction and Building Materials 53, p. 213–224, 2014. DOI: 10.1016/j.conbuildmat.2013.11.069
[38] X. Zhang, L. Wang, J. Zhang, Y. Ma, Y. Liu. Flexural behavior of bonded post-tensioned concrete beams under strand corrosion. Nuclear Engineering and Design, 313, p. 414–424, 2017. DOI: 10.1016/j.nucengdes.2017.01.004
[39] X. Chenga, H. Liub, J. Suc, L. Mad, G. Li. Flexural Performance of Prestressed Beams with Grouting Material of Various Compactnesses. KSCE Journal of Civil Engineering, 24(8), p. 2419-2434, 2020. DOI: 10.1007/s12205-020-0107-1
[40] L. Wanga, Z. Hua, J. Yia, L. Daia, Y. Maa, X. Zhang. Shear Behavior of Corroded Post-Tensioned Prestressed Concrete Beams with Full/Insufficient Grouting. KSCE Journal of Civil Engineering, 24(6), p. 1881-1892, 2020. DOI: 10.1007/s12205-020-1777-4

[41] T.M. Pape, R.E. Melchers. Performance of 45-year-old corroded prestressed concrete beams. ICE Structures and Buildings, 166, SB10. DOI: 10.1680/stbu.11.00016
[42] A. Menga, T. Kanstad, D. Cantero. Corrosion-induced failures of post-tensioned bridges. NTNU report, 2021.
[43] A. Menga, T. Kanstad, D. Cantero, L. Bathen, K. Hornbostel. Review of corrosion-induced failures of post-tensioned bridges. Capacity assessment of corroded reinforced concrete structures, CACRCS 2021, Online, 2021.
[44] U. Nurnberger. Corrosion induced failure mechanisms of prestressing steel. Material and Construction, 53, p. 591-601, 2002.
[45] L. Bertolini, B. Elsener, P. Pedeferri, R. Polder. Corrosion of steel in concrete. Prevention, diagnosis, repair. Wiley-vch Verlag GmbH, 2004.
[46] FIB. Factory applied corrosion protection of prestressing steel. Bulletin 11, 2001.
[47] S. Hurlebaus, M.B.D. Hueste, M.M. Karthik, T. Terzioglu. Condition assessment of bridge post-tensioning and stay cable systems using NDE methods. National Cooperative Highway Research Program NCHRP. The National Academies Press, 2016.
[48] F. Li, Y. Yuan, C.Q. Li. Corrosion propagation of prestressing steel strands in concrete subject to chloride attack. Construction and Building Materials, 25, p. 3878–3885, 2011. DOI: 10.1016/j.conbuildmat.2011.04.011
[49] M.J. Osmolska, K. Hornbostel, T. Kanstad, M.A.N. Hendriks, G. Markeset. Inspection and Assessment of Corrosion in Pretensioned Concrete Bridge Girders Exposed to Coastal Climate. Infrastructures, 5, 76, 2020. DOI: 10.3390/infrastructures5090076
[50] M.K. Hurst. Prestressed concrete design. Taylor & Francis e-Library, 2003.
[51] FIB. Influence of material and processing on stress corrosion cracking of prestressing steel. Bulletin 26, 2003.
[52] Luca Bertolini and Maddalena Carsana. High pH Corrosion of Prestressing Steel in Segregated Grout, in C. Andrade and G. Mancini (eds.), Modelling of Corroding Concrete Structures, RILEM Bookseries 5, DOI: 10.1007/978-94-007-0677-4_10
[53] M. Carsana, L. Bertolini. Corrosion failure of post-tensioning tendons in alkaline and chloride-free segregated grout: a case study. Structure and Infrastructure Engineering: Maintenance, Management, Life-Cycle Design and Performance, 11:3, p. 402-411, 2015. DOI: 10.1080/15732479.2014.887736
[54] S.K. Lee, J. Zielske. An FHWA Special Study: Post-Tensioning Tendon Grout Chloride Thresholds. The Federal Highway Administration, Report FHWA-HRT-14-039, 2014.
[55] Statens Vegvesen. Inspeksjonshåndbok for bruer. Hånbok V441. 2014. Link
[56] S. Hurlebaus, M.B.D. Hueste, M.M.Karthik, T. Terzioglu. Inspection Guidelines for Bridge Post-Tensioning and Stay Cable Systems Using NDE Methods. National Cooperative Highway Research Program NCHRP. The National Academies Press, 2017.
[57] A. Karlsson, E. Boström, P. Jilderda. Etterspente betongbruer Inspeksjonsmanual. FoU-programmet Bedre bruvedlikehold 2017-2021. Statens vegvesen rapporter Nr. 718, 2021. Link
[58] S.K.U. Rehman, Z. Ibrahim, S.A. Memon, M. Jameel. Nondestructive test methods for concrete bridges: A review. Construction and Building Materials, 107, p. 58–86, 2016. DOI: 10.1016/j.conbuildmat.2015.12.011
[59] International Atomic Energy Agency. Guidebook on non-destructive testing of concrete structures. Vienna, 2002.
[60] International Standard Organization. ISO 13822 Bases for design of structures – Assessment of existing structures. 2010.
[61] International Standard Organization. ISO 16311 Maintenance and repair of concrete standards. 2014.

[62] International Standard Organization. ISO 16204 Durability – Service life design of concrete structures. 2012.
[63] W.F. Chen, L. Duan. Bridge Engineering Handbook. Superstructure design. CRC Press, 2014.
[64] H. Mutsuyoshi, N.D. Hai, A. Kasuga. Recent technology of prestressed concrete bridges in Japan. IABSE-JSCE Joint Conference on Advances in Bridge Engineering, Dhaka, Bangladesh, 2010.
[65] R.M. Salas, A.J. Schokker, J.S. West, J.E. Breen, M.E. Kreger. Conclusions, Recommendations and Design Guidelines for Corrosion Protection of Post-Tensioned Bridges, Research Report (9/93-8/03), Center for Transportation Research, The University of Texas at Austin, 2004. Link
[66] Highways England. Design Manual for Roads and bridges – CS 450 Inspection of highway structures. 2021. Link
[67] Aas-Jakobsen A/S. NovaFrame User´s Guide Version 5, 2011. Link
[68] Simulia, Abaqus Analysis User's Guide. [Online], 2014. Link.
[69] DIANA. Manuals for DIANA FEA. [Online]. Link
[70] Python Software Foundation. Python Language Reference, version 3.10. Link.
[71] M.A.N. Hendriks, M.A. Roosen (editors). Guidelines for Nonlinear Finite Element Analysis of Concrete Structures. Rijkswaterstaat Centre for Infrastructure, Report RTD:1016-1:2019, 2019.
[72] G. Pinto, D. Cantero. Modelling post-tensioned structures with DIANA FEM software. NTNU report, 2021.
[73] M. Torrijos Musach. Post-tension concrete beam modelling and analysis in DIANA via Python. Bachelor thesis, NTNU, 2020. Link.
[74] M. Melaku. Framework for modelling post-tensioned structures in DIANA. Master thesis, 2022. (Tentative title)
[75] Standard Norge. Prosjektering av betongkonstruksjoner. Beregnings- og konstruksjonsregler. NS 3473:1973.
[76] M. Aasheim, L. Hangaard. Analyse av FFB-bru med korrosjonsskadet spennarmering. Tilstandsvurdering og kapasitetskontroll av Osstrupen bru. Master thesis, NTNU, 2018. Link
[77] H. Amiri, E. Moen. Styrkeberegning og analyse av eksisterende spennarmert buebru. Master thesis, NTNU, 2018. Link
[78] E. Ukvitne, H. Vangdal. Kapasitetskontroll og analyse av etteroppspent bru med redusert spennarmering. Master thesis, NTNU, 2021.
[79] H.R. Kvale, T.J. Opheim. Modellering av etteroppspente betongbruer med korrosjonsskadet spennarmering. Master thesis, NTNU, 2019. Link
[80] H. Gloppen. Sunmørsposten, [Online], 2017. Link
[81] MATLAB. version 9.8.0 (R2020a). Natick, Massachusetts: The MathWorks Inc.; 2020.
[82] J.J. Czesak. Behavior of Post-Tensioned concrete box girder with multiple damaged tendons. Master thesis, 2022. (Tentative title)
[83] European committee for standardization. Eurocode 1: Actions on structures – Part 2: Traffic loads on bridges, EN 1991-2:2003.

Appendix A

Steps to model a post-tensioned beam with bond-slip in DIANA software.

- Start project + Units
    - mm, T, N
- Define shapes
    - Beam
    - Tendons
    - Support blocks
    - Anchorage plates
    - Loading plates
- Additional points to place the reactions
    - Create point near desired location
    - Project the point
- Define element class for each object
    - All are standard solid
    - Except the tendons that are TRUSS_BOND_SLIP
- Create materials
    - Concrete
    - Steel for tendons
    - Steel for plates
    - No bond
    - Full bond
- Bond-slip modelling, two options
    - Linear, in terms of the normal and shear stiffness
    - Nonlinear, in terms of normal stiffness, and the points of the Doer curve
- Assign materials
- Define geometries
    - Tendon dimension
- Element data
    - Only if nonlinear bond-slip selected
    - Definition of interface type TRUSS
- Loads definitions
    - Self-weight
    - Prestressing force
        - Applied at the end of the tendon
        - Applied with the tendon’s angle at that point
    - Reactions on the concrete
        - Forces that are of opposite direction to the prestressing forces
        - Applied as distributed forces on the anchor plates
        - Applied at the exterior face of the plate
    - Any other loads
- Tying
    - A rigid connection between the end of the tendon and the anchor
    - Defined by master (tendon end) and slave (anchor plate) relation
- Supports
    - Define the boundary conditions at the lower face of the supporting blocks
- Interfaces
    - The connection between the beam and the plates

Define the interface material
- high stiffness in normal direction
- low stiffness in transverse directions

Define SHAPEFACES on the beam
- as projection of the plate to the beam

Create a connection
- Attach the source (plate) and target (beam)

- Load combinations
  - Give default factor 1 to all load sets separately
  - Add a new load combination with 3 loads (SelfWeight, Prestress, Reactions) each with factor 1
- Mesh
  - Define the options
    - HEXQUAD
    - size of element
  - Assign to all shapes in the model
  - Generate the mesh
- Analysis
  - Add phase step (Phase 1)
    - Add a structural nonlinear analysis
    - Calculate for the last load combination (the one including 3 loads)
  - Add phase step (Phase 2)
    - Add a structural nonlinear analysis
    - Remove existing execute block
    - New execute block of the type START
    - With additional load. Only self-weight.
    - Change number of iterations to 20. (or other)
  - Add phase step (Phase 2)
    - Add a structural nonlinear analysis
    - Addition execution block of the type LOAD
      - To add the eventual extra loads on the beam
- Change what materials are active in each phase
  - Make the reinforcement material in Phase 2 equal to the fully bonded one
  - Activate the tying (deactivate them in Phase 1)
- For the analysis it is better to have 3 phases
  - Phase 1: Prestressing and self-weight
  - Phase 2: Grouting and release
    - Changing bond “material”
    - Apply the tying
    - Remove the loads (prestress and reaction)
  - Phase 3: (Previous phase) + External load

Appendix B

Python framework introduced in Section 3 to model a PT beam in DIANA software.

B.1 Initial commands and definitions

```
# ----------------------------------------------------------
# ------------ Initial commands and definitions ------------
# ----------------------------------------------------------

# Close running project
closeProject(  )

# Importing libraries
import numpy as np
from math import *
import matplotlib
matplotlib.use('AGG')
import matplotlib.pyplot as plt
import time

# Class defintion
class StructType():
        pass

# Definitions
Project = StructType();
Project.Units = StructType();
Beam = StructType();
Beam.Prop = StructType();
Beam.Concrete = StructType();
Beam.Tendons = StructType();
Beam.Tendons.Coords = StructType();
Beam.Tendons.Opt = StructType();
Beam.Tendons.BS = StructType();
Beam.Tendons.BS.NoBond = StructType();
Beam.Tendons.BS.FullBond = StructType()
Beam.Tendons.Plates = StructType();
Beam.Tendons.Splits = StructType();
Beam.Tendons.Voids = StructType()
Beam.Reinf = StructType();
Beam.Reinf.Links = StructType();
Beam.Supports = StructType();
Beam.Supports.Plates = StructType()
Mesh = StructType();
Mesh.Element = StructType();
Mesh.Plates = {"Load": StructType(), "Anchor": StructType(), "Support": StructType() }
Loads = StructType();
Options = StructType(); Options.Solver = StructType()

# Default values
Beam.Tendons.f_p = 0;
Beam.Reinf.main_E = 210000;
Beam.Reinf.main_f_y = 0;
Beam.Reinf.links_E = 210000;
Beam.Reinf.links_f_y = 0;
Options.generate_mesh_on = 0;
Options.run_analysis_on = 0;
Options.phase2_on = 0;
Options.until_failure_on = 0;
Options.custom_output_on = 0;
Options.save_pictures_on = 0;
Options.phase3_on = 0;
Options.Solver.arc_length_on = 0;
Options.Solver.step_sizes = "";
Options.Solver.max_num_of_iterations = 30;
```

B.2 Particular model definitions

```
# --------------------------------------------------------
# -------------- Particular model definitions --------------
# --------------------------------------------------------

# -- Project path, folders, name and size --
Project.path_for_Diana_files = "C:/Diana_Models"
Project.path_for_plot_saving = "C:/Diana_Models"
Project.folder = "PT_Beam_Model"
Project.subfolder = "Model_01"
Project.size_in_m = 100              # Size of the workspace in [m]

# -- Units --
Project.Units.length = "mm"
Project.Units.mass = "T"
Project.Units.force = "N"

# -- Beam properties --
Beam.Prop.length = 5000              # in [mm]
Beam.Prop.width = 400        # in [mm]
Beam.Prop.height = 400       # in [mm]

# -- Concrete --
Beam.Concrete.class_num = 30
Beam.Concrete.model_type = "Dianas EC2"      # Select this for default concrete properties
Beam.Concrete.model_type = "Custom Properties"# Select this for custom concrete properties

# -- Tendons --
Beam.Tendons.num = 2                 # Number of tendons
Beam.Tendons.area = 100                      # Area of each tendon
Beam.Tendons.force_P = 1e5           # Prestressing force on each tendon
Beam.Tendons.Coords.x = np.array([0,0.5,1])*Beam.Prop.length
Beam.Tendons.Coords.z = (1-np.array([0,0.8,0]))*Beam.Prop.height/2
Beam.Tendons.Coords.num_of_points = 100
#Beam.Tendons.f_p = 1348             # Yield stress (Defaul is linear steel)

# -- Bond-slip options -- (linear)
Beam.Tendons.BS.type = "linear"
Beam.Tendons.BS.NoBond.normal_stiffness = 100
Beam.Tendons.BS.NoBond.shear_stiffness = 0.1
Beam.Tendons.BS.FullBond.normal_stiffness = Beam.Tendons.BS.NoBond.normal_stiffness
Beam.Tendons.BS.FullBond.shear_stiffness = 1e5
#Beam.Tendons.BS.FullBond.shear_stiffness = Beam.Tendons.BS.NoBond.shear_stiffness

# # -- Bond-slip options -- (nonlinear)
# Beam.Tendons.BS.type = "nonlinear"

# -- Splitting --
Beam.Tendons.Splits.num = 1          # Choose 1 for no splitting

# -- Voids in duct --
Beam.Tendons.Voids.tendon_IDs = []
Beam.Tendons.Voids.split_IDs = []
# Beam.Tendons.Voids.tendon_IDs = [1,1]
# Beam.Tendons.Voids.split_IDs = [1,Beam.Tendons.Splits.num]

# -- Anchor Plates --
#Beam.Tendons.Plates.thickness = 0           # No anchor plates option
Beam.Tendons.Plates.thickness = 50
Beam.Tendons.Plates.width = 100
Beam.Tendons.Plates.height = 100

# -- Passive reinforcement (Main) --
Beam.Reinf.Main = {}; k = 0
Beam.Reinf.main_E = 210000    # Elastic modulus (Default 210000)
#Beam.Reinf.main_f_y = 500    # Yield stress (Default is linear steel)
k = k + 1; Beam.Reinf.Main[k] = StructType()
Beam.Reinf.Main[k].num = 2
Beam.Reinf.Main[k].diameter = 12
Beam.Reinf.Main[k].cover = 20
Beam.Reinf.Main[k].z_coord = Beam.Reinf.Main[k].cover
k = k + 1; Beam.Reinf.Main[k] = StructType()
Beam.Reinf.Main[k].num = 2
Beam.Reinf.Main[k].diameter = 10
Beam.Reinf.Main[k].cover = 20
```

```
Beam.Reinf.Main[k].z_coord = Beam.Prop.height-Beam.Reinf.Main[k].cover

# -- Passive reinforcement (Links) --
Beam.Reinf.Links.num = 0              # Set to 0 for no reinforcement
Beam.Reinf.links_E = 210000   # Elastic modulus (Default 210000)
#Beam.Reinf.links_f_y = 500           # Yield stress (Default is linear steel)
Beam.Reinf.Links.diameter = 8
Beam.Reinf.Links.cover = 15

# -- Supports --
Beam.Supports.Plates.length = 200
Beam.Supports.Plates.thickness = 100

# -- Mesh --
Mesh.Element.size = 100              # in [mm]

# For denser meshes near plates make ratio > 1. (Default value = 1)
#Mesh.Plates["Anchor"].ratio = 2
#Mesh.Plates["Load"].ratio = 2
#Mesh.Plates["Support"].ratio = 2

# -- Loads --
# Self-weight
Loads.SelfWeight = StructType()
Loads.SelfWeight.on = 1              # Set to 0 for no self-weight

# Concentrated loads
Loads.Concentrated = {}; k = 0
k = k + 1; Loads.Concentrated[k] = StructType(); Loads.Concentrated[k].Plate = StructType()
Loads.Concentrated[k].coords = [Beam.Prop.length/2,Beam.Prop.width/2,Beam.Prop.height]
Loads.Concentrated[k].Plate.length = 200
Loads.Concentrated[k].Plate.width = 200
Loads.Concentrated[k].Plate.thickness = 20
Loads.Concentrated[k].value = -1e5

# Symmetry
Options.model_symmetry = 0
#Options.model_symmetry = 1/2
#Options.model_symmetry = 1/4 # Note: only working for even number of tendons

# Solver
Options.Solver.arc_length_on = 1
Options.Solver.step_sizes = "0.05 (25)"
#Options.Solver.max_num_of_iterations = 30   # Default 30

# Project name
Project.name = Project.subfolder + "_Symmetry_" + str(round(Options.model_symmetry*100)) +
"_Mesh_" + str(Mesh.Element.size)

# -- Other calculation options --
Options.generate_mesh_on = 1  # Only useful for developing purposes
Options.run_analysis_on = 1
Options.phase2_on = 1                # Grouting
Options.phase3_on = 1                # External loading
#Options.until_failure_on = 1
```

B.3 Main script

```
# ----------------------------------------------------------
# ----------------------- Main script ----------------------
# ----------------------------------------------------------

# Functions definitions

def my_Parabola3Points(Coords,num,model_symmetry):
	# Coords.x_i = array with x-coordinates of three points
	# Coords.y_i = array with y-coordinates of three points
	# Coords.z_i = array with x-coordinates of three points
	# num = number of points to evaluate in the interval x_i[0] to x_i[-1]
	# model_symmetry = 0 for full parabola, or other for half or quarter symmetry
	# Points to evaluate
	if model_symmetry == 0:
		x = np.linspace(Coords.x[0],Coords.x[-1],num)
		y = np.linspace(Coords.y[0],Coords.y[-1],num)
	elif model_symmetry > 0:
		x = np.linspace(Coords.x[0],Coords.x[1],num)
		y = np.linspace(Coords.y[0],Coords.y[1],num)
	# Evaluation
	z = Coords.z[0]*(x-Coords.x[1])*(x-Coords.x[2])/((Coords.x[0]-
Coords.x[1])*(Coords.x[0]-Coords.x[2])) + \
		Coords.z[1]*(x-Coords.x[0])*(x-Coords.x[2])/((Coords.x[1]-
Coords.x[0])*(Coords.x[1]-Coords.x[2])) + \
		Coords.z[2]*(x-Coords.x[0])*(x-Coords.x[1])/((Coords.x[2]-
Coords.x[0])*(Coords.x[2]-Coords.x[1]))
	# Merging arrays for Diana command input
	coordinates = [[x[0],y[0],z[0]]]
	coordinates_in_one_row = [x[0],y[0],z[0]]
	for k in range(1,num):
		coordinates.append([x[k],y[k],z[k]])
		coordinates_in_one_row.append(x[k])
		coordinates_in_one_row.append(y[k])
		coordinates_in_one_row.append(z[k])
	# Function output
	return [coordinates, coordinates_in_one_row]

def my_Parabola3PointsAngle(Coords,x):
	# Coords.x_i = array with x-coordinates of three points
	# Coords.y_i = array with y-coordinates of three points
	# Coords.z_i = array with x-coordinates of three points
	# x = x value where the angle is calculated
	theta = \
		(Coords.z[0]*(x - Coords.x[1]))/((Coords.x[0] - Coords.x[1])*(Coords.x[0] -
Coords.x[2])) + \
		(Coords.z[0]*(x - Coords.x[2]))/((Coords.x[0] - Coords.x[1])*(Coords.x[0] -
Coords.x[2])) - \
		(Coords.z[1]*(x - Coords.x[0]))/((Coords.x[0] - Coords.x[1])*(Coords.x[1] -
Coords.x[2])) - \
		(Coords.z[1]*(x - Coords.x[2]))/((Coords.x[0] - Coords.x[1])*(Coords.x[1] -
Coords.x[2])) + \
		(Coords.z[2]*(x - Coords.x[0]))/((Coords.x[0] - Coords.x[2])*(Coords.x[1] -
Coords.x[2])) + \
		(Coords.z[2]*(x - Coords.x[1]))/((Coords.x[0] - Coords.x[2])*(Coords.x[1] -
Coords.x[2]))
	# Function output
	return [theta]

def my_FrictionLoss(Coords,P,mu,k):
	# Calculation of friction loss for a parabola. Inputs:
	#	Coords.x_i = array with x-coordinates of three points
	#	Coords.y_i = array with y-coordinates of three points
	#	Coords.z_i = array with x-coordinates of three points
	#	P = Prestressing force
		#	mu = friction coefficient (around 0.2)
	#	k = unintentional additional curvature [m^-1] (around 0.01)
	# Adapting units of input x (because it has to be used in meters in the formula)
	if unit("Length")=="MM":
		Coords.x = Coords.x/1000
		Coords.y = Coords.y/1000
		Coords.z = Coords.z/1000
	# The location (x coordinate) where the friction loss is calculated
	x = Coords.x[1]
	# The angle change between the calculation point and the prestressing point
```

```
        theta = abs(my_Parabola3PointsAngle(Coords,x)[0]-my_Parabola3PointsAngle(Coords,0)[0])
        # To obtain the friction loss as a ratio we assume unit force prestressing (P = 1)
        friction_loss = 1-1*exp(-mu*(theta+k*x))
        # The actual prestress at section x
        P_at_x = P*(1-friction_loss)
        # Printing results
        round_digits = 2
        print("Theoretical friction loss: "+str(round(friction_loss*100,round_digits))+"%")
        print("Prestress at mid-span should be: "+str(round(P_at_x,round_digits)))
        # Outputs
        return [friction_loss, P_at_x]

def my_ParabolaLength(Coords):
        # Arc length of a symmetric parabola segment defined by
        # h = height of the segment
        # a = half the width of the segment
        # Definitions
        h = abs((Coords.z[0]+Coords.z[2])/2 - Coords.z[1])
        a = Coords.x[1]
        # Arc length
        if h>0:
                s = sqrt(a**2+4*h**2) + a**2/(2*h)*asinh(2*h/a)
        else:
                s = 2*a
        # Output
        return [s]

def my_EC2_Table_3_1(concrete_class):
        # Culculates all the values from Table 3.1 in EC2
        # Note all outputs in Mpa (also E values)
        # concrete_class = concrete class in MPa (N/mm2)
        # Table entries
        T31_f_ck = np.array([12,16,20,25,30,35,40,45,50,55,60,70,80,90])
        T31_f_ck_cube = np.array([15,20,25,30,37,45,50,55,60,67,75,85,95,105])
        # Output
        Output = StructType()
        Output.f_ck = concrete_class
        Output.f_ck_cube = np.interp(concrete_class,T31_f_ck,T31_f_ck_cube)
        Output.f_cm = concrete_class + 8
        Output.f_ctm = 0.3*concrete_class**(2/3)*(concrete_class<=50) +
2.12*log(1+(Output.f_cm/10))*(concrete_class>50)
        Output.f_ctk_0_05 = 0.7*Output.f_ctm
        Output.f_ctk_0_95 = 1.3*Output.f_ctm
        Output.E_cm = 22*(Output.f_cm/10)**0.3*1000
        Output.epsilon_c1 = min(0.7*Output.f_cm**0.31,2.8)/1000
        Output.epsilon_cu1 = (3.5*(concrete_class<=50) + (2.8+27*((98-
Output.f_cm)/100)**4)*(concrete_class>50))/1000
        Output.epsilon_c2 = (2.0*(concrete_class<=50) + (2.8+0.085*(Output.f_ck-
50)**0.53)*(concrete_class>50))/1000
        Output.epsilon_cu2 = (3.5*(concrete_class<=50) + (2.6+35*((90-
Output.f_ck)/100)**4)*(concrete_class>50))/1000
        Output.n = 2*(concrete_class<=50) + (1.4+23.4*((90-
Output.f_ck)/100)**4)*(concrete_class>50)
        Output.epsilon_c3 = (1.75*(concrete_class<=50) + (1.75+0.55*(Output.f_ck-
50)/40)*(concrete_class>50))/1000
        Output.epsilon_cu3 = (3.5*(concrete_class<=50) + (2.6+35*((90-
Output.f_ck)/100)**4)*(concrete_class>50))/1000
        # Function output
        return Output

# ---- Inputs processing ----

# Default values
# Plates mesh ratios
key_names = ["Anchor", "Load", "Support"]
for key_name in key_names:
        if not "ratio" in Mesh.Plates[key_name].__dir__():
                Mesh.Plates[key_name].ratio = 1
# Creating Options.SavePictures
Options.SavePictures = StructType();
Options.SavePictures.on = Options.save_pictures_on

# ---- Diana project file and properties ----

# Project file
```

```
newProject(
Project.path_for_Diana_files+"/"+Project.folder+"/"+Project.subfolder+"/"+Project.name,
Project.size_in_m )

# Project general properties
setModelAnalysisAspects( [ "STRUCT" ] )
setModelDimension( "3D" )
setDefaultMeshOrder( "QUADRATIC" )
setDefaultMesherType( "HEXQUAD" )
setDefaultMidSideNodeLocation( "ONSHAP" )

# Units
setUnit( "LENGTH", Project.Units.length.upper() )
setUnit( "MASS", Project.Units.mass.upper() )
setUnit( "FORCE", Project.Units.force.upper() )

# ---- Materials ----

# Dianas EC2 modelling
if Beam.Concrete.model_type == "Dianas EC2":
        addMaterial( "Material_Concrete", "CONCDC", "EN1992", [ "TOTCRK" ] )
        Concrete = my_EC2_Table_3_1(Beam.Concrete.class_num)
        setParameter( "MATERIAL", "Material_Concrete", "EC2CON/NORMAL/CLASS",
"C"+str(Concrete.f_ck)+"/"+str(Concrete.f_ck_cube) )

# With properties from EC2-Table 3.1
elif Beam.Concrete.model_type == "Custom Properties":
        Concrete = my_EC2_Table_3_1(Beam.Concrete.class_num)
        addMaterial( "Material_Concrete", "CONCR", "TSCR", [] )
        setParameter( "MATERIAL", "Material_Concrete", "LINEAR/ELASTI/YOUNG", Concrete.E_cm )
        setParameter( "MATERIAL", "Material_Concrete", "LINEAR/ELASTI/POISON", 0.2 )
        setParameter( "MATERIAL", "Material_Concrete", "LINEAR/MASS/DENSIT", 2.4e-09 )
        setParameter( "MATERIAL", "Material_Concrete", "MODTYP/TOTCRK", "ROTATE" )
        #setParameter( "MATERIAL", "Material_Concrete", "MODTYP/TOTCRK", "FIXED" )
        setParameter( "MATERIAL", "Material_Concrete", "TENSIL/TENCRV", "BRITTL" )
        setParameter( "MATERIAL", "Material_Concrete", "TENSIL/TENSTR", Concrete.f_ctm )
        setParameter( "MATERIAL", "Material_Concrete", "COMPRS/COMCRV", "EC2" )
        setParameter( "MATERIAL", "Material_Concrete", "COMPRS/COMSTR", Concrete.f_ck )
        setParameter( "MATERIAL", "Material_Concrete", "COMPRS/EPSC1", Concrete.epsilon_c1 )
        setParameter( "MATERIAL", "Material_Concrete", "COMPRS/EPSCU", Concrete.epsilon_cu1 )
        setParameter( "MATERIAL", "Material_Concrete", "COMPRS/YOUNCM", Concrete.E_cm )

# Anchor plates
if Beam.Tendons.Plates.thickness > 0:
        addMaterial( "Material_Steel_AnchorPlate", "MCSTEL", "ISOTRO", [] )
        setParameter( "MATERIAL", "Material_Steel_AnchorPlate", "LINEAR/ELASTI/YOUNG", 210000
)
        setParameter( "MATERIAL", "Material_Steel_AnchorPlate", "LINEAR/ELASTI/POISON", 0.3 )
        setParameter( "MATERIAL", "Material_Steel_AnchorPlate", "LINEAR/MASS/DENSIT", 8.05e-9
)

# Support plates
addMaterial( "Material_Steel_SupportPlate", "MCSTEL", "ISOTRO", [] )
setParameter( "MATERIAL", "Material_Steel_SupportPlate", "LINEAR/ELASTI/YOUNG", 210000 )
setParameter( "MATERIAL", "Material_Steel_SupportPlate", "LINEAR/ELASTI/POISON", 0.3 )
setParameter( "MATERIAL", "Material_Steel_SupportPlate", "LINEAR/MASS/DENSIT", 8.05e-9 )

# Load plates
addMaterial( "Material_Steel_LoadPlate", "MCSTEL", "ISOTRO", [] )
setParameter( "MATERIAL", "Material_Steel_LoadPlate", "LINEAR/ELASTI/YOUNG", 210000 )
setParameter( "MATERIAL", "Material_Steel_LoadPlate", "LINEAR/ELASTI/POISON", 0.3 )
setParameter( "MATERIAL", "Material_Steel_LoadPlate", "LINEAR/MASS/DENSIT", 8.05e-9 )

average_element_size =
Mesh.Element.size*my_ParabolaLength(Beam.Tendons.Coords)[0]/Beam.Prop.length;
#average_element_size = 1;

# Tendons (No bond)
addMaterial( "Material_Tendon_NoBond", "REINFO", "REBOND", [] )
setParameter( "MATERIAL", "Material_Tendon_NoBond", "REBARS/ELASTI/YOUNG", 195000 )
setParameter( "MATERIAL", "Material_Tendon_NoBond", "RESLIP/DSNY",
Beam.Tendons.BS.NoBond.normal_stiffness/average_element_size)
setParameter( "MATERIAL", "Material_Tendon_NoBond", "REBARS/POISON/POISON", 0.3 )
setParameter( "MATERIAL", "Material_Tendon_NoBond", "REBARS/MASS/DENSIT", 7.85e-09 )
if Beam.Tendons.BS.type == "linear":
        setParameter( "MATERIAL", "Material_Tendon_NoBond", "RESLIP/SHFTYP", "NONE" )
```

```
        setParameter( "MATERIAL", "Material_Tendon_NoBond", "RESLIP/DSSX",
Beam.Tendons.BS.NoBond.shear_stiffness/average_element_size)
else:
        setParameter( "MATERIAL", "Material_Tendon_NoBond", "RESLIP/DSSX", 0 )
        setParameter( "MATERIAL", "Material_Tendon_NoBond", "RESLIP/SHFTYP", "BONDS1" )
        setParameter( "MATERIAL", "Material_Tendon_NoBond", "RESLIP/BONDS1/SLPVAL", [ 2e-08,
0.06 ] )

# Tendons (stiff bond)
addMaterial( "Material_Tendon_FullBond", "REINFO", "REBOND", [] )
setParameter( "MATERIAL", "Material_Tendon_FullBond", "REBARS/ELASTI/YOUNG", 195000 )
setParameter( "MATERIAL", "Material_Tendon_FullBond", "RESLIP/DSNY",
Beam.Tendons.BS.FullBond.normal_stiffness/average_element_size)
setParameter( "MATERIAL", "Material_Tendon_FullBond", "REBARS/POISON/POISON", 0.3 )
setParameter( "MATERIAL", "Material_Tendon_FullBond", "REBARS/MASS/DENSIT", 7.85e-09 )
if Beam.Tendons.BS.type == "linear":
        setParameter( "MATERIAL", "Material_Tendon_FullBond", "RESLIP/SHFTYP", "NONE" )
        setParameter( "MATERIAL", "Material_Tendon_FullBond", "RESLIP/DSSX",
Beam.Tendons.BS.FullBond.shear_stiffness/average_element_size)
else:
        setParameter( "MATERIAL", "Material_Tendon_FullBond", "RESLIP/DSSX", 0 )
        setParameter( "MATERIAL", "Material_Tendon_FullBond", "RESLIP/SHFTYP", "BONDS1" )
        setParameter( "MATERIAL", "Material_Tendon_FullBond", "RESLIP/BONDS1/SLPVAL", [ 7.2,
0.06 ] )
        setParameter( "MATERIAL", "Material_Tendon_FullBond", "RESLIP/BONDS1/RESETU", True )

# Tendons plastic range
if Beam.Tendons.f_p > 0:
        list_of_materials = ["Material_Tendon_NoBond", "Material_Tendon_FullBond"]
        for material_name in list_of_materials:
                setParameter( "MATERIAL", material_name, "REBARS/PLATYP", "VMISES" )
                setParameter( "MATERIAL", material_name, "REBARS/PLASTI/TRESSH", "NONE" )
                setParameter( "MATERIAL", material_name, "REBARS/PLASTI/YLDSTR",
Beam.Tendons.f_p )

# Passive reinforcement (Links)
if Beam.Reinf.Links.num > 0:
        addMaterial( "Material_PassiveReinf_Link", "REINFO", "LINEAR", [] )
        setParameter( "MATERIAL", "Material_PassiveReinf_Link", "LINEAR/ELASTI/YOUNG",
Beam.Reinf.links_E )
        if Beam.Reinf.links_f_y > 0:
                setParameter( "MATERIAL", "Material_PassiveReinf_Link", "REBARS/PLATYP",
"VMISES" )
                setParameter( "MATERIAL", "Material_PassiveReinf_Link", "REBARS/PLASTI/TRESSH",
"NONE" )
                setParameter( "MATERIAL", "Material_PassiveReinf_Link", "REBARS/PLASTI/YLDSTR",
Beam.Reinf.links_f_y )

# Passive reinforcement (Main)
if len(Beam.Reinf.Main) > 0:
        addMaterial( "Material_PassiveReinf_Main", "REINFO", "LINEAR", [] )
        setParameter( "MATERIAL", "Material_PassiveReinf_Main", "LINEAR/ELASTI/YOUNG",
Beam.Reinf.main_E )
        if Beam.Reinf.main_f_y > 0:
                setParameter( "MATERIAL", "Material_PassiveReinf_Main", "REBARS/PLATYP",
"VMISES" )
                setParameter( "MATERIAL", "Material_PassiveReinf_Main", "REBARS/PLASTI/TRESSH",
"NONE" )
                setParameter( "MATERIAL", "Material_PassiveReinf_Main", "REBARS/PLASTI/YLDSTR",
Beam.Reinf.main_f_y )

# Connection Interface material - Anchor plates
addMaterial( "Material_Interface_AnchorPlates", "INTERF", "NONLIF", [] )
setParameter( "MATERIAL", "Material_Interface_AnchorPlates", "LINEAR/ELAS6/DSNZ", 1e3 )
setParameter( "MATERIAL", "Material_Interface_AnchorPlates", "LINEAR/ELAS6/DSSX", 1e3 )
setParameter( "MATERIAL", "Material_Interface_AnchorPlates", "NONLIN/IFNOTE", "NOTENS" )
setParameter( "MATERIAL", "Material_Interface_AnchorPlates", "NONLIN/NLEL7/NOTENS", [ 1e-3, 0
] )

# Connection Interface material - Support plates
addMaterial( "Material_Interface_SupportPlates", "INTERF", "NONLIF", [] )
setParameter( "MATERIAL", "Material_Interface_SupportPlates", "LINEAR/ELAS6/DSNZ", 1e3 )
setParameter( "MATERIAL", "Material_Interface_SupportPlates", "LINEAR/ELAS6/DSSX", 1e-3 )
setParameter( "MATERIAL", "Material_Interface_SupportPlates", "NONLIN/IFNOTE", "NOTENS" )
setParameter( "MATERIAL", "Material_Interface_SupportPlates", "NONLIN/NLEL7/NOTENS", [ 1e-3, 0
] )
```

```
# Connection Interface material - Load plates
addMaterial( "Material_Interface_LoadPlates", "INTERF", "NONLIF", [] )
setParameter( "MATERIAL", "Material_Interface_LoadPlates", "LINEAR/ELAS6/DSNZ", 1e3 )
setParameter( "MATERIAL", "Material_Interface_LoadPlates", "LINEAR/ELAS6/DSSX", 1e-3 )
setParameter( "MATERIAL", "Material_Interface_LoadPlates", "NONLIN/IFNOTE", "NOTENS" )
setParameter( "MATERIAL", "Material_Interface_LoadPlates", "NONLIN/NLEL7/NOTENS", [ 1e-3, 0 ]
)

# ---- Auxiliary varaibles ----

# Number of beam ends (with supports)
Beam.Prop.num_of_ends = 2
# Tendons y coordinates
Beam.Tendons.Coords.y =
Beam.Prop.width/(Beam.Tendons.num+1)*np.linspace(1,Beam.Tendons.num,Beam.Tendons.num)
# Initialice number of tendons considered half
Beam.Tendons.num_that_is_half = 0
# Initialice concentrated loads reduction factors
for k in range(1,len(Loads.Concentrated)+1):
        Loads.Concentrated[k].factor = 1
# Main passive definitions
for k in range(1,len(Beam.Reinf.Main)+1):
        Beam.Reinf.Main[k].x_coords = [Beam.Reinf.Main[k].cover, Beam.Prop.length-
Beam.Reinf.Main[k].cover]
        Beam.Reinf.Main[k].y_coords = np.linspace(Beam.Reinf.Main[k].cover,Beam.Prop.width-
Beam.Reinf.Main[k].cover,Beam.Reinf.Main[k].num)
        Beam.Reinf.Main[k].num_that_is_half = 0
# Lins definitions
Beam.Reinf.Links.x_coords = np.linspace(Beam.Reinf.Links.cover,Beam.Prop.length-
Beam.Reinf.Links.cover,Beam.Reinf.Links.num)
Beam.Reinf.Links.num_that_is_half = 0
# Figure saving variables
#Options.SavePictures.name = time.strftime("%Y%m%d%H%M%S") + "_" + Project.name
Options.SavePictures.name_prefix = time.strftime("%Y%m%d%H%M%S") + "_"
Options.SavePictures.name = Project.name
Options.SavePictures.root_folder = Project.path_for_plot_saving+"/"+Project.folder+"/"+"Plots"

# ---- Model symmetry ----

# Options for at least half models
if Options.model_symmetry > 0:
        # Updating number of beam ends
        Beam.Prop.num_of_ends = 1
        # Reducing beam length
        Beam.Prop.length = Beam.Prop.length/2
        # Adapting cocentrated load and plate values if at mid-span
        for k in range(1,len(Loads.Concentrated)+1):
                if Loads.Concentrated[k].coords[0] == Beam.Prop.length:
                        Loads.Concentrated[k].Plate.length =
Loads.Concentrated[k].Plate.length/2
                        Loads.Concentrated[k].coords[0] = Loads.Concentrated[k].coords[0]-
Loads.Concentrated[k].Plate.length/2
                        Loads.Concentrated[k].factor = Loads.Concentrated[k].factor*1/2
        # Adapting main passive reinforcement
        for k in range(1,len(Beam.Reinf.Main)+1):
                Beam.Reinf.Main[k].x_coords = [Beam.Reinf.Main[k].cover, Beam.Prop.length]
        # Updating number of links
        Beam.Reinf.Links.num = round(Beam.Reinf.Links.num/2+np.mod(Beam.Reinf.Links.num,2)/2)
        Beam.Reinf.Links.x_coords = Beam.Reinf.Links.x_coords[0:Beam.Reinf.Links.num]
        if Beam.Reinf.Links.num > 0:
                if Beam.Reinf.Links.x_coords[-1] == Beam.Prop.length:
                        Beam.Reinf.Links.num_that_is_half = Beam.Reinf.Links.num

# Options for quarter models
if Options.model_symmetry == 1/4:
        # Updating beam width
        Beam.Prop.width = Beam.Prop.width/2
        # Adapting tendons
        Beam.Tendons.num = round(Beam.Tendons.num/2+np.mod(Beam.Tendons.num,2)/2)
        Beam.Tendons.Coords.y = Beam.Tendons.Coords.y[0:Beam.Tendons.num]
        if Beam.Tendons.Coords.y[-1] == Beam.Prop.width:
                Beam.Tendons.num_that_is_half = Beam.Tendons.num
        # Adapting concentrated load and plate values
        for k in range(1,len(Loads.Concentrated)+1):
                if Loads.Concentrated[k].coords[1] == Beam.Prop.width:
                        Loads.Concentrated[k].Plate.width = Loads.Concentrated[k].Plate.width/2
```

```
                        Loads.Concentrated[k].coords[1] = Loads.Concentrated[k].coords[1]-
Loads.Concentrated[k].Plate.width/2
                        Loads.Concentrated[k].factor = Loads.Concentrated[k].factor*1/2
        # Adapting main passive reinforcement
        for k in range(1,len(Beam.Reinf.Main)+1):
                Beam.Reinf.Main[k].num =
round(Beam.Reinf.Main[k].num/2+np.mod(Beam.Reinf.Main[k].num,2)/2)
                Beam.Reinf.Main[k].y_coords =
Beam.Reinf.Main[k].y_coords[0:Beam.Reinf.Main[k].num]
                if Beam.Reinf.Main[k].y_coords[-1] == Beam.Prop.width:
                        Beam.Reinf.Main[k].num_that_is_half = k

# ---- Geometries ----

# Tendon
if Beam.Tendons.num_that_is_half != 1:
        addGeometry( "Geometry_Tendon", "RELINE", "REBAR", [] )
        setParameter( "GEOMET", "Geometry_Tendon", "REITYP", "REITRU" )
        setParameter( "GEOMET", "Geometry_Tendon", "REITRU/CROSSE", Beam.Tendons.area )

# Half tendon
if Beam.Tendons.num_that_is_half > 0:
        addGeometry( "Geometry_Tendon_half", "RELINE", "REBAR", [] )
        setParameter( "GEOMET", "Geometry_Tendon_half", "REITYP", "REITRU" )
        setParameter( "GEOMET", "Geometry_Tendon_half", "REITRU/CROSSE", Beam.Tendons.area/2 )

# Passive reinforcement (Main)
for main_set_num in range(1,len(Beam.Reinf.Main)+1):
        reinf_name = "PassiveReinf_Main_Phi_"+str(Beam.Reinf.Main[main_set_num].diameter)
        addGeometry( "Geometry_"+reinf_name, "RELINE", "REBAR", [] )
        setParameter( "GEOMET", "Geometry_"+reinf_name, "REITYP", "REIEMB" )
        setParameter( "GEOMET", "Geometry_"+reinf_name, "REIEMB/RDITYP", "RDIAME" )
        setParameter( "GEOMET", "Geometry_"+reinf_name, "REIEMB/DIAMET",
Beam.Reinf.Main[main_set_num].diameter)
        # Half passive reinforcement (Main)
        if Beam.Reinf.Main[main_set_num].num_that_is_half > 0:
                reinf_name =
"PassiveReinf_Main_Phi_"+str(Beam.Reinf.Main[main_set_num].diameter)+"_half"
                addGeometry( "Geometry_"+reinf_name, "RELINE", "REBAR", [] )
                setParameter( "GEOMET", "Geometry_"+reinf_name, "REITYP", "REIEMB" )
                setParameter( "GEOMET", "Geometry_"+reinf_name, "REIEMB/RDITYP", "RDIAME" )
                setParameter( "GEOMET", "Geometry_"+reinf_name, "REIEMB/DIAMET",
Beam.Reinf.Main[main_set_num].diameter/sqrt(2))

# Passive reinforcement (Links)
if Beam.Reinf.Links.num > 0:
        reinf_name = "PassiveReinf_Link_Phi_"+str(Beam.Reinf.Links.diameter)
        addGeometry( "Geometry_"+reinf_name, "RELINE", "REBAR", [] )
        setParameter( "GEOMET", "Geometry_"+reinf_name, "REITYP", "REIEMB" )
        setParameter( "GEOMET", "Geometry_"+reinf_name, "REIEMB/RDITYP", "RDIAME" )
        setParameter( "GEOMET", "Geometry_"+reinf_name, "REIEMB/DIAMET",
Beam.Reinf.Links.diameter )
        if Beam.Reinf.Links.num_that_is_half > 0:
                reinf_name = "PassiveReinf_Link_Phi_"+str(Beam.Reinf.Links.diameter)+"_half"
                addGeometry( "Geometry_"+reinf_name, "RELINE", "REBAR", [] )
                setParameter( "GEOMET", "Geometry_"+reinf_name, "REITYP", "REIEMB" )
                setParameter( "GEOMET", "Geometry_"+reinf_name, "REIEMB/RDITYP", "RDIAME" )
                setParameter( "GEOMET", "Geometry_"+reinf_name, "REIEMB/DIAMET",
Beam.Reinf.Links.diameter/sqrt(2) )

# ---- Shapes ----

Lists = StructType()
Lists.meshshapeset = []
Lists.elementset = []

# ShapeSets
addSet("SHAPESET","ShapeSet_Beam"); Lists.meshshapeset.append("ShapeSet_Beam")
addSet("SHAPESET","ShapeSet_Tendons"); Lists.meshshapeset.append("ShapeSet_Tendons")
addSet("SHAPESET","ShapeSet_PassiveReinf_Main");
Lists.meshshapeset.append("ShapeSet_PassiveReinf_Main")
addSet("SHAPESET","ShapeSet_PassiveReinf_Links");
Lists.meshshapeset.append("ShapeSet_PassiveReinf_Links")
addSet("SHAPESET","ShapeSet_Plates"); Lists.meshshapeset.append("ShapeSet_Plates")
remove("SHAPESET", [ "Shapes" ] )

# Beam
```

```
createBlock( "Shape_Beam", [ 0, 0, 0 ], [ Beam.Prop.length, Beam.Prop.width, Beam.Prop.height
] )
moveToShapeSet(["Shape_Beam"],"ShapeSet_Beam")
setElementClassType( "SHAPE", [ "Shape_Beam" ], "STRSOL" )
assignMaterial( "Material_Concrete", "SHAPE", [ "Shape_Beam" ] )

# Individual Tendons
Beam.Tendon = {}
for k in range(1,Beam.Tendons.num+1):

        # Tendon properties
        Beam.Tendon[k] = StructType()
        Beam.Tendon[k].Coords = StructType()
        #Beam.Tendon[k].Coords = Beam.Tendons.Coords # why this is not working?
        Beam.Tendon[k].Coords.x = Beam.Tendons.Coords.x
        Beam.Tendon[k].Coords.z = Beam.Tendons.Coords.z
        Beam.Tendon[k].Coords.y = (Beam.Tendon[k].Coords.x*0+1)*Beam.Tendons.Coords.y[k-1]
        Beam.Tendon[k].name = "Shape_Tendon_"+str(k)
        Beam.Tendon[k].is_half = 0
        Beam.Tendon[k].factor = 1
        Beam.Tendon[k].aux_width = 0
        if Beam.Tendon[k].Coords.y[0] == Beam.Prop.width:
                Beam.Tendon[k].is_half = 1
                Beam.Tendon[k].factor = 1/2
                Beam.Tendon[k].aux_width = Beam.Tendons.Plates.width/4

        # Shape
        [coordinates, coordinates_in_one_row] =
my_Parabola3Points(Beam.Tendon[k].Coords,Beam.Tendons.Coords.num_of_points,Options.model_symme
try)
        createCurve(Beam.Tendon[k].name, coordinates )
        moveToShapeSet([Beam.Tendon[k].name],"ShapeSet_Tendons")
        setShapeType( "REINFORCEMENTSHAPE", Beam.Tendon[k].name )
        setReinforcementType( "REINFORCEMENTSHAPE", Beam.Tendon[k].name, "TRUSS_BOND_SLIP" )
        assignMaterial( "Material_Tendon_NoBond", "REINFORCEMENTSHAPE", Beam.Tendon[k].name )
        if Beam.Tendon[k].is_half == 0:
                assignGeometry( "Geometry_Tendon", "REINFORCEMENTSHAPE", Beam.Tendon[k].name )
        elif Beam.Tendon[k].is_half == 1:
                assignGeometry( "Geometry_Tendon_half", "REINFORCEMENTSHAPE",
Beam.Tendon[k].name )

        # Anchor plates
        if Beam.Tendons.Plates.thickness > 0:
                createBlock( "Shape_AnchorPlate_"+str(k)+"_1",
                        [Beam.Tendon[k].Coords.x[0]-Beam.Tendons.Plates.thickness,
Beam.Tendon[k].Coords.y[0]-Beam.Tendons.Plates.width/2, Beam.Tendon[k].Coords.z[0]-
Beam.Tendons.Plates.height/2],
                        [Beam.Tendons.Plates.thickness,
Beam.Tendons.Plates.width*Beam.Tendon[k].factor, Beam.Tendons.Plates.height])
                moveToShapeSet(["Shape_AnchorPlate_"+str(k)+"_1"],"ShapeSet_Plates")
                setElementClassType( "SHAPE", [ "Shape_AnchorPlate_"+str(k)+"_1"], "STRSOL" )
                assignMaterial( "Material_Steel_AnchorPlate", "SHAPE", [
"Shape_AnchorPlate_"+str(k)+"_1"] )
                if Options.model_symmetry == 0:
                        createBlock( "Shape_AnchorPlate_"+str(k)+"_2",
                                [Beam.Tendon[k].Coords.x[-1], Beam.Tendon[k].Coords.y[-1]-
Beam.Tendons.Plates.width/2, Beam.Tendon[k].Coords.z[-1]-Beam.Tendons.Plates.height/2],
                                [Beam.Tendons.Plates.thickness, Beam.Tendons.Plates.width,
Beam.Tendons.Plates.height])
                        moveToShapeSet(["Shape_AnchorPlate_"+str(k)+"_2"],"ShapeSet_Plates")
                        setElementClassType( "SHAPE", [ "Shape_AnchorPlate_"+str(k)+"_2"],
"STRSOL" )
                        assignMaterial( "Material_Steel_AnchorPlate", "SHAPE", [
"Shape_AnchorPlate_"+str(k)+"_2" ] )

        # Tying points for the tendons
        createPointBody( "AuxPoint_1", [Beam.Tendon[k].Coords.x[0]-
Beam.Tendons.Plates.thickness*2, Beam.Tendon[k].Coords.y[0], Beam.Tendon[k].Coords.z[0]])
        createPointBody( "AuxPoint_2", [Beam.Tendon[k].Coords.x[-
1]+Beam.Tendons.Plates.thickness*2, Beam.Tendon[k].Coords.y[-1], Beam.Tendon[k].Coords.z[-1]])

        if Beam.Tendons.Plates.thickness == 0:
                projection( "SHAPEFACE", "Shape_Beam",
                        [[Beam.Tendon[k].Coords.x[0], Beam.Tendon[k].Coords.y[0],
Beam.Tendon[k].Coords.z[0]]],
                        [ "AuxPoint_1" ], [ 1, 0, 0 ], True )
                if Options.model_symmetry == 0:
```

```
                        projection( "SHAPEFACE", "Shape_Beam",
                                [[Beam.Tendon[k].Coords.x[-1], Beam.Tendon[k].Coords.y[0],
Beam.Tendon[k].Coords.z[0]]],
                                [ "AuxPoint_2" ], [ -1, 0, 0 ], True )
        elif Beam.Tendons.Plates.thickness > 0:
                projection( "SHAPEFACE", "Shape_AnchorPlate_"+str(k)+"_1",
                        [[Beam.Tendon[k].Coords.x[0]-Beam.Tendons.Plates.thickness,
Beam.Tendon[k].Coords.y[0], Beam.Tendon[k].Coords.z[0]]],
                        [ "AuxPoint_1" ], [ 1, 0, 0 ], True )
                if Options.model_symmetry == 0:
                        projection( "SHAPEFACE", "Shape_AnchorPlate_"+str(k)+"_2",
                                [[Beam.Tendon[k].Coords.x[-1]+Beam.Tendons.Plates.thickness,
Beam.Tendon[k].Coords.y[0], Beam.Tendon[k].Coords.z[0]]],
                                [ "AuxPoint_2" ], [ -1, 0, 0 ], True )
        removeShape( [ "AuxPoint_1" ] )
        removeShape( [ "AuxPoint_2" ] )

# Concentrated load plates
for load_num in range(1,len(Loads.Concentrated)+1):
        createBlock("Shape_LoadPlate_"+str(load_num),
                [Loads.Concentrated[load_num].coords[0]-
Loads.Concentrated[load_num].Plate.length/2, Loads.Concentrated[load_num].coords[1]-
Loads.Concentrated[load_num].Plate.width/2, Loads.Concentrated[load_num].coords[2]],
                [Loads.Concentrated[load_num].Plate.length,
Loads.Concentrated[load_num].Plate.width, Loads.Concentrated[load_num].Plate.thickness])
        moveToShapeSet(["Shape_LoadPlate_"+str(load_num)],"ShapeSet_Plates")
        setElementClassType( "SHAPE", [ "Shape_LoadPlate_"+str(load_num)], "STRSOL" )
        assignMaterial( "Material_Steel_LoadPlate", "SHAPE", [
"Shape_LoadPlate_"+str(load_num)] )

# Passive reinforcement (Main)
for main_set_num in range(1,len(Beam.Reinf.Main)+1):
        for bar_num in range(1,Beam.Reinf.Main[main_set_num].num+1):
                reinf_name = "PassiveReinf_Main_"+str(main_set_num)+"_"+str(bar_num)
                start_coord = [Beam.Reinf.Main[k].x_coords[0],
Beam.Reinf.Main[main_set_num].y_coords[bar_num-1], Beam.Reinf.Main[main_set_num].z_coord]
                end_coord = [Beam.Reinf.Main[k].x_coords[1],
Beam.Reinf.Main[main_set_num].y_coords[bar_num-1], Beam.Reinf.Main[main_set_num].z_coord]
                createLine("Shape_"+reinf_name, start_coord, end_coord)
                moveToShapeSet(["Shape_"+reinf_name],"ShapeSet_PassiveReinf_Main")
                setShapeType( "REINFORCEMENTSHAPE", "Shape_"+reinf_name )
                #convertToReinforcement( [ "Shape_"+reinf_name ] )
                assignMaterial( "Material_PassiveReinf_Main", "SHAPE", [ "Shape_"+reinf_name ]
)
                geometry_name =
"Geometry_PassiveReinf_Main_Phi_"+str(Beam.Reinf.Main[main_set_num].diameter)
                if bar_num == Beam.Reinf.Main[main_set_num].num_that_is_half:
                        geometry_name =  geometry_name+"_half"
                assignGeometry( geometry_name, "SHAPE", [ "Shape_"+reinf_name ] )
                resetElementData( SHAPE, [ "Shape_"+reinf_name] )

# Passive reinforcement (Links)
if Beam.Reinf.Links.num > 0:
        for link_num in range(1,Beam.Reinf.Links.num+1):
                reinf_name = "PassiveReinf_Link_"+str(link_num)
                if Options.model_symmetry == 0 or Options.model_symmetry == 1/2:
                        createPolyline( "Shape_"+reinf_name,
                                [ [ Beam.Reinf.Links.x_coords[link_num-1],
Beam.Reinf.Links.cover, Beam.Reinf.Links.cover ],
                                [ Beam.Reinf.Links.x_coords[link_num-1], Beam.Reinf.Links.cover,
Beam.Prop.height-Beam.Reinf.Links.cover ],
                                [ Beam.Reinf.Links.x_coords[link_num-1], Beam.Prop.width-
Beam.Reinf.Links.cover, Beam.Prop.height-Beam.Reinf.Links.cover ],
                                [ Beam.Reinf.Links.x_coords[link_num-1], Beam.Prop.width-
Beam.Reinf.Links.cover, Beam.Reinf.Links.cover ] ], True )
                elif Options.model_symmetry == 1/4:
                        createPolyline( "Shape_"+reinf_name,
                                [ [ Beam.Reinf.Links.x_coords[link_num-1], Beam.Prop.width,
Beam.Reinf.Links.cover ],
                                [ Beam.Reinf.Links.x_coords[link_num-1], Beam.Reinf.Links.cover,
Beam.Reinf.Links.cover ],
                                [ Beam.Reinf.Links.x_coords[link_num-1], Beam.Reinf.Links.cover,
Beam.Prop.height-Beam.Reinf.Links.cover ],
                                [ Beam.Reinf.Links.x_coords[link_num-1], Beam.Prop.width,
Beam.Prop.height-Beam.Reinf.Links.cover ] ], False )
                moveToShapeSet(["Shape_"+reinf_name],"ShapeSet_PassiveReinf_Links")
                setShapeType( "REINFORCEMENTSHAPE", "Shape_"+reinf_name )
```

```
                #convertToReinforcement( [ "Shape_"+reinf_name ] )
                assignMaterial( "Material_PassiveReinf_Link", "SHAPE", [ "Shape_"+reinf_name ]
)
                geometry_name =
"Geometry_PassiveReinf_Link_Phi_"+str(Beam.Reinf.Links.diameter)
                if link_num == Beam.Reinf.Links.num_that_is_half:
                        geometry_name = geometry_name+"_half"
                assignGeometry( geometry_name, "SHAPE", [ "Shape_"+reinf_name ] )
                resetElementData( SHAPE, [ "Shape_"+reinf_name] )

# ---- Element data ----
if Beam.Tendons.BS.type != "linear":
        addElementData( "ElementData_Interface" )
        setParameter( "DATA", "ElementData_Interface", "INTERF", "TRUSS" )
        for k in range(1,Beam.Tendons.num+1):
                assignElementData( "ElementData_Interface", "REINFORCEMENTSHAPE",
Beam.Tendon[k].name )

# ---- Loads ----

current_LC_num = 0

# Self weight
if Loads.SelfWeight.on==1:
        addSet( "GEOMETRYLOADSET", "LoadSet_SelfWeight" )
        current_LC_num = current_LC_num + 1
        Loads.SelfWeight.LC_num = current_LC_num
        createModelLoad( "Load_Global", "LoadSet_SelfWeight" )

# Prestress
addSet( "GEOMETRYLOADSET", "LoadSet_Prestress" )
Loads.Prestress = StructType()
current_LC_num = current_LC_num + 1
Loads.Prestress.LC_num = current_LC_num
# To handle parabolic or a straight prestressing profile
[theta_left] = my_Parabola3PointsAngle(Beam.Tendons.Coords,Beam.Tendons.Coords.x[0])
[theta_right] = my_Parabola3PointsAngle(Beam.Tendons.Coords,Beam.Tendons.Coords.x[-1])
if theta_left == theta_right:
        # This is straight profile
        P_1_sign_factor = -1
        P_2_sign_factor = 1
        P_1_direction_num = 1
        P_2_direction_num = 1
else:
        # This is a parabolic profile
        addDirection( "Direction_P_1", [ cos(theta_left+pi), 0, sin(theta_left+pi) ] )
        addDirection( "Direction_P_2", [ cos(theta_right), 0, sin(theta_right) ] )
        P_1_sign_factor = 1
        P_2_sign_factor = 1
        P_1_direction_num = 4
        P_2_direction_num = 5
# Left side
if Beam.Tendons.num_that_is_half != 1:
        createPointLoad( "Load_P_1", "LoadSet_Prestress" )
        setParameter( "GEOMETRYLOAD", "Load_P_1", "FORCE/VALUE",
P_1_sign_factor*Beam.Tendons.force_P )
        setParameter( "GEOMETRYLOAD", "Load_P_1", "FORCE/DIRECT", P_1_direction_num )
if Beam.Tendons.num_that_is_half > 0:
        createPointLoad( "Load_P_1_half", "LoadSet_Prestress" )
        setParameter( "GEOMETRYLOAD", "Load_P_1_half", "FORCE/VALUE",
P_1_sign_factor*Beam.Tendons.force_P/2 )
        setParameter( "GEOMETRYLOAD", "Load_P_1_half", "FORCE/DIRECT", P_1_direction_num )
for k in range(1,Beam.Tendons.num+1):
        if Beam.Tendon[k].is_half == 0:
                attach( "GEOMETRYLOAD", "Load_P_1", Beam.Tendon[k].name, [ [
Beam.Tendon[k].Coords.x[0], Beam.Tendon[k].Coords.y[0], Beam.Tendon[k].Coords.z[0] ] ] )
        elif Beam.Tendon[k].is_half == 1:
                attach( "GEOMETRYLOAD", "Load_P_1_half", Beam.Tendon[k].name, [ [
Beam.Tendon[k].Coords.x[0], Beam.Tendon[k].Coords.y[0], Beam.Tendon[k].Coords.z[0] ] ] )
# Right side
if Options.model_symmetry == 0:
        createPointLoad( "Load_P_2", "LoadSet_Prestress" )
        setParameter( "GEOMETRYLOAD", "Load_P_2", "FORCE/VALUE",
P_2_sign_factor*Beam.Tendons.force_P )
        setParameter( "GEOMETRYLOAD", "Load_P_2", "FORCE/DIRECT", P_2_direction_num )
        for k in range(1,Beam.Tendons.num+1):
```

```
                attach( "GEOMETRYLOAD", "Load_P_2", Beam.Tendon[k].name, [ [
Beam.Tendon[k].Coords.x[-1], Beam.Tendon[k].Coords.y[-1], Beam.Tendon[k].Coords.z[-1] ] ] )

# Prestress reaction
addSet( "GEOMETRYLOADSET", "LoadSet_Reactions" )
Loads.Reactions = StructType()
current_LC_num = current_LC_num + 1
Loads.Reactions.LC_num = current_LC_num
if Beam.Tendons.Plates.thickness == 0:
        # Left side
        if Beam.Tendons.num_that_is_half != 1:
                createPointLoad( "Load_R_1", "LoadSet_Reactions" )
                setParameter( "GEOMETRYLOAD", "Load_R_1", "FORCE/VALUE", -
P_1_sign_factor*Beam.Tendons.force_P )
                setParameter( "GEOMETRYLOAD", "Load_R_1", "FORCE/DIRECT", P_1_direction_num )
        if Beam.Tendons.num_that_is_half > 0:
                createPointLoad( "Load_R_1_half", "LoadSet_Reactions" )
                setParameter( "GEOMETRYLOAD", "Load_R_1_half", "FORCE/VALUE", -
P_1_sign_factor*Beam.Tendons.force_P/2 )
                setParameter( "GEOMETRYLOAD", "Load_R_1_half", "FORCE/DIRECT",
P_1_direction_num )
        for k in range(1,Beam.Tendons.num+1):
                if Beam.Tendon[k].is_half == 0:
                        attach( "GEOMETRYLOAD", "Load_R_1", "Shape_Beam", [ [
Beam.Tendon[k].Coords.x[0], Beam.Tendon[k].Coords.y[0], Beam.Tendon[k].Coords.z[0] ] ] )
                elif Beam.Tendon[k].is_half == 1:
                        attach( "GEOMETRYLOAD", "Load_R_1_half", "Shape_Beam", [ [
Beam.Tendon[k].Coords.x[0], Beam.Tendon[k].Coords.y[0], Beam.Tendon[k].Coords.z[0] ] ] )
        # Right side
        if Options.model_symmetry == 0:
                createPointLoad( "Load_R_2", "LoadSet_Reactions" )
                setParameter( "GEOMETRYLOAD", "Load_R_2", "FORCE/VALUE", -
P_2_sign_factor*Beam.Tendons.force_P )
                setParameter( "GEOMETRYLOAD", "Load_R_2", "FORCE/DIRECT", P_2_direction_num )
                for k in range(1,Beam.Tendons.num+1):
                        attach( "GEOMETRYLOAD", "Load_R_2", "Shape_Beam", [ [
Beam.Tendon[k].Coords.x[-1], Beam.Tendon[k].Coords.y[-1], Beam.Tendon[k].Coords.z[-1] ] ] )
elif Beam.Tendons.Plates.thickness > 0:
        # Left reactions
        if Beam.Tendons.num_that_is_half != 1:
                createSurfaceLoad( "Load_R_1", "LoadSet_Reactions" )
                setParameter( "GEOMETRYLOAD", "Load_R_1", "FORCE/VALUE", -
P_1_sign_factor*Beam.Tendons.force_P/(Beam.Tendons.Plates.width * Beam.Tendons.Plates.height)
)
                setParameter( "GEOMETRYLOAD", "Load_R_1", "FORCE/DIRECT", P_1_direction_num )
        if Beam.Tendons.num_that_is_half > 0:
                createSurfaceLoad( "Load_R_1_half", "LoadSet_Reactions" )
                setParameter( "GEOMETRYLOAD", "Load_R_1_half", "FORCE/VALUE", -
P_1_sign_factor*Beam.Tendons.force_P/(Beam.Tendons.Plates.width *
Beam.Tendons.Plates.height)/2 )
                setParameter( "GEOMETRYLOAD", "Load_R_1_half", "FORCE/DIRECT",
P_1_direction_num )
        for k in range(1,Beam.Tendons.num+1):
                if Beam.Tendon[k].is_half == 0:
                        attach( "GEOMETRYLOAD", "Load_R_1", "Shape_AnchorPlate_"+str(k)+"_1",
                                [ [ Beam.Tendon[k].Coords.x[0]-Beam.Tendons.Plates.thickness,
Beam.Tendon[k].Coords.y[0], Beam.Tendon[k].Coords.z[0] ] ] )
                elif Beam.Tendon[k].is_half == 1:
                        attach( "GEOMETRYLOAD", "Load_R_1_half",
"Shape_AnchorPlate_"+str(k)+"_1",
                                [ [ Beam.Tendon[k].Coords.x[0]-Beam.Tendons.Plates.thickness,
Beam.Tendon[k].Coords.y[0], Beam.Tendon[k].Coords.z[0] ] ] )
        # Right reactions
        if Options.model_symmetry == 0:
                createSurfaceLoad( "Load_R_2", "LoadSet_Reactions" )
                setParameter( "GEOMETRYLOAD", "Load_R_2", "FORCE/VALUE", -
P_2_sign_factor*Beam.Tendons.force_P/(Beam.Tendons.Plates.width * Beam.Tendons.Plates.height)
)
                setParameter( "GEOMETRYLOAD", "Load_R_2", "FORCE/DIRECT", P_2_direction_num )
                for k in range(1,Beam.Tendons.num+1):
                        attach( "GEOMETRYLOAD", "Load_R_2", "Shape_AnchorPlate_"+str(k)+"_2",
                                [ [ Beam.Tendon[k].Coords.x[-1]+Beam.Tendons.Plates.thickness,
Beam.Tendon[k].Coords.y[-1], Beam.Tendon[k].Coords.z[-1] ] ] )

# Concentrated loads
if len(Loads.Concentrated) > 0:
        addSet( "GEOMETRYLOADSET", "LoadSet_ConcentratedLoads")
```

```
        current_LC_num = current_LC_num + 1
        Loads.Concentrated[1].LC_num = current_LC_num
        for load_num in range(1,len(Loads.Concentrated)+1):
                createSurfaceLoad( "Load_Concentrated_"+str(load_num),
"LoadSet_ConcentratedLoads")
                setParameter( "GEOMETRYLOAD", "Load_Concentrated_"+str(load_num),
                        "FORCE/VALUE",
Loads.Concentrated[load_num].value/(Loads.Concentrated[load_num].Plate.length*Loads.Concentrat
ed[load_num].Plate.width)*Loads.Concentrated[load_num].factor)
                setParameter( "GEOMETRYLOAD", "Load_Concentrated_"+str(load_num),
"FORCE/DIRECT", 3 )
                attach( "GEOMETRYLOAD", "Load_Concentrated_"+str(load_num),
"Shape_LoadPlate_"+str(load_num),
                        [[ Loads.Concentrated[load_num].coords[0],
Loads.Concentrated[load_num].coords[1],
Loads.Concentrated[load_num].coords[2]+Loads.Concentrated[load_num].Plate.thickness]] )

# ---- Tying ----

addSet( "GEOMETRYTYINGSET", "TyingSet" )

for k in range(1,Beam.Tendons.num+1):
        for j in range(1,Beam.Prop.num_of_ends+1):
                suffix = str(k)+"_"+str(j)

                createPointTying( "Tying_"+suffix , "TyingSet" )
                setParameter( "GEOMETRYTYING", "Tying_"+suffix, "AXES", [ 1, 2 ] )
                setParameter( "GEOMETRYTYING", "Tying_"+suffix, "TRANSL", [ 1, 1, 1 ] )
                setParameter( "GEOMETRYTYING", "Tying_"+suffix, "ROTATI", [ 0, 0, 0 ] )
                if j==1:
                        index = 0
                        sign_factor = -1
                elif j== 2:
                        index = -1
                        sign_factor = +1
                if Beam.Tendons.Plates.thickness == 0:
                        attachTo( "GEOMETRYTYING", "Tying_"+suffix, "SLAVE", "Shape_Beam" ,
                                [ [ Beam.Tendon[k].Coords.x[index],
Beam.Tendon[k].Coords.y[index], Beam.Tendon[k].Coords.z[index] ] ] )
                        attachTo( "GEOMETRYTYING", "Tying_"+suffix, "MASTER",
Beam.Tendon[k].name,
                                [ [ Beam.Tendon[k].Coords.x[index],
Beam.Tendon[k].Coords.y[index], Beam.Tendon[k].Coords.z[index] ] ] )
                elif Beam.Tendons.Plates.thickness > 0:
                        attachTo( "GEOMETRYTYING", "Tying_"+suffix, "SLAVE",
"Shape_AnchorPlate_"+suffix ,
                                [ [
Beam.Tendon[k].Coords.x[index]+sign_factor*Beam.Tendons.Plates.thickness,
Beam.Tendon[k].Coords.y[index], Beam.Tendon[k].Coords.z[index] ] ] )
                        attachTo( "GEOMETRYTYING", "Tying_"+suffix, "MASTER",
Beam.Tendon[k].name,
                                [ [ Beam.Tendon[k].Coords.x[index],
Beam.Tendon[k].Coords.y[index], Beam.Tendon[k].Coords.z[index] ] ] )

# ---- Supports with steel plates ----

addSet( "GEOMETRYSUPPORTSET", "SupportSet" )

# Left support
createBlock( "Shape_SupportPlate_1", [ 0, 0, -Beam.Supports.Plates.thickness ],
        [ Beam.Supports.Plates.length, Beam.Prop.width, Beam.Supports.Plates.thickness ] )
moveToShapeSet(["Shape_SupportPlate_1"],"ShapeSet_Plates")
createSurfaceSupport( "Support_1", "SupportSet" )
setParameter( "GEOMETRYSUPPORT", "Support_1", "AXES", [ 1, 2 ] )
setParameter( "GEOMETRYSUPPORT", "Support_1", "TRANSL", [ 1, 1, 1 ] )
setParameter( "GEOMETRYSUPPORT", "Support_1", "ROTATI", [ 0, 0, 0 ] )
attach( "GEOMETRYSUPPORT", "Support_1", "Shape_SupportPlate_1",
        [[ Beam.Supports.Plates.length/2, Beam.Prop.width/2, -Beam.Supports.Plates.thickness
]] )
setElementClassType( "SHAPE", [ "Shape_SupportPlate_1" ], "STRSOL" )
assignMaterial( "Material_Steel_SupportPlate", "SHAPE", [ "Shape_SupportPlate_1" ] )

# Right support
if Options.model_symmetry == 0:
        createBlock( "Shape_SupportPlate_2", [ Beam.Prop.length-Beam.Supports.Plates.length,
0, -Beam.Supports.Plates.thickness ],
```

```
                [ Beam.Supports.Plates.length, Beam.Prop.width, Beam.Supports.Plates.thickness
] )
        moveToShapeSet(["Shape_SupportPlate_2"],"ShapeSet_Plates")
        createSurfaceSupport( "Support_2", "SupportSet" )
        setParameter( "GEOMETRYSUPPORT", "Support_2", "AXES", [ 1, 2 ] )
        setParameter( "GEOMETRYSUPPORT", "Support_2", "TRANSL", [ 0, 0, 1 ] )
        setParameter( "GEOMETRYSUPPORT", "Support_2", "ROTATI", [ 0, 0, 0 ] )
        attach( "GEOMETRYSUPPORT", "Support_2", "Shape_SupportPlate_2",
                [[ Beam.Prop.width-Beam.Supports.Plates.length/2, Beam.Prop.width/2, -
Beam.Supports.Plates.thickness ]] )
        setElementClassType( "SHAPE", [ "Shape_SupportPlate_2" ], "STRSOL" )
        assignMaterial( "Material_Steel_SupportPlate", "SHAPE", [ "Shape_SupportPlate_2" ] )

# ---- Symmetry supports ----

# Mid-span symmetry plane conditions
if Options.model_symmetry > 0:
        createSurfaceSupport( "Support_MidSpanSymmetry_Face", "SupportSet" )
        setParameter( "GEOMETRYSUPPORT", "Support_MidSpanSymmetry_Face", "AXES", [ 1, 2 ] )
        setParameter( "GEOMETRYSUPPORT", "Support_MidSpanSymmetry_Face", "TRANSL", [ 1, 0, 0 ]
)
        setParameter( "GEOMETRYSUPPORT", "Support_MidSpanSymmetry_Face", "ROTATI", [ 0, 1, 1 ]
)
        attach( "GEOMETRYSUPPORT", "Support_MidSpanSymmetry_Face", "Shape_Beam", [[
Beam.Prop.length, Beam.Prop.width/2, Beam.Prop.height/2 ]] )
        for k in range(1,len(Loads.Concentrated)+1):
                if Loads.Concentrated[k].coords[0] == (Beam.Prop.length-
Loads.Concentrated[k].Plate.length/2):
                        attach( "GEOMETRYSUPPORT", "Support_MidSpanSymmetry_Face",
"Shape_LoadPlate_"+str(k), [[ Beam.Prop.length, Loads.Concentrated[k].coords[1],
Beam.Prop.height+Loads.Concentrated[k].Plate.thickness/2 ]] )
        createPointSupport( "Support_MidSpanSymmetry_Tendons", "SupportSet" )
        setParameter( "GEOMETRYSUPPORT", "Support_MidSpanSymmetry_Tendons", "AXES", [ 1, 2 ] )
        setParameter( "GEOMETRYSUPPORT", "Support_MidSpanSymmetry_Tendons", "TRANSL", [ 1, 0,
0 ] )
        setParameter( "GEOMETRYSUPPORT", "Support_MidSpanSymmetry_Tendons", "ROTATI", [ 0, 1,
1 ] )
        for k in range(1,Beam.Tendons.num+1):
                attach( "GEOMETRYSUPPORT", "Support_MidSpanSymmetry_Tendons",
"Shape_Tendon_"+str(k), [[ Beam.Tendon[k].Coords.x[1], Beam.Tendon[k].Coords.y[1],
Beam.Tendon[k].Coords.z[1] ]] )

# Mid-Width symmetry plane conditions
if Options.model_symmetry == 1/4:
        createSurfaceSupport( "Support_MidWidthSymmetry_Face", "SupportSet" )
        setParameter( "GEOMETRYSUPPORT", "Support_MidWidthSymmetry_Face", "AXES", [ 1, 2 ] )
        setParameter( "GEOMETRYSUPPORT", "Support_MidWidthSymmetry_Face", "TRANSL", [ 0, 1, 0
] )
        setParameter( "GEOMETRYSUPPORT", "Support_MidWidthSymmetry_Face", "ROTATI", [ 1, 0, 1
] )
        attach( "GEOMETRYSUPPORT", "Support_MidWidthSymmetry_Face", "Shape_Beam", [[
Beam.Prop.length/2, Beam.Prop.width, Beam.Prop.height/2 ]] )
        attach( "GEOMETRYSUPPORT", "Support_MidWidthSymmetry_Face", "Shape_SupportPlate_1", [[
Beam.Supports.Plates.length/2, Beam.Prop.width, -Beam.Supports.Plates.thickness/2 ]] )
        for k in range(1,len(Loads.Concentrated)+1):
                attach( "GEOMETRYSUPPORT", "Support_MidWidthSymmetry_Face",
"Shape_LoadPlate_"+str(k),
                        [[ Loads.Concentrated[k].coords[0], Beam.Prop.width,
Beam.Prop.height+Loads.Concentrated[k].Plate.thickness/2 ]] )
        if Beam.Tendons.num_that_is_half > 0:
                attach( "GEOMETRYSUPPORT", "Support_MidWidthSymmetry_Face",
"Shape_AnchorPlate_"+str(Beam.Tendons.num_that_is_half)+"_1",
                        [[ Beam.Tendon[Beam.Tendons.num_that_is_half].Coords.x[0]-
Beam.Tendons.Plates.thickness/2,  Beam.Tendon[Beam.Tendons.num_that_is_half].Coords.y[0],
Beam.Tendon[Beam.Tendons.num_that_is_half].Coords.z[0] ]] )

# ---- Interfaces between plates and beam ----

# New shapefaces for the beam supports
k = 1
projection( "SHAPEFACE", "Shape_Beam", [ [ Beam.Prop.length/2, Beam.Prop.width/2, 0 ] ],
        [ "Shape_SupportPlate_"+str(k) ], [ 0, 0, 1 ], True )
if Options.model_symmetry == 0:
        k = 2
        projection( "SHAPEFACE", "Shape_Beam", [ [ Beam.Prop.length/2, Beam.Prop.width/2, 0 ]
],
                [ "Shape_SupportPlate_"+str(k) ], [ 0, 0, 1 ], True )
```

```
# New shapefaces for the anchor plates
if Beam.Tendons.Plates.thickness > 0:
	for k in range(1,Beam.Tendons.num+1):
		for j in range(1,Beam.Prop.num_of_ends+1):
			suffix = str(k)+"_"+str(j)
			if j==1:
				index = 0
				sign_factor = 1
				length_factor = 0
			elif j== 2:
				index = -1
				sign_factor = -1
				length_factor = 1
			projection( "SHAPEFACE", "Shape_Beam", [ [
Beam.Tendon[k].Coords.x[index] , Beam.Tendon[k].Coords.y[index] - Beam.Tendon[k].aux_width,
Beam.Tendon[k].Coords.z[index]] ],
				[ "Shape_AnchorPlate_"+suffix ], [ sign_factor, 0, 0 ], True )

# New shapefaces for the load plates
for load_num in range(1,len(Loads.Concentrated)+1):
	projection( "SHAPEFACE", "Shape_Beam", [Loads.Concentrated[load_num].coords],
		["Shape_LoadPlate_"+str(load_num)], [0,0,-1], True)

# Create a Connection Interface
# for the beam supports
for k in range(1,Beam.Prop.num_of_ends+1):
	connection_name = "Connection_Interface_SupportPlate_"+str(k)
	createConnection( connection_name, "INTER", "SHAPEFACE", "SHAPEFACE" )
	Lists.elementset.append(connection_name)
	setParameter( "GEOMETRYCONNECTION", connection_name, "MODE", "CLOSED" )
	setElementClassType( "GEOMETRYCONNECTION", connection_name, "STPLIF" )
	assignMaterial( "Material_Interface_SupportPlates", "GEOMETRYCONNECTION",
connection_name )
	setParameter( "GEOMETRYCONNECTION", connection_name, "FLIP", False )
	if k==1:
		sign_factor = 1
		length_factor = 0
	elif k== 2:
		sign_factor = -1
		length_factor = 1
	attachTo( "GEOMETRYCONNECTION", connection_name, "SOURCE",
"Shape_SupportPlate_"+str(k),
		[ [ Beam.Prop.length*length_factor + sign_factor*Beam.Supports.Plates.length/2,
Beam.Prop.width/2, 0] ] )
	attachTo( "GEOMETRYCONNECTION", connection_name, "TARGET", "Shape_Beam",
		[ [ Beam.Prop.length*length_factor + sign_factor*Beam.Supports.Plates.length/2,
Beam.Prop.width/2, 0] ] )
# for the anchor plates
if Beam.Tendons.Plates.thickness > 0:
	for k in range(1,Beam.Tendons.num+1):
		for j in range(1,Beam.Prop.num_of_ends+1):
			suffix = str(k)+"_"+str(j)
			connection_name = "Connection_Interface_AnchorPlate_"+suffix
			createConnection( connection_name, "INTER", "SHAPEFACE", "SHAPEFACE" )
			Lists.elementset.append(connection_name)
			setParameter( "GEOMETRYCONNECTION", connection_name, "MODE", "CLOSED" )
			setElementClassType( "GEOMETRYCONNECTION", connection_name, "STPLIF" )
			assignMaterial( "Material_Interface_AnchorPlates",
"GEOMETRYCONNECTION", connection_name )
			setParameter( "GEOMETRYCONNECTION", connection_name, "FLIP", False )
			if j==1:
				index = 0
			elif j== 2:
				index = -1
			attachTo( "GEOMETRYCONNECTION", connection_name, "SOURCE",
"Shape_AnchorPlate_"+suffix,
				[ [ Beam.Tendon[k].Coords.x[index] ,
Beam.Tendon[k].Coords.y[index] - Beam.Tendon[k].aux_width , Beam.Tendon[k].Coords.z[index]] ]
)
			attachTo( "GEOMETRYCONNECTION", connection_name, "TARGET",
"Shape_Beam",
				[ [ Beam.Tendon[k].Coords.x[index] ,
Beam.Tendon[k].Coords.y[index] - Beam.Tendon[k].aux_width , Beam.Tendon[k].Coords.z[index]] ]
)
# for the load plates
for load_num in range(1,len(Loads.Concentrated)+1):
	connection_name = "Connection_Interface_LoadPlate_"+str(load_num)
```

```
        createConnection( connection_name, "INTER", "SHAPEFACE", "SHAPEFACE" )
        Lists.elementset.append(connection_name)
        setParameter( "GEOMETRYCONNECTION", connection_name, "MODE", "CLOSED" )
        setElementClassType( "GEOMETRYCONNECTION", connection_name, "STPLIF" )
        assignMaterial( "Material_Interface_LoadPlates", "GEOMETRYCONNECTION", connection_name
)
        setParameter( "GEOMETRYCONNECTION", connection_name, "FLIP", False )
        attachTo( "GEOMETRYCONNECTION", connection_name, "SOURCE",
"Shape_LoadPlate_"+str(load_num),
                [ Loads.Concentrated[load_num].coords  ] )
        attachTo( "GEOMETRYCONNECTION", connection_name, "TARGET", "Shape_Beam",
                [ Loads.Concentrated[load_num].coords  ] )

# ---- Splitting tendons ----

for k in range(1,Beam.Tendons.num+1):

        if Beam.Tendons.Splits.num > 1:
                # Splitting
                splitWire( [ "Shape_Tendon_"+str(k) ], "DIVIDE", [ Beam.Tendons.Splits.num ] )
                # Renaming
                for i in range(Beam.Tendons.Splits.num,1,-1):
                        renameShape( "Shape_Tendon_"+str(k)+" "+str(i-1),
"Shape_Tendon_"+str(k)+"_"+str(i) )
        renameShape( "Shape_Tendon_"+str(k), "Shape_Tendon_"+str(k)+"_"+str(1) )
        # Saving name into a "subclass"
        for i in range(1,Beam.Tendons.Splits.num+1,1):
                Beam.Tendon[k,i] = StructType()
                Beam.Tendon[k,i].name = "Shape_Tendon_"+str(k)+"_"+str(i)

# ---- Voids processing ----

Beam.Tendons.Voids.num = len(Beam.Tendons.Voids.tendon_IDs)
Beam.Tendons.Void = {}
for void_num in range(1,Beam.Tendons.Voids.num+1):
        Beam.Tendons.Void[void_num] = StructType()
        Beam.Tendons.Void[void_num].tendon_ID = Beam.Tendons.Voids.tendon_IDs[void_num-1]
        Beam.Tendons.Void[void_num].split_ID = Beam.Tendons.Voids.split_IDs[void_num-1]

# ---- Load combinations ----

# All loads separatelly with a unit factor
setDefaultGeometryLoadCombinations(  )

# Additional load combination (Prestress + Reactions)
addGeometryLoadCombination( "" )
current_LC_num = current_LC_num + 1
Loads.P_and_R = StructType()
Loads.P_and_R.LC_num = current_LC_num
setGeometryLoadCombinationFactor( "Load combination "+str(Loads.P_and_R.LC_num),
"LoadSet_Prestress", 1 )
setGeometryLoadCombinationFactor( "Load combination "+str(Loads.P_and_R.LC_num),
"LoadSet_Reactions", 1 )

# ---- Mesh ----

# Mesh options
# Beam
setElementSize( [ "Shape_Beam" ], Mesh.Element.size, 0.5, True )
setMesherType( [ "Shape_Beam" ], "HEXQUAD" )
clearMidSideNodeLocation( [ "Shape_Beam" ] )
for k in range(1,Beam.Tendons.num+1):
        # Tendons
        for i in range(1,Beam.Tendons.Splits.num+1,1):
                setElementSize( [ Beam.Tendon[k,i].name ], Mesh.Element.size, 0.5, True )
                setMesherType( [ Beam.Tendon[k,i].name ], "HEXQUAD" )
                clearMidSideNodeLocation( [ Beam.Tendon[k,i].name ] )
        # Passive reinforcement (Main)
        for reinf_set_num in range(1,len(Beam.Reinf.Main)+1):
                for bar_num in range(1,Beam.Reinf.Main[reinf_set_num].num+1):
                        setElementSize( [
"Shape_PassiveReinf_Main_"+str(reinf_set_num)+"_"+str(bar_num) ], Mesh.Element.size, 0.5, True
)
                        setMesherType( [
"Shape_PassiveReinf_Main_"+str(reinf_set_num)+"_"+str(bar_num) ], "HEXQUAD" )
                        clearMidSideNodeLocation( [
"Shape_PassiveReinf_Main_"+str(reinf_set_num)+"_"+str(bar_num) ] )
```

```
        # Passive reinforcement (Links)
        for reinf_num in range(1,Beam.Reinf.Links.num+1):
                setElementSize( [ "Shape_PassiveReinf_Link_"+str(reinf_num) ],
Mesh.Element.size, 0.5, True )
                setMesherType( [ "Shape_PassiveReinf_Link_"+str(reinf_num) ], "HEXQUAD" )
                clearMidSideNodeLocation( [ "Shape_PassiveReinf_Link_"+str(reinf_num) ] )
        # Support plates
        for j in range(1,Beam.Prop.num_of_ends+1):
                suffix = str(j)
                setElementSize( [ "Shape_SupportPlate_"+suffix ],
Mesh.Element.size/Mesh.Plates["Support"].ratio, 0.5, True )
                setMesherType( [ "Shape_SupportPlate_"+suffix ], "HEXQUAD" )
                clearMidSideNodeLocation( [ "Shape_SupportPlate_"+suffix ] )
        # Anchor plates
        if Beam.Tendons.Plates.thickness > 0:
                for j in range(1,Beam.Prop.num_of_ends+1):
                        suffix = str(k)+"_"+str(j)
                        setElementSize( [ "Shape_AnchorPlate_"+suffix ],
Mesh.Element.size/Mesh.Plates["Anchor"].ratio, 0.5, True )
                        setMesherType( [ "Shape_AnchorPlate_"+suffix ], "HEXQUAD" )
                        clearMidSideNodeLocation( [ "Shape_AnchorPlate_"+suffix ] )
        # Load plates
        for load_num in range(1,len(Loads.Concentrated)+1):
                setElementSize( [ "Shape_LoadPlate_"+str(load_num) ],
Mesh.Element.size/Mesh.Plates["Load"].ratio, 0.5, True )
                setMesherType( [ "Shape_LoadPlate_"+str(load_num) ], "HEXQUAD" )
                clearMidSideNodeLocation( [ "Shape_LoadPlate_"+str(load_num) ] )

# Mesh generation
if Options.generate_mesh_on == 1:
        generateMesh( [] )

        # View mesh
        hideView( "GEOM" )
        showView( "MESH" )

# ---- Analysis definition ----

# Analysis name
addAnalysis( "Analysis1" )

# -- Phase 1 -- Ungrouted
addAnalysisCommand( "Analysis1", "PHASE", "Phase 1" )
addAnalysisCommand( "Analysis1", "NONLIN", "Ungrouted" )
exe_num = 0

# Prestressing operation
exe_num = exe_num + 1
cebt = "EXECUT("+str(exe_num)+")"     # Current Execute Block Text
setAnalysisCommandDetail( "Analysis1", "Ungrouted", cebt+"/LOAD/LOADNR", Loads.P_and_R.LC_num
)
renameAnalysisCommandDetail( "Analysis1", "Ungrouted", cebt, "Prestressing operation" )

# Activation of self-weight
if Loads.SelfWeight.on == 1:
        exe_num = exe_num + 1
        cebt = "EXECUT("+str(exe_num)+")"     # Current Execute Block Text
        setAnalysisCommandDetail( "Analysis1", "Ungrouted", cebt+"/EXETYP", "LOAD" )
        renameAnalysisCommandDetail( "Analysis1", "Ungrouted", cebt, "Self-weight activation"
)
        setAnalysisCommandDetail( "Analysis1", "Ungrouted", cebt+"/LOAD/LOADNR",
Loads.SelfWeight.LC_num )
        setAnalysisCommandDetail( "Analysis1", "Ungrouted", cebt+"/ITERAT/MAXITE", 20 )

# -- Phase 2 -- Grouted
if Options.phase2_on == 1:
        addAnalysisCommand( "Analysis1", "PHASE", "Phase 2" )
        addAnalysisCommand( "Analysis1", "NONLIN", "Grouted" )
        exe_num = 0

        # Tendons release
        exe_num = exe_num + 1
        cebt = "EXECUT("+str(exe_num)+")"     # Current Execute Block Text
        removeAnalysisCommandDetail( "Analysis1", "Grouted", cebt )
        setAnalysisCommandDetail( "Analysis1", "Grouted", cebt+"/EXETYP", "START" )
        renameAnalysisCommandDetail( "Analysis1", "Grouted", cebt, "Tendons release" )
        setAnalysisCommandDetail( "Analysis1", "Grouted", cebt+"/START/LOAD/PREVIO", False )
```

```
        setAnalysisCommandDetail( "Analysis1", "Grouted", cebt+"/START/LOAD/ADD", False )
        setAnalysisCommandDetail( "Analysis1", "Grouted", cebt+"/ITERAT/MAXITE", 20 )
        # Maintainging self-weight (if active earlier)
        if Loads.SelfWeight.on == 1:
                setAnalysisCommandDetail( "Analysis1", "Grouted", cebt+"/START/LOAD/ADD", True
)
                setAnalysisCommandDetail( "Analysis1", "Grouted",
cebt+"/START/LOAD/ADD/LOADNR", Loads.SelfWeight.LC_num )

        # Change of tendon material in Phase 2 to fully bonded
        for k in range(1,Beam.Tendons.num+1):
                for i in range(1,Beam.Tendons.Splits.num+1,1):
                        assignMaterialForPhase( "Analysis1", "REINFORCEMENTSHAPE", [
Beam.Tendon[k,i].name ], [ "Phase 2" ], "Material_Tendon_FullBond" )

        # Change of tendon material in Phase 2 to damaged
        for void_num in range(1,Beam.Tendons.Voids.num+1):
                assignMaterialForPhase( "Analysis1", "REINFORCEMENTSHAPE", [
Beam.Tendon[Beam.Tendons.Void[void_num].tendon_ID,Beam.Tendons.Void[void_num].split_ID].name
], [ "Phase 2" ], "Material_Tendon_NoBond" )

# -- Phase 3 -- External Loading
if Options.phase3_on == 1:
        addAnalysisCommand( "Analysis1", "PHASE", "Phase 3" )
        addAnalysisCommand( "Analysis1", "NONLIN", "ExternalLoad" )
        exe_num = 0

        # Start step
        exe_num = exe_num + 1
        cebt = "EXECUT("+str(exe_num)+")"
        setAnalysisCommandDetail( "Analysis1", "ExternalLoad", cebt+"/EXETYP", "START" )
        renameAnalysisCommandDetail( "Analysis1", "ExternalLoad", cebt, "Previous Phase loads"
)
        setAnalysisCommandDetail( "Analysis1", "ExternalLoad", cebt+"/START/LOAD/PREVIO", True
)

        # Block - Concentrated load
        if len(Loads.Concentrated) > 0:
                exe_num = exe_num + 1
                cebt = "EXECUT("+str(exe_num)+")"     # Current Execute Block Text
                setAnalysisCommandDetail( "Analysis1", "ExternalLoad", cebt+"/EXETYP", "LOAD" )
                renameAnalysisCommandDetail( "Analysis1", "ExternalLoad", cebt, "Concentrated
load" )
                setAnalysisCommandDetail( "Analysis1", "ExternalLoad", cebt+"/LOAD/LOADNR",
Loads.Concentrated[1].LC_num )

                if Options.until_failure_on == 1:

                        if len(Options.Solver.step_sizes) > 0:
                                # Custom stepping
                                setAnalysisCommandDetail( "Analysis1", "ExternalLoad",
cebt+"/LOAD/STEPS/STEPTY", "EXPLIC" )
                                setAnalysisCommandDetail( "Analysis1", "ExternalLoad",
cebt+"/LOAD/STEPS/EXPLIC/SIZES", Options.Solver.step_sizes )
                        else:
                                # Automatic stepping
                                setAnalysisCommandDetail( "Analysis1", "ExternalLoad",
cebt+"/LOAD/STEPS/STEPTY", "AUTOMA" )

                        # Maximum number of iterations
                        setAnalysisCommandDetail( "Analysis1", "ExternalLoad",
cebt+"/ITERAT/MAXITE", Options.Solver.max_num_of_iterations )

                        if Options.Solver.arc_length_on == 1:
                                # Arc-length
                                setAnalysisCommandDetail( "Analysis1", "ExternalLoad",
cebt+"/LOAD/STEPS/EXPLIC/ARCLEN", True )
                                addAnalysisCommandDetail( "Analysis1", "ExternalLoad",
cebt+"/LOAD/STEPS/EXPLIC/ARCLEN/REGULA/SET" )

                                # Reference node selection
                                # # Under load plate
                                # node_id = findNodesCloseTo(Loads.Concentrated[1].coords);
node_id = node_id[0]
                                # Top centre of beam
                                node_id =
findNodesCloseTo([Beam.Prop.length,Beam.Prop.width,Beam.Prop.height]); node_id = node_id[0]
```

```
                              # Reference node options
                              setAnalysisCommandDetail( "Analysis1", "ExternalLoad",
cebt+"/LOAD/STEPS/EXPLIC/ARCLEN/REGULA/SET(1)/NODES(1)/RNGNRS", str(node_id) )
                              setAnalysisCommandDetail( "Analysis1", "ExternalLoad",
cebt+"/LOAD/STEPS/EXPLIC/ARCLEN/REGULA/SET(1)/DIRECT", 3 )
                              setAnalysisCommandDetail( "Analysis1", "ExternalLoad",
cebt+"/LOAD/STEPS/EXPLIC/ARCLEN/REGULA/SET(1)/ALPHA", 1 )

                              # Equilibrium iteration
                              setAnalysisCommandDetail( "Analysis1", "ExternalLoad",
cebt+"/ITERAT/CONVER/DISPLA", False )
                              setAnalysisCommandDetail( "Analysis1", "ExternalLoad",
cebt+"/ITERAT/CONVER/FORCE", False )
                              addAnalysisCommandDetail( "Analysis1", "ExternalLoad",
cebt+"/ITERAT/CONVER/ENERGY" )
                              setAnalysisCommandDetail( "Analysis1", "ExternalLoad",
cebt+"/ITERAT/CONVER/ENERGY", True )
                              setAnalysisCommandDetail( "Analysis1", "ExternalLoad",
cebt+"/ITERAT/CONVER/ENERGY/TOLCON", 0.005 )
                              setAnalysisCommandDetail( "Analysis1", "ExternalLoad",
cebt+"/ITERAT/CONVER/ENERGY/TOLABT", 10000 )
                              setAnalysisCommandDetail( "Analysis1", "ExternalLoad",
cebt+"/ITERAT/CONVER/ENERGY/NOCONV", "CONTIN" )

        # Change of tendon material in Phase 3 to fully bonded
        for k in range(1,Beam.Tendons.num+1):
                for i in range(1,Beam.Tendons.Splits.num+1,1):
                        assignMaterialForPhase( "Analysis1", "REINFORCEMENTSHAPE", [
Beam.Tendon[k,i].name ], [ "Phase 3" ], "Material_Tendon_FullBond" )

        # Change of tendon material in Phase 3 to damaged
        for void_num in range(1,Beam.Tendons.Voids.num+1):
                assignMaterialForPhase( "Analysis1", "REINFORCEMENTSHAPE", [
Beam.Tendon[Beam.Tendons.Void[void_num].tendon_ID,Beam.Tendons.Void[void_num].split_ID].name
], [ "Phase 3" ], "Material_Tendon_NoBond" )

# Desactivation of tying for Phase 1
setActiveInPhase( "Analysis1", "GEOMETRYTYINGSET", [ "TyingSet" ], [ "Phase 1" ], False )

# Displaying the whole beam
fitAll(  )

# Saving project
saveProject(  )

# Existing analysis name
analysis_name = analyses(); analysis_name = analysis_name[0]

# Existing commands (without the phase)
command_names = analysisCommands(analysis_name); command_names =
command_names[1:len(command_names):2]

# Running analysis
if Options.run_analysis_on == 1:
        runSolver( [ analysis_name ] )

# ---- End of script ----
```

NTNU

Norwegian University of
Science and Technology