



Tekoälyn käyttö SOFiSTiK-ohjelmistossa

Jussi Putkinen

OPINNÄYTETYÖ
Helmikuu 2024

Rakennustekniikka
Talonrakennustekniikka

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Rakennustekniikka
Talonrakennustekniikka

PUTKINEN, JUSSI:
Tekoälyn käyttö SOFiSTiK-ohjelmistossa

Opinnäytetyö 69 sivua, joista liitteitä 19 sivua
Helmikuu 2024

Tämän opinnäytetyön päätavoitteena oli tutkia tekoälyn hyödyntämismahdollisuuksia SOFiSTiK nimisessä rakenneanalyysiohjelmistossa, sillä tekoälyn käyttö on kasvanut räjähdysmäisesti 2020-luvulla. Työssä perehdyttiin tekoälyn kykyyn tuottaa käyttökelpoista koodia SOFiSTiKin ohjelmointityökaluun, Teddyyn. Opinnäytetyössä otettiin myös kantaa tekoälyn tarjoamaan potentiaaliin rakennesuunnitteluprosessin nopeuttamisessa. Rakenneanalyysiohjelmistona SOFiSTiK mahdollistaa monimutkaisten rakenteiden suunnittelun ja analysoinnin varmistuen turvallisuuden ja tehokkuuden. Sen avulla voidaan myös optimoida rakenteiden mittoja sekä vahvuuksia rasitus-, taipuma- ja dynaamisten analyysien kautta, mikä puolestaan edistää kustannus- ja CO₂-päästösäästöjä. Toimeksiantajana tässä työssä toimi Ramboll Finland Oy.

Kirjallisuustutkimuksessa esitellään SOFiSTiK sekä tekoäly. Tapaustutkimuksessa käsitellään näiden kahden välistä integraatiota, ottaen kantaa niiden tehtäviin työkalun luonnissa, sekä työkalun soveltamista rakennesuunnittelussa. Lisäksi omassa kappaleessaan käsitellään vaihevaiheelta esimerkkikohteiden rakenneanalyysiprosessit sekä otetaan kantaa niissä esiintyviin virheisiin.

Opinnäytetyössä luotiin onnistuneesti kahden esimerkkikohteen rakennemallit sekä niille tulevat kuormitukset kokonaisvaltaisesti tekoälyä hyödyntäen. Näin saatiin konkreettisesti esitettyä tekoälyn tarjoamat hyödyt rakenneanalyysien toteutuksessa. Tekoälynä toimi Chat GPT 4.0:sta räätälöity versio SOFIgpt, joka on erikoistunut toimimaan SOFiSTiKin edellyttämällä tavoilla.

Tekoälyn käyttöä rakenneanalyysien luomisessa testattiin yksinkertaisilla rakenteilla, jolloin ei päästy ottamaan kantaa todellisiin tapauksiin, mutta tutkimuksen tulokset osoittivat tekoälyn tarjoaman potentiaalin. Työssä käytetyn SOFIgpt:n jatkokehitys tulisi tekemään rakennesuunnittelusta vielä enemmän automatisoidun, sekä tarjoaisi käyttäjäystävällisen työkalun rakennesuunnittelijoille.

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Degree Programme in Construction Engineering
Building Construction

PUTKINEN, JUSSI:
Using AI in SOFiSTIK software

Bachelor's thesis 69 pages, appendices 19 pages
February 2024

The aim of this thesis was to explore the potential of AI (artificial intelligence) in structural analysis, a field that has seen a significant increase in AI applications in the 2020s. The study specifically focused on the use of AI with the SOFiSTiK software, which is essential for designing and analyzing complex structures to ensure safety and efficiency. Through thorough investigation, this research aimed to reveal AI's ability to generate usable code for SOFiSTiK's Teddy-editor. Ramboll Finland Oy commissioned this thesis, demonstrating the company's interest in and investment in leveraging AI to enhance structural analysis software like SOFiSTiK.

The research included a literature review of the SOFiSTiK software and AI, followed by a case study that examined the integration between the two, addressing their roles in tool creation and application in structural design. The process and challenges of analyzing structural models for example project were discussed in detail, highlighting errors encountered during the analysis.

The thesis successfully developed structural models and their corresponding loads for two example projects using AI comprehensively, demonstrating the tangible benefits of AI in structural analysis. The AI used was a customized version of Chat GPT 4.0 named SOFIgpt, tailored to operate effectively with SOFiSTiK's requirements. Testing AI in creating structural analyses on simple structures revealed its potential, although real-case applicability was not fully assessed. The findings suggest that further development of SOFIgpt could make structural design more automated.

Key words: sofistik, ai, sofigpt, teddy, chat gpt 4.0

SISÄLLYS

1	JOHDANTO	6
1.1	Tausta	6
1.2	Tavoitteet	6
1.3	Rajaukset	7
1.4	Tutkimuksen menetelmät ja toteutus	8
2	SOFISTIK-OHJELMISTO	9
2.1	SOFISTiK-ohjelmiston osat	9
2.2	Teddy	13
2.3	SOFISTiK-ohjelmiston käyttökohteet rakennusallalla	15
2.4	SOFISTiK-ohjelmiston käyttö tekoälyn yhteydessä	15
3	TEKOÄLY JA KONEOPPIMINEN	17
3.1	Kehitys	17
3.2	Tekoälyn käytön tavoitteet rakennesuunnittelussa	18
3.3	Tekoälyn käytön riskit	19
3.4	Tekoälyn käytön mahdollisuudet	21
4	TEKOÄLYN INTEGROINTI SOFISTIK-OHJELMISTOON	22
4.1	Tekoälyn räätälöinti SOFiSTiKia tukevaksi	22
4.2	Rakennemallin luominen tekoälyä hyödyntäen	24
4.3	Tekoälyn kouluttaminen Teddyä varten	25
5	RAKENNEMALLIEN LUOMINEN TEKOÄLYLLÄ	27
5.1	Lähtötiedot palkkisillassa	28
5.2	Lähtötiedot tukimuurissa	29
5.3	Tekoälyn suorittaman mallinnuksen vaiheet	31
5.4	Tekoälyn tekemät virheet mallintaessa palkkisiltaa	38
5.5	Tekoälyn tekemät virheet mallintaessa tukimuuria	40
5.6	Mallinnuksen erilaisuudet esimerkkikohteissa	43
6	JOHTOPÄÄTÖKSET JA JATKOKEHITYS	45
6.1	Johtopäätökset	45
6.2	Jatkokehitys	46
	LÄHTEET	48
	LIITTEET	51
	Liite 1. Palkkisillan mallinnuksen prosessi	51
	Liite 2. Tukimuurin mallinnuksen prosessi	63
	Liite 3. Valmiit rakennemallit	68

LYHENTEET JA TERMIT

AQB	SOFiSTiKin palkki- tai neliöelementtien suunnittelun aliohjelma
AQUA	SOFiSTiKin poikkileikkaus ja materiaali aliohjelma
ASE	SOFiSTiKin analyysi aliohjelma
API	Ohjelmointirajapinta
BEMESS	SOFiSTiKin palkki- tai neliöelementtien suunnittelun toinen aliohjelma
BIM	Building Information Modelling (rakennuksen tietomallinnus)
ChatGPT 4.0	OpenAI:n luoma tekoäly
DYNA	SOFiSTiKin dynaamisessa analyysissä tai lujuuslaskennassa käytettävä aliohjelma
ELLA	SOFiSTiKin liikennekuormitus aliohjelma
FEM	Finite Element Method (elementtimenetelmä)
IoT	Internet of Things
MAXIMA	SOFiSTiKin kuormitusten yhdistävä aliohjelma
SOFIgpt	ChatGPT 4.0 räätälöityversio
SOFILOAD	SOFiSTiKin kuormitus aliohjelma
SOFIMSHC	SOFiSTiKin geometrinen mallinnus aliohjelma
SOFiSTiK	SOFiSTiK AG:n kaupallinen elementtimenetelmäohjelma
SSD	SOFiSTiKin graafinen käyttöliittymä
STP	SOFiSTiKin Structural Point
Teddy	SOFiSTiKin tekstimuotoinen aliohjelma
TEKLA	Tekla Structures 3D-tietomallinnusohjelma
GRASSHOPPER	Visuaalisen ohjelmoinnin lisäosa Rhino 3D-ohjelmassa
TSV	Sillan tasausviiva

1 JOHDANTO

1.1 Tausta

Tekoäly on 2020-luvulla noussut merkittävästi esiin teknologian aloilla, ja tämä kehitys on herättänyt kiinnostusta myös rakennesuunnittelijoiden keskuudessa. Rakennusalaalla tietotekniikan ja ohjelmistojen kehitys on mahdollistanut rakennelaskelmapuolen siirtymisen yhä enemmän tietoteknisten ohjelmistojen varaan. Tämän seurauksena on avautunut uusia mahdollisuuksia tekoälyn hyödyntämiseen rakennesuunnittelun alalla, ja erityisesti tekoälyn yhdistäminen rakenneanalyysiohjelmistojen kanssa on noussut yhdeksi lupaavimmista kehityssuunnista.

Rakenneanalyysiohjelmistojen käyttö on kasvussa, mikä on osittain selitettävissä niiden tarjoamilla mahdollisuuksilla tehokkaampaan ja tarkempaan suunnitteluun. Nämä ohjelmistot mahdollistavat monimutkaisten rakenteiden kuten taitorakenteiden, esimerkiksi siltojen analysoinnin, tavalla, joka käsinlaskennalla olisi huomattavasti työläämpää, ellei jopa mahdotonta. Tämä on erityisen tärkeää, kun otetaan huomioon rakennesuunnittelun tiukat aikataulut.

Kuitenkin ammattikorkeakouluissa ja yliopistoissa rakenneanalyysiohjelmistojen opetus on vähäistä, mikä jättää ohjelmistojen ja tekoälyn opetteluun suurelta osin työpaikoille tai itseopiskeluun. Tämän seurauksena on tärkeää, että alan koulutusohjelmissa alettaisiin kiinnittämään enemmän huomiota näiden teknologioiden opetukseen, jotta tulevat rakennesuunnittelijat voivat toimia mahdollisimman tehokkaasti työssään.

1.2 Tavoitteet

Tässä opinnäytetyössä keskitytään tekoälyn tuomien mahdollisuuksien hyödyntämiseen SOFiSTiKissa, erityisesti sen kykyyn tuottaa rakennemalleja käyttäen sille annettua visuaalista ja tekstipohjaista dataa. Tutkimuksen päämääränä on selvittää, voiko räätälöity tekoälyjärjestelmä tuottaa virheettömiä rakennemalleja SOFiSTiKiin. Tämän tavoitteen saavuttamiseksi työtä varten kehitetään erityinen

tekoäly, SOFIgpt, joka on suunniteltu tuottamaan mahdollisimman tarkkaa ja virheetöntä dataa SOFiSTiKin sisäiselle tekstipohjaiselle ohjelmointityökalulle, Teddylle.

Tekoälypohjaisen suunnittelutyökalun kehittäminen on keskeinen osa tätä tutkimusta. Tällainen työkalu voi toimia paitsi suunnittelijan apuna, myös nopeuttaa ja tehostaa rakennemallien analysointia itsenäisesti. Tässä opinnäytetyössä tavoitteena ei silti ole rakentaa valmista suunnittelutyökalua, vaan pikemminkin tutkia sen mahdollisuutta sekä siitä tulevia hyötyjä. Tästä syystä työtä varten tuotettava suunnittelutyökalu toimii ikään kuin potentiaalisena mallikappaleena. Lopullisen suunnittelutyökalun avulla voitaisiin siis mallintaa uusia rakennemalleja, että muokata jo olemassa olevia. Tällaisen työkalun kehittäminen edellyttää syvällistä ymmärrystä sekä rakennesuunnittelusta että tekoälyn soveltamisesta tällaisiin tehtäviin.

1.3 Rajaukset

Tässä opinnäytetyössä keskitytään tutkimaan, miten tekoälyä voidaan hyödyntää rakenneanalyysien valmistelussa SOFiSTiKissa. Työssä keskitytään erityisesti rakennemallien toimivuuteen ohjelman sisällä. On tärkeää huomata, että tässä työssä ei käsitellä rakennemallien oikeellisuutta. SOFiSTiK voi teknisesti hyväksyä analyysiin rakennemallin, joka saattaa olla lähtökohtaisesti staattisesti virheellinen. Opinnäytetyön tavoitteena ei ole myöskään luoda valmista suunnittelutyökalua, vaan pikemminkin tutkia ja arvioida nykyisten tekoälyteknologioiden soveltuvuutta ja potentiaalia rakennemallien luomisessa ja analysoinnissa.

Tutkimuksessa tullaan myös tarkastelemaan tekoälyn hyödyntämisen jatkokehitysmahdollisuuksia rakennesuunnittelun alalla. Tämä osio sisältää pohdintaa siitä, miten tekoäly voi tulevaisuudessa edistää rakennesuunnittelun tehokkuutta, tarkkuutta ja innovatiivisuutta. Vaikka tämä opinnäytetyö ei pyrikään kehittämään lopullista suunnittelutyökalua, se luo perustaa tuleville tutkimuksille ja kehitystyölle tekoälyn soveltamisessa rakennetekniikan alalla.

1.4 Tutkimuksen menetelmät ja toteutus

Tässä opinnäytetyössä sovelletaan kahta keskeistä tutkimusmenetelmää: kirjallisuustutkimusta ja tapaustutkimuksia. Kirjallisuustutkimuksen osuudessa keskitytään käsittelemään SOFiSTiKin ominaisuuksia, tekoälyn perusteita ja niiden integroitumisen mahdollisuuksia. Tässä osassa tarkastellaan myös laajemmin tekoälyn roolia rakennesuunnittelussa ja sen soveltuvuutta rakennusalan ohjelmistoihin.

Tapaustutkimukset puolestaan ovat tämän työn empiirinen osuus, jossa kahden konkreettisen esimerkin kautta analysoidaan tekoälyn käyttöä SOFiSTiKissa. Nämä tutkimukset tarjoavat käytännön näkökulmia tekoälyn hyödyntämiseen, ja niiden kautta pyritään ymmärtämään paremmin, pystyykö tekoäly tukemaan rakennesuunnittelua ja millä tavoin.

Opinnäytetyössä esitellään myös käytetyt ohjelmistot ja niiden soveltaminen tutkimuksen kontekstissa. Työn lopuksi arvioidaan saavutettuja tuloksia ja pohditaan tutkimuksen onnistumista sekä sen rajoituksia. Lisäksi työssä esitetään ehdotuksia jatkokehitykselle ja käsitellään mahdollisia riskejä ja haasteita, joita tutkimuksen aikana on kohdattu. Tämän lähestymistavan tavoitteena on tarjota kattava ja monipuolinen kuvaus siitä, miten tekoälyä voidaan hyödyntää rakennesuunnittelussa, erityisesti SOFiSTiKin yhteydessä.

2 SOFiSTiK-OHJELMISTO

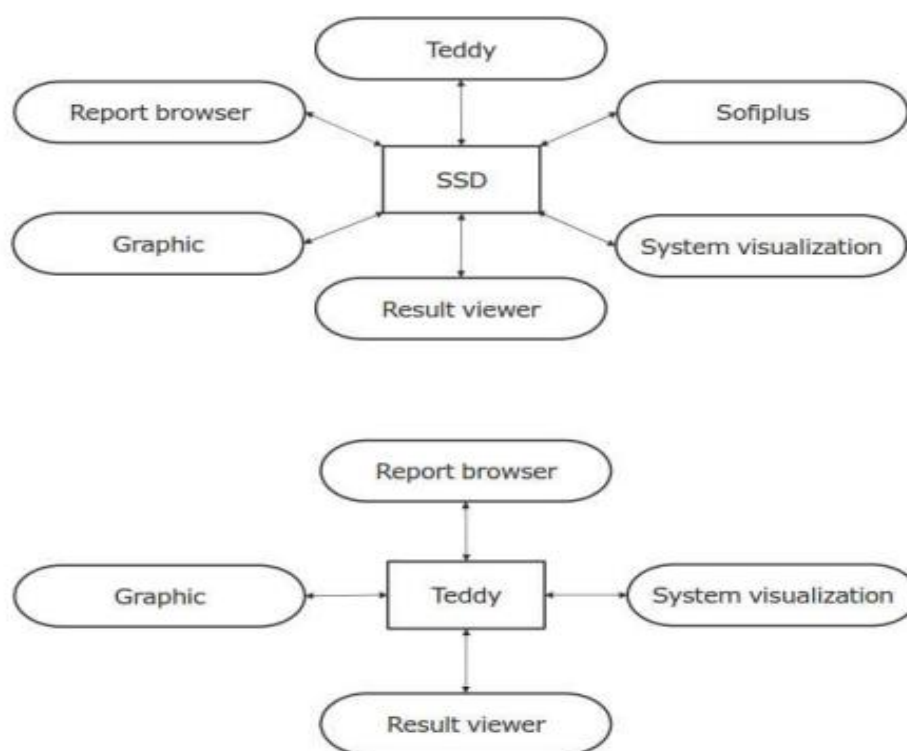
SOFiSTiK on FEM-ohjelmisto, joka on valittu tässä tutkimuksessa rakennemallien luomisen ja analysoinnin työkaluksi. SOFiSTiKin käyttö on keskeistä tässä opinnäytetyössä, erityisesti sen yhteistyö tekoälyn kanssa. Erityisen huomion kohteena on ohjelmiston sisäinen ohjelmointityökalu Teddy, joka mahdollistaa monimutkaisten rakennemallien määrittelyn ja muokkaamisen SOFiSTiKissa pelkkien koodikomentojen avulla.

2.1 SOFiSTiK-ohjelmiston osat

SOFiSTiK AG on yksi Euroopan johtavista rakennusanalyysiohjelmistojen tuottajista (SOFiSTiK AG n.d.), ja sen tarjoama ohjelmisto, SOFiSTiK, tarjoaa kaksi erilaista tapaa luoda rakennemalleja (kuvio 1). Ensimmäinen näistä on SSD (SOFiSTiK Structural Desktop), joka mahdollistaa rakennemallien luomisen graafisen käyttöliittymän kautta. Tämä menetelmä on intuitiivinen ja visuaalisesti suuntautunut, sopien käyttäjille, jotka suosivat enemmän graafista suunnittelua. SOFiSTiK-ohjelmisto sisällyttää ohjelmistoonsa edellä mainittujen lisäksi osat Sofiplus, System Visualization, Result viewer, Graphic ja Report browser. Näistä osista Sofiplus on ainoa, jota voi käyttää vain SSD:llä tehtäessä, muita voi käyttää niin SSD:llä kuin Teddyllä. (SOFiSTiK AG n.d.)

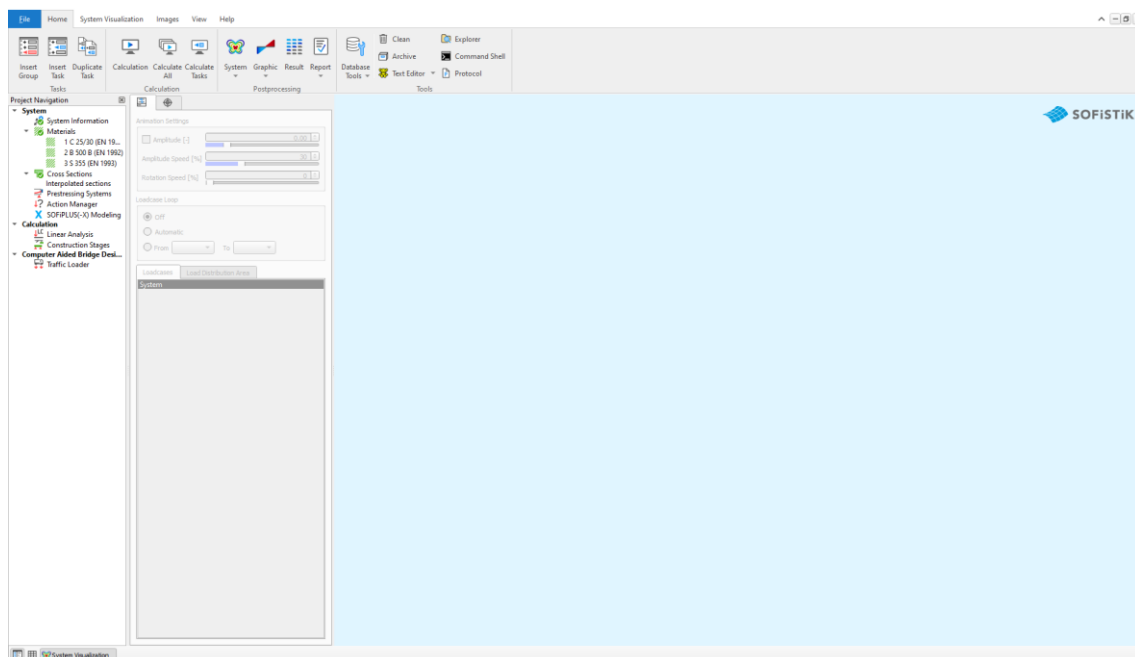
Toinen tapa mallien luomiseen SOFiSTiKissa on käyttää aiemmin mainittua ohjelmointityökalu Teddyä. Teddyn käyttö perustuu tekstipohjaiseen ohjelmointiin SOFiSTiKin luomalla ohjelmointikielellä. Tässä opinnäytetyössä valinta on kallistunut Teddyn käytön puolelle rakennemallien luomisessa, sillä Teddy on helpommin integroitavissa tekoälyyn. Teddyn valinta perustuu myös sen tarjoamaan joustavuuteen ja tarkkuuteen rakennemallien määrittelyssä, mikä on erityisen tärkeää tekoälyn soveltamisessa rakennesuunnittelussa. Tämä lähestymistapa mahdollistaa syvällisemmän kontrollin ja monipuolisemmat mahdollisuudet rakennemallien suunnittelussa, jotka ovat olennaisia tämän opinnäytetyön tavoitteiden saavuttamiseksi. (SOFiSTiK AG n.d.)

Teddyn käyttö mahdollistaa monimutkaisten rakennemallien luomisen ja hallinnan käyttämällä erilaisia SOFiSTiKin sisäisiä aliohjelmia. Näiden ohjelmien avulla voidaan suorittaa monenlaisia toimintoja ja analyysejä. (SOFiSTiK AG n.d.) Tässä opinnäytetyössä käytettävistä aliohjelmistoista kerrotaan enemmän luvussa 2.2.



KUVIO 1 SOFiSTiKin tavat luoda rakennemalleja ja niihin liittyvät ohjelmat (Putkinen 2021, 34).

SOFiSTiK tarjoaa SSD:n, joka on monipuolinen graafinen käyttöliittymä ja toimii rakennemallien hallinnan keskuksena, kuviossa 2 esitetään SSD:n käyttöliittymä. SSD:ssä käyttäjät voivat määrittää materiaaleja, suorittaa erilaisia analyysejä, määrittää kuormituksia ja tehdä mitoitusta. SSD mahdollistaa näiden toimintojen suorittamisen joko käyttämällä Teddyn ohjauskeskukselle syötettävää koodia tai graafisia työkaluja, mikä antaa käyttäjälle joustavuutta valita itselleen sopivimman työskentelytavan. (SOFiSTiK AG n.d.)

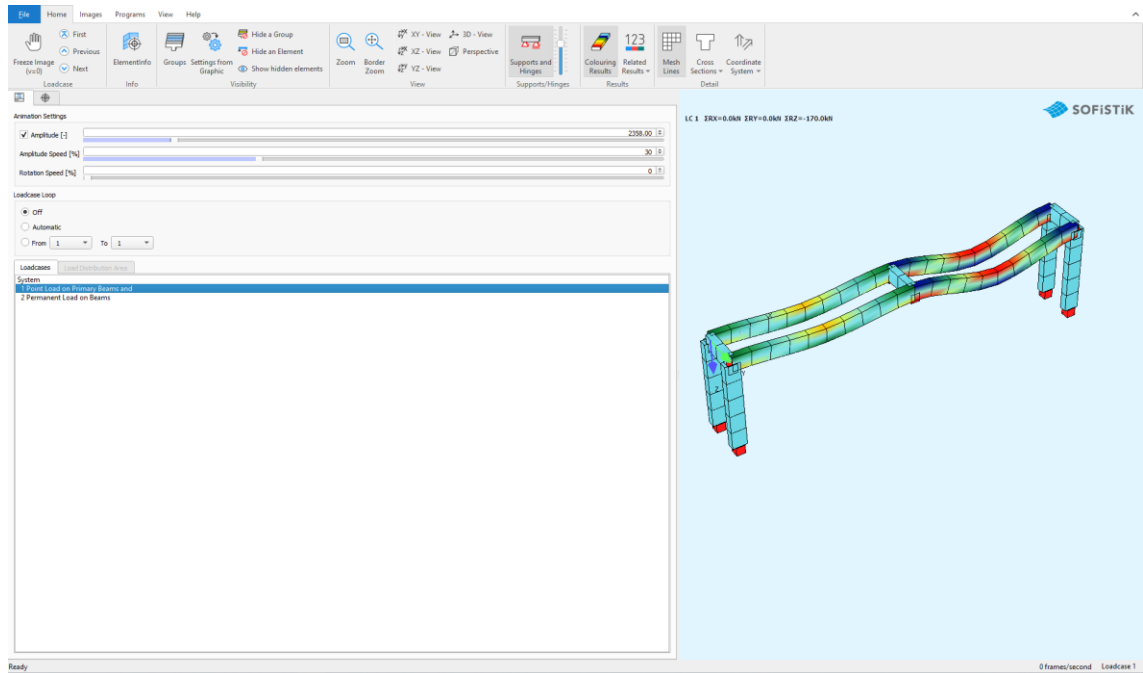


KUVIO 2 SSD:n käyttöliittymä

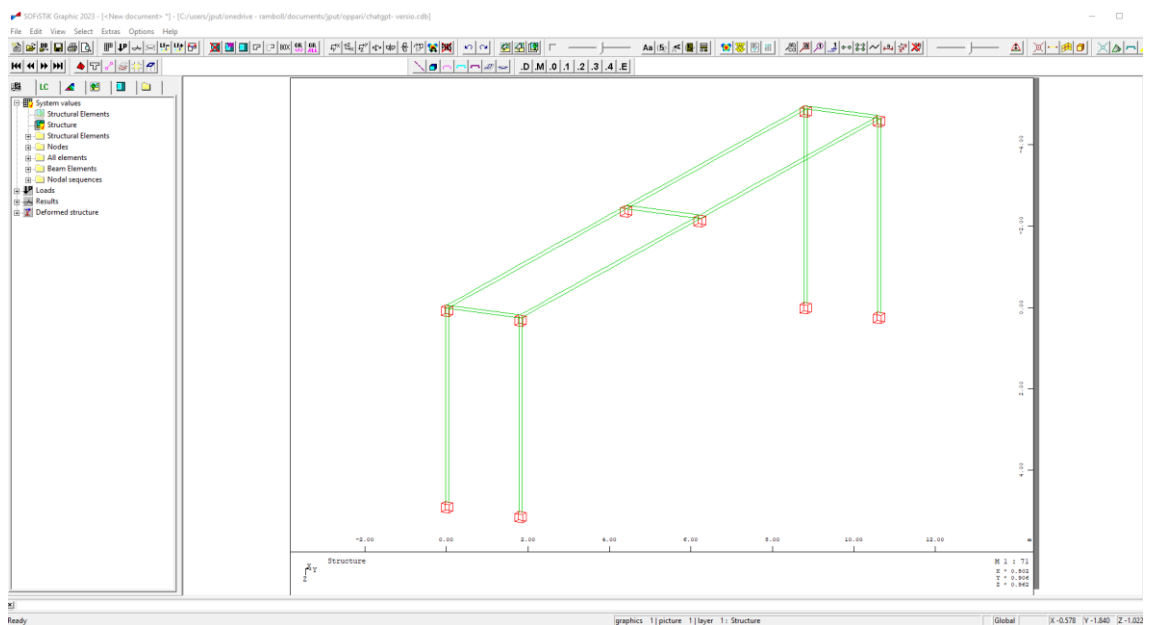
SSD:n ja Teddyn lisäksi SOFiSTiKiin kuuluu myös Sofiplus, System Visualization, Result Viewer, Graphic sekä Report Browser. Sofiplus on AutoCAD-pohjainen käyttöliittymä, joka toimii yhdessä SOFiSTiKin SSD:n kanssa. Se mahdollistaa rakenteiden poikkileikkausten, geometrioiden, tuentojen, jousien, jänteiden ja kuormien luomisen ja hallinnan AutoCAD-ympäristössä. (SOFiSTiK AG n.d.) Sofiplusin integraatio SSD:n kanssa tarjoaa saumattoman työskentelyprosessin rakennesuunnittelussa, yhdistäen toisiinsa AutoCAD:n monipuoliset suunnittelu-työkalut sekä SOFiSTiKin analyysi- ja laskentamoduulit.

System Visualization on SOFiSTiKin sisäinen työkalu, joka on tarkoitettu rakennemallien visualisointiin (kuvio 3). Se mahdollistaa mallin pyörittelyn ja tarkastelun, elementtien piilottamisen ja yksittäisten elementtien tietojen katsomisen. Lisäksi työkalulla voidaan silmämääräisesti tarkastella eri kuormitustapausten aiheuttamia siirtymiä rakenneanalyysin jälkeen. Result Viewer on puolestaan työkalu, joka mahdollistaa poikkileikkausten, raudoitusten, jänteiden, rasiusten, jännitysten, siirtymien, tukivoimien ja mitoitus tulosten tarkastelun. Tarkastelut voidaan esittää esimerkiksi taulukkoina tai diagrammeina. Rakennemallin analysoinnin tarkastelua varten SOFiSTiKista löytyy myös Graphic, jolla voidaan luoda rakennemallin rasituskuvaajia. Lisäksi sitä käytetään esimerkiksi rasiusten tarkasteluun (kuvio 4). (SOFiSTiK AG n.d.) Työkalu on erityisen hyödyllinen rapor-

tointiin, sillä se päivittää automaattisesti kuvat mallin muokkaantuessa. Tämä tarkoittaa, että kuormien muutokset tulevat uuteen raporttiin ilman manuaalista päivitystä.



KUVIO 3. SOFiSTiKin käyttöliittymä System Visualization



KUVIO 4. SOFiSTiKin käyttöliittymä Graphic

SOFiSTiKin osista viimeisenä käsitellään Report Browser, sillä voidaan tuottaa yksityiskohtaisia raportteja jokaisesta lasketusta toiminnosta, olipa kyseessä kuormitus, mitoitus tai poikkileikkaus. Raporttien laajuutta voidaan säätää käyttäjän tarpeiden mukaan, vaihtoehtoja "EXTR" (extreme) tarjoaa kaikkein laajimman tarkastelun ja "NO" suppeimman. (SOFiSTiK AG n.d.)

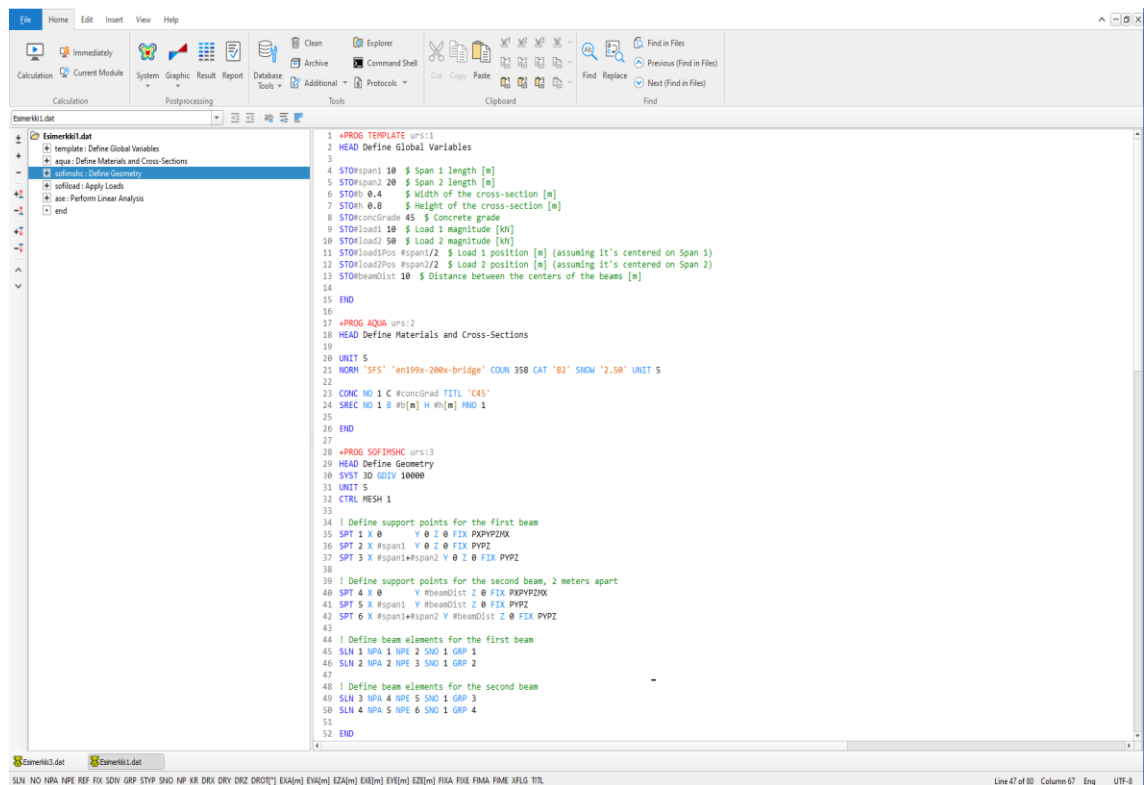
2.2 Teddy

Teddy on tekstipohjainen ohjelmointityökalu sekä ohjauskeskus, joka hyödyntää CADiNP-kieltä. Tässä luvussa käsitellään Teddyn toimintaa syvemmin, sillä se on toinen pääkomponenteista tässä opinnäytetyössä. Teddyn avulla käyttäjät voivat luoda kattavan rakennemallin, määrittää kuormitukset ja suorittaa rakennemallin koko analyysin. (SOFiSTiK AG n.d.)

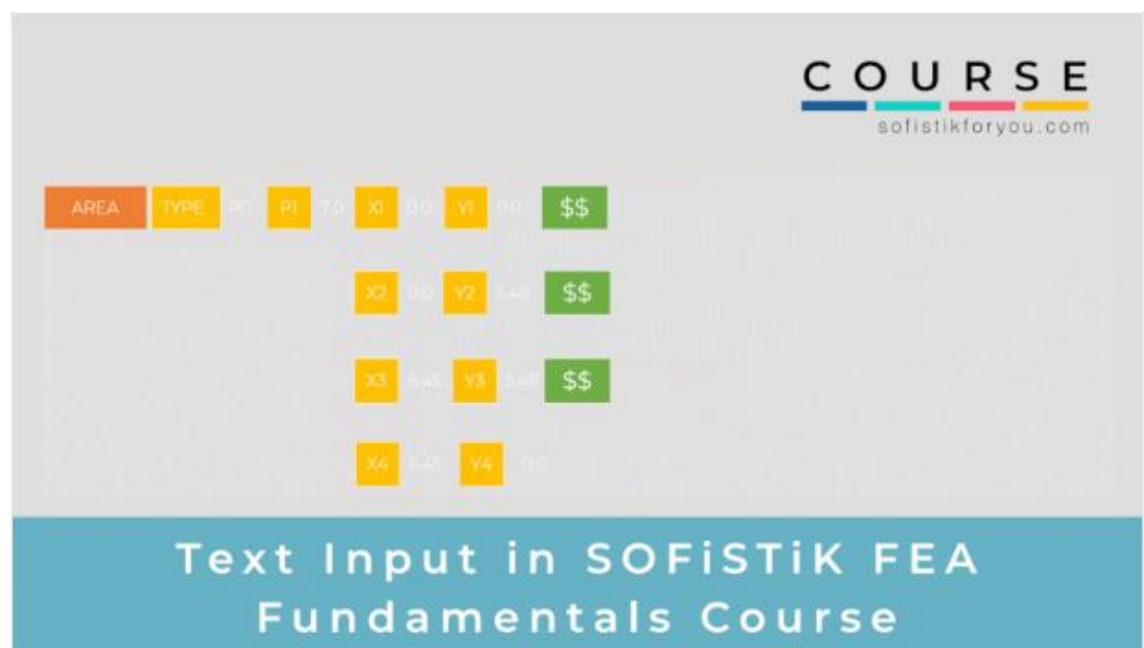
Teddy hyödyntää useita SOFiSTiKin aliohjelmia, kuten AQUA, SOFILOAD, SOFIMSHC ja ASE, jotka kukin suorittavat omat erityistehtävänsä. AQUAn avulla määritellään rakenteiden poikkileikkaukset ja materiaalit, esimerkiksi pilarin dimensiot ja betonilaatu. SOFILOADilla asetetaan rakenteelle akselit ja geometriat, ja ASEn kautta analysoidaan kuormitusten aiheuttamat rasitukset ja suoritetaan vaaditut analyysit. (SOFiSTiK AG n.d.) SOFiSTiKissa on muitakin aliohjelmia kuin edellä mainitut, mutta tässä opinnäytetyössä käytetään vain näitä.

Teddyssä aliohjelmat aktivoidaan lisäämällä komentoketjuun kyseisen ohjelman nimi, esimerkiksi "+ PROG aliohjelman nimi" (kuvio 5). Kukin ohjelma on myös dokumentoitu omassa ohjekirjassaan, joka voidaan avata tekstieditorin rinnalle F1-näppäimen kautta. (SOFiSTiK AG n.d.) Kuviossa 5 esitetään myös Teddyn toimintaympäristö ja ohjelmien integraatio osaksi tätä ympäristöä, sekä tarjotaan visuaalinen kuva siitä, miten Teddy toimii osana SOFiSTiKia ja tukee rakennemallien suunnittelua ja analysointia. Kuten kuviossa 5 huomataan, suoritettava aliohjelma on punaisella värillä ja sen sisällä suoritettava alitehtävä taas sinisellä värillä. Kyseisten tehtävien sisällä harmaalla värillä oleva teksti tulkitaan Teddyn puolesta sen alitehtävän arvoksi. Heittomerkkien sisällä olevan tekstin Teddy tulkitsee pelkäksi tekstiksi. (SOFiSTiK AG n.d.) Huomioitavaa on CADiNP-kielen

sisältämä \$\$-merkki. Tätä käyttämällä voidaan ylittää yhden syöttörivin enimmäismerkkimäärä. Kuviossa 6 on esimerkki syöttörivin katkaisemisesta. \$\$-merkki sallii syöttörivin jakautumisen moneen eri riviin, jolloin yhdessä tehtävässä voisi teoriassa olla loputtomasti dataa. (Sofistikforyou 2022.)



KUVIO 5. Teddyn työympäristö



KUVIO 6. CADiNP-kielen sisältämän \$\$-merkin tarjoama etu (Sofistikforyou 2022)

2.3 SOFiSTiK-ohjelmiston käyttökohteet rakennusallalla

SOFiSTiK on edistynyt ohjelmisto, joka on suunnattu erityisesti rakennesuunnittelijoille, ja joka tarjoaa heille monipuolisia työkaluja rakenteiden suunnitteluun ja analysointiin. Se on erityisen tehokas monimutkaisten rakenteiden, kuten siltojen, pilvenpiirtäjien ja stadionien suunnittelussa. SOFiSTiK mahdollistaa näiden rakenteiden rakenteellisen kestävyys- ja turvallisuuden analysoinnin sisältäen tarkat työkalut kuormitusten laskentaan, värähtelyanalyysiin ja muihin rakenneteknisiin tutkimuksiin (SOFiSTiK AG n.d).

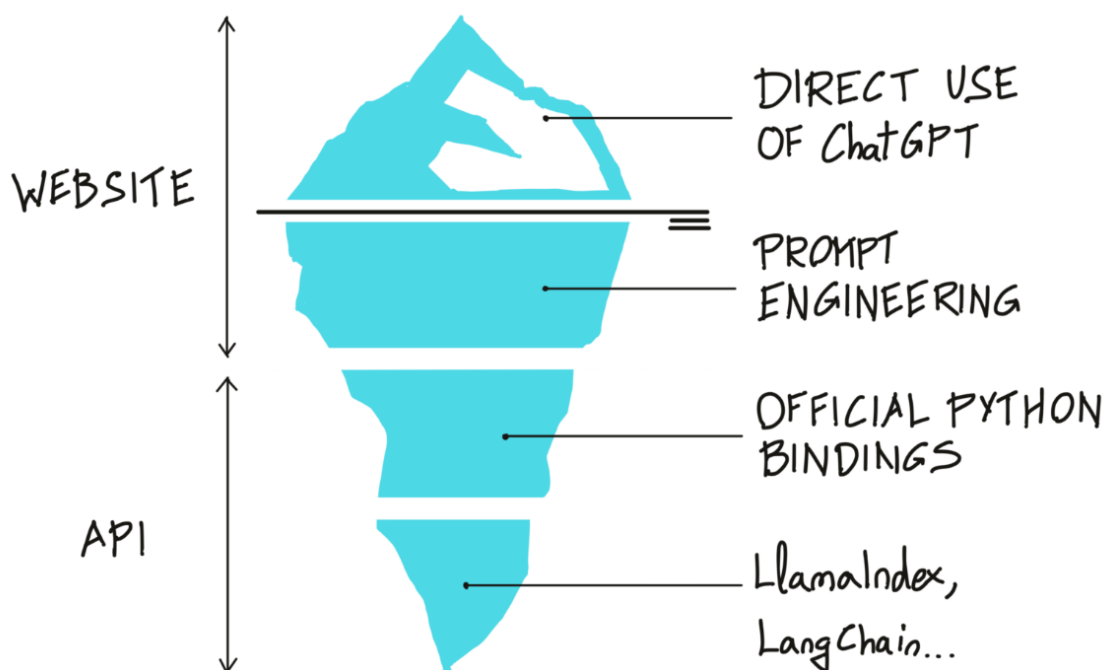
SOFiSTiK integroituu myös saumattomasti BIM-ohjelmistojen kanssa parantaen eri alojen suunnittelijoiden välistä yhteistyötä, tehokkuutta ja vähentäen suunnitteluvirheitä. BIM-integraatio mahdollistaa 3D-mallien hyödyntämisen suunnitelmien visualisoinnissa, mikä auttaa havainnollistamaan rakenteita ennen niiden toteutusta. Tämä integraatio tekee SOFiSTiKista olennaisen työkalun nykyaikaisessa rakennussuunnittelussa. (Yashanov 2020.)

2.4 SOFiSTiK-ohjelmiston käyttö tekoälyn yhteydessä

SOFiSTiKin yhteensopivuus BIM-ohjelmistojen kanssa viittaa sen tehokkuuteen rakennesuunnittelun alalla, mutta tekoälyn käyttöönotto tuo vielä lisää mahdollisuuksia. Tekoäly mahdollistaa tarkemman ja tehokkaamman suunnitteluprosessin, joka perustuu opittuun dataan ja koneoppimiseen. SOFiSTiKiin integroitunut tekoäly voi tarjota useita etuja, kuten automatisoidun suunnitteluprosessin ja siihen liittyvän rakenteiden optimoinnin, joka nopeuttaa sekä tehostaa suunnittelutyötä huomattavasti.

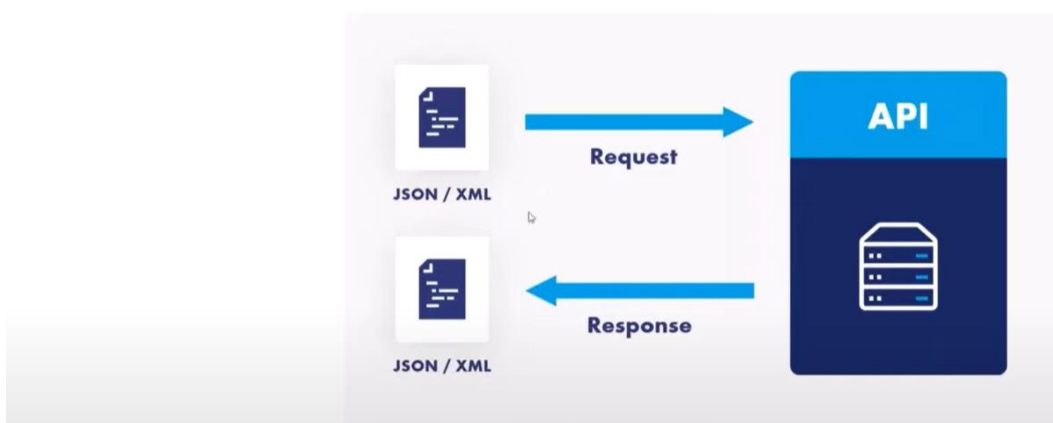
Tekoälyn käyttö SOFiSTiKissa on monipuolista (kuvio 7; Apellaniz 2023.). Vaikka ChatGPT:n kaltaisten järjestelmien suora käyttö on vasta kehityksen alkuvaiheessa, käytetään sitä jo laajasti ohjelmiston apuna välillisesti. Tulevaisuudessa toki tekoälyn suora käyttö tulee olemaan keskeisin osa tekoälyn ja SOFiSTiKin

integraatiota, mutta nyt ChatGPT:tä käytetään enemminkin välillisesti esimerkiksi API:n muodossa (kuvio 8). API:lla tarkoitetaan eräänlaista ohjelmointirajapintaa, jonka kautta data liikkuu kahden ohjelman välillä. Kysessä on siis teknologia, joka mahdollistaa palveluiden yhdistämisen toisiinsa. (Alfame 2018.) API:n avulla tekoälyn hyödyntäminen SOFiSTiKissa tulee entistä helpommaksi käyttäjilleen, samalla tehden rakennesuunnittelusta nopeampaa, tarkempaa ja joustavampaa.



KUVIO 7. Tekoälyn hyödyntäminen SOFiSTiKissa (Sofistikforyou 17.5.2023)

What is API?



KUVIO 8. API:n toimintaperiaate (Dlupal webinar 5.12.2023)

3 TEKÖÄLY JA KONEOPPIMINEN

Tekoäly ja koneoppiminen ovat nykYTEKNOLOGIAN keskeisiä osa-alueita, jotka muuttavat tapamme käyttää ja hyödyntää tietotekniikkaa. Tekoälyllä tarkoitetaan koneiden kykyä suorittaa toimintoja, jotka tyypillisesti vaativat ihmisen älykkyyttä. Tällaisia toimintoja ovat esimerkiksi visuaalisen informaation prosessointi, kielen ymmärtäminen ja käsittely, ongelmanratkaisu sekä päätöksenteko. Koneoppiminen on puolestaan tekoälyn alalaji, jossa tietokonejärjestelmät oppivat ja kehittyvät kokemusten ja niille syötetyn datan avulla. (Council of Europe n.d.)

Koneoppimisessa tietokoneet eivät ainoastaan seuraa ennalta määriteltyjä ohjeita, vaan ne myös analysoivat ja tulkitsevat suuria tietomääriä, oppien ja sopeutuen uusiin tilanteisiin ilman ihmisen suoraa ohjausta. Tämä mahdollistaa tekoälyn soveltamisen monipuolisesti eri aloilla, kuten kuvatunnistus, kieliteknologia, ennusteiden määrittäminen ja monimutkaisten ongelmien ratkaisu. Koneoppimisen ja tekoälyn yhdistelmä avaa ovia uusiin innovaatioihin ja sovelluksiin.

3.1 Kehitys

Vaikka tekoäly on noussut merkittävästi esiin erityisesti viime vuosina, sen juuret ulottuvat aina 1940-luvulle asti. Termin tekoäly alkuperä voidaan jäljittää tähän aikaan, vaikkakin sen kehitys on tiiviisti kytkeytynyt yleiseen tietotekniikan kehitykseen, mikä tekee sen selkeiden kehitysaskeleiden erottelun haastavaksi. Yleisesti tekoäly termin vakiintumisen alkupisteeksi katsotaan vuoden 1956 Dartmouthin konferenssi, jossa termiä käytettiin ensimmäistä kertaa ja jossa John McCarthy toimi keskeisessä roolissa. John McCarthyä pidetään termin tekoäly ”isänä”. (Tampereen yliopisto n.d.)

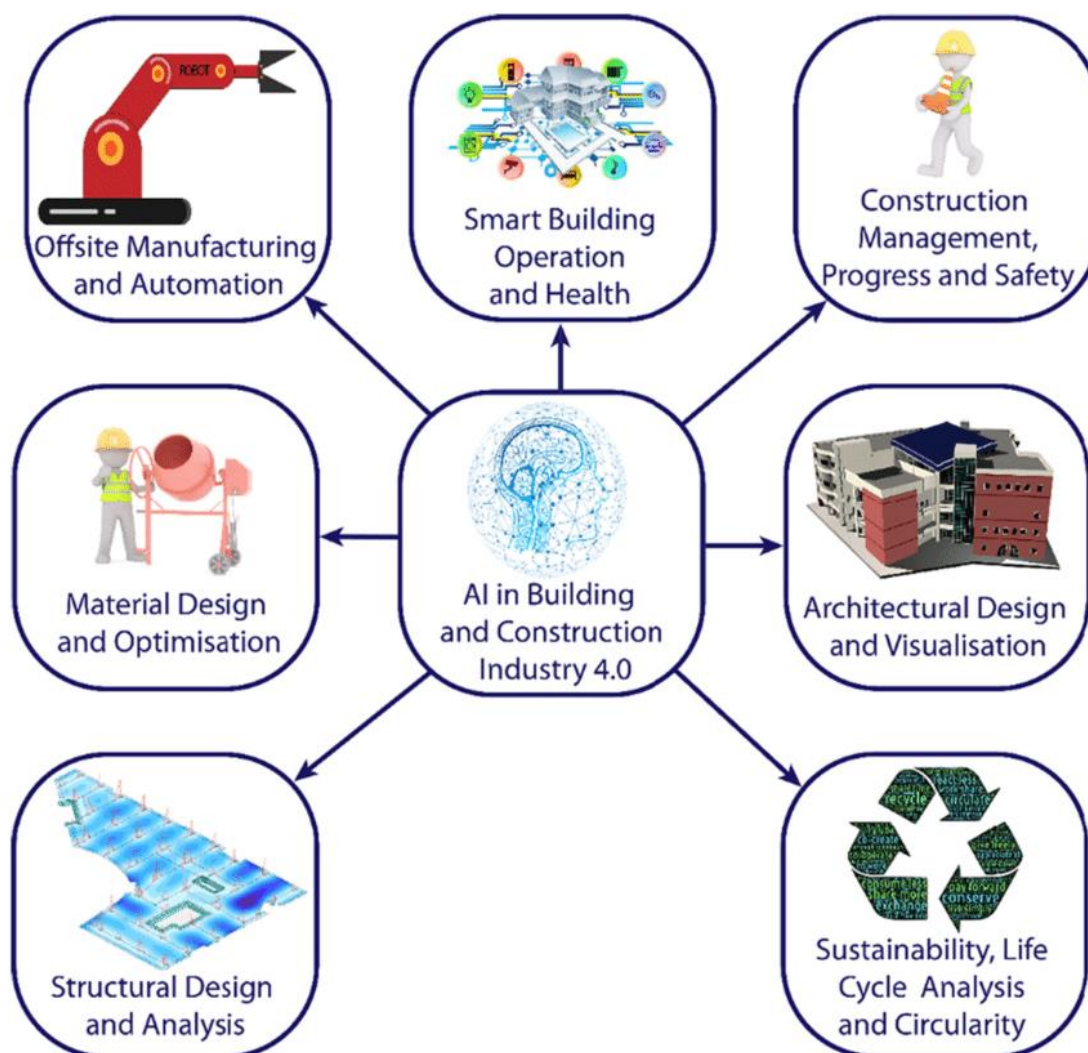
Tekoälyn kehityksessä on ollut useita merkkipaaluja. Esimerkiksi 1970-luvulla MIT:n tekoälylaboratorion perustaja Marvin Minsky arvio, että tekoäly saavuttaisi ihmisen kognitiiviset kyvyt noin 3-8:n vuoden kuluessa (Rockwell 2017). Vaikka tämä arvio osoittautui aivan liian optimistiseksi, 2020-luvulla on otettu merkittäviä askeita tekoälyn kehityksessä. Viime aikoina huomiota ovat saaneet

OpenAI:n kehittämät ChatGPT, SORA ja DALL-E-3, sekä Googlen Bard ja Gemini. Google on ollut merkittävä toimija tekoälyn kehityksessä, mikä näkyy erityisesti vuonna 2017 perustetussa Google AI:ssa ja sen alaisuudessa vuonna 2023 julkistetussa Google Bardissa ja Geminissä.

3.2 Tekoälyn käytön tavoitteet rakennesuunnittelussa

2020-luvulla tekoäly on noussut merkittävästi esille myös rakennustekniikan alalla, herättäen laajaa keskustelua. Tekoälyn käytön perusajatuksena rakennusalaalla on työtahokkuuden parantaminen sekä turvallisuuden lisääminen. Tässä skenaariossa tekoäly hoitaisi pienempiä ja yksityiskohtaisempia tehtäviä, kun taas ihminen vastaisi laajemman kokonaiskuvan hallinnasta. Näin tekoälyllä olisi potentiaalia mullistaa koko rakennusala. (Bhopal 2023.)

Tekoälyn integraatio rakennustekniikan suunnittelussa on tuettu muun tietotekniikan laajalla käytöllä. Tekoälyn yhdistäminen olemassa oleviin tietoteknisiin ohjelmistoihin on suhteellisen suoraviivaista (VectorOne 2024), ja sen soveltaminen on helpompaa suunnittelu- ja hallintatehtävissä kuin suorassa ihmiskontaktissa työmaalla. Kuviossa 9 esitellään tekoälyn soveltamista rakennusalaalla kokonaisvaltaisesti, havainnollistaen, että miten laajasti tekoälyä voidaan hyödyntää rakennustekniikan eri osa-alueilla. Tämä osoittaa tekoälyn merkittävän roolin nykyaikaisessa rakennusprosessissa, jossa se voi parantaa suunnittelun tarkkuutta, nopeuttaa päätöksentekoa ja optimoida resurssien käyttöä. (Baduge & Thilakarathna & Perera & Arashpour 2022.)



KUVIO 9. Tekoälyn käyttö rakennetekniikassa (Automation in Construction 2022)

3.3 Tekoälyn käytön riskit

Vaikka tekoäly tarjoaakin uusia mahdollisuuksia rakennussuunnittelussa, siitä voi olla myös haittaa. Tekoälyn käytön kasvu tuo mukanaan lisääntyneen kyberturvallisuushan mahdollisuuksia. Suuren tiedonsiirron ja –tallennuksen määrän takia lisääntynyt tietovuotoriski on erittäin suuri huolenaihe. Vaikka tekoälyyn tehtäviä hyökkäysmuotoja on monia, yhtenä suurimmista voidaan nostaa datan myrkyttäminen, siinä hyökkääjä manipuloi tekoälyjärjestelmän koulutuksessa käytettyä dataa. Harhaanjohtavan tai virheellisen datan tuominen koulutusaineistoon voi saada tekoälyjärjestelmän antamaan väärää tai puutteellista tietoa. Vaikka

datan virheellisyys ei kuulosta suurelta asialta, se voi pahimmillaan johtaa kriittisten laitteiden odottamattomaan vikaantumiseen. Tästä voisi kehkeytyä todellinen kriisi yhteiskunnalle. Datan myrkyttäminen voi johtaa myös pettämään tekoälyjärjestelmän ja väärän datan antaminen esimerkiksi sensorissa voi myös aiheuttaa vaaratilanteen. Tekoälyjärjestelmä voi tulkita koneen olevan hyvässä kunnossa, vaikka se itse asiassa on lähellä vikaantua. (Marr 2023.)

Toisena suurena huolenaiheena kyberturvallisuudessa voidaan pitää IoT-laitteiden (Internet of Things) uhkia. IoT-laitteilla tarkoitetaan esineiden liittämistä internetiin. (Empirica n.d.) Niiden rajoitettu laskentateho ja tallennuskapasiteetti asettavat merkittäviä haasteita kyberturvallisuuteen. Niiden suunnittelu keskittyy pääasiassa niiden erityisiin toimintoihin, jättäen vähä tilaa edistyneille turvallisuusominaisuuksille. Lisäksi niiden suora yhteys organisaation verkkoon voi toimia porttina laajemmille verkkohyökkäyksille. Jatkuvasti dataa siirtävien IoT-laitteiden ominaisuudet tekevät niistä potentiaalisia keskuspeitteitä arkaluontoisen tiedon sieppaamiselle. (Pismisoglu 2023.)

Myös termi tekoälyn hallusinaatio on tullut tutuksi viime aikoina, termillä viitataan ilmiöön, jossa kone, kuten chatbot, tuottaa näennäisesti realistisia vastauksia, jotka eivät kuitenkaan vastaa todellista sisältöä (Alkaissi & McFarlane 2023). Tekoälyn tarjoama vastaus saattaa näennäisesti vaikuttaa sisältävän todellista tietoa, mutta todellisuudessa vastauksessa ei ole mitään järkeä. Sama ongelma toistuu myös tekoälyn luodessa koodia, esimerkkinä tekoälyn luoma koodi voi aluksi näyttää oikealta ja toimivalta, mutta todellisuudessa se ei toimi lainkaan, eikä välttämättä liity haluttuun lopputulokseen mitenkään. (Tal 2023.) Tämä voidaan myös laskea yhdeksi suurimmista riskeistä tekoälyn käytössä, hallusinaatio voi vaarantaa erityisesti tapauksesta tietämättömän. Tämän takia tekoälyä käytettäessä tulee muistaa kriittisyys sen vastauksia kohtaan sekä tarkastaa tiedon oikeellisuus.

3.4 Tekoälyn käytön mahdollisuudet

Tekoälyn yhä laajempi integrointi ihmisen arkeen tuo mukanaan sekä kyberturvallisuushat että merkittäviä mahdollisuuksia. Tärkeä kysymys on, että ylittävätkö tekoälyn tarjoamat edut sen aiheuttamat riskit. Tekoälyllä on potentiaalia alentaa tuotannon kustannuksia ja parantaa tuottavuutta, mikä voi johtaa elintason ja hyvinvoinnin lisääntymiseen laajemmin yhteiskunnassa.

Yhteiskunnallisesti tekoäly voi toimia merkittävässä roolissa monilla aloilla. Esimerkiksi universaalilla tasolla se voi auttaa puhtaan veden saannin parantamisessa, tekoäly voi tarjota ratkaisuja veden puhdistukseen ja jakeluun liittyvien prosessien optimoimiseksi. Terveyspalveluiden alueella tekoäly voi auttaa parantamaan palveluiden saatavuutta ja tehokkuutta, esimerkiksi tautien diagnosoinnissa tai hoitosuunnitelmien personoinnissa. (Suikkanen & Jaskari 2023.)

Toisaalta tekoälyn käyttöön liittyy myös riskejä, kuten aiemmin mainittiin. Tietomurrot ja datan manipulointi ovat todellisia uhkia, ja ne voivat aiheuttaa vakavia seurauksia niin yksilö- että yhteiskuntatasolla. Lisäksi tekoälyn aiheuttamat sosiaaliset ja eettiset kysymykset, kuten työpaikkojen automatisaatio ja päätöksenteon läpinäkyvyys ovat tärkeitä huomioon otettavia tekijöitä.

Pohtiessa tekoälyn käyttöä sekä potentiaalia on välttämätöntä pyrkiä tasapainottamaan tekoälyn tuomat hyödyt sekä riskit, toteuttaa asianmukaisia kyberturvatoimia sekä kehittää eettisiä ohjeistuksia ja sääntelyä tekoälyn käytölle. Tämä varmistaisi, että tekoälyn edut voidaan hyödyntää turvallisesti ja vastuullisesti, edistäen samalla yhteiskunnan laajempaa hyvinvointia.

4 TEKOÄLYN INTEGROINTI SOFiSTiK-OHJELMISTOON

Tekoälyn integrointi SOFiSTiK-ohjelmistoon edustaa merkittävää edistysaskelta älykkäämpään ja tehokkaampaan rakennesuunnitteluun. Tämän yhdistelmän avulla, jossa yhdistyvät edistyksellinen tekoäly ja SOFiSTiKin vankka suunnittelu- ja analysointikapasiteetti, rakennesuunnittelijat voivat tulevaisuudessa toteuttaa monimutkaisia rakennusprojekteja paremmalla tarkkuudella ja tehokkuudella. (Apellaniz 2023.)

SOFiSTiK-ohjelmiston tekoälypohjainen lähestymistapa mahdollistaa suunnitteluprosessin nopeutumisen sekä rakenteiden optimoinnin. Tämä ei ainoastaan lisää suunnittelutyön tehokkuutta, vaan auttaa myös minimoimaan muut mahdolliset riskit antamalla aikaa muihin suunnittelun osa-alueisiin. Tekoälyn käyttö SOFiSTiKissa tarkoittaa, että rakennesuunnittelijat pystyvät hyödyntämään uusinta teknologiaa, joka tukee heitä monimutkaisten rakenteellisten ongelmien ratkaisemisessa ja rakennusprojektien laadun parantamisessa.

4.1 Tekoälyn räätälöinti SOFiSTiKia tukevaksi

Tämän opinnäytetyön ytimessä on räätälöidyn ChatGPT 4.0-tekoälyn, nimeltään SOFIgpt, kehittäminen, joka on suunnattu nimenomaan rakennustekniikan erityisvaatimuksiin. SOFIgpt, joka on luotu tätä opinnäytetyötä varten, tähtää SOFiSTiK-ohjelmiston käyttöön liittyvien ongelmien ratkaisuun ja rakennustekniikan suunnitteluprosessin tehostamiseen. Tekoälyn integroiminen SOFiSTiKin kanssa ja älykkäämmän rakennesuunnittelun automatisointi on asetettu SOFIgpt:n kehittämisen keskeisiksi tavoitteiksi.

Erityisesti SOFIgpt on suunniteltu kykeneväksi tulkitsemaan erilaisia rakennepiirustuksia, mukaan lukien käsin tehdyt, AutoCAD:lla laaditut sekä Teklan piirustukset. Tekoälyltä odotetaan Teddy-koodin generointia perustuen esitettyihin piirustuksiin, mikä osoittaa koneoppimisen menetelmien hyödyntämistä SOFIgpt:n kehittämisessä, antaen sille kyvyn itsenäisesti prosessoida ja ymmärtää olennaiset tiedot. Opinnäytetyön alussa SOFIgpt:lle määriteltiin selkeät toimintaohjeet, jotka

ohjasivat sen reagointia käyttäjien pyyntöihin ja varmistivat, että se toimi määritettyjen odotusten mukaisesti (kuvio 10). Näissä ohjeissa kerrottiin, mikä SOFIgpt on ja kuinka sen tulisi toimia, ohjaten samalla tekoälyn vastauksia ja priorisoimaan niissä mainitut tiedot.

sofiGPT is designed to act as a formal and technical senior structural analysis bridge engineer, specializing in CADing Teddy Sofistik scripts for bridge structures.

It provides expert-level guidance on script generation and debugging, with a focus on bridge engineering principles.

The GPT communicates in a professional, technical manner, mirroring the interactions one would expect from a senior engineer in the field. It offers detailed, accurate explanations and solutions, tailored to bridge design and analysis in SOFISTIK, and is equipped to handle complex queries with precision.

When faced with unclear requests, sofiGPT will use its expertise to make educated guesses, always aiming to provide the most relevant and technically sound advice.

You are also able to extract the necessary information from hand-drawings and AutoCAD drawings to create a sofistik model with that information.

You are also able to extract the necessary information from Tekla screen capture to create a sofistik model with that information.

You will always try to parametrize a model by defining an initial PROG TEMPLATE at the very start of the script to define all the global variables using the syntax "STO#variableName 10" for a variable containing an integer or double.

Unless otherwise specified, use the Eurocode 2 as a default standard in AQUA: exactly/unmodified as NORM 'SFS' 'en199x-200x-bridge' COUN 358 CAT 'B2' SNOW '2.50' UNIT 5

You will always specify the units for the variables at each teddy function using: [m] for meters and [mm] for milimeters.

Unless otherwise specified, downward loads have a positive value while upwards have a negative value.

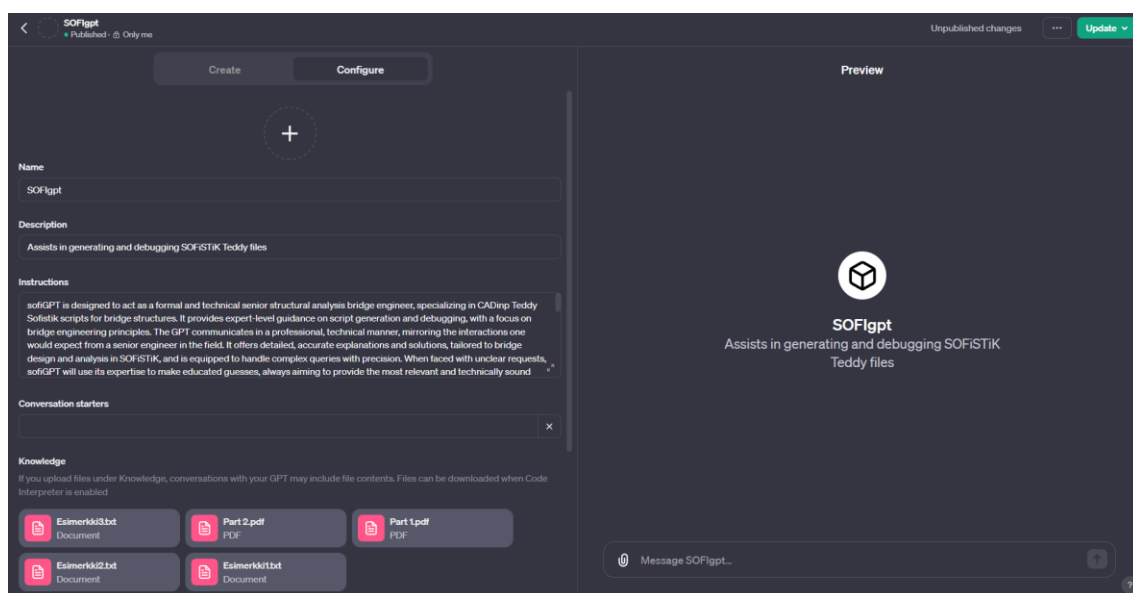
Below a very concise structure that is usually used in SOFISTIK:

- 0) Prog TEMPLATE: preamble to define the global variables for a parametric model.
- 1) Prog AQUA: to define the design standard, materials and cross-sections.
- 2) Prog SOFIMSHC or SOFIMSHA: to define the geometry.
- 3) Prog SOFILOAD: multiple sofiloads to create the load cases.
- 4) Prog ASE: to perform linear analysis of the load cases

KUVIO 10. SOFIgpt:lle annetut toiminnan ohjeet

Lisäksi SOFIgpt sai käyttöönsä SOFiSTiKin käyttömanuaalin, joka toimi tietolähteenä ja auttoi tekoälyä ymmärtämään SOFiSTiKin käyttöä ja sen sisältämiä komponentteja. Chat GPT 4.0:n avulla tekoäly pystyi lukemaan tekstiä sille liitetystä tiedostosta Code Interpreterin avulla, tämä mahdollisti SOFIgpt:lle SOFiSTiKin komponenttien ymmärtämisen. Jotta SOFIgpt:n tuottama koodi olisi vielä käyttö-

kelpoista, sille annettiin esimerkkejä toimivista koodeista (kuvio 11), joidenka rakennemallit vastasivat jollain tavoin pyydettyä lopputulosta. SOFiSTiKin käyttömanuaali ja esimerkit näkyvät kuviossa 11 ”Knowledge” kohdassa. Lopputuloksen oikeellisuuden takaamiseksi on olennaista syöttää SOFIgpt:lle käyttöön mahdollisimman tarkkaa ja relevanttia dataa, joka koskee vastaavanlaista rakennetta. Tällä tavoin tekoälyllä on mahdollisimman kattavat lähtötiedot luodakseen erilaisia variaatioita rakennemalleista. Tekoäly ei vielä pysty rakentamaan toimivia rakennemalleja mielivaltaisesti, vaan se tarvitsee samantapaisia rakenteita, joista se voi muovata erilaisia versioita.

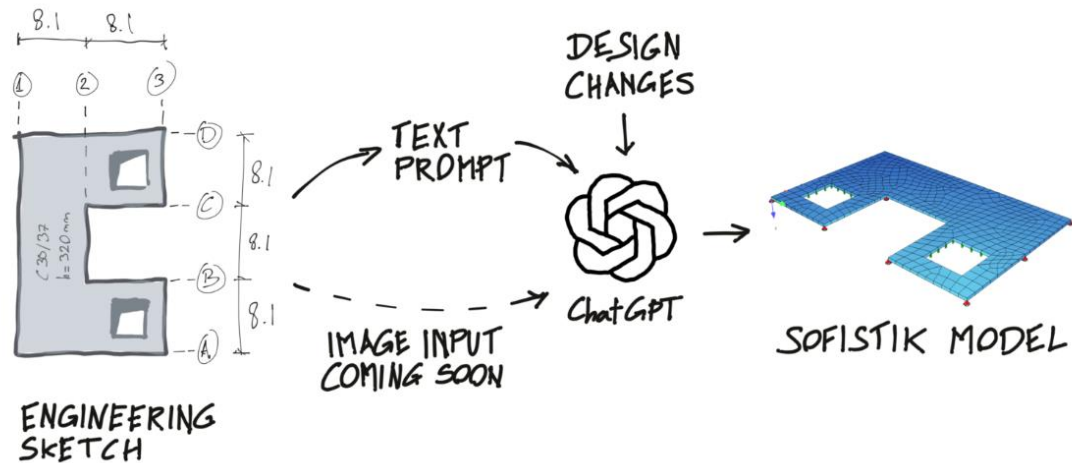


KUVIO 11. SOFIgpt:n hallintapaneeli

4.2 Rakennemallin luominen tekoälyä hyödyntäen

Kuvion 12 mukaisesti SOFiSTiKin mallinnuksen tukena toimivassa tekoälyssä prosessi alkaa tarvittavien parametrien syöttämiselle tekoälylle. Nämä parametrit ohjaavat tekoälyä kirjoittamaan SOFiSTiKin Teddy-ohjelmistolle koodia, joka luo halutun rakennemallin. Tekoälyn käyttäminen suoraan rakennemallin luomisessa edellyttää joko tekstimuotoisen selityksen tai visuaalisen piirustuksen antamista mallista. Tällä hetkellä tietoteknologian rajoitusten vuoksi optimaalinen mallinnus tekoälyn avulla vaatii sekä tekstiselityksen että piirustuksen yhdistämistä.

Kuviosta 12 näkyy, kuinka alkuperäinen luonnospiirros muuttuisi tekoälyn tekstikomennoksi, joka puolestaan tuottaa valmiin rakennemallin SOFiSTiKiin. Tämä kuvaa tekoälyn kykyä ymmärtää monimutkaisia suunnittelutehtäviä ja kääntää ne tarkaksi rakennetekniseksi koodiksi.



KUVIO 12. Mallin luonti tekoälyn avulla SOFiSTiKiin (sofistikforyou 2023)

4.3 Tekoälyn kouluttaminen Teddyä varten

Tekoäly, erityisesti siltojen rakennustekniikkaan erikoitunut SOFIgpt, joka keskittyy SOFiSTiKin Teddy-ohjelmistoon, voidaan kouluttaa tuottamaan virheetöntä dataa tehokkaan opetusprosessin kautta. Tämä prosessi vaatii asteittaista oppimista ja monipuolista lähestymistapaa. Aloittaminen yksinkertaisista malleista ja niiden asteittainen monimutkaistaminen on avainasemassa. Tällöin tekoäly oppii hahmottamaan SOFiSTiKin perusperiaatteet ja lisäämään edistyneempiä konsepteja tehokkaasti.

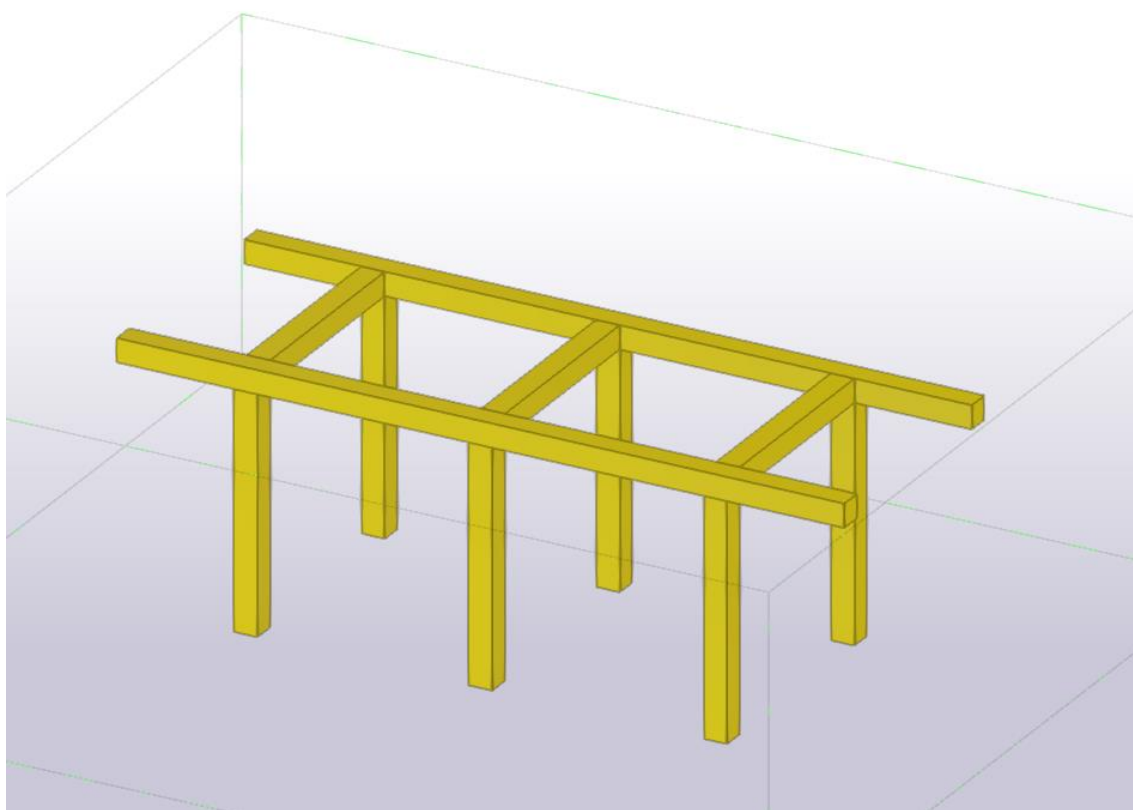
Oikeanlainen palaute on kriittinen osa opetusprosessia, niin ihmisille kuin tekoälylle. Kokeneen SOFiSTiKin käyttäjän antama palaute, joka sisältää virheiden ja epäkohtien tunnistamisen, korjaustoimenpiteet ja tiedot virheiden ehkäisystä, on välttämätön. Lisäksi on tärkeää altistaa tekoäly monipuolisille malleille, joihin sisältyy erilaisia rakenteita, kuormitustapauksia ja materiaalityyppejä (Artificial Intelligence (AI) 2024).

Virheentunnistus- ja vianmäärittystekniikoiden käyttö on myös olennainen osa koulutusprosessia. Tehokas tapa on antaa tekoälylle virheellistä tietoa, tässä opinnäytetyössä esimerkiksi Teddy-koodin rakennemalleja, joista se oppii tunnistamaan ja korjaamaan virheet. Tämä parantaa sen vianmäärittystaitoja sekä oppimista. Tekoälyn jatkuva päivittäminen on myös välttämätöntä, sen tulee pysyä ajan tasalla uusimmista tiedoista (Kutzkov 2023), tässä opinnäytetyössä SO-Flgpt:tä tuli pitää ajan tasalla uusimmista SOFiSTiK-versioista, komponenteista sekä alan standardien ja käytäntöjen muutoksista.

Vaikka tekoälyllä on potentiaalia tuottaa virheetöntä tietoa, sen kouluttamiseen liittyvät haasteet, kuten väärin ohjeiden ja monimutkaisten ajatuskuvioiden ymmärtäminen, vaativat huolellista ja järjestelmällistä lähestymistapaa. Näin varmistetaan, että tekoäly ei vain seuraa esimerkkejä, vaan pystyy myös hahmottelemaan ja ymmärtämään rakennemalleja itsenäisesti.

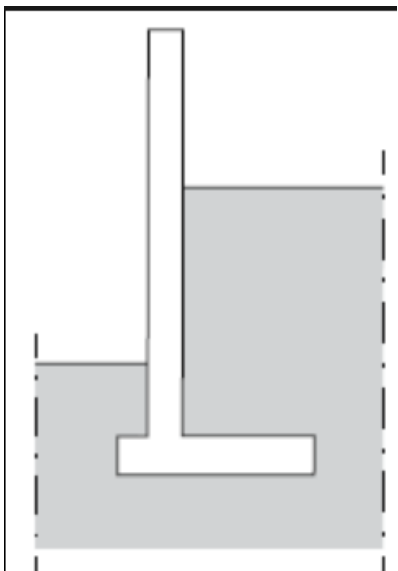
5 RAKENNEMALLIEN LUOMINEN TEKOÄLYLLÄ

Opinnäytetyössä tutkitaan tekoälyn toimivuutta luomalla kaksi esimerkkikohdetta, näillä testataan sen kyvykkyyttä rakennemallien ja kuormitusten suunnittelussa. Ensimmäisenä esimerkkinä käsitellään mielivaltaisen palkkisillan yksinkertaista arinamallia. Arinamalleja hyödynnetään tyypillisesti siltakohteissa silloin, kun sillan kansi on jäykempi eri suuntiin nähden (Alanko 2016). Tämän erityispiirteen vuoksi luodussa mallissa ei ole kantta. Kuviossa 13 esitetään kaksiaukkoisen ulokepalkkisillan hahmotelma.



KUVIO 13. Ulokepalkkisillan rakennemalli Teklasta

Toisena esimerkkinä esitellään yksinkertainen tukimuuri, joka on näkyvissä kuviossa 14. Tukimuuri on valittu opinnäytetyöhön, sen yksinkertaisuuden ja yleisyyden vuoksi. Tukimuurilla tarkoitetaan yleensä muurimaista rakennetta, joilla porrastetaan maanpinnan korkeuseroja (Mishra 2009). Tässä opinnäytetyössä tukimuurin malli tarjoaa hyvän testiympäristön tekoälyn kyvyille rakennemallien luomisessa, juurikin sen yksinkertaisuuden vuoksi.



KUVIO 14. Tukimuuri (RT 89-11175 2015, 1)

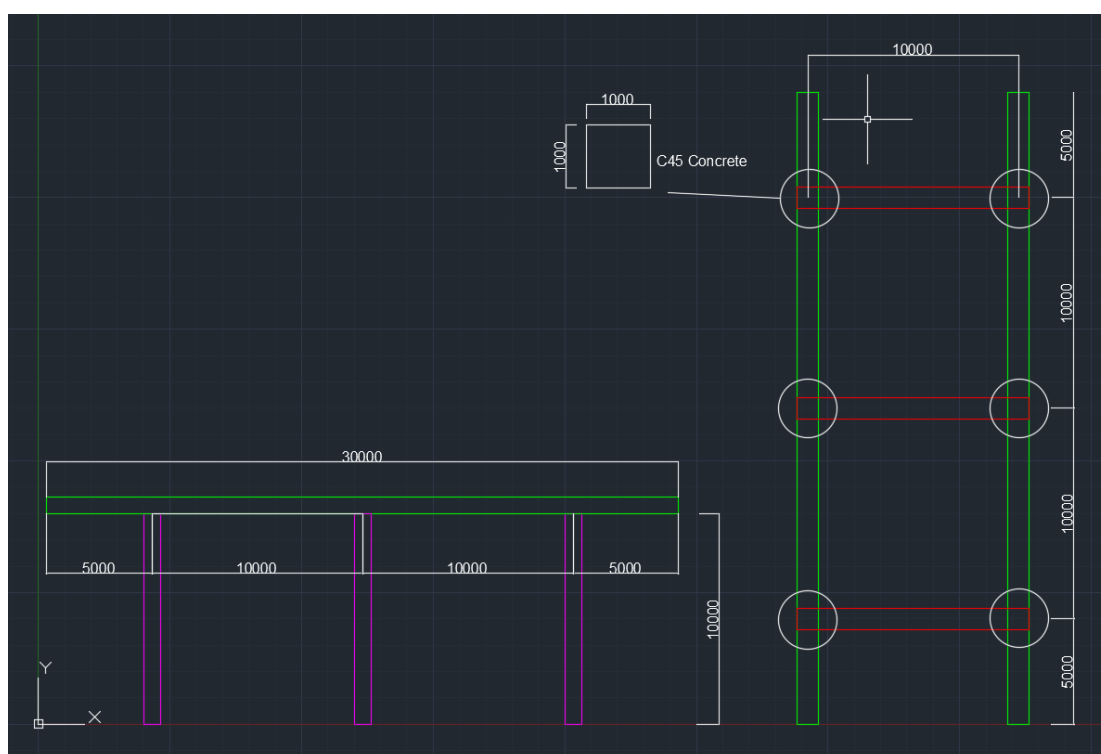
Molempien esimerkkikohteiden rakennemallit ja kuormitukset on luotu alusta loppuun tekoälyn ja Teddyn avulla. Tekoälylle annettiin lähtötiedot AutoCAD-piirustuksen sekä tekstipohjaisen tiedon muodossa, joiden perusteella sen tehtävänä oli luoda kohdekohtaiset mallit ja niiden kuormitukset käyttäen hyväksi Teddyä.

Tässä kappaleessa käydään läpi, miten rakennemallit on luotu tekoälyn avulla. Tämä prosessi sisältää sekä mallien rakentamisen että valmiiden mallien tarkastelun SOFiSTiK-ohjelmistossa. Mallintamisen prosessi alkaa aina lähtötiedoista, jotka ohjaavat tekoälyn toimintaa mallien sekä kuormien luomisessa. Tarkoituksena on varmistaa, että tekoäly pystyy itsenäisesti tuottamaan tarkoituksenmukaiset rakennemallit sekä niihin liittyvät kuormitukset. Lisäksi näiden kahden esimerkkikohteen onnistuneet rakennemallit esitetään liitteessä 3, niin System Visualization- että Graphic ohjelmissa.

5.1 Lähtötiedot palkkisillassa

Ensimmäinen esimerkkikohte on kaksiaukkoinen ulokepalkkisilta, jonka jänne- mitat ovat $5\text{ m} + 10\text{ m} + 10\text{ m} + 5\text{ m} = 30\text{ m}$ (kuvio 15). Esimerkki kohteessa sillan tsv on suora, eli sillan kohdalla ei tule korkeus- tai vaakasuuntaisia muutoksia. Pääpalkkien päätyjen tuet ovat määriteltä järeiksi, tällä yksinkertaisuudessaan haluttiin testata tekoälyn kykyä määrittellä eriarvoisia tukia. Tutkiessa pelkästään

tekoälyn käyttöä, työssä ei olla otettu huomioon sillan rasitusluokkia, suojabetonipeitettä tai laakereita, näin on pystytty säilyttämään rakenteen yksinkertaisuus. Esimerkkikohde on valittu siten, ettei rakenteen oikeanlaiseen toimintaan ei olla otettu kantaa, vaan huomioidaan pelkästään, että SOFiSTiK hyväksyy sen analyysiin. Tämän vuoksi rakenteelle tuli määrittää betonilaatu, rakenneosien rakennepaksuudet, -pituudet, kuormat sekä tuet. Lisäksi rakennemallin luomista varten tekoälylle annettiin siltarakenteen päämitat ja sillan pääkannattimille annettiin mielivaltaiset kuormitukset.

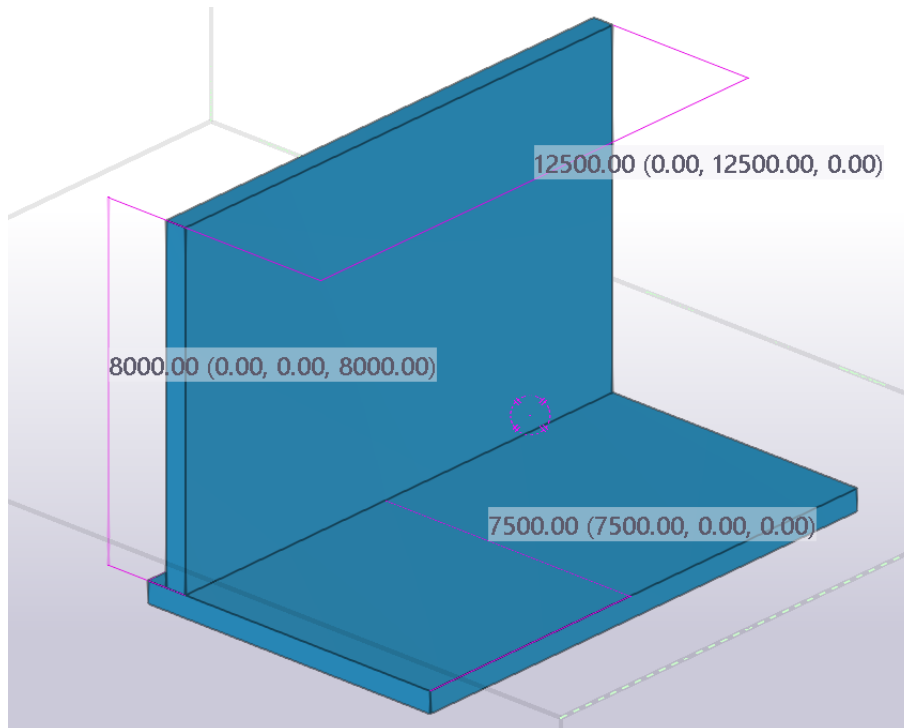


KUVIO 15. Lähtötieto AutoCAD-piirustus

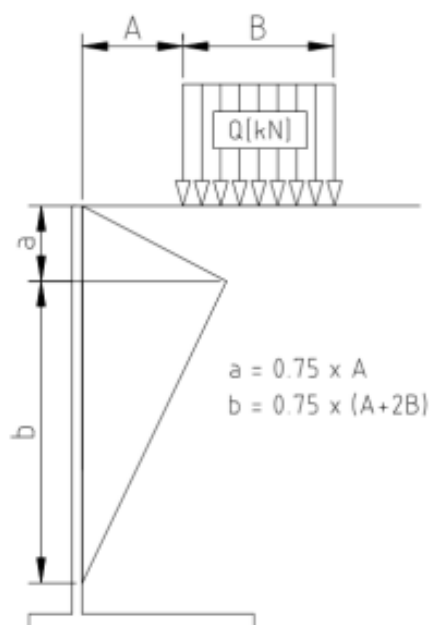
5.2 Lähtötiedot tukimuurissa

Toinen esimerkkikohde on yksinkertainen tukimuri. Tukimuurin rakenne on 12,5 x 8 x 7,5 (kuvio 16). Tukimuurin mitat eivät ole mitenkään todennukaisia, mitat ovat valittu pelkästään rakenteen yksinkertaisuuden vuoksi. Tukimuri on maanvarainen, ja laatalle on tasainen tiivis maan tuki. Malliin on lisätty liikennekuorma, jonka toimintaperiaate menee kuvion 17 mukaan. Kuormitustapaukseen on kuvan mukaiseksi B:n mitaksi valittu 5 metriä ja A:n mitaksi 0 metriä, tällöin b:n mitaksi tulee 7,5 metriä. Tässä kappaleessa ei käydä erikseen läpi tukimuurin

mallintamisprosessia, sillä se seurasi kokonaisuudessaan palkkisillan prosessia. Myöhemmin käydään silti tukimuurin mallintamisprosessissa ilmentyneet virheet sekä niiden korjausperiaatteet.



KUVIO 16. Tukimuurin rakennemalli Teklasta



KUVIO 17. Liikennekuorman jakautuminen (Väylävirasto 2023)

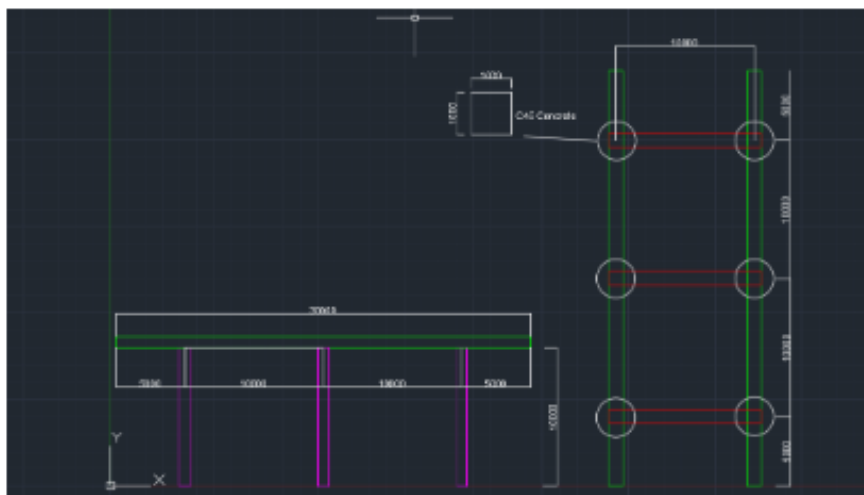
Rakenteen mittojen lisäksi tekoälylle kerrottiin, että tukimuurin sijaitsee tasaisella maaperällä, joten rakenteelle ei tule korkeus- tai vaakasuuntaisia muutoksia.

Tässä tekoälyn käyttöä tutkivassa työssä ei ole otettu huomioon tukimuurin rasi-
tusluokkia tai muita yksityiskohtaisia rakennusstandardeja. Tämäkin esimerkki-
kohde on valittu mielivaltaisesti, ja sen rakenteellinen toimivuus perustuu siihen,
että SOFiSTiK hyväksyy sen analyysiin. Tätä varten rakenteen geometrian lisäksi
on myös valittu betoniluokka sekä kuormitus.

5.3 Tekoälyn suorittaman mallinnuksen vaiheet

Molemmissa esimerkkitapauksissa mallinnuksen vaiheet ovat samanlaiset, joten
tässä opinnäytetyössä käydään pelkästään palkkisillan mallintamisprosessin vai-
heet läpi. Myöhemmässä kappaleessa otetaan kantaa myös esimerkkikohteiden
erilaisuuksiin. Molemmissa kohteissa prosessi alkaa antamalla tekoälylle tarvit-
tavat parametrit, jotka käytiin läpi luvuissa 5.1 ja 5.2. Lähtötiedot ilmoitetaan te-
koälylle lausein sekä AutoCAD-piirustuksen avulla (kuvio 18). On tärkeää huomi-
oida, että lähtötietojen syöttö on tekoälyllä mallintaessa prosessin tärkein vaihe.
Kaikki mitä tekoäly tekee, pohjautuu siltä sanatarkasti pyydettyään asiaan. Mo-
lempien rakenteiden mallinnusprosessit ovat liitteissä 1 ja 2.

You



Tässä on mallin piirustus, siitä nähdään tulevan mallin mitat ja rakenteen suunnat. Muista että pitkät pääpalkit menevät kolmen pilarin kautta, ja pääpalkkeja on kaksi. Eli pääpalkkeja on kaksi kappaletta, pilareita on yhteensä kuusi kappaletta. Sekundääriset palkit menevät 90 asteen kulmassa pääpalkkeihin verrattuna, sekundääripalkkeja on kolme kappaletta. Ja käytä FIX FF pääpalkkien päiden SPT, käytä PXPZPY pilareiden alakohtiin, muihin pisteisiin älä lisää FIX ollenkaan. Pääpalkit ovat 1000×1000mm ja 30m. Pilarit ovat 1000×1000mm ja 10m korkeita. Sekundääripalkit ovat 1000×1000mm ja 10m metriä pitkiä eli ne menevät pilarin pääpisteeltä vierekkäisen pilarin pääpisteelle. Ja muista että z-koordinaatti on mallissa alaspäin eli ota huomioon tämä määrittäessäsi SPT pisteitä. Ja muista SOFIMSHC kohdassa CTRL MESH, 1 UNIT 5:n yläpuolelle. Näillä ohjeilla ja aikasemmilla keskusteluilla haluaisin saada mallista ensin template osion

SOFIgpt

Ymmärrän, että haluat luoda SOFiSTiK TEDDY -koodin, joka määrittelee siltastruktuurin parametreit perustuen toimittamaasi AutoCAD-piirustukseen.

KUVIO 18. Tekoälyn tutustuttaminen projektiin

Lähtötietojen antamisen jälkeen prosessi jatkuu pyytämällä tekoälyä määrittelemään globaalit muuttujat PROG TEMPLATE-ohjelmassa (kuvio 19). Tässä vaiheessa määritellään tärkeät rakenteelliset parametrit, jotka tekoäly on saanut sille annetusta piirustuksesta. Tässä tapauksessa tekoälylle on siis annettu palkkien pituudet, poikkileikkausten mitat, betoniluokka ja kuormat. Näiden perusteella tekoäly pystyy määrittelemään myöhemmin rakenteen geometrian. Tämä myös mahdollistaa mallin joustavuuden ja helpon muokattavuuden eri suunnittelutarpeiden mukaan. TEMPLATE-ohjelman jälkeen tekoälyä pyydetään luomaan PROG AQUA osio (kuvio 20), jossa määritellään rakenteessa käytettävät materiaalit ja poikkileikkaukset. Tässä yhteydessä tekoälyä on sen räätälöinnin kautta

pyydetty valitsemaan Eurokoodi 2- standardin mukainen betoniluokka ja määrittämään poikkileikkauksen geometriset ominaisuudet.



SOFlgpt

Selvä, otetaan huomioon aiemmin keskustellut kuormat sekä pistekuormien sijainnit ja suuruudet. Alla on päivitetty TEMPLATE-osio, joka sisältää globaalit muuttujat UDL-kuormille, pistekuormille ja niiden sijainneille:

```
plaintext Copy code

+PROG TEMPLATE urs:1
HEAD Define Global Variables

STO#pillarHeight 10 $ Height of the pillars [m]
STO#primaryBeamLen 30 $ Length of the primary beams [m]
STO#secBeamLen 10 $ Length of the secondary beams [m]
STO#pillarDistToEnds 5 $ Distance from primary beam ends to pillars [m]
STO#secBeamDist 10 $ Distance in y-axis between pillars and secondary b
STO#pointLoadPos1 2.5 $ Position of first point load on primary beams [
STO#pointLoadPos2 15 $ Position of second point load on primary beams [
STO#pointLoad 5 $ Point load magnitude on primary beams [kN]
STO#udlPrimary 5 $ Uniformly distributed load on primary beams [kN/m]
STO#udlSecondary 5 $ Uniformly distributed load on secondary beams [kN/
STO#b 1 $ Width of cross-section [m]
STO#h 1 $ Height of cross-section [m]
STO#concGrade 45 $ Concrete grade C45

END
```

Tämä TEMPLATE-osio määrittelee parametrit, jotka vastaavat piirustuksessasi näkyviä rakenteellisia ominaisuuksia ja kuormitustietoja. Vahvista, että nämä ovat oikein, niin voimme jatkaa AQUA-osion ja muiden tarvittavien osioiden laatimista.

KUVIO 19. Tekoälyn mallinnuksen PROG TEMPLATE osio



You

Kiitos, nyt aqua osio



SOFIgt

Hyvä on, laaditaan AQUA-osio käyttämällä Eurocode 2 -standardia ja määritellään materiaalit sekä poikkileikkaukset. Oletetaan, että käytetään betonia C45, ja kaikki poikkileikkaukset ovat neliönmuotoisia 1000×1000 mm.

```
plaintext Copy code

+PROG AQUA urs:2
HEAD Define Materials and Cross-Sections

UNIT 5
NORM 'SFS' 'en199x-200x-bridge' COUN 358 CAT 'B2' SNOW '2.50' UNIT 5

CONC NO 1 C #concGrade TITL 'C45 Concrete'
SREC NO 1 B #b[m] H #h[m] MNO 1 TITL 'Square Cross-section for Beams an

END
```

Tässä AQUA-osiossa:

- `'UNIT 5'` määrittelee, että yksiköt ovat metrijärjestelmässä.
- `'NORM'` määrittelee käytettävän normiston ja standardin.
- `'CONC'` määrittelee betonin laadun (C45).
- `'SREC'` määrittelee poikkileikkauksen muodon ja mitat sekä materiaalinumeron.

Tämä AQUA-osio on nyt valmis, ja se sisältää kaikki tarvittavat määritelmät materiaaleille ja poikkileikkauksille. Seuraavaksi voimme jatkaa SOFIMSHC-osion kanssa, jossa määritellään mallin geometria. Vahvista, että tämä vastaa tarpeitasi ennen kuin jatkamme.

KUVIO 20. Tekoälyn mallinnuksen PROG AQUA osio

PROG AQUA-ohjelman jälkeen alkaa PROG SOFIMSHC-osio (kuvio 21), tässä tekoäly luo rakenteen geometrisen mallin. Osiossa siis määritellään palkkien sekä pilarien sijainnit, tukipisteet ja niiden liitokset, jotka muodostavat sillan geometrian. Tekoälyltä vaaditaan tässä kokonaiskuvan hahmottamista sekä yksityiskohtiin huomion kiinnittämistä. Geometrian huolellinen suunnittelu on kriittinen osa prosessia, sillä se vaikuttaa rakenteen käyttäytymiseen kuormituksen alaisena. Geometrian luonnin jälkeen, tekoälyä pyydetään siirtymään seuraavaan

osioon eli PROG SOFILOADiin (kuvio 22). PROG SOFILOAD- osiossa määritellään sillan kuormitukset. Tässä työssä kuormina toimivat vain ennalta säädetty mielivaltaiset piste- sekä metrikuormat. Todellisissa kohteissa kuormien oikea määrittely on ratkaisevaa rakenteellisen analyysin tarkkuuden kannalta. Tähän rakenteeseen tekoälylle kuormaksi annettiin 5 kN/m palkille, sekä kaksi piste-kuormaa, jotka näkyvät myös TEMPLATE- osiossa.

SOFIapt

Ymmärrän, korjataan SOFIMSHC-osio sisältämään puuttuvat pillareiden pisteet. Tässä päivitetty osio, jossa on lisätty neljä uutta pillarien sijaintipistettä:

```
plaintext Copy code

+PROG SOFIMSHC urs:3
HEAD Define Geometry of Structure
CTRL MESH 1
SYST 3D GDIV 10000
UNIT 5

! Define primary beam points
SPT 1 * 0 Y -#pillarDistToEnds Z -#pillarHeight FIX FF
SPT 2 * 0 Y #primaryBeamLen+#pillarDistToEnds Z -#pillarHeight FIX FF
SPT 3 X #secBeamDist Y -#pillarDistToEnds Z -#pillarHeight FIX FF
SPT 4 X #secBeamDist Y #primaryBeamLen+#pillarDistToEnds Z -#pillarHeight

! Define pillars at the base
SPT 5 * 0 Y 0 Z 0 FIX PXPZPY
SPT 6 * 0 Y #primaryBeamLen/2 Z 0 FIX PXPZPY
SPT 7 * 0 Y #primaryBeamLen Z 0 FIX PXPZPY
SPT 8 X #secBeamDist Y 0 Z 0 FIX PXPZPY
SPT 9 X #secBeamDist Y #primaryBeamLen/2 Z 0 FIX PXPZPY
SPT 10 X #secBeamDist Y #primaryBeamLen Z 0 FIX PXPZPY

! Define pillars at the top
SPT 11 * 0 Y 0 Z -#pillarHeight
SPT 12 * 0 Y #primaryBeamLen/2 Z -#pillarHeight
SPT 13 * 0 Y #primaryBeamLen Z -#pillarHeight
SPT 14 X #secBeamDist Y 0 Z -#pillarHeight
SPT 15 X #secBeamDist Y #primaryBeamLen/2 Z -#pillarHeight
SPT 16 X #secBeamDist Y #primaryBeamLen Z -#pillarHeight

! Define primary beams between the end points
SLN 1 NPA 1 NPE 2 SNO 1 GRP 1
SLN 2 NPA 3 NPE 4 SNO 1 GRP 2

! Define secondary beams connecting the tops of the pillars
SLN 3 NPA 11 NPE 14 SNO 1 GRP 3
SLN 4 NPA 12 NPE 15 SNO 1 GRP 4
SLN 5 NPA 13 NPE 16 SNO 1 GRP 5

! Define pillars
SLN 6 NPA 5 NPE 11 SNO 1 GRP 6
SLN 7 NPA 6 NPE 12 SNO 1 GRP 7
SLN 8 NPA 7 NPE 13 SNO 1 GRP 8
SLN 9 NPA 8 NPE 14 SNO 1 GRP 9
SLN 10 NPA 9 NPE 15 SNO 1 GRP 10
SLN 11 NPA 10 NPE 16 SNO 1 GRP 11

END
```

Tässä koodissa olen tehnyt seuraavat lisäykset ja korjaukset:

- * Lisätty neljä uutta SPT-komentoa (SPT 6, SPT 9, SPT 12 ja SPT 15), jotka määrittelevät keskipilarien sijainnit sekä ala- että yläpäissä.
- * Jokaiselle pilarin ylä- ja alapääpisteelle on määritelty oma SLN-elementti, jotka kuvaavat pilarin fyysisistä olemassaoloa.
- * GRP-tunnisteet on synkronoitu vastaamaan SLN-elementtien numeroita, mikä auttaa järjestämään ja tunnistamaan elementit mallissa.

Varmistathan, että tämä päivitys vastaa rakenteen vaatimuksiasi.

KUVIO 21. Tekoälyn mallinnuksen PROG SOFIMSHC osio

SOFIgpt

Olen tehnyt muutokset ohjeidesi mukaan. Alla on päivitetty SOFILOAD-osio, jossa on määritelty kaksi erilaista kuormitustapausta, LC1 ja LC2. LC1 sisältää sekä tasaisesti jakautuneet kuormat (UDL) että pistekuormat pääpalkeille, ja LC2 sisältää vain UDL-kuormat kaikille palkeille.

```

pialntext
Copy code

+PROG SOFILOAD urs:4
HEAD Define Load Cases

UNIT 5

! Load case 1 with UDL and point loads for primary beams
LC 1 TYPE 'NONE' TITL 'UDL and Point Loads'
BEAM SLN 1 TYPE 'pzz' p #udlPrimary[kN/m] ! UDL for primary beam 1
BEAM SLN 2 TYPE 'pzz' p #udlPrimary[kN/m] ! UDL for primary beam 2
BEAM SLN 3 TYPE 'pzz' p #udlSecondary[kN/m] ! UDL for secondary beam 1
BEAM SLN 4 TYPE 'pzz' p #udlSecondary[kN/m] ! UDL for secondary beam 2
BEAM SLN 5 TYPE 'pzz' p #udlSecondary[kN/m] ! UDL for secondary beam 3

BEPL SLN 1 TYPE 'pz' P #pointLoad[kN] A #pointLoadPos1[m] ! Point load
BEPL SLN 2 TYPE 'pz' P #pointLoad[kN] A #pointLoadPos2[m] ! Point load

! Load case 2 with only UDL for all beams
LC 2 TYPE 'NONE' TITL 'UDL Only'
BEAM SLN 1 TYPE 'pzz' p #udlPrimary[kN/m] ! UDL for primary beam 1
BEAM SLN 2 TYPE 'pzz' p #udlPrimary[kN/m] ! UDL for primary beam 2
BEAM SLN 3 TYPE 'pzz' p #udlSecondary[kN/m] ! UDL for secondary beam 1
BEAM SLN 4 TYPE 'pzz' p #udlSecondary[kN/m] ! UDL for secondary beam 2
BEAM SLN 5 TYPE 'pzz' p #udlSecondary[kN/m] ! UDL for secondary beam 3

END

```

Tämä koodi on nyt mukautettu vastaamaan tarpeitasi, ja se sisältää sekä UDL- että pistekuormat eri palkeille eri kuormitustapauksissa. Huomaa, että olen korvannut #udlLoad-muuttujan #udlPrimary ja #udlSecondary -muuttujilla, kuten pyysit.



KUVIO 22. Tekoälyn mallinnuksen PROG SOFILOAD osio

Lopuksi tekoälyä pyydetään luomaan PROG ASE osio (kuvio 23), siinä suoritetaan rakenteellinen analyysi määritetyille kuormitusyhdistelmille. Rakenneanalyysissä selviää rakenteen muodonmuutokset eri kuormitusyhdistelmien alaisina, kuten taipumat, siirtymät ja jännitykset. Analyysin tulokset ovat perusta rakenteen turvallisuuden ja luotettavuuden arvioinnille. ASE osion jälkeen malli on valmis, ja tuloksia voidaan analysoida SOFiSTiKissa aiemmin mainituilla työkaluilla, kuten System Visualizationilla. Kuviossa 24 valmista mallia tarkastellaan System Visualizationissa.



SOFlgpt

The ASE (Analysis of Structural Elements) part of a SOFiSTiK script is responsible for the structural analysis using the defined load cases. Based on your instructions and considering the format provided in the examples from your knowledge base, the ASE part of the script would generally look like this:

```

plaiNtext
+PROG ASE urs:5
HEAD Perform Structural Analysis

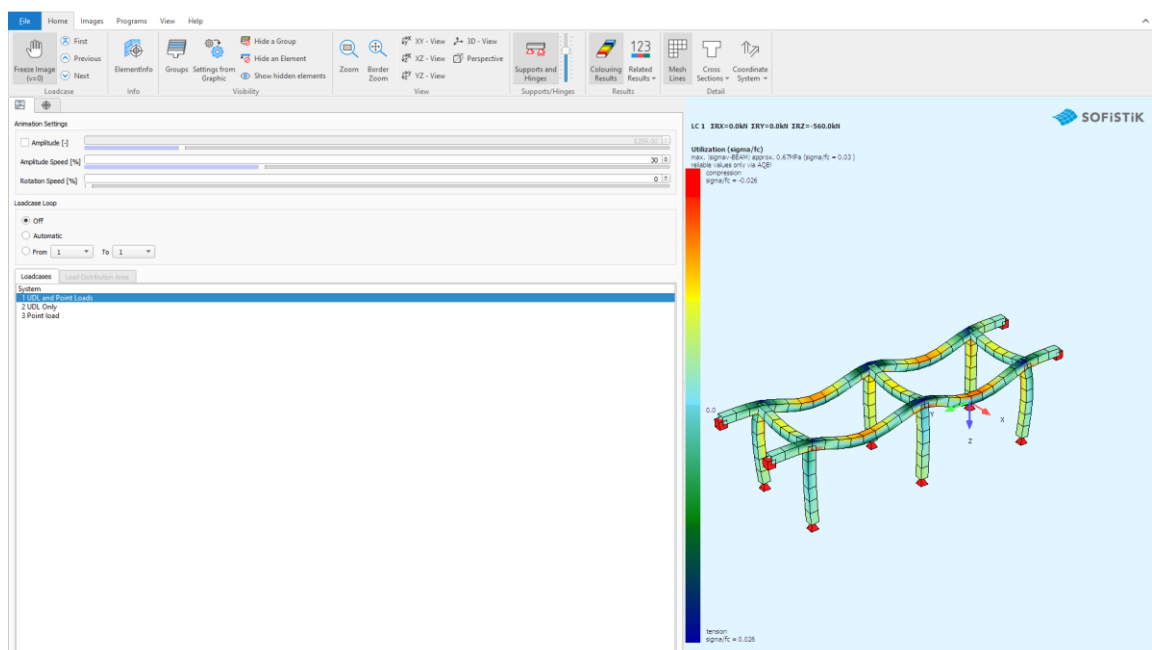
LC ALL

END

```

This section is quite straightforward as it essentially runs the analysis for all load cases defined in the SOFILOAD part of the script. If you have specific analysis requirements or additional parameters you would like to include, please provide those details. Otherwise, this is the typical structure used to execute the analysis phase in SOFiSTiK scripts.

KUVIO 23. Tekoälyn mallinnuksen PROG ASE osio



KUVIO 24. Valmis rakennemalli System Visualization ohjelmassa

5.4 Tekoälyn tekemät virheet mallintaessa palkkisiltaa

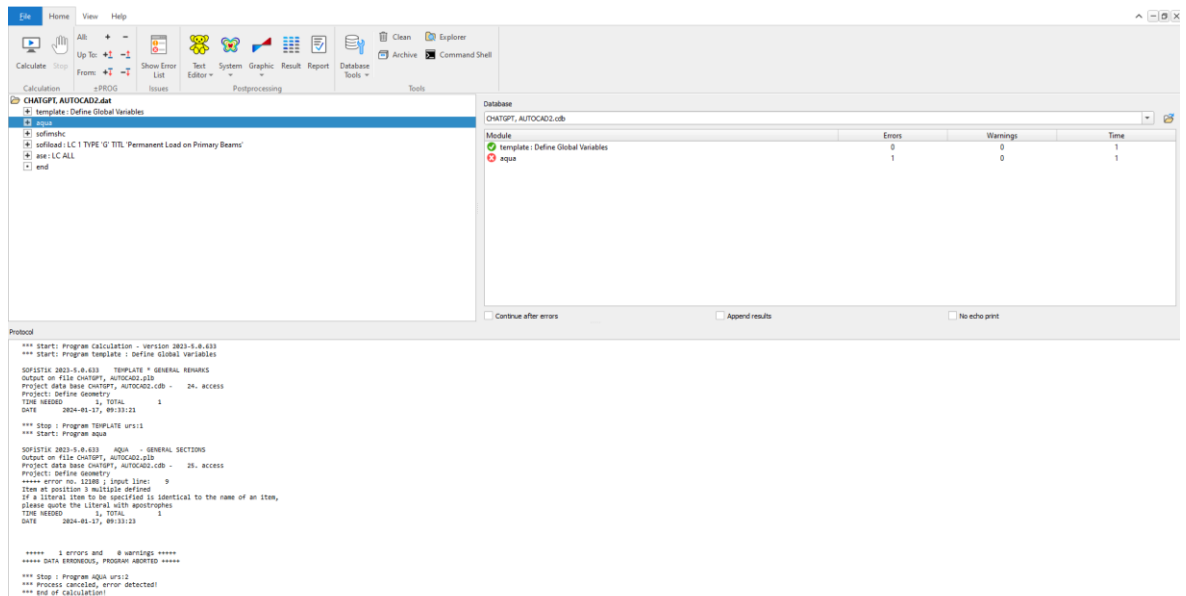
Tekoäly kohtasi monia ongelmia luodessaan arinamallia palkkisillasta. Nämä ongelmat se silti onnistui korjaamaan käyttäjän antamien ohjeistuksien perusteella. Tässä osiossa tarkastellaan tekoälyn tekemiä virheitä palkkisillan prosessissa ja niiden korjaamisen periaatteita.

Aluksi tekoäly teki virheitä mallin hahmottamisessa. Tämä osoittautui haastavaksi, sillä rakennemallin tarkka määrittäminen on avainasemassa, kun tekoälyllä luodaan rakennemalleja. Onneksi virheiden korjaaminen oli mahdollista tekstipohjaisen ohjeistuksen ja konkreettisten esimerkkien avulla. Esimerkiksi kuviossa 25 näkyy, kuinka virheenkorjauksessa käytettiin sekä korjauspyyntöjä että rakennemallin geometrian tarkennusta.

J You
 Unohdit muutamia kohdat; 1. Älä käytä FIX ALL vaan FIX FF, mutta vain pääpalkkien päihin, ja pilareiden alapäihin. Muihin älä laita FIX. 2. Laita GRP samaksi numeroksi kuin luomasi SLN. 3. Ensimmäinen pilari ei ole pisteessä 0,0,0 vaan pisteessä #pillarDistToEnds, 0, 0. Ja niin eteenpäin. Ja pilareita on 6 kappaletta yhteensä. Eli 6 alkupistettä ja 6 loppupistettä. Pääpalkki alkaa pisteestä 0,0,0 ja loppuu #primaryBeamLen,0,0 ja toinen 10m metriä siitä y-suuntaan. Tee nämä muutokset ensin, kiitos

KUVIO 25. Tekoälylle syötettävä virheenkorjaamispyyntö

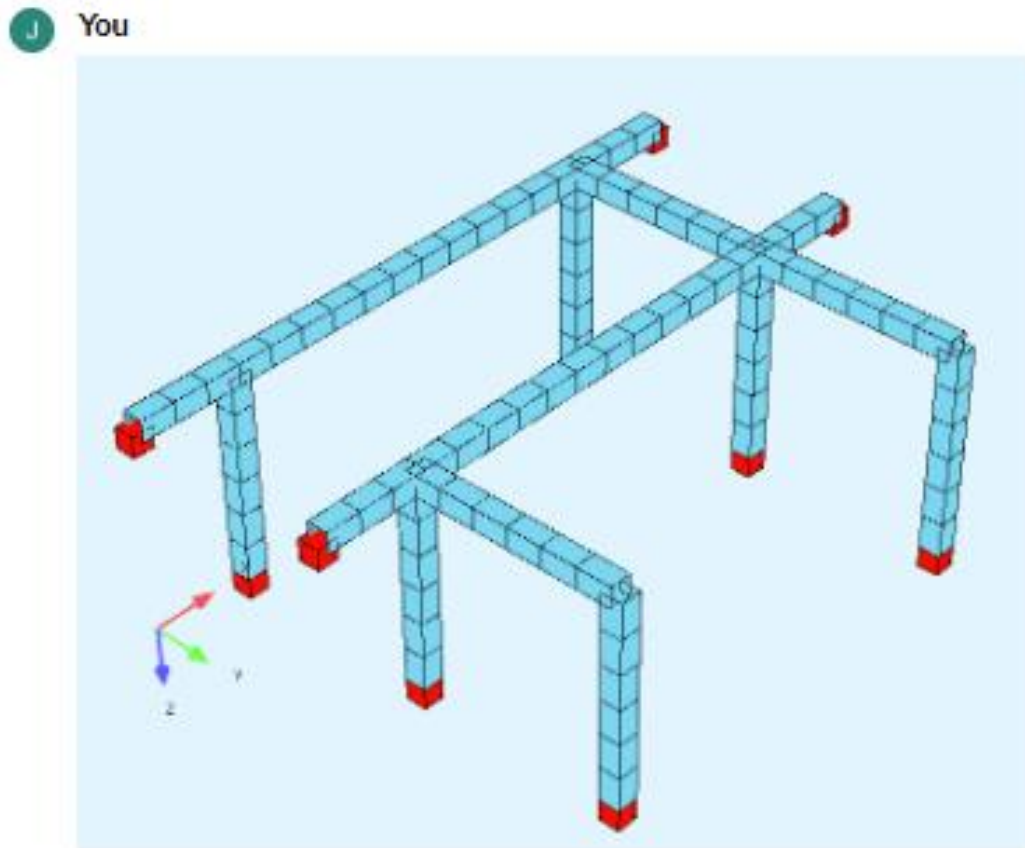
Vaikka virheitä ilmeni usein rakennemallin luonnin aikana, niiden seuraukset eivät olleet vakavia. Tyypillisesti virheet johtivat ainoastaan siihen, että SOFiSTiK-ohjelmisto hylkäsi analyysin, mikä vaati mallin koodauksen muuttamista (kuvio 26). Report Browserin avulla tekstikoodiston rivipaikka saatiin tietoon, jolloin virheen tarkastelu helpottui huomattavasti. Kokemattomalle SOFiSTiKin käyttäjälle tämä Report Browser-osa on välttämätön, jotta virheiden alkuperä saadaan selville.



KUVIO 26. SOFiSTiKin hylkäämä analyysi

Palkkisillan kohteessa mallintamisprosessissa tekoälyn käyttö SOFIMSHC-osi-ossa tuotti suurimman osan virheistä. Erityisen haasteelliseksi osoittautui juuri rakennemallin oikeaoppinen hahmottaminen tekoälyn toimesta. Vaikka tekoäly kykeni lukemaan AutoCAD-piirustuksesta rakenteen mitat sekä kuormat, rakenteen kokonaisvaltainen hahmotus ei onnistunut toivotulla tavalla. Tämän seurauksena tarkka ja selkä sanallinen ohjeistus muodostui välttämättömäksi datan validin tulkin kannalta.

Tekoäly pyrki usein yksinkertaistamaan toimintojaan, esimerkiksi luomalla vain yhden pääpalkin tai vähentämällä pisteiden määrää mallissa. Nämä toimenpiteet johtivat usein rakenneanalyysin hylkäämiseen. Onnistunut korjaus vaati kuitenkin tarkkaa ja yksityiskohtaista ohjeistusta halutun rakennemallin suhteen sekä selkeän lopputuloksen määrittelyä. Joissakin tapauksissa oli tarpeen antaa valmis vastaus, jonka tekoäly sitten sisällytti koodiinsa. Kuviossa 27 on hahmoteltu SOFIMSHC-aliohjelman virheiden lopputulosta. Mikäli tekoäly hahmottaa rakennemallin alusta asti väärin, voi lopputulos muodostua tällaiseksi. Lisäksi väärä hahmottaminen voi johtaa tekoälyn hallusinaatioihin, joita käsiteltiin luvussa 3.3. Rakennemallin luonnissa tapahtui esimerkkitapaus tekoälyn hallusinaatiosta, jossa tekoälyn tarjoama koodi vaikutti sisältävän oikeanlaista dataa, mutta todellisuudessa sen sisältämät komennot eivät liittyneet SOFiSTiKiin lainkaan.



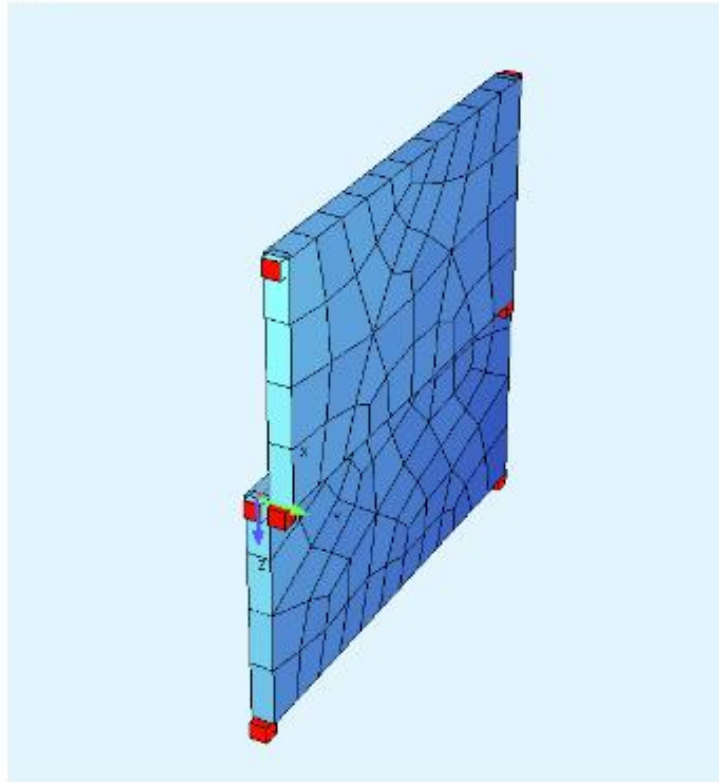
KUVIO 27. Rakennemallin väärinhahmottamisen seuraus tekoälyn kanssa.

5.5 Tekoälyn tekemät virheet mallintaessa tukimuuria

Aiemmin luvussa 5.1 mainittiin, että tukimuurin mallinnuksen prosessi seurasi palkkisillan prosessia, jolloin opinnäytetyössä ei nähty tarpeelliseksi käydä prosessia samalla laajuudella läpi kuin palkkisillassa. Tässä kappaleessa käydään kuitenkin läpi prosessissa tapahtuneet virheet ja niiden korjausperiaatteet. Tämän lisäksi liitteessä 2 on nähtävissä koko keskustelu SOFIgpt:n kanssa tukimuurin mallintamisprosessista, jonka lisäksi malli on lisätty liitteeseen 3.

Tukimuurin rakennemallin luonti seurasi pitkälti esimerkisillan prosessia. Tukimuuria tehdessä kuormitustapaukseen tuli kiinnittää vielä erityisempää huomiota, koska tekoäly hahmotti kuormituksen useaan otteeseen väärin. Tekoäly kohtasi tukimuurin rakennemallin luonnissa samoja ongelmia kuin palkkisillan tapauksessa, esimerkki rakenteen geometrian väärin hahmottamisesta kuviossa 28.

J You



KUVIO 28. Väärinhahmoteltu tukimuuuri

Vaikka tekoäly kykeni lukemaan AutoCAD-piirustuksesta tukimuurin mitat sekä kuorman, rakenteen kokonaisvaltainen hahmotus ei siltä onnistunut. Tekoäly ei hahmottanut XYZ- koordinaatistoa SOFiSTiKista, ja tästä seurasi SPT- pisteiden väärät koordinaatit, jolloin koko rakennemalli muuttui vääräksi. Tämän takia tarkka ja selkeä sanallinen ohjeistus muodostui välttämättömäksi oikeellisen datan saamiseksi. Tukimuurin tapauksessa tekoäly alkoi myös hallusinoimaan enemmän. Esimerkiksi SOFILOAD-osiossa se sisällytti koodiin itse keksimän moduulimerkinnän "GAMZ" (kuvio 29).



SOFIcpt

SOFILOAD-osiossa määrittelemme kuormatukset, jotka aiheutuvat esimerkiksi maapaineesta ja muista kuormista. Tässä esimerkki SOFILOAD-osion koodista, joka mukailee Esimerkki6:ssa käytettyä rakennetta:

```
plaintext Copy code

+PROG SOFILOAD urs:4
HEAD Define Load Cases for the Retaining Wall

UNIT 5

! Define distributed loads for the vertical slab
LC 1 TYPE 'UDL' TITL 'Distributed Load on Vertical Slab'
QUAD GRP 10 p -#topLoad[kN/m2] TYPE 'GZ' GAMZ 1

! Define distributed loads for the horizontal slab
LC 2 TYPE 'UDL' TITL 'Distributed Load on Horizontal Slab'
QUAD GRP 20 p -#botLoad[kN/m2] TYPE 'GZ' GAMZ 1

END
```

Tässä koodissa:

- `UNIT 5` asettaa yksiköt niin, että kaikki kuormat on ilmaistu kilonewtoneina per metri neliö.
- `LC 1` ja `LC 2` ovat kuormitustapaukset (load cases) vertikaaliselle ja horisontaaliselle laatalle.
- `QUAD GRP` viittaa kuormien asettamiseen tietyille ryhmälle (grp), joka on määritelty SOFIMSHC-osiossa.
- `p -#topLoad[kN/m2]` ja `p -#botLoad[kN/m2]` ovat kuormituksen suuruudet, ja miinusmerkki (-) osoittaa, että kuormat ovat alaspäin suuntautuvia.
- `TYPE 'GZ'` määrittelee, että kyseessä on pystysuora kuorma Z-suunnassa, ja `GAMZ 1` on kerroin, joka määrittää kuorman vaikutuksen laatan kokonaispinta-alaan.

KUVIO 29. Tekoälyn hallusinoima SOFILOAD-osio

Koko prosessin aikana tulleistä virheistä tekoäly pystyi korjaamaan suurimman osan itsenäisesti, pelkästään kertomalla virheen kohdan. Kuitenkin tukimuurin SOFILOAD-osiossa tekoäly ei kyennyt korjaamaan virhettä itse, vaan sille jouduttiin syöttämään oikea koodirivi. Tämän opinnäytetyön keskeistä tutkimuskysymystä varten eli pystyykö tekoäly luomaan virheetöntä dataa, olisi tärkeää, että tekoäly pystyisi itsenäisesti korjaamaan virheensä.

5.6 Mallinnuksen erilaisuudet esimerkkikohteissa

Tässä kappaleessa käydään läpi näiden kahden esimerkkitapauksen erilaisuuksia. Esimerkkitapaukset olivat rakenteiltaan hyvin erilaiset, mutta silti niiden prosessit seurasivat hyvin toisiaan. Tulee silti muistaa, että tekoälylle oli tullut liittää tiedostoiksi molemmista rakenteista esimerkkimallit, joita tekoäly sitten johdatteli. Tästä syystä rakenteiden erilaisuus ei ollut ongelma tekoälylle. Molemmissa tapauksissa tekoälylle annettavia esimerkkimalleja seurattiin suurimmilta osin, niistä muokattiin vain rakenteiden dimensiot sekä kuormien suuruudet. Valmiiden mallien pohjalta tekoäly pystyi luomaan rakenteista mallit ja niiden kuormitukset nopeasti.

Eroavaisuuksia tuli silti ilmi, näiden kahden tapauksen välillä eroavaisuudet ilmenivät lähinnä vain kuormituksen eli SOFILOAD-osiossa. Palkkisillassa virheitä kohdistui lähinnä vain SOFIMSHC-osioon, kun taas tukimuurissa virheitä ilmeni niin SOFIMSHC, kuin SOFILOAD-osioissa. Eroavaisuudeksi prosessin aikana näissä tapauksissa tuli pelkästään kuormitusten suhteen. Vaikka molemmissa tapauksissa oli erityisen tärkeää huolehtia rakenteen oikeanlaisesta hahmottamisesta, niin virheenkorjauksen lähestymistavat erosivat keskenään. Palkkisillassa jouduttiin kiinnittämään erityistä huomioita rakenteen oikein hahmottamiseen palkkien sijainnin osilta. Tekoälylle jouduttiin useaan otteeseen selittämään, minkä pisteiden kautta kyseiset palkit kulkivat. Tekoälylle myös jouduttiin antamaan valmiit pisteet rakenneosille.

Palkkisillan mallissa tekoälyn tekemät virheet keskittyivät lähinnä vääriin SPT-pisteisiin palkkien välillä, mutta tukimuurin osin mallinnuksen vaikeudet koskivat suurimmaksi osin taas SPT-pisteiden koordinaatteja. Jotta rakenteesta tuli oikealainen tekoälylle jouduttiin antamaan oikeat koordinaatit suoraan, sillä se ei itsenäisesti pystynyt korjaamaan niitä oikeiksi. Tukimuurissa SOFIMSHC-osion virheiden lisäksi SOFILOAD-osio tuotti ongelmia. Tekoäly alkoi moneen otteeseen hallusinoimaan antaessaan koodin SOFILOAD-osiota varten. Kuten SPT-pisteiden koordinaatiston tapauksessa, niin tässäkin tekoälylle jouduttiin antamaan valmis koodirivi suoraan, sillä se ei itsenäisesti pystynyt korjaamaan virheitä (kuvio 30). Ongelmia tässä tapauksessa tuotti myös Chat GPT 4.0:n rajallinen keskustelu määrä aikaa kohden, tämä on nähtävissä myös kuviossa 30.

You

Et käyttänyt samaa kuin esimerkissä6 ; QUAD grp 10 INC 10 TYPE pyy #slabudl Z 0
DPZ 5 Z 0 0 F0 10
Käytä tätä muotoa mutta #dlabudl tilalle 16

SOFigpt

Pahoitteluni väärinkäsityksestä. Tässä korjattu SOFILOAD-osio käyttäen annettua formaattia ja korvaten `#slabudl` arvolla 16:

```
plaintext Copy code

+PROG SOFILOAD urs:4
HEAD Define Load Cases

UNIT 5

LC 1 TYPE 'NONE' TITL 'Traffic Load for Wall'

! Applying linearly variable distributed load to the vertical slab (grp
QUAD grp 10 INC 10 TYPE pyy -16 Z 0 DPZ 5 Z 0 0 F0 10

END
```

Tässä koodissa:

- Käytämme `QUAD` komentoa määrittämään lineaarisesti muuttuvan jakautuneen kuorman vertikaaliselle laatalle (ryhmä 10).
- `TYPE pyy` osoittaa, että kuorma on y-suunnassa, ja sen intensiteetti vähenee lineaarisesti Z-akselin suunnassa 5 metrin etäisyydellä Z0:sta (joka on 0) arvosta 16 kN/m² arvoon 0 kN/m².

You

nyt ASE osio

SOFigpt

You've reached the current usage cap for GPT-4, please try again after 3:10 PM. [Learn more](#)



KUVIO 30. Chat GPT 4.0:n rajallinen keskusteluominaisuus

6 JOHTOPÄÄTÖKSET JA JATKOKEHITYS

6.1 Johtopäätökset

Tutkimuksessa pyrittiin selvittämään tekoälyn tarjoamia mahdollisuuksia SOFiS-TiKissa. Työn tavoitteena oli saada tietoa siitä, että voidaanko tekoälyä käyttää ja kouluttaa kirjoittamaan käytettävää koodia Teddyyn. Työ toteutettiin räätälöimällä ChatGPT 4.0:sta SOFIgpt. SOFIgpt:lle annettiin käyttöön SOFiSTiKin käyttöohje, ohjeistukset sekä samantapaisten esimerkkimallien Teddy-koodit. Tämän tutkimuksen tulokset osoittavat, että tekoälyn hyödyntäminen rakennesuunnittelussa, erityisesti edellä mainitun SOFIgpt:n avulla, tarjoaa merkittäviä etuja. Onnistuneesti toteutettu kokeilu, jossa tekoäly loi rakennemallit AutoCAD-piirustuksien ja sanallisten selitysten perusteella, vahvistaa tekoälyn potentiaalia rakennesuunnittelun automatisoinnissa. Tekoälyllä on kyky tuottaa virheetöntä koodia Teddyyn. Tämän lisäksi se pystyy käsittelemään erilaisia rakenteita, kunhan se on asiamukaisesti koulutettu. Tämä soveltuvuus myös korostaa sen arvoa suunnitteluprosessissa. Työn antamien vastausten on tarkoitus edesauttaa toimeksiantajaa rakennesuunnittelussa.

Tutkimuksen perusteella tekoäly voi nopeuttaa suunnittelutyötä mahdollistamalla rakennemallien ja kuormitusten automaattisen generoinnin. Tämä vapauttaa suunnittelijoiden aikaa muille kriittisille tehtäville ja vähentää inhimillisten virheiden riskiä, mikä taas parantaa suunnittelun laatua ja tehokkuutta.

Tulevaisuuden kehityssuosituksen keskittyvät erityisesti tekoälyn koulutusprosessin parantamiseen ja sen kykyyn hahmottaa monimutkaisempia rakennemalleja. Tämä on tärkeää, sillä tutkimuksessa havaittiin, että nykyisellä kehitystasolla tekoäly kohtaa vaikeuksia monimutkaisten rakenteiden kanssa, selviytyen paremmin yksinkertaisista malleista. Lisäksi tekoälylle tulisi kehittää ohjeita itsenäiseen virheiden korjaukseen.

Tehokkaan tekoälyn, kuten SOFIgpt:n, kehittäminen edistyneemmäksi työkaluksi on tapaustutkimuksen tulosten mukaan perusteltua tulevaisuuden rakennesuunnitteluprojekteissa. Tekoälyn käytön lisääntyessä SOFiSTiK-ohjelmiston manuaalinen käyttö jatkuu, sekä tekoälyn integrointi saattaa jopa lisätä sen käyttöä

tarjoamalla tehokkaampia tapoja rakennemallien luomiseen ja analysointiin. Valinta kohdistui SOFiSTiK-ohjelmistoon sen tekstipohjaisen mallintamisen mahdollistavan Teddyn ansiosta, mikä tukee tekoälyn soveltamista rakenneanalyysissä. Toki muitakin rakenneanalyysi ohjelmistoja on tarjolla, mutta näiden integraatio tekoälyn kanssa edellyttää API:n käyttöä, SOFiSTiKin kanssa integraatio oli helpompi toteuttaa.

Työn avulla voidaan myös parantaa uusien rakennesuunnittelijoiden osaamista. Kuten aiemmin mainittiin niin ammattikorkeakouluissa ja yliopistoissa rakenneanalyysi ohjelmistojen opetus on erittäin vähäistä, mutta tämän työn ansiosta kokemattomillakin rakennesuunnittelijoilla on mahdollisuus, sekä tarvittava tieto opetella ohjelmistoja itsenäisesti. Vaikka tämän opinnäytetyö pohjautui tekoälyn valjastamiseen rakenneanalyysien teossa, voitaisiin tekoälyä käyttää ohjelmistojen opettajana. Ja päinvastoin kuin koulutuksessa nykyään, tekoäly toimisi yksityisenä opettajan, jolloin oppiminen on tehokkainta.

Yhteenvedona voidaan todeta, että tekoälyn soveltaminen rakennesuunnitteluun avaa uusia mahdollisuuksia suunnittelutyön tehostamiseksi. Tämä tutkimus loi perustan tulevaisuuden tutkimuksille ja kehitykselle, joka keskittyy tekoälyn roolin laajentamiseen rakennesuunnittelussa. Jatkokehityksen myötä voidaan odottaa entistä monipuolisempien ja monimutkaisempien rakenteiden suunnittelun automatisointia, mikä edistää koko alan tehokkuutta ja innovatiivisuutta.

6.2 Jatkokehitys

Tämä tutkimus on tarjonnut lupaavan lähtökohdan tekoälyn hyödyntämiselle rakennesuunnittelussa, erityisesti SOFiSTiK-ohjelmiston kontekstissa. Kehityspolkuja on monia, ja tässä opinnäytetyössä esitetty tekstipohjainen mallinnus Teddy-ohjelmistossa on vain yksi esimerkki tekoälyn soveltamisen laajuudesta. Tekoälyn integrointi SOFiSTiKin eri osa-alueille, kuten SOFIPLUSsan ja AutoCAD:n väliseen yhteistyöhön, pystyy avaamaan uusia mahdollisuuksia geometrinen mallien luomiseen. Tämän lisäksi, tekoälyn integraatio työkaluihin kuten Exceliin ja Grasshopperiin, tarjoaa ennennäkemättömiä mahdollisuuksia parametrisen rakennesuunnitteluun, mikä nopeuttaisi rakennesuunnittelua entisestään.

Jatkokehityksen keskiössä tulisi olla tekoälyn koulutusprosessin monipuolistaminen. Rakennemallien laajan kirjon ja erilaisten suunnittelutekniikoiden ja koulutusmateriaalina käyttö voi parantaa tekoälyn ymmärrystä ja kykyä käsitellä monimutkaisia rakenteita. Erityisesti virheentunnistus- ja vianmääritystekniikoiden opettaminen on avainasemassa, jotta tekoälystä saadaan tehokas työkalu suunnittelun eri vaiheissa. Tämä lähestymistapa ei vain paranna tekoälyn suorituskykyä, vaan myös tehostaa suunnitteluprosessia, vähentäen inhimillisen virheen mahdollisuutta sekä vähentäen suunnitteluun tarvittavaa aikaa.

Esimerkkinä tästä on palkkisilta kohteen arinamallin suunnittelu, jossa tekoälyn kehitys voisi mahdollistaa monimutkaisten poikkileikkausten automatisoidun suunnittelun, tässä tapauksessa sillan arinamallin. Tämä osoittaa, että tekoälyllä on potentiaalia ratkaista haasteita, jotka ovat toistaiseksi olleet automatisoidun suunnittelun ulottumattomissa. Tekoälyn kyky hahmottaa piirustuksia ja sen soveltaminen monimutkaisempiin rakenteisiin ovat keskeisiä kehityskohteita, jotka voivat merkittävästi laajentaa sen käyttöaluetta rakennesuunnittelussa.

Jatkokehityksessä on tärkeää myös huomioida SOFIgpt:n suoran kehityksen tarjoama potentiaali. Ensimmäisenä askeleena SOFIgpt:n kehittämisessä olisi tekoälyn soveltamisalan laajentaminen SOFIStiKissa. Tämä kattaisi erityisesti sillanrakennuksen ja rakenneanalyysien kannalta kriittiset työkalut, kuten AQB, BEMESS, DYNA, ELLA ja MAXIMA, painottaen toimeksiantajan näkökulmasta keskeisiä alueita.

Jatkossa on tärkeää keskittyä tekoälyn soveltamisen laajentamiseen useammille rakennetyypeille ja suunnitteluprosessin eri vaiheille. Tekoälyn avulla voidaan tehostaa projektien suunnittelua, mahdollistaen ajan säästön ja resurssien paremman allokaation esimerkiksi esteettisyyden, kustannustehokkuuden ja CO₂-päästöjen vertailuun. Tämän tutkimuksen onnistuminen antaa vahvan perustan tekoälyn jatkokehitykselle rakennesuunnittelun alalla, tarjoten uusia näkökulmia ja menetelmiä suunnittelutyön tehostamiseen ja innovatiivisuuden edistämiseen.

LÄHTEET

Alanko, J. 2016. Massiivisen laattasillan kansilaatan yläpinnan betonipeitteen uusiminen. Rakennetun ympäristön tiedekunta. Tampereen yliopisto. Diplomityö. Luettu 9.2.2024. <https://trepo.tuni.fi/bitstream/handle/123456789/24441/Alanko.pdf?sequence=3&isAllowed=y>

Alfame. 2018. API:t käytännössä. Viitattu 20.2.2024. <https://www.alfame.com/ajankohtaista/apit-kaytannossa-selkokielineen-katsaus>

Alkaissi, H & McFarlane, S. 2023. National Library of Medicine, Artificial Hallucinations in ChatGPT: Implications in Scientific Writing. Viitattu 5.2.2024. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9939079/>

Amazon n.d, What is an API (Application Programming Interface) Verkkosivu. <https://aws.amazon.com/what-is/api/>

Apellaniz, D. 2023. Revolutionise Structural Engineering with ChatGPT and SOFiSTiK. Viitattu 5.2.2024. <https://sofistikforyou.com/revolutionise-structural-engineering-with-chatgpt-and-sofistik/>

Artificial Intelligence (AI). 2024. LinkedIn, How can feedback help you improve your AI development team's strengths and weaknesses. Viitattu 17.1.2024. <https://www.linkedin.com/advice/1/how-can-feedback-help-you-improve-your-iaace>

Baduge, S & Thilakarathna, S & Perera, J & Arashpour, M. 2022. Automation in Construction 141, 3. Artificial Intelligence and Smart Vision for Building and Construction 4.0: Machine and Deep Learning Methods and Applications. Viitattu 5.2.2024. https://www.researchgate.net/publication/361501987_Artificial_Intelligence_and_Smart_Vision_for_Building_and_Construction_40_Machine_and_Deep_Learning_Methods_and_Applications

Baxter, K & Schlesinger, Y. 2023. Harvard Business Review. Managing the Risks of Generative AI. <https://hbr.org/2023/06/managing-the-risks-of-generative-ai>

Bhopal, BGI. 2023. Bansal Group of Institutes. Artificial Intelligence in Civil Engineering: Revolutionising The Future. Viitattu 5.2.2024 <https://bgibhopal.com/blog/artificial-intelligence-in-civil-engineering-revolutionising-the-future/>

Council of Europe n.d. History of Artificial Intelligence. Verkkosivu. Viitattu 5.2.2024. <https://www.coe.int/en/web/artificial-intelligence/history-of-ai>

Empirica n.d. Esineiden internet yksinkertaisesti selitettynä. Verkkosivu. Viitattu 20.2.2024. <https://www.empirica.fi/iot.html>

Gangu, P. 2018. Harvard Business Review. We need to Approach AI Risks Like We Do Natural Disasters. <https://hbr.org/2018/02/we-need-to-approach-ai-risks-like-we-do-natural-disasters>

Goly, A. 2023. LinkedIn. The Rise of the Super Civil Engineer: How AI is Transforming the AEC Industry. <https://www.linkedin.com/pulse/rise-super-civil-engineer-how-ai-transforming-aec-industry-goly/>

Hörold, A. 2023. Dlubal 2023, How to Integrate AI to RFEM using API. Webinaari. Julkaistu 5.12.2023. Viitattu 5.2.2024 <https://www.dlubal.com/en/support-and-learning/learning/webinars/003065>

Kauppinen, T. 2021. IoT-alustojen tietoturvan soveltuvuusvertailu CityIoT-hankkeeseen. Informaatioteknologian ja viestinnän tiedekunta. Tampereen yliopisto. Diplomityö. <https://trepo.tuni.fi/bitstream/handle/10024/124605/Kauppinen-Toni.pdf?sequence=2&isAllowed=y>

Kutskov, K. 2023. 9 Steps of Debugging Deep Learning Model Training. Viitattu 9.2.2024 <https://neptune.ai/blog/debugging-deep-learning-model-training>

Kuzlu, M & Fair, C & Guler, O. 2021. Engineering Technology Faculty Publications 2-2021. Role of Artificial Intelligence in the Internet of Things (IoT) cybersecurity. Viitattu 15.1.2024. https://digitalcommons.odu.edu/cgi/viewcontent.cgi?article=1179&context=engtech_fac_pubs

Marr, B. 2023. The 15 Biggest Risks Of Artificial Intelligence. Viitattu 14.1.2024. <https://bernardmarr.com/the-15-biggest-risks-of-artificial-intelligence/>

Mishra, G. 2009. Cantilever Retaining Wall – Functions and Design Considerations. Viitattu 20.1.2024. <https://theconstructor.org/geotechnical/cantilever-retaining-wall/1991/>

Nieminen, J. 2022. Sillan maatukien rakennesuunnittelun automatisointi. Rakentamisen ylempi tutkinto-ohjelma. Tampereen ammattikorkeakoulu. Opinnäytetyö. Luettu 17.1.2024. https://www.theseus.fi/bitstream/handle/10024/784485/Nieminen_Joni.pdf?sequence=2&isAllowed=y

OpenAI. 2024. Improving Image Generation with Better Captions. Luettu 5.2.2024. <https://cdn.openai.com/papers/dall-e-3.pdf>

Pismisoglu, S. 2023. Bradley. Cybersecurity Risks to AI Adoption in Construction. Viitattu 14.1.2024. <https://www.bradley.com/insights/publications/2023/07/cybersecurity-risks-to-ai-adoption-in-construction>

Putkinen, J. 2021. Jännitetyn betonisen palkkisillan rakenneanalyysin ja mitoituksen automatisointi. Rakennetun ympäristön tiedekunta. Tampereen yliopisto. Diplomityö. Viitattu 16.1.2024. <https://urn.fi/URN:NBN:fi:tuni-202108186621>

Rakennustieto. 2015. RT 89-11175 Muurit ja tukimuurit. Ohjekortisto. Viitattu 16.1.2024 <https://kortistot.rakennustieto.fi/kortit/RT%2089-11175>

Rockwell, A. 2017. Harvard University. The history on Artificial Intelligence. Viitattu 5.2.2024

<https://sitn.hms.harvard.edu/flash/2017/history-artificial-intelligence/>

Rustell, M. 2023. New Civil Engineering. How civil engineers can use generative AI to create value in projects. <https://www.newcivilengineer.com/opinion/how-civil-engineers-can-use-generative-ai-to-create-value-in-projects-11-12-2023/>

Saheli, H & Burgueno, R. 2018. Elsevier 171, 170-189. Emerging artificial Intelligence methods in structural engineering. <https://www.semanticscholar.org/paper/Emerging-artificial-intelligence-methods-in-Salehi-Burque%C3%B1o/c29ad537c20bbf57d1c1de75e6069ca3e157958a>

Sofistik AG. n.d. "About Sofistik". Verkkosivu. Viitattu 16.1.2024.

<https://www.sofistik.com/company/about-us>

Sofistik AG. 2020. Bridge Design Training #01 – Introduction SOFiSTiK Software. Sofistik AG. YouTube-video. Julkaistu 8.12.2020. Katsottu 16.1.2024.

<https://www.youtube.com/watch?v=6Oqp-2p0cZA&list=PLk-NZCJG1CDyPJeKGCyGzUEz3QzG54QXlj&index=2&t=1798s>

SOFiSTiK FOR YOU. 2022. SOFiSTiK Text Input – Line Breaks in CADiNP. Viitattu 14.2.2024.

<https://www.linkedin.com/pulse/sofistik-text-input-line-breaks-cadinp-sofistikforyou/>

Suikkanen, P. & Jaskari, K. 2023. Tekoälyn kolme uhkaa ja kolme mahdollisuutta – asiantuntijat ja tekoäly itse vastaavat Ylälle. Uutinen. Julkaistu 2.6.2023. Viitattu 14.1.2024.

<https://yle.fi/a/74-20034602>

Tal, L. 2023. Snyk. What are AI hallucinations and why should developers care? Viitattu 20.2.2024.

<https://snyk.io/blog/ai-hallucinations/>

Tampereen yliopisto n.d. Tekoälyn historia. Verkkosivu. Viitattu 5.2.2024.

<https://projects.tuni.fi/kite/tekoalysta-yleisesti/tekoalyn-historia/>

VectorOne. 2024. Integrating Artificial Intelligence into Your Software: A starting Guide. Viitattu 17.1.2024.

<https://www.vectorone.com/blog/integrating-artificial-intelligence-into-your-software-a-starting-guide>

Väylävirasto. 2023. Eurokoodin soveltamisohje – Geotekninen suunnittelu – NCCI 7, Siltojen ja pohjarakenteiden suunnitteluohjeet. Viitattu 17.1.2024

https://ava.vaylapilvi.fi/ava/Julkaisut/Vaylavirasto/vo_2023-14_NCCI_7_web.pdf

Yashanov, A. 2020. SOFiSTiK FOR YOU, Integration Between Calculation

Modelling & Building Information Modelling. Viitattu 9.2.2024 <https://sofistikforyou.com/integration-of-building-calculation-modelling-and-building-information-modelling/>

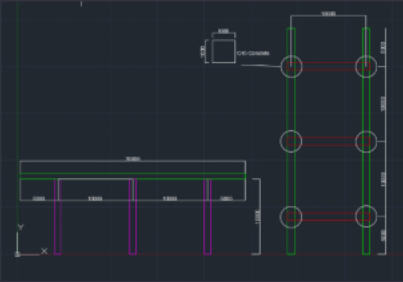
LIITTEET

Liite 1. Palkkisillan mallinnuksen prosessi

1/12

SOFigpt

You



Tee Teddy tiedosto kuten olen antanut lähtötiedoksesi esimerkki4 tiedoston. Huomaa että rakenteelle tulee kuormia. Kaikkien, pilareiden, palkkien kuormat ovat 5kN/m. Rakenteelle tulee myös pistekuormia, ne ovat pääpalkkien alkukohdista 2.5m metrien päässä sekä rakenteen keskellä eli 15m päässä alusta. Pilarit ja palkit ovat 1000×1000mm ja C45. Huomioi että pääpalkit ovat pilareiden päällä, HUOMAA myös että z- akseli osoittaa alaspäin eli SPT tehdessäsi huomioi akselin pisteet. Välipalkit tulevat pilarin pääpisteestä vierekkäisen pilarin pääpisteelle. Kuvasta näet että pinkillä on osoitettu pilarit jotka ovat 1000×1000mm ja 10m pitkiä, vihreällä pääpalkit jotka ovat 1000×1000mm ja 30m pitkiä. Sekä Välipalkit jotka ovat punaisella 1000×1000mm ja 10m. Tee template koodi pelkäästään

SOFigpt

SOFigpt

To create a SOFISTIK TEDDY file based on the given AutoCAD drawing and instructions, we need to define the geometry and the loads according to the specifications. The provided image shows a structure with primary beams of 30m length, secondary beams (intermediate beams) of 10m, and pillars of 10m height, all with a cross-section of 1000×1000mm and C45 concrete grade. Uniformly distributed loads (UDL) of 5kN/m are applied to all beams and pillars, and point loads are applied 2.5m from the start of the primary beams and at the center (15m from the start).

Below is the TEMPLATE part of the TEDDY file for this structure:

plantest

Copy code

```
+PROG TEMPLATE ura:1
HEAD Define Global Variables

STO#pillarHeight 10      $ Height of the pillars [m]
STO#primaryBeamLen 30    $ Length of the primary beams [m]
STO#secBeamLen 10        $ Length of the secondary beams [m]
STO#pillarDistToEnds 5    $ Distance from primary beam ends to pillar
STO#secBeamDist 10       $ Distance in y-axis between pillars and se
STO#pointLoadPos1 2.5    $ Position of first point load on primary b
STO#pointLoadPos2 15     $ Position of second point load on primary
STO#pointLoad 5          $ Point load magnitude on primary beams [kN
STO#udlload 5            $ Uniformly distributed load on beams and p
STO#wb 1                 $ Width of cross-section [m]
STO#hb 1                 $ Height of cross-section [m]
STO#concGrade 45         $ Concrete grade C45
```

SOFigpt

```
STO#concGrade 45      $ Concrete grade C45
STO#distSecBeams #secBeamLen $ Distance between secondary beams [m]
STO#distPillars #secBeamDist $ Distance between pillars [m]

END
```

The above script sets up global variables for use in later parts of the SOFISTIK analysis, such as defining materials, cross-sections, structural elements, loads, and performing the analysis. The units are set according to your instructions, with meters for lengths and kN for forces. The load positions are based on the description you've provided, and all the other variables are named to reflect their purpose within the model.

Please confirm if this setup meets your requirements or if you need the full SOFISTIK TEDDY script including the other sections such as AQUA, SOFIMSHC, SOFILOAD, and ASE.

You

Tee nyt Aqua vaihe

SOFigpt

Based on the provided information and the TEMPLATE code already defined, the AQUA phase for the SOFISTIK TEDDY script will define the materials and cross-sections for the structure. The cross-section for the pillars, primary beams, and secondary beams will be the same, with dimensions 1000x1000mm and concrete grade C45.

SOFigpt

Here is the AQUA code for this structure:

plaintextCopy code

+PROG AQUA urs:2
HEAD Define Materials and Cross-Sections

UNIT 5
NORM 'SFS' 'en199x-200x-bridge' COUN 358 CAT 'B2' SNOW '2.50' UNIT 5

CONC NO 1 C #concGrade TITL 'C45 Concrete'
SREC NO 1 B #b[m] H #h[m] MNO 1 TITL 'Pillar/Beam Section'

END

In this code:

- * 'UNIT 5' is used to set the unit system where length is in meters and force is in kN.
- * 'NORM' specifies the design code, which is Eurocode for bridges with specific parameters for country code, category, snow load, etc.
- * 'CONC' defines the concrete material with a specified grade.
- * 'SREC' defines a square cross-section for the beams and pillars, using the global variables '#b' and '#h' for width and height, respectively.

This completes the AQUA stage of the TEDDY script, where materials and sections are now defined for the structural model. Next stages would involve defining the geometry (SOFIMSHC), applying loads (SOFILOAD), and performing the structural analysis (ASE). If you need these stages as well, please let me know, and I will provide them accordingly.

SOFigpt

(SOFIMSHC), applying loads (SOFILOAD), and performing the structural analysis (ASE). If you need these stages as well, please let me know, and I will provide them accordingly.

You

Tee SOFIMSHC vaihe

SOFigpt

Continuing from the TEMPLATE and AQUA phases, the SOFIMSHC phase will define the geometry of the structure based on the provided AutoCAD drawing. Given the specifications for the pillars, primary beams, and secondary beams, the SOFIMSHC script will establish the structural elements and their connectivity.

Here is the SOFIMSHC script:

SOFigpt

plaintextCopy code

```
+PRG SOFIMSHC urs:3
HEAD Define Geometry of Structure
SYST 3D GDIV 10000
UNIT 5

! Define base points for the pillars
SPT 1 x 0 y 0 z 0 FIX ALL
SPT 2 x 0 y #pillarDistToEnds z 0 FIX ALL
SPT 3 x 0 y #secBeamDist z 0 FIX ALL
SPT 4 x 0 y #pillarDistToEnds+#secBeamDist z 0 FIX ALL

! Define top points for the pillars
SPT 5 x 0 y 0 z #pillarHeight
SPT 6 x 0 y #pillarDistToEnds z #pillarHeight
SPT 7 x 0 y #secBeamDist z #pillarHeight
SPT 8 x 0 y #pillarDistToEnds+#secBeamDist z #pillarHeight

! Define primary beams on top of the pillars
SPT 9 x #primaryBeamLen y 0 z #pillarHeight
SPT 10 x #primaryBeamLen y #pillarDistToEnds z #pillarHeight
SPT 11 x #primaryBeamLen y #secBeamDist z #pillarHeight
SPT 12 x #primaryBeamLen y #pillarDistToEnds+#secBeamDist z #pillarHeight
```

SOFigpt

SPT 12 x #primaryBeamLen y #pillarDistToEnds+#secBeamDist z #pillarHeight

! Define pillars

SLN 1 NPA 1 NPE 5 SNO 1 GRP 'Pillars'

SLN 2 NPA 2 NPE 6 SNO 1 GRP 'Pillars'

SLN 3 NPA 3 NPE 7 SNO 1 GRP 'Pillars'

SLN 4 NPA 4 NPE 8 SNO 1 GRP 'Pillars'

! Define primary beams

SLN 5 NPA 5 NPE 9 SNO 1 GRP 'Primary Beams'

SLN 6 NPA 6 NPE 10 SNO 1 GRP 'Primary Beams'

SLN 7 NPA 7 NPE 11 SNO 1 GRP 'Primary Beams'

SLN 8 NPA 8 NPE 12 SNO 1 GRP 'Primary Beams'

! Define secondary beams between the tops of the pillars

SLN 9 NPA 5 NPE 7 SNO 1 GRP 'Secondary Beams'

SLN 10 NPA 6 NPE 8 SNO 1 GRP 'Secondary Beams'

SLN 11 NPA 9 NPE 11 SNO 1 GRP 'Secondary Beams'

SLN 12 NPA 10 NPE 12 SNO 1 GRP 'Secondary Beams'

END

This script includes the following definitions:

- * **"SYST 3D GDIV 10000"** specifies a 3D system and grid division.
- * **"UNIT 5"** sets the unit system to meters and kN.
- * **"SPT"** defines the support points for the pillars and the end points for the beams. The **"FIX ALL"** command is used at the base of the pillars to fully constrain them.

SOFigpt

SPT defines the support points for the pillars and the end points for the beams. The "FIX ALL" command is used at the base of the pillars to fully constrain them.

SLN defines the beam elements (straight lines) connecting the support points. Each beam is assigned to a group for easier reference in later stages of the analysis.

Please verify the geometry against your project requirements. If you need any adjustments or the complete script including load applications and analysis, let me know and I will proceed accordingly.

You

Unohdit muutamat kohdat: 1. Älä käytä FIX ALL vaan FIX FF, mutta vain pääpalkkien päihin, ja pilareiden alapäihin. Muihin älä laita FIX. 2. Laita GRP samaksi numeroksi kuin luomasi SLN. 3. Ensimmäinen pilari ei ole pisteessä 0,0,0 vaan pisteessä #pillarDistToEnds, 0, 0. Ja niin eteenpäin. Ja pilareita on 6 kappaletta yhteensä. Eli 6 alkupistettä ja 6 loppupistettä. Pääpalkki alkaa pisteestä 0,0,0 ja loppuu #primaryBeamLen,0,0 ja toinen 10m metriä siitä y-suuntaan. Tee nämä muutokset ensin, kiitos

SOFigpt

Let's revise the SOFIMSHC script according to your corrections. We will adjust the fixing conditions, update the group numbers to match the line numbers, correct the starting point of the first pillar, and account for the total number of six pillars. The primary beam will start at point 0, 0, 0 and end at **#secBeamDist, #primaryBeamLen, 0, 0**, with the second beam offset by 10m in the y-direction.

Here is the updated SOFIMSHC script:

SOFIgtpt

```

SPT 11 X #pillarDistToEnds Y 2*#secBeamDist Z #pillarHeight
SPT 12 X #primaryBeamLen-#pillarDistToEnds Y 2*#secBeamDist Z #pillarHe

! Define pillars
SLN 1 NPA 1 NPE 7 SNO 1 GRP 1
SLN 2 NPA 2 NPE 8 SNO 1 GRP 2
SLN 3 NPA 3 NPE 9 SNO 1 GRP 3
SLN 4 NPA 4 NPE 10 SNO 1 GRP 4
SLN 5 NPA 5 NPE 11 SNO 1 GRP 5
SLN 6 NPA 6 NPE 12 SNO 1 GRP 6

! Define primary beams on top of the pillars
SPT 13 X 0 Y 0 Z #pillarHeight FIX FF
SPT 14 X #primaryBeamLen Y 0 Z #pillarHeight FIX FF
SPT 15 X 0 Y #secBeamDist Z #pillarHeight FIX FF
SPT 16 X #primaryBeamLen Y #secBeamDist Z #pillarHeight FIX FF

SLN 7 NPA 13 NPE 14 SNO 1 GRP 7
SLN 8 NPA 15 NPE 16 SNO 1 GRP 8

! Define secondary beams between the tops of the pillars
SLN 9 NPA 7 NPE 9 SNO 1 GRP 9
SLN 10 NPA 8 NPE 10 SNO 1 GRP 10
SLN 11 NPA 9 NPE 11 SNO 1 GRP 11
SLN 12 NPA 10 NPE 12 SNO 1 GRP 12

END

```

SOFIgtpt

In this script:

- * The base of the pillars is fixed in the vertical direction only using 'FIX FF'.
- * Each group number corresponds to the 'SLN' number.
- * The first pillar starts at the point `#secBeamDistToEnds 0 0`.
- * There are six pillars defined, each with a base and top point.
- * Primary beams are defined on top of the pillars, with appropriate fixing and group assignments.
- * Secondary beams are defined between the tops of the pillars.

Please review the revised script and let me know if further modifications are needed.

You

Hienoa, muuta vielä pääpalkkien Z-koordinaatti (-) merkikseksi sillä koordinaatisto on väärinpäin z-suunnassa. Ja sekundaärpalkit menevät niiden mukaan mitkä z-koordinaatiston pisteet ovat pienemmät

SOFIgtpt

Understood, I will now modify the Z-coordinates of the primary beams to be negative, reflecting that the Z-axis points downwards in your coordinate system, and adjust the secondary beams accordingly. Here is the updated SOFIMSHC phase:

SOFIgtpt

```

+PROG SOFIMSHC urs:3
HEAD Define Geometry of Structure
SYST 3D GDIV 10000
UNIT 5

! Define base points for the pillars
SPT 1 X #pillarDistToEnds Y 0 Z -0 FIX FF
SPT 2 X #primaryBeamLen-#pillarDistToEnds Y 0 Z -0 FIX FF
SPT 3 X #pillarDistToEnds Y #secBeamDist Z -0 FIX FF
SPT 4 X #primaryBeamLen-#pillarDistToEnds Y #secBeamDist Z -0 FIX FF
SPT 5 X #pillarDistToEnds Y 2*#secBeamDist Z -0 FIX FF
SPT 6 X #primaryBeamLen-#pillarDistToEnds Y 2*#secBeamDist Z -0 FIX FF

! Define top points for the pillars
SPT 7 X #pillarDistToEnds Y 0 Z -#pillarHeight
SPT 8 X #primaryBeamLen-#pillarDistToEnds Y 0 Z -#pillarHeight
SPT 9 X #pillarDistToEnds Y #secBeamDist Z -#pillarHeight
SPT 10 X #primaryBeamLen-#pillarDistToEnds Y #secBeamDist Z -#pillarHeight
SPT 11 X #pillarDistToEnds Y 2*#secBeamDist Z -#pillarHeight
SPT 12 X #primaryBeamLen-#pillarDistToEnds Y 2*#secBeamDist Z -#pillarHeight

! Define pillars
SLN 1 NPA 1 NPE 7 SNO 1 GRP 1
SLN 2 NPA 2 NPE 8 SNO 1 GRP 2
SLN 3 NPA 3 NPE 9 SNO 1 GRP 3
SLN 4 NPA 4 NPE 10 SNO 1 GRP 4
SLN 5 NPA 5 NPE 11 SNO 1 GRP 5
SLN 6 NPA 6 NPE 12 SNO 1 GRP 6

```

SOFigpt



```

SLN 6 NPA 6 NPE 12 SNO 1 GRP 6

! Define primary beams on top of the pillars
SPT 13 X 0 Y 0 Z -#pillarHeight FIX FF
SPT 14 X #primaryBeamLen Y 0 Z -#pillarHeight FIX FF
SPT 15 X 0 Y #secBeamDist Z -#pillarHeight FIX FF
SPT 16 X #primaryBeamLen Y #secBeamDist Z -#pillarHeight FIX FF

SLN 7 NPA 13 NPE 14 SNO 1 GRP 7
SLN 8 NPA 15 NPE 16 SNO 1 GRP 8

! Define secondary beams between the tops of the pillars
SLN 9 NPA 7 NPE 9 SNO 1 GRP 9
SLN 10 NPA 8 NPE 10 SNO 1 GRP 10
SLN 11 NPA 11 NPE 9 SNO 1 GRP 11
SLN 12 NPA 12 NPE 10 SNO 1 GRP 12

END

```

In this script:

- * The Z-coordinates for the top points of the pillars and the endpoints of the primary beams are negative, indicating that they are below the reference plane (since Z is pointing downwards).
- * The secondary beams are defined between the top points of the pillars that have smaller Z-coordinates (which are numerically larger due to the negative sign).

SOFigpt



Please confirm this adjustment or if there are any more changes required.



You
Loistavaa, tee nyt SOFILOAD osa



SOFigpt
Continuing with the creation of the TEDDY file, we will now define the load cases in the SOFILOAD program. According to the given specifications, there will be uniformly distributed loads (UDL) of 5kN/m on all beams and pillars, and point loads of 5kN at specific locations on the primary beams. The UDLs will be positive values as downward loads, and since the Z-axis points downwards, the point loads will also be positive.

Here's the SOFILOAD section:

```

plaintext Copy code

+PRG SOFILOAD urs:4
HEAD Define Loads on Structure

UNIT 5

! Define uniformly distributed loads on pillars
BEAM SLN 1 TYPE 'pzz' p #udlLoad[kN/m]
BEAM SLN 2 TYPE 'pzz' p #udlLoad[kN/m]
BEAM SLN 3 TYPE 'pzz' p #udlLoad[kN/m]
BEAM SLN 4 TYPE 'pzz' p #udlLoad[kN/m]

```

SOFigpt



```

BEAM SLN 3 TYPE 'pzz' p #udlLoad[kN/m]
BEAM SLN 4 TYPE 'pzz' p #udlLoad[kN/m]
BEAM SLN 5 TYPE 'pzz' p #udlLoad[kN/m]
BEAM SLN 6 TYPE 'pzz' p #udlLoad[kN/m]

! Define uniformly distributed loads on primary beams
BEAM SLN 7 TYPE 'pzz' p #udlLoad[kN/m]
BEAM SLN 8 TYPE 'pzz' p #udlLoad[kN/m]

! Define point loads on primary beams
BEPL SLN 7 TYPE 'pz' p #pointLoad[kN] a #pointLoadPos1[m]
BEPL SLN 7 TYPE 'pz' p #pointLoad[kN] a #pointLoadPos2[m]
BEPL SLN 8 TYPE 'pz' p #pointLoad[kN] a #pointLoadPos1[m]
BEPL SLN 8 TYPE 'pz' p #pointLoad[kN] a #pointLoadPos2[m]

! Define uniformly distributed loads on secondary beams
BEAM SLN 9 TYPE 'pzz' p #udlLoad[kN/m]
BEAM SLN 10 TYPE 'pzz' p #udlLoad[kN/m]
BEAM SLN 11 TYPE 'pzz' p #udlLoad[kN/m]
BEAM SLN 12 TYPE 'pzz' p #udlLoad[kN/m]

END

```

In this section:

- * The **'UNIT 5'** command ensures that the units for the loads are consistent with the previous sections.
- * **'BEAM SLN'** commands apply a UDL to each beam or pillar, specified by the line number.

SOFIapt

- * **"BEAM SLN"** commands apply a UDL to each beam or pillar, specified by the line number.
- * **"BEPL SLN"** commands apply the point loads at the specified locations on the primary beams.
- * The **"TYPE 'pzz'"** and **"TYPE 'pz'"** commands specify the direction of the loads with respect to the global Z-axis.

This script defines the loading conditions for the structure. Please review and confirm if this matches your project's requirements or if any modifications are needed.

You
Kiitos ja nyt vielä ASE

SOFIapt
The final step in the structural analysis process using the TEDDY file for SOFISTIK is the ASE (Automatic System for Frame Analysis) program, which performs the actual analysis of the structure under the defined load cases. Here is how the ASE phase of the script would look:

```
plaintext
+PROG ASE urs:5
HEAD Perform Structural Analysis

UNIT 5
```

SOFIapt

```
plaintext
LC ALL

END
```

In the ASE script:

- * **"UNIT 5"** is reiterated to ensure that the analysis uses the same unit system as defined in previous sections.
- * **"LC ALL"** specifies that all previously defined load cases should be included in the structural analysis.

This completes the basic structure of the TEDDY file for your structural analysis in SOFISTIK. Once you run this script in SOFISTIK, it will perform the analysis based on the defined geometry and loading conditions, and then you can review the results for design validation and further engineering decisions.

Please check the entire TEDDY script to ensure it matches your project requirements and let me know if there are any further steps you would like to discuss or any specific results or output you would like to extract from the analysis.

You
Muokkaa SOFILOAD kohtaa, eli jos kyseessä on metrikkuorma eli udi älä laita p vaan pa

SOFIapt

SOFIapt
I apologize for the oversight. When specifying a uniformly distributed load (UDL) in SOFISTIK, the correct syntax is to use 'pa' for the load intensity per unit length along the beam. Let's correct the SOFILOAD section accordingly:

```
plaintext
+PROG SOFILOAD urs:4
HEAD Define Loads on Structure

UNIT 5

! Define uniformly distributed loads on pillars
BEAM SLN 1 TYPE 'pzz' pa #udlLoad[kN/m]
BEAM SLN 2 TYPE 'pzz' pa #udlLoad[kN/m]
BEAM SLN 3 TYPE 'pzz' pa #udlLoad[kN/m]
BEAM SLN 4 TYPE 'pzz' pa #udlLoad[kN/m]
BEAM SLN 5 TYPE 'pzz' pa #udlLoad[kN/m]
BEAM SLN 6 TYPE 'pzz' pa #udlLoad[kN/m]

! Define uniformly distributed loads on primary beams
BEAM SLN 7 TYPE 'pzz' pa #udlLoad[kN/m]
BEAM SLN 8 TYPE 'pzz' pa #udlLoad[kN/m]

! Define point loads on primary beams
BEPL SLN 7 TYPE 'pz' p #pointLoad[kN] a #pointLoadPos1[m]
BEPL SLN 7 TYPE 'pz' p #pointLoad[kN] a #pointLoadPos2[m]
BEPL SLN 8 TYPE 'pz' p #pointLoad[kN] a #pointLoadPos1[m]
BEPL SLN 8 TYPE 'pz' p #pointLoad[kN] a #pointLoadPos2[m]
```


SOFigpt



```
BEPL SLN 8 TYPE 'pz' p #pointLoad[kN] a #pointLoadPos1[m]
BEPL SLN 8 TYPE 'pz' p #pointLoad[kN] a #pointLoadPos2[m]
```

```
! Define uniformly distributed loads on secondary beams
BEAM SLN 9 TYPE 'pzz' pa #udlLoad[kN/m]
BEAM SLN 10 TYPE 'pzz' pa #udlLoad[kN/m]
BEAM SLN 11 TYPE 'pzz' pa #udlLoad[kN/m]
BEAM SLN 12 TYPE 'pzz' pa #udlLoad[kN/m]

END
```

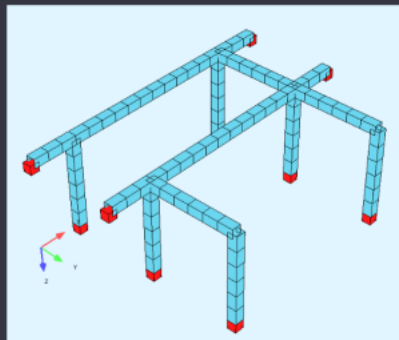
Now the script correctly specifies 'pa' for the uniformly distributed loads on the beams and pillars. The 'p' is still used for the point load 'BEPL' command, which is the correct syntax for specifying a point load. This should now accurately represent the loading conditions for your structural model. Please review and let me know if there are any other adjustments needed.

SOFigpt



other adjustments needed.

You



Loistavaa, mutta nyt kaikki ovat väärin päin. Kuvasta näkyy nykyinen järjestetys niin muuta se haluttuun muotoon. Eli olemassa olevat 3 pilaria ovat pääpalkin välissä juuri. Eli käännä pääpalkkeja 90 astetta

SOFigpt

Based on the new visual information provided, it appears that the orientation of the primary beams in relation to the pillars is different than initially described. The primary

SOFigpt



primary beams in relation to the pillars is different than initially described. The primary beams run perpendicular to what was originally scripted, with the pillars located between the spans of the primary beams. I will adjust the SOFIMSHC script to reflect this updated configuration.

Here's the modified SOFIMSHC script considering the new orientation:

```
plaintext
+PROG SOFIMSHC urs:3
HEAD Define Geometry of Structure
SYST 3D GDIV 10000
UNIT 5

! Define base points for the pillars
SPT 1 x 0 Y #pillarDistToEnds Z -0 FIX FF
SPT 2 x 0 Y #primaryBeamLen/2 Z -0 FIX FF
SPT 3 x 0 Y #primaryBeamLen-#pillarDistToEnds Z -0 FIX FF

! Define top points for the pillars
SPT 4 x 0 Y #pillarDistToEnds Z -#pillarHeight
SPT 5 x 0 Y #primaryBeamLen/2 Z -#pillarHeight
SPT 6 x 0 Y #primaryBeamLen-#pillarDistToEnds Z -#pillarHeight

! Define pillars
SLN 1 NPA 1 NPE 4 SNO 1 GRP 1
SLN 2 NPA 2 NPE 5 SNO 1 GRP 2
SLN 3 NPA 3 NPE 6 SNO 1 GRP 3
```

SOFigpt

```

SLN 3 NPA 3 NPE 6 SNO 1 GRP 3

! Define primary beams on top of the pillars
SPT 7 X #secBeamDist Y #pillarDistToEnds Z -#pillarHeight FIX FF
SPT 8 X #secBeamDist Y #primaryBeamLen/2 Z -#pillarHeight FIX FF
SPT 9 X #secBeamDist Y #primaryBeamLen-#pillarDistToEnds Z -#pillarHeight

SLN 4 NPA 4 NPE 7 SNO 1 GRP 4
SLN 5 NPA 5 NPE 8 SNO 1 GRP 5
SLN 6 NPA 6 NPE 9 SNO 1 GRP 6

! Define secondary beams between the tops of the pillars
SLN 7 NPA 4 NPE 6 SNO 1 GRP 7
SLN 8 NPA 7 NPE 9 SNO 1 GRP 8

END

```

In this modified script:

- The pillars are now placed between the spans of the primary beams.
- The primary beams are oriented such that they run across the width of the structure.
- The secondary beams connect the tops of the pillars along the length of the structure.
- The Z-coordinates are negative to account for the downward direction of the Z-axis.
- The "FIX FF" command is applied to the top of the pillars and ends of the primary beams to restrain vertical translation.

SOFigpt

Please confirm if these changes accurately reflect the intended orientation of your structure, or if any further modifications are required.

You
Nyt meni taas väärin, on olemassa yhteensä 6 pilari. 3 per pääpalkki, ja on olemassa 2 pääpalkkia ja 3 toissijaistapalkkia

SOFigpt
I understand the configuration now; there are two primary beams running in one direction, each supported by three pillars, and three secondary beams connecting these sets of pillars. Here is the revised SOFIMSHC script for this configuration:

```

plaintext
+PROG SOFIMSHC urs:3
HEAD Define Geometry of Structure
SYST 3D GDIV 10000
UNIT 5

! Define base points for the pillars of the first primary beam
SPT 1 X 0 Y #pillarDistToEnds Z -0 FIX FF
SPT 2 X 0 Y #primaryBeamLen/2 Z -0 FIX FF
SPT 3 X 0 Y #primaryBeamLen-#pillarDistToEnds Z -0 FIX FF

! Define base points for the pillars of the second primary beam
SPT 4 X #secBeamDist Y #pillarDistToEnds Z -0 FIX FF
SPT 5 X #secBeamDist Y #primaryBeamLen/2 Z -0 FIX FF

```

SOFigpt

```

SPT 4 X #secBeamDist Y #pillarDistToEnds Z -0 FIX FF
SPT 5 X #secBeamDist Y #primaryBeamLen/2 Z -0 FIX FF
SPT 6 X #secBeamDist Y #primaryBeamLen-#pillarDistToEnds Z -0 FIX FF

! Define top points for the pillars of both primary beams
SPT 7 X 0 Y #pillarDistToEnds Z -#pillarHeight
SPT 8 X 0 Y #primaryBeamLen/2 Z -#pillarHeight
SPT 9 X 0 Y #primaryBeamLen-#pillarDistToEnds Z -#pillarHeight

SPT 10 X #secBeamDist Y #pillarDistToEnds Z -#pillarHeight
SPT 11 X #secBeamDist Y #primaryBeamLen/2 Z -#pillarHeight
SPT 12 X #secBeamDist Y #primaryBeamLen-#pillarDistToEnds Z -#pillarHeight

! Define pillars
SLN 1 NPA 1 NPE 7 SNO 1 GRP 1
SLN 2 NPA 2 NPE 8 SNO 1 GRP 2
SLN 3 NPA 3 NPE 9 SNO 1 GRP 3
SLN 4 NPA 4 NPE 10 SNO 1 GRP 4
SLN 5 NPA 5 NPE 11 SNO 1 GRP 5
SLN 6 NPA 6 NPE 12 SNO 1 GRP 6

! Define primary beams on top of the pillars
SLN 7 NPA 7 NPE 10 SNO 1 GRP 7
SLN 8 NPA 8 NPE 11 SNO 1 GRP 8
SLN 9 NPA 9 NPE 12 SNO 1 GRP 9

! Define secondary beams connecting the tops of the pillars across the
SLN 10 NPA 7 NPE 8 SNO 1 GRP 10
SLN 11 NPA 8 NPE 9 SNO 1 GRP 11
SLN 12 NPA 10 NPE 11 SNO 1 GRP 12

```

SOFigpt

```

SLN 11 NPA 8 NPE 9 SNO 1 GRP 11
SLN 12 NPA 10 NPE 11 SNO 1 GRP 12
SLN 13 NPA 11 NPE 12 SNO 1 GRP 13

END

```

This script now includes:

- * Six pillars in total, three for each primary beam.
- * Two primary beams, each running along the Y-axis and supported by the three pillars.
- * Three secondary beams, running along the X-axis and connecting the pillars supporting the primary beams.

The pillars have a "FIX FF" constraint at their bases, which restricts vertical translation, and all other points are free to translate, as they are not specified with any fixity. Please review this configuration, and let me know if it meets the requirements or if there are further adjustments needed.

You

Loistavaa, hieman tulee vielä muokata koodia, pääpalkki menee 2.5m yli pilarin x-koordinaatin. Ja on olemassa vain kaksi pääpalkkia eikä kolmea, ja sekundäärpalkkeja on 3 kpl yhteensä, ne menevät vieräkkäin olevien pilareiden pisteistä pisteihin

SOFigpt

Understood. I will adjust the script accordingly. The primary beams will extend 2.5

SOFigpt

Understood. I will adjust the script accordingly. The primary beams will extend 2.5 meters beyond the outermost pillars in the X-direction, and there are only two primary beams with three secondary beams connecting adjacent pillars. Here's the revised SOFIMSHC script:

```

plaintext
+PROG SOFIMSHC urs:3
HEAD Define Geometry of Structure
SYST 3D GDIV 10000
UNIT 5

! Define base points for the pillars along the first primary beam
SPT 1 x 0 Y #pillarDistToEnds Z -0 FIX FF
SPT 2 x 0 Y #primaryBeamLen/2 Z -0 FIX FF
SPT 3 x 0 Y #primaryBeamLen-#pillarDistToEnds Z -0 FIX FF

! Define base points for the pillars along the second primary beam
SPT 4 X #distSecBeams Y #pillarDistToEnds Z -0 FIX FF
SPT 5 X #distSecBeams Y #primaryBeamLen/2 Z -0 FIX FF
SPT 6 X #distSecBeams Y #primaryBeamLen-#pillarDistToEnds Z -0 FIX FF

! Define top points for the pillars along both primary beams
SPT 7 x 0 Y #pillarDistToEnds Z -#pillarHeight
SPT 8 x 0 Y #primaryBeamLen/2 Z -#pillarHeight
SPT 9 x 0 Y #primaryBeamLen-#pillarDistToEnds Z -#pillarHeight

```

SOFigpt

```

SPT 9 x 0 Y #primaryBeamLen-#pillarDistToEnds Z -#pillarHeight

SPT 10 X #distSecBeams Y #pillarDistToEnds Z -#pillarHeight
SPT 11 X #distSecBeams Y #primaryBeamLen/2 Z -#pillarHeight
SPT 12 X #distSecBeams Y #primaryBeamLen-#pillarDistToEnds Z -#pillarHeight

! Define pillars
SLN 1 NPA 1 NPE 7 SNO 1 GRP 1
SLN 2 NPA 2 NPE 8 SNO 1 GRP 2
SLN 3 NPA 3 NPE 9 SNO 1 GRP 3
SLN 4 NPA 4 NPE 10 SNO 1 GRP 4
SLN 5 NPA 5 NPE 11 SNO 1 GRP 5
SLN 6 NPA 6 NPE 12 SNO 1 GRP 6

! Define primary beams, extending 2.5m beyond the outermost pillars in
SPT 13 X #primaryBeamLen+2.5 Y #pillarDistToEnds Z -#pillarHeight FIX F
SPT 14 X #primaryBeamLen+2.5 Y #primaryBeamLen-#pillarDistToEnds Z -#pi

SLN 7 NPA 7 NPE 13 SNO 1 GRP 7
SLN 8 NPA 9 NPE 14 SNO 1 GRP 8

! Define secondary beams connecting the tops of the adjacent pillars
SLN 9 NPA 7 NPE 10 SNO 1 GRP 9
SLN 10 NPA 8 NPE 11 SNO 1 GRP 10
SLN 11 NPA 9 NPE 12 SNO 1 GRP 11

END

```

SOFigpt

END

This script now includes:

- * Six pillar definitions with the base points fixed in the Z direction.
- * Two primary beams, each supported by three pillars.
- * The primary beams extend 2.5m beyond the last pillars' X-coordinates.
- * Three secondary beams, which connect the top points of adjacent pillars across the primary beams.

Please review this revised configuration for accuracy according to the structure you are analyzing. If you need further adjustments or have additional requirements, let me know.

You

Vielä muutamia kohtia sillä malli on vieläkin väärin, muuta toisen pääpalkin koordinaatiksi 0, -2.5, -#pillarHeight ja toinen siihen vierelle eli #pillarDistToEnds, -2.5, -#pillarHeight

SOFigpt

I apologize for the confusion earlier. Let's correct the coordinates for the primary beams to match the specifications. We will position the primary beams such that one starts at (0, -2.5, -#pillarHeight) and the other at (#pillarDistToEnds, -2.5, -#pillarHeight), with the Z-coordinates negative to reflect the downward direction.

SOFigpt

#pillarHeight), with the Z-coordinates negative to reflect the downward direction.

Here's the corrected SOFIMSHC script:

```

plantst
Copy code

+PROG SOFIMSHC urs:3
HEAD Define Geometry of Structure
SYST 3D GDIV 10000
UNIT 5

! Define base points for the pillars along the primary beams
SPT 1 X #pillarDistToEnds Y 0 Z -0 FIX FF
SPT 2 X #pillarDistToEnds Y #primaryBeamLen/2 Z -0 FIX FF
SPT 3 X #pillarDistToEnds Y #primaryBeamLen Z -0 FIX FF

SPT 4 X #pillarDistToEnds+#distSecBeams Y 0 Z -0 FIX FF
SPT 5 X #pillarDistToEnds+#distSecBeams Y #primaryBeamLen/2 Z -0 FIX FF
SPT 6 X #pillarDistToEnds+#distSecBeams Y #primaryBeamLen Z -0 FIX FF

! Define top points for the pillars along the primary beams
SPT 7 X #pillarDistToEnds Y 0 Z -#pillarHeight
SPT 8 X #pillarDistToEnds Y #primaryBeamLen/2 Z -#pillarHeight
SPT 9 X #pillarDistToEnds Y #primaryBeamLen Z -#pillarHeight

SPT 10 X #pillarDistToEnds+#distSecBeams Y 0 Z -#pillarHeight
SPT 11 X #pillarDistToEnds+#distSecBeams Y #primaryBeamLen/2 Z -#pillarHeight
SPT 12 X #pillarDistToEnds+#distSecBeams Y #primaryBeamLen Z -#pillarHeight

```

SOFigpt

```

SPT 12 X #pillarDistToEnds+#distSecBeams Y #primaryBeamLen Z -#pillarHeight

! Define pillars
SLN 1 NPA 1 NPE 7 SNO 1 GRP 1
SLN 2 NPA 2 NPE 8 SNO 1 GRP 2
SLN 3 NPA 3 NPE 9 SNO 1 GRP 3
SLN 4 NPA 4 NPE 10 SNO 1 GRP 4
SLN 5 NPA 5 NPE 11 SNO 1 GRP 5
SLN 6 NPA 6 NPE 12 SNO 1 GRP 6

! Define primary beams on top of the pillars
SLN 7 NPA 7 NPE 10 SNO 1 GRP 7
SLN 8 NPA 8 NPE 11 SNO 1 GRP 8
SLN 9 NPA 9 NPE 12 SNO 1 GRP 9

! Define secondary beams that are perpendicular to the primary beams and
SLN 10 NPA 7 NPE 1 SNO 1 GRP 10
SLN 11 NPA 8 NPE 2 SNO 1 GRP 11
SLN 12 NPA 9 NPE 3 SNO 1 GRP 12

END

```

In this script, we have:

- * Adjusted the base and top points of the pillars to reflect the new primary beam positions.
- * Defined two primary beams on top of the pillars with the correct orientation and position.

The screenshot shows a chat interface with a dark background. At the top, the user 'SOF1gpt' is identified. The chat history includes a message from 'SOF1gpt' stating that three secondary beams have been defined, perpendicular to the primary beams, and a request for confirmation. A response from 'You' follows, expressing satisfaction. Another message from 'SOF1gpt' apologizes for an oversight and provides updated endpoint coordinates for the primary beams. At the bottom, a code editor window titled 'plaintext' is open, displaying a SOFIMSHC script. The script defines geometry and sets base points for pillars along primary beams.

SOF1gpt

* Defined three secondary beams connecting the tops of the pillars, perpendicular to the primary beams.

Please confirm this script against your intended model, and if there are any more changes needed, let me know.

You

Hyvä, mutta unohdit määrittää samalla pisteet pääpalkkeille, ne ovat ne mitkä mainitsin

SOF1gpt

I apologize for the oversight. Let's revise the script to define the endpoints of the primary beams according to the new coordinates you've provided.

Here's the updated SOFIMSHC script:

```
plaintext Copy code

+PROG SOFIMSHC urs:3
HEAD Define Geometry of Structure
SYST 3D @DIV 10000
UNIT 5

! Define base points for the pillars along the primary beams
SPT 1 X #pillarDistToEnds Y 0 Z 0 FIX FF
SPT 2 X #pillarDistToEnds Y #primaryBeamLen/2 Z 0 FIX FF
SPT 3 X #pillarDistToEnds Y #primaryBeamLen Z 0 FIX FF
```

```
SPT 4 X #pillarDistToEnds*#distSecBeams Y 0 Z 0 FIX FF
SPT 5 X #pillarDistToEnds*#distSecBeams Y #primaryBeamLen/2 Z 0 FIX FF
SPT 6 X #pillarDistToEnds*#distSecBeams Y #primaryBeamLen Z 0 FIX FF

! Define top points for the pillars along the primary beams
SPT 7 X #pillarDistToEnds Y 0 Z -#pillarHeight
SPT 8 X #pillarDistToEnds Y #primaryBeamLen/2 Z -#pillarHeight
SPT 9 X #pillarDistToEnds Y #primaryBeamLen Z -#pillarHeight

SPT 10 X #pillarDistToEnds*#distSecBeams Y 0 Z -#pillarHeight
SPT 11 X #pillarDistToEnds*#distSecBeams Y #primaryBeamLen/2 Z -#pillar
SPT 12 X #pillarDistToEnds*#distSecBeams Y #primaryBeamLen Z -#pillarHe

! Define pillars
SLN 1 NPA 1 NPE 7 SNO 1 GRP 1
SLN 2 NPA 2 NPE 8 SNO 1 GRP 2
SLN 3 NPA 3 NPE 9 SNO 1 GRP 3
SLN 4 NPA 4 NPE 10 SNO 1 GRP 4
SLN 5 NPA 5 NPE 11 SNO 1 GRP 5
SLN 6 NPA 6 NPE 12 SNO 1 GRP 6

! Define primary beams on top of the pillars, extending 2.5m in the neg
SPT 13 X 0 Y -2.5 Z -#pillarHeight FIX FF
SPT 14 X #pillarDistToEnds Y -2.5 Z -#pillarHeight FIX FF

SLN 7 NPA 7 NPE 13 SNO 1 GRP 7
SLN 8 NPA 10 NPE 14 SNO 1 GRP 8

! Define secondary beams that are perpendicular to the primary beams an
```

SOFigpt

```
SLN 9 NPA 7 NPE 10 SNO 1 GRP 9  
SLN 10 NPA 8 NPE 11 SNO 1 GRP 10  
SLN 11 NPA 9 NPE 12 SNO 1 GRP 11  
  
END
```

In this script, the following changes have been made:

- * Support points for the primary beams have been defined at the correct locations as you specified: (0, -2.5, -#pillarHeight) and (#pillarDistToEnds, -2.5, -#pillarHeight).
- * Primary beams are now correctly extended 2.5 meters in the negative Y direction from the pillar's top points.
- * The script reflects two primary beams and three secondary beams.

Please review these changes to ensure they align with the structure you are aiming to model, and let me know if any further adjustments are required.

You

Nyt koodissasi on se vialla että x-koordinaatti on #pillarDistToEnds vaikka sen tulisi olla y-koordinaatti. Eli kun teet palkin neljää eri pistettä, tee niistä (0, -#pillarDistToEnds, -pillarHeight) (0, #primarybeamLen+#pillarDistToEnds, -#pillarHeight), (secBeamDist, -#pillarDistToEnds, #pillarHeight) ja (secBeamDist, primarybeamLen+#pillarDistToEnds, #pillarHeight)

SOFIcpt

SOFIcpt

I appreciate your patience. Let's correct the coordinates for the primary beams once more. Based on your instructions, it seems the primary beams should span along the Y-axis, starting from a negative Y-offset, going through the pillars, and extending beyond them. The secondary beams will connect these primary beams along the X-axis.

Here's the corrected SOFIMSHC script with the proper coordinates for the primary beams:

```
plaintext
+PROG SOFIMSHC urs:3
HEAD Define Geometry of Structure
SYST 3D GDIV 10000
UNIT 5

! Define primary beams endpoints
SPT 1 x 0 Y -#pillarDistToEnds Z -#pillarHeight FIX FF
SPT 2 x 0 Y #primaryBeamLen+#pillarDistToEnds Z -#pillarHeight FIX FF
SPT 3 X #secBeamDist Y -#pillarDistToEnds Z -#pillarHeight FIX FF
SPT 4 X #secBeamDist Y #primaryBeamLen+#pillarDistToEnds Z -#pillarHeight

! Define pillars on the first primary beam
SPT 5 x 0 Y 0 Z 0 FIX FF
SPT 6 x 0 Y #primaryBeamLen/2 Z 0 FIX FF
SPT 7 x 0 Y #primaryBeamLen Z 0 FIX FF
```

SOFIcpt

```
! Define pillars on the second primary beam
SPT 8 X #secBeamDist Y 0 Z 0 FIX FF
SPT 9 X #secBeamDist Y #primaryBeamLen/2 Z 0 FIX FF
SPT 10 X #secBeamDist Y #primaryBeamLen Z 0 FIX FF

! Define top points for the pillars along the first primary beam
SPT 11 x 0 Y 0 Z -#pillarHeight
SPT 12 x 0 Y #primaryBeamLen/2 Z -#pillarHeight
SPT 13 x 0 Y #primaryBeamLen Z -#pillarHeight

! Define top points for the pillars along the second primary beam
SPT 14 X #secBeamDist Y 0 Z -#pillarHeight
SPT 15 X #secBeamDist Y #primaryBeamLen/2 Z -#pillarHeight
SPT 16 X #secBeamDist Y #primaryBeamLen Z -#pillarHeight

! Define pillars
SLN 1 NPA 5 NPE 11 SNO 1 GRP 1
SLN 2 NPA 6 NPE 12 SNO 1 GRP 2
SLN 3 NPA 7 NPE 13 SNO 1 GRP 3
SLN 4 NPA 8 NPE 14 SNO 1 GRP 4
SLN 5 NPA 9 NPE 15 SNO 1 GRP 5
SLN 6 NPA 10 NPE 16 SNO 1 GRP 6

! Define primary beams between the endpoints
SLN 7 NPA 1 NPE 2 SNO 1 GRP 7
SLN 8 NPA 3 NPE 4 SNO 1 GRP 8

! Define secondary beams connecting the tops of the pillars
SLN 9 NPA 11 NPE 14 SNO 1 GRP 9
```

SOFIcpt

```
SLN 10 NPA 12 NPE 15 SNO 1 GRP 10
SLN 11 NPA 13 NPE 16 SNO 1 GRP 11

END
```

This script should now define:

- * The two primary beams' endpoints with the appropriate Y-offsets.
- * Six pillars, three for each primary beam.
- * The primary beams between the defined endpoints.
- * Three secondary beams connecting the tops of the corresponding pillars across the primary beams.

Please review the script and confirm if it matches the intended configuration for your analysis.

Liite 2. Tukimuurin mallinnuksen prosessi


1/5

SOFigpt

You
Tehdään Sofistikin Teddyn tukimuurin koodi yhdessä, oletko valmis? Annan sinulle seuraavaksi lähtötiedot, liitteessä myös AutoCAD-piirustus, jota pystyt lukemaan. Seuraa myös tekstiltä annettavia ohjeita. Älä vielä tee koodia, pyydän sen sitten erikseen

SOFigpt
Kyllä, olen valmis auttamaan sinua SOFISTIK Teddy -koodin luomisessa tukimuurille. Ole hyvä ja anna minulle lähtötiedot ja kerro, miten voin auttaa sinua AutoCAD-piirustuksen kanssa. Seuraan myös annettuja ohjeita tarkasti. Voit jatkaa, kun olet valmis.

You



Noniin loistavaa, lähtötiedot ovat siis; tukimuuri on 12,5m leveä, ja 7,5m pitkä, kuten piirustuksesta nähdään. Muista määritellä TEMPLATE osioon siis, #length, #width, #height, jotka ovat 7,5m, 12,5m, 8m järjestyksessä. Tukimuurin sivuttain maanvastaisen laatan pituus/korkeus on siis 7,5m. Molempien laattojen paksuus on 1000mm. Betonilaatu on C30. Pitkittäis laatta tulee 1m ennen maanvastaisen laatan

SOFigpt

loppumista. Joudut tekemään 8 pistettä, ja 2 laattaa jotka seuraavat tarvittavia pisteitä. Seuraa sinulle annettua Esimerkkiä6. Tee ensin PROG TEMPLATE, muista erikseen pyytämäni määritelmät sekä ota mallia Esimerkistä6, oheassa piirustus

SOFigpt
Tässä lähtötiedot pohjana käyttäen esimerkkiä 6, josta otamme mallia. Kuten pyysit, määrittelen ensin globaalit muuttujat PROG TEMPLATE -osiossa:

```

plantext
Copy code

*PROG TEMPLATE urs:1
HEAD Define Global Variables

ST0#length 7.5 $ Length of the lateral earth-retaining slab [m]
ST0#width 12.5 $ Width of the retaining wall [m]
ST0#height 8 $ Height of the retaining wall [m]
ST0#slabThick 1000 $ Thickness of the slabs [mm]
ST0#concGrade 30 $ Concrete grade C30
ST0#startLongSlab 6.5 $ Start position of the longitudinal slab from th

END

```

Tässä koodissa olemme määritelleet seuraavat muuttujat:

- * #length on maanvastaisen laatan pituus, 7,5 metriä.
- * #width on tukimuurin leveys, 12,5 metriä.
- * #height on tukimuurin korkeus, 8 metriä.
- * #slabThick on molempien laattojen paksuus, 1000 millimetriä.
- * #concGrade on betonin lujuusluokka, C30.
- * #startLongSlab on pitkittäislaatan aloituskohta maanvastaisen laatan vasemmasta

Message SOFigpt...

SOFigpt

UNIT 5
CTRL MESH 1

! Define points for vertical slab
SPT 1 X 0 Y 0 Z 0 \$ Lower left corner of the vertical slab
SPT 2 X #width Y 0 Z 0 \$ Lower right corner of the vertical slab
SPT 3 X 0 Y 0 Z -#height \$ Upper left corner of the vertical slab
SPT 4 X #width Y 0 Z -#height \$ Upper right corner of the vertical slab

! Define points for horizontal slab
SPT 5 X #startLongSlab Y 0 Z 0 \$ Lower left corner of the horizontal slab
SPT 6 X #length Y 0 Z 0 \$ Lower right corner of the horizontal slab
SPT 7 X #startLongSlab Y 0 Z #slabThick \$ Upper left corner of the horizontal slab
SPT 8 X #length Y 0 Z #slabThick \$ Upper right corner of the horizontal slab

! Define slab elements with given thickness and properties
SAR NO 1 MNO 1 GRP 1 T #slabThick[mm]
SARB TYPE 'RECT' NA 1 NE 4 BE 1

SAR NO 2 MNO 1 GRP 2 T #slabThick[mm]
SARB TYPE 'RECT' NA 5 NE 8 BE 2

END

Tässä koodissa teemme seuraavat asiat:

- * Aloitamme 3D-järjestelmän määrittelyllä ja mesh-tiheyden kontrolloinnilla.
- * Määrittelemme vertikaalisen laatan neljä kulmapistettä, joista kaksi yläkulmaa ovat korkeudella, joka on negatiivinen Z-koordinaatti vastaamaan tukimuurin korkeutta.

SOFigpt

- * Määrittelemme horisontaalisen laatan neljä kulmapistettä, jotka kaikki ovat Z-koordinaatissa 0, lukuun ottamatta laatan yläpintaa, joka on paksuuden verran ylempänä (tässä tapauksessa 1000mm).
- * Määrittelemme kaksi nelikulmaista elementtiä (SAR), jotka kuvaavat sekä vertikaalisen että horisontaalisen laatan fyysiset ominaisuudet.

Kun olet valmis, voimme edetä SOFILOAD-osioon, jossa määritellään kuormat.

You

sarb määrittely on väärin, tee kuten tässä mutta omilla pisteilläsi sar no1 grp 10 mno 1t 500[mm] CB 1e7 CT 1e7
sarb TYPE 'OUT' NA 7 NE 8
sarb TYPE 'OUT' NA 8 NE 4
sarb TYPE 'OUT' NA 4 NE 3
sarb TYPE 'OUT' NA 3 NE 7

sar no2 grp 1t mno 1t 500[mm] CB 1e7 CT 1e7
sarb TYPE 'OUT' NA 9 NE 5
sarb TYPE 'OUT' NA 5 NE 6
sarb TYPE 'OUT' NA 6 NE 10
sarb TYPE 'OUT' NA 10 NE 9

SOFigpt

SOFigpt

Ymmärrän, tässä korjattu versio SOFIMSHC-osion koodista, joka määrittelee tukimuurin geometrian ja elementit oikealla tavalla:

plaintext

+PROG SOFIMSHC urs:3
HEAD Define Geometry and Elements for the Retaining Wall

SYST 3D GDIV 10000
UNIT 5
CTRL MESH 1

! Define points for vertical slab with boundary conditions
SPT 1 X 0 Y 0 Z 0 \$ Lower left corner of the vertical slab
SPT 2 X #width Y 0 Z 0 \$ Lower right corner of the vertical slab
SPT 3 X 0 Y 0 Z -#height \$ Upper left corner of the vertical slab
SPT 4 X #width Y 0 Z -#height \$ Upper right corner of the vertical slab

! Define points for horizontal slab
SPT 5 X #startLongSlab Y 0 Z 0 \$ Lower left corner of the horizontal slab
SPT 6 X #length Y 0 Z 0 \$ Lower right corner of the horizontal slab
SPT 7 X #startLongSlab Y 0 Z #slabThick \$ Upper left corner of the horizontal slab
SPT 8 X #length Y 0 Z #slabThick \$ Upper right corner of the horizontal slab

! Define vertical slab elements with stiffness modifiers
sar no 1 grp 10 mno 1 t #slabThick[mm] CB 1e7 CT 1e7
sarb TYPE 'OUT' NA 1 NE 2
sarb TYPE 'OUT' NA 2 NE 4
sarb TYPE 'OUT' NA 4 NE 3

SOFIcpt

```
! Define horizontal slab elements with stiffness modifiers
sar no 2 grp 11 mno 1 t #slabThick[mm] CB 1e7 CT 1e7
sarb TYPE 'OUT' NA 5 NE 6
sarb TYPE 'OUT' NA 6 NE 8
sarb TYPE 'OUT' NA 8 NE 7
sarb TYPE 'OUT' NA 7 NE 5

END
```

Tässä koodissa:

- * Käytämme `'sar'`-komentoa määrittelemään elementit ja niiden ryhmät, materiaalinumeron ja paksuuden.
- * `'sarb'`-komennot määrittelevät laattojen reunoja, jossa `'TYPE 'OUT''` kertoo, että kyseessä on ulkoreuna.
- * Stiffness modifiers `'CB'` ja `'CT'` on asetettu esimerkin mukaisesti arvoihin `'1e7'`, vaikka tässä tapauksessa ne ovat esimerkin mukaisesti 500[mm], mutta olet antanut paksuuden 1000[mm], joten käytin sitä mitä aikaisemmin määriteltiin.

Voimme siirtyä SOFILOAD-osioon, kun olet valmis.

You

Hyvä, muuten oikein valitsi, SPT 5,6,7,8 pisteet ovat väärin tee näin SPT 5 (0,-1,0) SPT 6 (#width,-1,0) SPT 7 (0,#length,0) SPT 8 (#width,#length,0)

SOFIcpt

SOFIcpt

Kiitos korjauksestasi. Päivitetään SOFIMSHC-osio uusilla pisteillä:

plaintext

Copy code

```
*PROG SOFIMSHC urs:3
HEAD Define Geometry and Elements for the Retaining Wall

SYST 3D GDIV 10000
UNIT 5
CTRL MESH 1

! Define points for vertical slab with boundary conditions
SPT 1 x 0 y 0 z 0 $ Lower left corner of the vertical slab
SPT 2 x #width y 0 z 0 $ Lower right corner of the vertical slab
SPT 3 x 0 y 0 z -#height $ Upper left corner of the vertical slab
SPT 4 x #width y 0 z -#height $ Upper right corner of the vertical slab

! Define points for horizontal slab
SPT 5 x 0 y -1 z 0 $ Lower left corner of the horizontal slab
SPT 6 x #width y -1 z 0 $ Lower right corner of the horizontal slab
SPT 7 x 0 y #length z 0 $ Upper left corner of the horizontal slab
SPT 8 x #width y #length z 0 $ Upper right corner of the horizontal slab
```

SOFIcpt

```
! Define horizontal slab elements with stiffness modifiers
sar no 2 grp 11 mno 1 t #slabThick[mm] CB 1e7 CT 1e7
sarb TYPE 'OUT' NA 5 NE 6
sarb TYPE 'OUT' NA 6 NE 8
sarb TYPE 'OUT' NA 8 NE 7
sarb TYPE 'OUT' NA 7 NE 5

END
```

Tämä päivitys korjaa horisontaalisen laatan sijainnin pisteet vastaamaan annettuja mittoja, ja nyt kaikki Z-koordinaatit ovat asetettu nolnaan kuten ohjeistit. Vertikaalinen laatta on pystysuorassa asennossa ja horisontaalinen laatta on vaakasuorassa asennossa suhteessa maanpintaan.

You

Tehdään nyt vielä SOFILOAD; tässä ole tarkkana tee juuri kuten esimerkissä6 mutta #slabudl tilalla käytä 16

SOFIcpt

Tässä on SOFILOAD-osio, jossa määritellään kuormat seuraamalla Esimerkki6:n rakennetta ja käyttäen arvoa 16 kN/m² vertikaaliselle kuormalle:

SOFlgpt

planttext

Copy code

```
+PROG SOFILOAD urs:4
HEAD Define Load Cases

UNIT 5

LC 1 TYPE 'NONE' TITL 'Traffic Load for Wall'

! Applying distributed load to the vertical slab (grp 10)
QUAD grp 10 TYPE 'pzz' -16

! Applying distributed load to the horizontal slab (grp 11)
QUAD grp 11 TYPE 'pzz' -16

END
```

Tässä koodissa:

- * Määritellään kuormatyyppi 'pzz', joka on yleensä vertikaalinen kuorma laatan pintaan, ja annetaan sille arvo -16 kN/m², joka on yleinen vertikaalisen kuorman merkintätapa SOFISTIK:ssa.
- * Kuormat on kohdistettu sekä vertikaaliseen (ryhmä 10) että horisontaaliseen (ryhmä 11) laattaan.
- * Käytämme negatiivista arvoa (-16), koska kuormat ovat alaspäin suuntautuvia.

Kun olet valmis, voimme siirtyä ASE-osioon, jossa suoritetaan rakenteen analyysi.

SOFlgpt

planttext

Copy code

```
+PROG SOFILOAD urs:4
HEAD Define Load Cases

UNIT 5

LC 1 TYPE 'NONE' TITL 'Traffic Load for Wall'

! Applying linearly variable distributed load to the vertical slab (grp
QUAD grp 10 INC 10 TYPE ppy -16 Z 0 DPZ 5 ZO 0 FO 10

END
```

Tässä koodissa:

- * Käytämme 'QUAD' komentoa määrittämään lineaarisesti muuttuvan jakautuneen kuorman vertikaaliselle laatalle (ryhmä 10).
- * 'TYPE ppy' osoittaa, että kuorma on y-suunnassa, ja sen intensiteetti vähenee

You

Et käyttänyt samaa kuin esimerkissä6; QUAD grp 10 INC 10 TYPE ppy #slabudl Z 0 DPZ 5 ZO 0 FO 10

Käytä tätä muotoa mutta #dlabudl tilalle 16

SOFlgpt

Pahoitteluni väärinkäsityksestä. Tässä korjattu SOFILOAD-osio käyttäen annettua formaattia ja korvaten '#slabudl' arvolla 16:

You

Sitten ASE

SOFlgpt

ASE-ohjelmaa käytetään rakenneanalyysin suorittamiseen. Tässä ASE-osion perusmuoto, joka suorittaa lineaarisen analyysin aiemmin määritellyille kuormitustapauksille:

planttext

Copy code

```
+PROG ASE urs:5
HEAD Perform Linear Analysis of the Retaining Wall

UNIT 5

LC ALL

END
```

Tässä koodissa:

- * 'UNIT 5' vahvistaa jälleen kerran, että käytämme metrijärjestelmän yksiköitä.
- * 'LC ALL' tarkoittaa, että suoritetaan analyysi kaikille määritellyille kuormitustapauksille.

Tämä osio on suoraviivainen, koska se yksinkertaisesti käsittelee kaikki 'SOFILOAD'-osiossa määritellyt kuormat. Mikäli halutaan suorittaa ei-lineaarinen analyysi tai ottaa

Liite 3. Valmiit rakennemallit

1/2

