

KANDIDATARBETE 2026

**Tillämpning av artificiell intelligens inom mekanisk
konstruktion**

Andreas Brunzell
Ingemar Edelvik
Tim Olsson



CHALMERS

Institutionen för Mechanical Engineering
CHALMERS TEKNISKA HÖGSKOLA
Göteborg, Sverige 2026

Tillämpning av artificiell intelligens inom mekanisk konstruktion

Andreas Brunzell, Ingemar Edelvik, Tim Olsson

© Andreas Brunzell, Ingemar Edelvik, Tim Olsson, 2026

Handledare: Magnus Evertsson, Mechanical Engineering
Examinator: Gauti Asbjörnsson, Mechanical Engineering

Kandidatarbete 2026
Institutionen för Mechanical Engineering
Chalmers tekniska högskola
SE-412 96 Göteborg
Telefon: 031-772 1000

Satt med L^AT_EX
Tryckt av Chalmers Reproservice
Göteborg, Sverige 2026

Tillämpning av artificiell intelligens inom mekanisk konstruktion
Andreas Brunzell, Ingemar Edelvik, Tim Olsson
Institutionen för Mechanical Engineering
Chalmers tekniska högskola
Göteborg, Sverige 2026

Sammandrag

Den ökade tillgängligheten av stora språkmodeller och AI-baserade utvecklingsverktyg har öppnat nya möjligheter för att integrera artificiell intelligens i ingenjörsbaserade arbetsflöden. Detta arbete undersöker hur AI kan tillämpas som ett stödverktyg i en computer-aided design (CAD)-baserad konstruktionsprocess, med fokus på geometrigenerering, automatiserade sammanställningsoperationer samt dimensionering och komponentval. Studien genomfördes genom en praktisk implementation i Autodesk Fusion, där en AI-agent i utvecklingsmiljön Cursor kopplades till CAD-systemet via en Model Context Protocol-server, vilket möjliggjorde direkt interaktion mellan AI-agenten och CAD-miljön.

Resultaten visar att AI-assisterade CAD-arbetsflöden är praktiskt genomförbara och kan bidra till ökad effektivitet vid väldefinierade och återkommande konstruktionsmoment. Tillförlitligheten visade sig dock vara starkt beroende av tydligheten i användarens instruktioner, och manuell verifiering var nödvändig genom hela processen. En central iakttagelse är att verifieringstiden i många fall överstiger den tidsbesparing som automatiseringen medför, särskilt vid komplexa uppgifter. För att AI-baserade arbetsflöden ska bli konkurrenskraftiga krävs att modellernas tillförlitlighet når en nivå där användaren kan lita på resultaten utan att granska varje enskilt steg. Sammantaget bedöms AI ha betydande potential för framtidens CAD-baserade ingenjörarbete, men denna potential förutsätter fortsatta förbättringar av både modellkvalitet och kostnadseffektivitet.

Abstract

The increasing availability of large language models and AI-based development tools has opened new possibilities for integrating artificial intelligence into engineering workflows. This thesis investigates how AI can be applied as a support tool in a CAD-based design process, with focus on geometry generation, automated assembly operations, and component dimensioning. The study was conducted through a practical implementation in Autodesk Fusion, where an AI agent in the development environment Cursor was connected to the CAD system via a Model Context Protocol server, enabling direct interaction between the AI agent and the CAD environment.

The results demonstrate that AI-assisted CAD workflows are practically feasible and can contribute to increased efficiency in well-defined and recurring design tasks. However, the reliability of AI-generated results was found to be highly dependent on the clarity of user instructions, and manual verification remained necessary throughout the process. A key finding is that the time required to verify AI-generated outputs in many cases exceeds the time savings gained through automation, particularly for more complex tasks. For AI-based CAD workflows to become genuinely competitive with traditional methods, model reliability must improve to a level where users can trust generated results without reviewing each step individually. The study concludes that while AI holds significant promise for the future of CAD-based engineering, its full potential is contingent on continued improvements in both model quality and cost-effectiveness.

Innehåll

Sammandrag	3
Abstract	4
Innehållsförteckning	5
1 Inledning	7
1.1 Bakgrund	7
1.2 Problemställning	8
1.3 Syfte	8
1.4 Mål	8
1.5 Avgränsningar	9
2 Teori	10
2.1 Beskrivning av programvaran Autodesk Fusion	10
2.2 Beskrivning av programvaran Cursor	11
2.3 Beskrivning av MCP-server	11
2.4 Kullager och lagerdimensionering	12
2.4.1 Bestämning av belastningar	12
2.4.2 Ekvivalent dynamisk lagerbelastning	13
2.4.3 Livslängdsberäkning för kullager	13
2.4.4 Utökad livslängdsmodell	13
2.4.5 Generell beräkningsgång vid lagerdimensionering	14
2.4.6 Praktiskt lagerval	14
3 Metod	15
3.1 Fas 1 - Generering av geometri	15
3.1.1 Geometrigenerering genom skript	15
3.1.2 Integrering mellan MCP, Cursor och Autodesk Fusion	15
3.2 Fas 2 - Automatiserad sammanställning med AI-genererade skript	17
3.3 Fas 3 - AI-baserad dimensionering och val av kullager	17
4 Resultat	18
4.1 Resultat – Fas 1: Generering av geometri	18
4.1.1 Skriptbaserad geometrigenerering	18
4.1.2 Geometrigenerering med AI-agent i Cursor	19
4.1.2.1 Generering av skript från grunden	19
4.1.2.2 Användning av fördefinierade skript och add-ins	20
4.2 Resultat – Fas 2: Automatiserad sammanställning	21
4.3 Resultat – Fas 3: AI-baserad dimensionering	22
5 Diskussion	27
5.1 Samhälle och etik	27
5.1.1 Ansvar och beslutsfattande	27
5.1.2 Förändringar av ingenjörnsrollen	27
5.1.3 Arbetsmarknad och kompetensutveckling	28
5.1.4 Hållbarhet och resurseffektivitet	28
5.1.5 Sammanfattande reflektion	28
5.2 Tillförlitlighet och noggrannhet	29

5.2.1	Tillförlitlighet inom skriptgenerering av geometrier	29
5.2.2	Tillförlitlighet med Cursor-MCP-lösningen	30
5.2.3	Tillförlitlighet och noggrannhet i AI-baserad lagerdimensionering	31
5.3	Resursanvändning och modellvals påverkan på resultat	33
5.4	Säkerhetsaspekter och risker	34
5.5	Utvärdering av metod	35
5.6	Vidareutveckling och framtida arbete	36
6	Slutsats	38
	Referenser	40
	Appendix	40

1 Inledning

Utvecklingen inom digitala verktyg och artificiell intelligens har i allt större utsträckning påverkat hur konstruktion och produktutveckling bedrivs. Inom maskinteknik innebär detta att traditionella CAD-baserade arbetsflöden kompletteras med intelligenta system som kan stödja utformning, analys, optimering och beslutsfattande. Detta skapar nya möjligheter för effektivisering av konstruktionsprocesser, men ställer även krav på hur dessa tekniker kan integreras i praktiskt ingenjörsarbete.

1.1 Bakgrund

Användningen av artificiell intelligens (AI) och maskininlärning inom maskinteknik har ökat markant under de senaste decennierna, där flera studier påvisar hur tekniken bidrar till förbättrad designoptimering, beslutsstöd samt effektivisering av ingenjörers arbetsflöden (Puttegowda & Ballupete Nagaraju, 2025). Parallellt med denna utveckling har Computer-Aided Design (CAD) genomgått en betydande transformation, från enklare ritverktyg till avancerade system som stödjer simulering, analys och optimering (Enahoro & Ekiugbo, 2025). Utvecklingen har därmed skapat förutsättningar för mer avancerade och integrerade designprocesser, där flera moment kan utföras inom samma systemmiljö (Allen & Kouppas, 2012).

Samtidigt har AI, som tidigare främst varit begränsad till områden såsom dataanalys, prediktion och automation, successivt börjat tillämpas i mer kreativa och problemlösningsorienterade delar av ingenjörprocessen (Yüksel m. fl., 2023). Inom generativ design har detta blivit särskilt tydligt, där AI-baserade metoder används för att automatiskt generera och utvärdera designalternativ baserat på givna krav och begränsningar (Peckham m. fl., 2025). Detta möjliggör inte bara en ökad effektivitet, utan bidrar även till att bredda lösningsrymden och därigenom främja innovation inom produktutveckling (Regenwetter m. fl., 2022).

Integration av AI i CAD-miljöer har även visat potential att automatisera enkla, repetitiva och tidskrävande moment, såsom dimensionering, modelljustering och validering av konstruktionslösningar (Buga m. fl., 2025). Genom att reducera behovet av manuellt arbete i dessa steg kan ingenjörer i större utsträckning fokusera på övergripande designbeslut och komplex problemlösning. Detta är särskilt relevant i takt med att moderna produktutvecklingsprocesser blir allt mer komplexa och multidisciplinära, där krav på prestanda, hållbarhet och kostnadseffektivitet måste balanseras samtidigt (Tomiyama m. fl., 2019).

Den ökade tillgången till stora datamängder samt kraftfulla beräkningsresurser har ytterligare accelererat möjligheterna att integrera AI i designprocesser. Detta har bidragit till framväxten av intelligenta system som inte enbart fungerar som passiva verktyg, utan som aktiva stöd i beslutsfattande och designutforskning. I detta sammanhang uppstår ett växande behov av mer avancerade assistanssystem som kan kombinera automatisering med ett mer kontextmedvetet och interaktivt stöd för ingenjören (Verganti m. fl., 2020).

Sammantaget befinner sig utvecklingen idag, våren 2026, i ett skede där gränsen mellan verktyg och samarbetspartners blir allt mer diffus, vilket öppnar upp för nya sätt att arbeta med konstruktion och produktutveckling.

1.2 Problemställning

Trots den snabba utvecklingen inom artificiell intelligens och dess ökade användning inom ingenjörskärl arbete kvarstår flera utmaningar vid praktisk integration i CAD-baserade konstruktionsprocesser. Dagens CAD-verktyg erbjuder avancerade funktioner för modellering, simulering och analys, men många moment i arbetsflödet är fortfarande manuella, repetitiva och tidskrävande.

Samtidigt har AI-baserade arbetsmetoder visat stor potential att stödja och automatisera delar av designprocessen. Trots detta är dessa tekniker ofta svåra att integrera i befintliga arbetsflöden på ett sätt som är praktiskt användbart i vardaglig konstruktionsverksamhet (Mustapha, 2025). Det saknas därmed praktiskt tillämpbara arbetssätt för hur AI kan användas som stödverktyg i CAD-miljöer för att effektivisera avgränsade konstruktionsmoment.

1.3 Syfte

Huvudsyftet med detta kandidatarbete är att analysera och utvärdera hur artificiell intelligens kan användas för att automatisera och förenkla delar av CAD-arbetet vid konstruktion av enkla mekaniska system. Vidare analyseras hur AI kan tillämpas för enkla sammanställningsoperationer samt dimensionering utifrån givna laster och begränsningar. Genom detta vill arbetet bidra till en ökad förståelse hur moderna AI-baserade metoder kan integreras i en produktutvecklingsprocess och vilken praktisk nytta detta kan ge i ett ingenjörsmässigt sammanhang.

Arbetet fokuserar på AI som stödverktyg i konstruktionsprocessen snarare än som en fullständig ersättning av traditionella metoder och mänsklig expertis. Utgångspunkten är att AI kan effektivisera delmoment som är särskilt repetitiva och/eller tidskrävande, samtidigt som den övergripande tekniska bedömningen fortfarande ligger på ingenjören.

1.4 Mål

Det övergripande målet med arbetet är att utveckla och utvärdera minst en praktiskt tillämpbar lösning där AI på ett tydligt avgränsat sätt bidrar till en mer effektiv CAD-process. Detta innefattar att undersöka om AI kan minska den manuella arbetsinsatsen i olika konstruktionskedor, exempelvis vid framtagning av grundgeometri, val av standardkomponenter samt stöd vid enklare dimensioneringsuppgifter.

Lösningen ska utformas så att den kan integreras i ett typiskt arbetsflöde inom CAD-konstruktion, vilket möjliggör en bedömning av hur AI kan användas för att minska manuellt arbete i praktiken.

1.5 Avgränsningar

Kandidatarbetet avgränsas till att undersöka hur en AI-baserad agent kan användas som stöd i en CAD-miljö genom ett externt gränssnitt mot Autodesk Fusion. Arbetet fokuserar därmed på en specifik implementationsmiljö, vilket innebär att jämförelser mellan olika CAD- eller CAE-system inte genomförs och resultaten kommer ej att generaliseras till andra plattformar. Vidare baseras arbetet på användning av befintliga Large Language Models (LLMs) via en agentarkitektur, där fokus ligger på tillämpning snarare än utveckling av AI-modellerna. Träning, finjustering och optimering av modeller behandlas därför inte.

Arbetet avgränsas till de AI-baserade verktyg och integrationslösningar som var tillgängliga och praktiskt användbara inom projektets tidsram. Funktioner som tillkom sent under arbetets gång har därför inte inkluderats i studien.

Implementationen sker genom en befintlig Model Context Protocol-server (MCP), vilket innebär att agenten fungerar som ett externt "hard-shell", ovanpå Fusions gränssnitt. Arbetet omfattar därav inte djupgående integrationer mot Fusions interna Application Programming Interface (API). Fokus ligger dessutom främst på att undersöka genomförbarhet och konceptuell funktionalitet, snarare än att utveckla en fullständig färdig produkt. Ur denna synvinkel kommer därmed aspekter såsom robusthet, skalbarhet eller kommersiell implementering inte beaktas.

Slutligen begränsas arbetet till ett urval av grundläggande CAD-operationer, exempelvis geometriskapande, enklare sammanställnings-operationer samt skriptgenerering via agenten. Avancerade funktioner som simulering och optimering med flera, samt användarstudier, kvantitativa utvärderingar inom User Experience (UX), ingår ej i arbetet.

2 Teori

I denna del presenteras den teoretiska bakgrund som ligger till grund för arbetet. Inledningsvis beskrivs de programvaror och verktyg som används samt hur de fungerar. Vidare behandlas grundläggande principer inom dimensionering.

2.1 Beskrivning av programvaran Autodesk Fusion

Fusion är ett CAD-program som används för att konstruera, analysera och tillverkningsanpassa komponenter och system i en integrerad miljö. Programmet bygger på modellering, där användaren skapar tvådimensionella skisser som därefter används för att generera tredimensionella geometrier genom exempelvis extrudering eller rotation. Se figur 11 i appendix för Fusions utseende i standardvyn.

En central del av CAD-modellering är användningen av constraints, vilket kan översättas till geometriska och dimensionella begränsningar. Constraints används för att styra hur skisser och geometrier får förändras, exempelvis genom att ange att linjer ska vara parallella, vinkelräta, koncentriska eller ha ett bestämt avstånd. I figur 12 visas en enkel måttsättning på en vanlig rektangel i ett plan med specifika mått och relation till planet.

Samma princip används även vid sammanställningar, där flera komponenter kopplas samman genom relationer mellan deras geometrier. I Fusion används exempelvis joints för att definiera hur komponenter sitter ihop och hur de får röra sig i förhållande till varandra. Detta kan ses som constraints på sammanställningsnivå, där komponenters position och rörelse begränsas för att motsvara den verkliga mekaniska funktionen.

Utöver traditionell modellering, så har Fusion även möjligheten att automatisera arbetsflöden genom programmering. Detta kan göras med hjälp av så kallade skript och add-ins. Skript körs en gång med fördefinierade operationer, medan add-ins är avsedda att köras kontinuerligt i bakgrunden efter att de har aktiverats. Add-ins kan ses som funktioner som kan slås på eller av beroende på behov. Programmering i Fusion kan utföras i antingen C++ eller Python. I detta projekt har Python valts som programmeringsspråk.

Under arbetets gång, våren 2026, introducerades även Autodesk Assistant i Fusion på marknaden, vilket är en inbyggd AI-funktion för interaktion via vanlig text (Autodesk, 2026a). Eftersom funktionen blev tillgänglig sent i projektet inkluderades den inte i detta arbete.

Autodesk Fusion valdes som CAD-miljö då programmet erbjuder en kostnadsfri version för personlig och icke-kommersiell användning, vilket möjliggjorde genomförandet av arbetet utan ytterligare programkostnader. Programmet stöds även på både Windows och macOS, vilket var en viktig faktor då flera etablerade CAD-system har mer begränsat plattformstöd. Fusion bedömdes därmed vara ett praktiskt och lättillgängligt val för projektets genomförande (Autodesk, 2026b, 2026c).

2.2 Beskrivning av programvaran Cursor

Cursor är en AI-baserad kodeditor. Den är specifikt bra på att integrera AI för att assistera vid programmering. Cursor kan analysera kod, generera förslag samt automatiskt skapa eller modifiera programkod baserat på prompts givna av användaren.

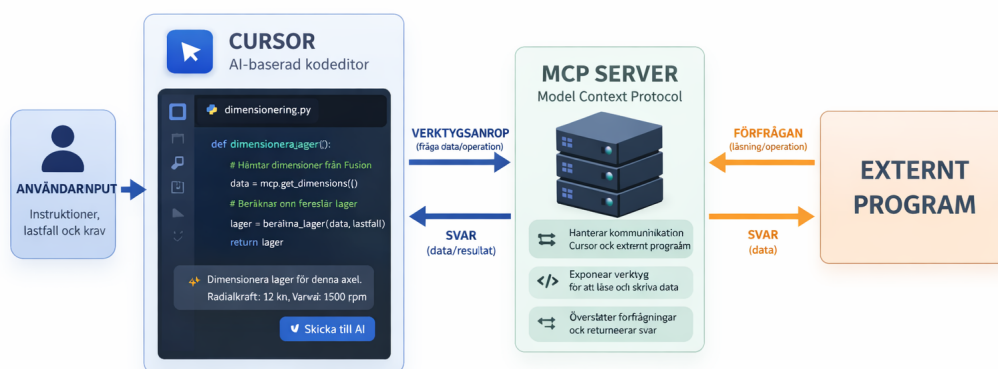
En central funktion i Cursor är möjligheten att använda en AI-agent för att analysera, generera och modifiera kod utifrån textbaserade prompts (Cursor, 2026). Detta skapar en enkel brygga mellan användare och kodskrivning utan att användaren behöver ha en bra förståelse för programmering.

2.3 Beskrivning av MCP-server

Model Context Protocol (MCP) fungerar som ett protokoll för att möjliggöra kommunikation mellan AI-modeller och externa system. Genom att implementera en MCP-server kan en AI-modell inte enbart generera text eller kod, utan även interagera med andra programvaror samt hämta och manipulera data via definierade gränssnitt (Model Context Protocol, 2026).

MCP-servern gör så att AI-modellen får tillgång till den externa programvarans information och kan göra ändringar i det externa programmet utan att någon användarinteraktion krävs. Kommunikationen sker genom speciella anrop där AI:n skickar en förfrågan och servern returnerar ett svar baserat på det aktuella tillståndet i den externa programvaran.

MCP-servern möjliggör därmed ett dynamiskt arbetsflöde där AI:n inte enbart arbetar med statisk information, utan kontinuerligt kan få uppdaterad data efter att ett kommando har getts. Se figur 1 för en förenklad förklaring av hur MCP-servern fungerar som en brygga mellan AI-agenten (i detta fall Cursor) och ett externt program.



Figur 1: Arbetsflöde mellan användarinput, Cursor, MCP-server och externt program.

2.4 Kullager och lagerdimensionering

Kullager är maskinelement som används för att möjliggöra relativ rotation mellan två komponenter med låg friktion. De används främst för att stödja roterande axlar och överföra radiella och axiella laster, samtidigt som slitage och energiförluster begränsas. Vid dimensionering av kullager är syftet att välja ett lager som med tillräcklig säkerhet klarar de belastningar och driftförhållanden som uppstår under den avsedda livslängden.

Dimensionering av kullager bygger vanligtvis på en explicit och iterativ metod. Detta innebär att ett initialt lager väljs ur standardiserade katalogserier, varefter det kontrolleras mot givna krav avseende belastning, livslängd och driftförhållanden. Om lagret inte uppfyller kraven väljs ett nytt lager och beräkningen upprepas tills en lämplig lösning erhålls.

2.4.1 Bestämning av belastningar

Det första steget i lagerdimensioneringen är att bestämma de krafter som verkar på axeln. Dessa krafter kan exempelvis uppstå från kugghjul, remdrifter, kedjor, tyngdkrafter eller externa laster. För att kunna dimensionera lagren måste axeln analyseras med hjälp av jämviktsekvationer. För statisk jämvikt gäller

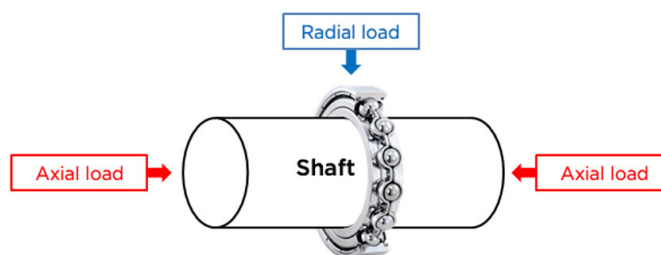
$$\sum \mathbf{F} = \mathbf{0} \quad (1)$$

och

$$\sum \mathbf{M} = \mathbf{0} \quad (2)$$

där summan av samtliga krafter och moment i respektive riktning ska vara noll. Genom att ställa upp kraft- och momentjämvikt kan lagerreaktionerna i lagringspunkterna bestämmas. Dessa reaktionskrafter utgör sedan de grundläggande belastningarna för respektive lager.

I praktiska tillämpningar delas belastningen ofta upp i en radiell komponent F_r och en axiell komponent F_a . Den radiella lasten verkar vinkelrätt mot axelns centrumlinje, medan den axiella lasten verkar längs axelns riktning. Se figur 2 nedan för visuell förklaring på radiell och axial last.



Figur 2: Vy av laster på en axel

2.4.2 Ekvivalent dynamisk lagerbelastning

Eftersom kullager ofta utsätts för både radiella och axiella laster används begreppet ekvivalent dynamisk lagerbelastning, P , för att uttrycka den sammansatta belastningen som en jämförbar storhet (Mägi m. fl., 2017). För rullningslager skrivs denna generellt som

$$P = XF_r + YF_a \quad (3)$$

där X och Y är faktorer som beror på lagertyp, belastningsförhållande och relationen mellan axiell och radiell last. Dessa faktorer hämtas normalt från tillverkarens katalogdata. För vissa lastfall, där den axiella lasten är mycket liten eller exakt noll, kan den ekvivalenta dynamiska lagerbelastningen i praktiken approximeras som

$$P \approx F_r \quad (4)$$

Detta gäller dock endast under specifika förutsättningar.

2.4.3 Livslängdsberäkning för kullager

Den grundläggande livslängdsberäkningen för kullager baseras på den nominella livslängden L_{10} . Denna definieras som den livslängd som 90 % av en grupp identiska lager statistiskt förväntas uppnå eller överstiga innan utmattningsskador uppstår. Livslängden uttrycks ofta i miljoner varv och beräknas enligt

$$L_{10} = \left(\frac{C}{P}\right)^p \quad (5)$$

där

C = dynamisk bärförmåga (specifikt för varje lager)

P = ekvivalent dynamisk lagerbelastning

p = livslängdsexponent

För kullager gäller vanligen

$$p = 3 \quad (6)$$

medan rullager har exponenten

$$p = \frac{10}{3} \quad (7)$$

Om livslängden istället önskas i drifttimmar används sambandet

$$L_{10h} = \frac{10^6}{60n} \left(\frac{C}{P}\right)^p \quad (8)$$

där n är varvtalet i revolutions per minute (rpm). Denna form är ofta mer praktisk i konstruktionssammanhang, eftersom krav vanligtvis anges i antal drifttimmar.

2.4.4 Utökad livslängdsmodell

Utöver den grundläggande nominella livslängden finns även korrigerade livslängdsmodeller som syftar till att bättre beskriva verkliga driftförhållanden. En sådan modell används av SKF och kan skrivas som

$$L_{10,skf} = a_1 a_{SKF} \left(\frac{C}{P}\right)^p, \quad \text{där } a_{SKF} \in [0.1, 50] \quad (9)$$

där a_1 är en tillförlitlighetsfaktor och a_{SKF} är en korrigeringsfaktor som används i SKF:s livslängdsmodell (SKF, 2019). Faktorn a_{SKF} tar hänsyn till bland annat smörjförhållanden, föroreningsnivå och materialegenskaper, och används för att ge en mer realistisk uppskattning av lagrets livslängd än den ideala L_{10} -modellen. Det bör dock poängteras att a_{SKF} inte utgör ett generellt uttryck för all lagerdimensionering, utan är specifik för SKF:s beräkningsmetodik.

2.4.5 Generell beräkningsgång vid lagerdimensionering

En generell lösningsgång för dimensionering av kullager kan sammanfattas enligt följande steg:

1. Bestäm yttre laster och moment som verkar på axeln.
2. Ställ upp jämviktsekvationer och beräkna lagerreaktionerna i respektive lagringspunkt.
3. Dela upp belastningarna i radiella och axiella komponenter för varje lager.
4. Välj ett preliminärt lager ur en standardiserad lagerkatalog, exempelvis baserat på axeldiameter eller tillgängligt inbyggnadsutrymme.
5. Beräkna den ekvivalenta dynamiska lagerbelastningen med hjälp av

$$P = XF_r + YF_a$$

6. Beräkna lagrets nominella livslängd enligt

$$L_{10} = \left(\frac{C}{P}\right)^p \quad \text{eller} \quad L_{10h} = \frac{10^6}{60n} \left(\frac{C}{P}\right)^p$$

7. Jämför beräknad livslängd med ställda krav på drifttid eller tillförlitlighet.
8. Kontrollera vid behov statisk hållfasthet, smörjning, förorening, temperatur och monteringsförhållanden. Eventuellt beräkna $L_{10,skf}$
9. Iterera valet av lager om kraven inte uppfylls, exempelvis genom att välja ett lager med högre dynamisk bärförmåga eller annan lagertyp.

Denna metod är explicit eftersom ingen sluten direkt lösning normalt används för att bestämma det optimala lagret. I stället bygger dimensioneringen på att konstruktören väljer ett initialt standardlager, utför kontroller och därefter stegvis justerar valet tills samtliga krav uppfylls.

2.4.6 Praktiskt lagerval

I praktiken sker lagerval genom en iterativ jämförelse mellan beräknade krav, önskad livslängd och katalogdata från tillverkare. Ett första lager väljs utifrån exempelvis axeldiameter, lastfall, varvtal och lagertyp. Därefter beräknas lagrets livslängd, exempelvis som nominell livslängd L_{10} eller modifierad livslängd enligt SKF:s metod $L_{10,skf}$. Den beräknade livslängden jämförs sedan med det krav på livslängd som ställs för tillämpningen.

Om den beräknade livslängden inte uppfyller kravet behöver ett annat lager väljas, exempelvis med högre dynamisk bärförmåga eller annan dimension. Relevanta parametrar vid detta val är bland annat lagrets invändiga och utvändiga dimensioner, dynamisk bärförmåga, maximalt tillåtna varvtal samt lagrets förmåga att ta upp radiella respektive axiella laster. Eftersom kullager generellt är standardiserade komponenter begränsas valet normalt till befintliga dimensioner och serier, vilket gör att dimensioneringsprocessen ofta kräver flera iterationer innan ett lämpligt lager kan väljas.

3 Metod

I detta kapitel beskrivs den metod som användes för att genomföra arbetet. Metoden var uppdelad i flera faser. Fokuset låg på att automatisera delar av konstruktionsprocessen genom generering av geometri samt vidare analys och dimensionering av systemet.

3.1 Fas 1 - Generering av geometri

Den första fasen i arbetet fokuserade på att generera geometrier med AI. Arbetet omfattade uppbyggnaden av parametriska modeller i Fusion där geometrin skapades genom olika former av AI arbetsflöden.

3.1.1 Geomtrigenerering genom skript

För att undersöka hur AI kunde användas som stöd vid CAD-konstruktion användes AI för att generera skript som automatiskt skapade geometrier i CAD-miljön. Med skript avses programmeringskod som utförde operationer i CAD-programmet för att skapa geometrier. I stället för att manuellt utveckla denna kod användes AI för att producera de skript som krävdes för att generera parametriska modeller.

Arbetsprocessen bestod av en iterativ interaktion mellan utvecklare och AI. Inledningsvis fokuserade processen på att generera den grundläggande geometrin. Därefter utökades skriptens funktionalitet för att ge användaren större möjlighet att påverka genereringen. Detta gjordes genom att införa användarstyrda inmatningar kopplade till exempelvis antal objekt och specifika parametervärden.

De genererade skripten testades därefter i CAD-miljön för att verifiera att geometrin skapades korrekt. I de fall där skripten inte fungerade som avsett gjordes justeringar i prompten till AI-modellen genom att förtydliga instruktionerna eller inkludera felmeddelanden från CAD-miljön i den efterföljande prompten. Denna process upprepades tills ett fungerande skript erhöles.

Metoden kan därmed beskrivas som en iterativ utvecklingsprocess där AI användes som verktyg för kodgenerering. Genom denna metod kunde skript tas fram utan traditionell manuell programmering eller djupgående förkunskaper inom Fusions Python-API.

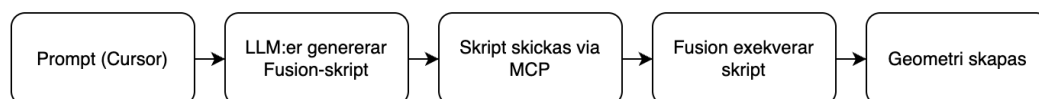
3.1.2 Integrering mellan MCP, Cursor och Autodesk Fusion

För att möjliggöra kommunikation mellan utvecklingsmiljön Cursor och CAD-miljön i Autodesk Fusion användes en befintlig MCP-baserad integration utvecklad av Christopher Drake, implementationen fanns tillgänglig via ett offentligt GitHub-repository (Drake, 2026). Lösningen fungerade som en server som gjorde funktioner i Fusion tillgängliga genom Model Context Protocol (MCP), vilket gjorde det möjligt för externa verktyg och AI-agenter att kommunicera med CAD-systemet genom strukturerade anrop.

Integrationen installerades via Autodesk App Store och konfigurerades i enlighet med den dokumentation som tillhandahölls av utvecklaren (Drake, 2026). Att källkoden var publikt tillgänglig möjliggjorde en översiktlig granskning av hur servern exponerade funktioner i Fusion samt hur kommunikationen mellan klient och CAD-miljö var uppbyggd.

Efter installation startades MCP-servern som en lokal tjänst tillsammans med Fusion. Servern tillhandahöll då ett gränssnitt som kunde anropas från externa utvecklingsmiljöer. I detta arbete användes detta gränssnitt för att etablera en koppling mellan Cursor och Fusion, vilket gjorde det möjligt att överföra och exekvera skript direkt i CAD-miljön utan manuell filhantering.

Integrationen utgjorde därmed den tekniska grund som krävdes för att genomföra experiment med AI-genererade skript för modellering i Fusion. MCP-servern fungerade som ett mellanlager mellan AI-agenten och CAD-programmet och möjliggjorde ett arbetsflöde där kod genererades i utvecklingsmiljön, överfördes via MCP och därefter exekverades direkt i CAD-miljön. Arbetsflödet för denna process illustreras i Figur 3.



Figur 3: Översikt av arbetsflödet.

I arbetet användes AI-funktionalitet i utvecklingsmiljön Cursor, där modellvalet i huvudsak hanterades automatiskt genom inställningen "Auto". Detta innebär att olika LLM:er kan ha använts beroende på uppgift och kontext, utan att en specifik modell explicit valdes. Utöver detta testades även en specifik modell (Anthropics Opus 4.6) i syfte att undersöka hur modellvalet påverkade den genererade kodens kvalitet.

3.2 Fas 2 - Automatiserad sammanställning med AI-genererade skript

I denna fas undersöktes hur AI kunde användas för att generera skript för sammanställning i Autodesk Fusion. Arbetet genomfördes genom att låta AI ta fram skript som satte villkor (constraints) mellan geometrier i CAD-miljön.

Inledningsvis utvecklades ett skript där två kuber sammanfogades genom att en vald väggyta på den ena kuben kopplades till en motsvarande väggyta på den andra kuben. Syftet var att skapa en fast koppling mellan objekten baserat på deras geometri.

Därefter utvecklades ett ytterligare skript där en cylinder placerades i ett motsvarande hål i en annan komponent. I detta fall användes geometriska referenser för att säkerställa att cylindern och hålet var korrekt positionerade och orienterade i förhållande till varandra.

De genererade skripten testades i CAD-miljön och justerades iterativt genom att modifiera instruktionerna till AI tills önskat resultat uppnåddes.

3.3 Fas 3 - AI-baserad dimensionering och val av kullager

I denna fas undersöktes hur AI kunde användas för att dimensionera en befintlig geometri baserat på givna lastfall. Arbetet genomfördes med hjälp av integrationen mellan Cursor och Autodesk Fusion via MCP, vilket möjliggjorde direkt interaktion med CAD-modellen. En befintlig geometri i Fusion användes som utgångspunkt, där relevanta mått och geometriska egenskaper hämtades automatiskt via MCP-integrationen och användes som indata i beräkningarna.

Lastfall definierades i form av krafter och moment som applicerades på modellen. Dessa inkluderades tillsammans med geometrisk information i prompten till AI-modellen, vilket möjliggjorde att dimensioneringsberäkningar kunde utföras baserat på de givna förutsättningarna. Resultaten från dessa beräkningar användes därefter som underlag för att bestämma lämpliga dimensioner samt för val av komponenter i konstruktionen.

För att möjliggöra ett automatiserat komponentval utvecklades två alternativa angreppssätt. I det första skapades ett lokalt bibliotek med kullager från SKF, innehållande relevanta tekniska data samt tillhörande Standard Triangle Language (STL)-filer. Detta möjliggjorde ett snabbt och kontrollerat val av komponenter som direkt kunde importeras i CAD-modellen.

Som alternativ implementerades en mer dynamisk metod där AI-agenten i Cursor gavs tillgång till både lokala filer och internetbaserade resurser. Genom denna åtkomst kunde agenten själv identifiera lämpliga kullager, ladda ner motsvarande STL-filer och importera dessa direkt i modellen. Detta eliminerade behovet av ett fördefinierat bibliotek och möjliggjorde istället en successiv uppbyggnad av komponentdata över tid.

Baserat på de beräknade lastfallen användes AI-agenten för att välja ett lämpligt kullager genom att antingen matcha mot det lokala biblioteket eller hämta en komponent från externa källor. Den valda komponenten importerades därefter till CAD-miljön som en STL-fil och integrerades i modellen. Processen genomfördes iterativt, där indata såsom lastfall, parametrar och promptformuleringar justerades vid behov för att säkerställa ett rimligt och tekniskt motiverat komponentval.

4 Resultat

I detta kapitel redovisas resultaten från de genomförda experimenten. Resultaten presenteras uppdelat efter de olika faserna i arbetet och omfattar geometrigeniering, automatiserad sammanställning samt AI-baserad dimensionering.

4.1 Resultat – Fas 1: Generering av geometri

I denna fas utvärderades hur AI kunde användas för att generera geometri i CAD-miljön. Resultaten omfattar både skriptbaserad geometrigeniering samt användning av en AI-agent i Cursor i kombination med MCP-integrationen.

De två tillvägagångssätten testades separat för att undersöka hur geometri kunde skapas, modifieras och parameteriseras med hjälp av AI-stöd. Resultaten från respektive metod presenteras i följande avsnitt.

4.1.1 Skriptbaserad geometrigeniering

Den skriptbaserade geometrigenieringen resulterade i att AI kunde generera fungerande skript för att skapa enklare geometrier i CAD-miljön. Skripten kunde användas för att generera grundläggande former samt parametriskt styrda modeller baserade på definierade indata.

Vid initial generering innehöll vissa skript fel, exempelvis relaterade till referenser i CAD-miljön eller felaktigt definierade operationer. Genom att iterativt justera prompten till AI-modellen och inkludera felmeddelanden från CAD-systemet kunde dessa problem åtgärdas, vilket resulterade i fungerande skript.

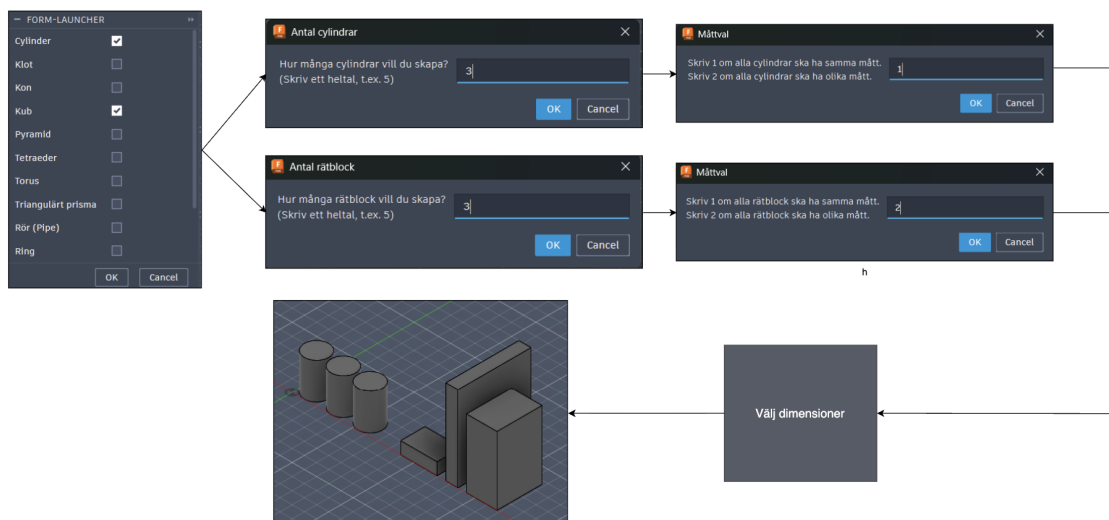
Skripten utvecklades successivt till att inkludera parametrar som kunde styras av användaren, vilket möjliggjorde variation av geometrier utan att ändra i själva koden. Detta gjorde det möjligt att generera flera varianter av en geometri baserat på samma skript.

Ytterligare funktionalitet implementerades, såsom att genererade geometrier automatiskt placerades utan att överlappa varandra vid samtidig generering av flera objekt. Vidare infördes möjligheten för användaren att välja om geometrier av samma typ skulle ha identiska dimensioner eller individuellt varierande mått.

Resultaten visade att skriptbaserad generering kunde användas för att skapa reproducerbara och parametriska modeller, men att processen i flera fall krävde flera iterationer innan ett korrekt resultat uppnåddes.

Genereringen var begränsad till enklare geometrier såsom kuber, pyramider, cylindrar och sfärer, vilket innebar en begränsning i metodens tillämpbarhet för mer komplexa konstruktioner.

För att underlätta användningen utvecklades ett gränssnitt i form av en meny där användaren kunde välja mellan de olika geometrierna. Det var möjligt att välja flera geometrier samtidigt samt ange antal och dimensioner för respektive objekt. Se figur 4 för en översiktsbild på hur användarflödet ser ut med denna lösning.



Figur 4: Användarflöde för skriptbaserad generering av geometri.

4.1.2 Geometrigenerering med AI-agent i Cursor

Geometrigenerering med AI-agenten i Cursor kunde genomföras på två huvudsakliga sätt: genom generering av skript från grunden eller genom användning av fördefinierade skript och add-ins. Dessa två angreppssätt resulterade i olika typer av funktionalitet och arbetsflöden, vilka presenteras separat nedan.

4.1.2.1 Generering av skript från grunden

Vid generering av skript från grunden användes AI-agenten för att tolka textbaserade promptar och därefter skapa skript anpassade för Fusions API. De genererade skripten överfördes via MCP-kopplingen och exekverades direkt i CAD-miljön för att skapa den efterfrågade geometrin. Arbetsflödet för denna process visualiseras i figur 3.

Detta arbetsflöde möjliggörs av att MCP-servern tillhandahåller funktioner i Fusion via dess API, vilket gör att AI-agenten i Cursor kan generera, modifiera och exekvera skript direkt i CAD-miljön utan manuell hantering.

Resultaten visade att AI-agenten kunde generera fungerande skript för enklare geometrier, såsom kuber, cylindrar och andra grundläggande former. I de fall ett genererat skript innehöll fel, exempelvis relaterade till referenser i CAD-miljön eller felaktigt definierade operationer, återfördes felmeddelandet automatiskt från Fusion till AI-agenten. Detta möjliggjorde att skriptet kunde revideras och exekveras på nytt inom samma arbetsflöde, se figur 3, tills ett fungerande resultat uppnåddes.

Det observerades även att kvaliteten på de genererade skripten varierade beroende på hur specifik prompten var. Mer detaljerade och strukturerade promptar resulterade generellt i mer korrekta skript, medan mer generella instruktioner oftare ledde till att flera iterationer krävdes innan önskat resultat uppnåddes.

De skript som genererades kunde inkludera parametrar för att styra modellens dimensioner samt namn på komponenter och andra relevanta egenskaper. Detta innebar att modellen kunde anpassas redan vid genereringstillfället. Om resultatet trots detta inte motsvarade den avsedda geometrin kunde användaren formulera en ny prompt, varefter AI-agenten justerade skriptet utifrån den uppdaterade instruktionen.

I genomförda tester observerades att generering av skript från grunden i flera fall krävde upprepade iterationer innan ett korrekt skript kunde exekveras, vilket medförde en ökad resursanvändning mätt via tokenförbrukning i Cursors användningsstatistik. Utöver användning av automatiskt modellval i Cursor testades även Anthropic Opus 4.6. Det observerades att denna modell i större utsträckning genererade korrekta skript samt resulterade i mer korrekt definierad geometri jämfört med standardinställningen.

4.1.2.2 Användning av fördefinierade skript och add-ins

Vid användning av fördefinierade skript och add-ins användes AI-agenten för att styra exekveringen av redan implementerade funktioner för geometrigeniering, vilka hade utvecklats i en tidigare fas av arbetet, se sektion 4.1.1. Till skillnad från generering av skript från grunden innebar detta att AI-agenten inte behövde skapa ny kod, utan istället anropa ett befintligt add-in som i sin tur exekverade de underliggande skripten.

Genom promptar i Cursor kunde användaren specificera vilken typ av geometri som skulle skapas samt ange relevanta parametrar. Dessa parametrar tolkades av AI-agenten och skickades vidare via MCP-kopplingen till add-inet. Add-inet exekverades därefter, där de fördefinierade skripten kördes i bakgrunden med de angivna parametrarna, vilket resulterade i att den önskade geometrin genererades i CAD-miljön.

Vid detta angreppssätt användes fördefinierade skript som redan var verifierade i CAD-miljön. Detta innebar att AI-agenten inte behövde generera ny kod, utan istället konfigurerade och initierade exekveringen av befintlig funktionalitet.

Genom detta upplägg kunde användaren styra genereringen av geometri genom att ange parametrar såsom dimensioner, antal objekt samt variation mellan dessa direkt i prompten. Flera geometrier kunde även genereras i samma arbetsflöde, där objekten automatiskt placerades i modellen utan att överlappa varandra.

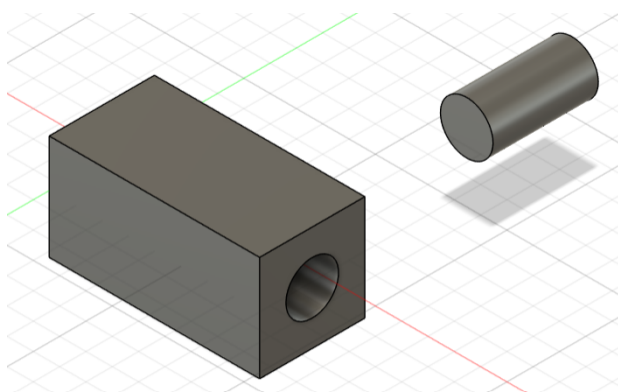
Vid användning av fördefinierade skript observerades att genereringen i de flesta fall kunde genomföras utan behov av iteration, då skripten redan var anpassade för den aktuella CAD-miljön. Vid generering av enklare geometrier registrerades även en ungefär halverad resursanvändning jämfört med generering av skript från grunden. Processen var dock begränsad till den funktionalitet som implementerats i dessa skript, vilket innebar att nya eller mer komplexa geometrier krävde vidareutveckling av den befintliga koden.

4.2 Resultat – Fas 2: Automatiserad sammanställning

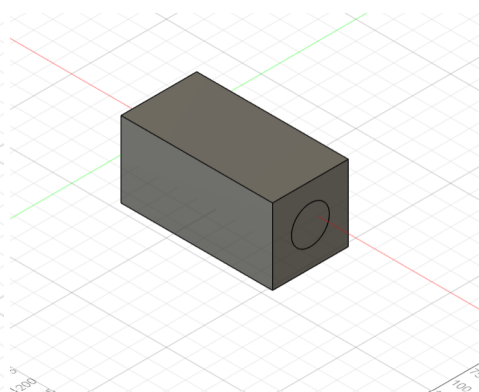
I denna fas utvärderades hur AI kunde användas för att generera skript för att sammanfoga geometrier i CAD-miljön. Resultaten baseras på tester där olika typer av geometrier placerades samman genom AI-genererade skript.

Resultaten visade att det var möjligt att skapa fungerande skript för att sammanfoga två kuber genom att koppla motsvarande ytor till varandra. De genererade kopplingarna resulterade i att objekten placerades i korrekt relation till varandra baserat på deras geometri.

Vidare kunde skripten genereras för att placera en cylinder i ett motsvarande hål i en annan komponent. I detta fall identifierades cylindriska ytor och centrumaxlar hos både cylindern och hålet för att möjliggöra automatisk sammanställning. Cylindern roterades och förflyttades därefter så att dess centrumaxel linjerades med hålets centrumaxel, samtidigt som komponentens position justerades axiellt relativt hålets öppning. Detta resulterade i korrekt positionering och orientering mellan komponenterna, se figur 5 och 6.



Figur 5: Cylinder och hålkomponent före skriptgenererad sammanfogning



Figur 6: Cylinder och hålkomponent efter skriptgenererad sammanfogning

I likhet med tidigare fas krävdes i vissa fall flera iterationer för att uppnå fungerande skript. Fel som uppstod var främst kopplade till val av referenser samt hur dessa definierades i relation till geometrin. Genom att justera instruktionerna till AI-modellen kunde dessa problem åtgärdas.

Resultaten visade att AI kunde användas för att generera skript som möjliggjorde automatiserad sammanfogning av enklare geometrier i CAD-miljön.

4.3 Resultat – Fas 3: AI-baserad dimensionering

Resultatet visade att AI:n kunde kombinera CAD-generering med en grundläggande lagerdimensionering baserad på ett specificerat lastfall. I detta test fick AI:n skapa en cylindrisk axel i Fusion, placera två lager på axeln och därefter dimensionera lagren utifrån radiell och axiell belastning, varvtal samt önskad livslängd. Den radiella lasten var inte placerad symmetriskt mellan lagren, vilket innebar att lastfördelningen beräknades med hjälp av statisk jämvikt enligt ekvation 1 och 2.

Prompten nedan gavs till AI:n:

"Create a cylindrical shaft in Fusion. Diameter 40 mm and length 200 mm. I want the shaft created in Fusion to have two bearings, each 10 mm from the edge on either side. The left bearing is the locating bearing and the right bearing is the floating bearing. There is a radial force of 1000 N (F_r) acting normal to the shaft surface at 70 mm from the left end of the shaft. There is also an axial force $F_a = 500$ N acting along the shaft at the same position. Make a decision if it should be a ball bearing or not. Tell me exactly what bearing I need for this to work for 100000 hours at 1500 rpm. Do not assume equal load sharing between the bearings. Use SKF theory for calculations"

Axeln modellerades med diametern 40 mm och längden 200 mm. Två lager placerades 10 mm från respektive axelände. Det vänstra lagret definierades som lokalisering lager och det högra som flytande lager. Positionerna definierades längs axelns längdriktning, där axelns vänstra kant sattes till $x = 0$ och axelns högra kant därmed låg vid $x = 200$ mm. Lagerpositionerna avser lagrens centrumpositioner längs denna x -axel. Detta gav lagerpositionerna

$$x_1 = 10 \text{ mm}$$

$$x_2 = 190 \text{ mm}$$

och därmed ett lageravstånd på

$$L = 190 - 10 = 180 \text{ mm}$$

Lastfallet bestod av en radiell kraft

$$F_r = 1000 \text{ N}$$

placerad vid

$$x = 70 \text{ mm}$$

från vänster axelände. Avståndet från det vänstra lagret till lasten blev

$$a = 70 - 10 = 60 \text{ mm}$$

och avståndet från lasten till det högra lagret blev

$$b = 190 - 70 = 120 \text{ mm}$$

Reaktionskraften i det vänstra lagret beräknades enligt ekvation 1 och 2 till

$$F_{r1} = F_r \frac{b}{a + b}$$

$$F_{r1} = 1000 \cdot \frac{120}{180}$$

$$F_{r1} = 666.7 \text{ N}$$

Reaktionskraften i det högra lagret beräknades enligt ekvation 1 och 2 till

$$F_{r2} = F_r \frac{a}{a+b}$$

$$F_{r2} = 1000 \cdot \frac{60}{180}$$

$$F_{r2} = 333.3 \text{ N}$$

Den axiella kraften var

$$F_a = 500 \text{ N}$$

Eftersom det vänstra lagret definierades som lokalisering antogs detta lager ta upp den axiella lasten. Det högra lagret behandlades som ett flytande lager och antogs därför inte ta upp någon axiell last vid dimensioneringen. Lastfallet för respektive lager blev därmed

$$F_{a1} = 500 \text{ N}$$

$$F_{a2} = 0 \text{ N}$$

AI:n valde ett enradigt spårkullager och föreslog SKF 6208. Lagret har dimensionerna

$$d = 40 \text{ mm}$$

$$D = 80 \text{ mm}$$

$$B = 18 \text{ mm}$$

För SKF 6208 användes den dynamiska bärförmågan

$$C = 32.5 \text{ kN}$$

och den statiska bärförmågan

$$C_0 = 19.0 \text{ kN}$$

För det vänstra, lokalisering lagret beräknades förhållandet mellan axiell last och statisk bärförmåga som

$$\frac{F_a}{C_0} = \frac{0.5}{19.0}$$

$$\frac{F_a}{C_0} = 0.0263$$

Värdena X , Y och e hämtades från SKF:s tabell för enradiga spårkullager. Dessa faktorer beror på förhållandet F_a/C_0 samt på om förhållandet F_a/F_r är större eller mindre än gränsvärdet e . Förhållandet $F_a/C_0 = 0.0263$ låg mellan tabellvärdena $F_a/C_0 = 0.025$ och $F_a/C_0 = 0.04$. Genom interpolation mellan dessa tabellrader erhöles

$$e \approx 0.222$$

och

$$Y \approx 1.98$$

Förhållandet mellan axiell och radiell last för det vänstra lagret blev

$$\frac{F_a}{F_r} = \frac{500}{666.7}$$

$$\frac{F_a}{F_r} = 0.75$$

Eftersom

$$\frac{F_a}{F_r} > e$$

användes den ekvivalenta dynamiska lagerbelastningen enligt ekvation 3

$$P = XF_r + YF_a$$

För detta lastfall gav SKF-tabellen

$$X = 0.56$$

Med $X = 0.56$ och $Y \approx 1.98$ blev den ekvivalenta dynamiska belastningen för det vänstra lagret

$$P_1 = 0.56F_{r1} + YF_{a1}$$

$$P_1 = 0.56 \cdot 666.7 + 1.98 \cdot 500$$

$$P_1 \approx 1365 \text{ N}$$

För det högra, flytande lagret fanns ingen axiell last i dimensioneringen. Den ekvivalenta dynamiska belastningen blev därför enligt ekvation 4

$$P_2 = F_{r2}$$

$$P_2 = 333.3 \text{ N}$$

För kullager användes livslängdsexponenten enligt ekvation 6

$$p = 3$$

Den nominella livslängden beräknades enligt ekvation 8 med

$$L_{10h} = \frac{10^6}{60n} \left(\frac{C}{P} \right)^3$$

där varvtalet var

$$n = 1500 \text{ rpm}$$

För att uppnå 100000 timmar krävs enligt ekvation 8

$$\left(\frac{C}{P}\right)^3 = \frac{100000 \cdot 60 \cdot 1500}{10^6}$$

$$\left(\frac{C}{P}\right)^3 = 9000$$

$$\frac{C}{P} = 9000^{1/3}$$

$$\frac{C}{P} \approx 20.8$$

Den erforderliga dynamiska bärförmågan för det vänstra lagret blev

$$C_{\text{krav},1} = 20.8 \cdot P_1$$

$$C_{\text{krav},1} = 20.8 \cdot 1365$$

$$C_{\text{krav},1} \approx 28400 \text{ N}$$

$$C_{\text{krav},1} \approx 28.4 \text{ kN}$$

För SKF 6208 och det vänstra lagret beräknades den nominella livslängden enligt ekvation 8 till

$$L_{10h,1} = \frac{10^6}{60 \cdot 1500} \left(\frac{32500}{1365}\right)^3$$

$$L_{10h,1} \approx 150000 \text{ h}$$

AI:ns slutliga svar var att använda två SKF 6208 enradiga spårkullager, där det vänstra lagret är lokaliserande och det högra lagret är flytande. AI:n beräknade den erforderliga dynamiska bärförmågan för det vänstra lagret, men beräknade inte C_{krav} separat för det högra lagret. I svaret antogs därmed samma lagertyp på båda sidor. För att kunna jämföra detta i diskussionen genomfördes en kompletterande handberäkning av det högra lagrets erforderliga dynamiska bärförmåga.

För det högra, flytande lagret fanns ingen axiell last i dimensioneringen. Därför blev

$$F_{a2} = 0 \text{ N}$$

Den radiella lasten på det högra lagret var

$$F_{r2} = 333.3 \text{ N}$$

Eftersom ingen axiell last verkar på det högra lagret blir den ekvivalenta dynamiska lagerbelastningen enligt ekvation 4

$$P_2 = F_{r2}$$

$$P_2 = 333.3 \text{ N}$$

För att uppnå livslängdskravet på 100000 timmar vid 1500 rpm användes samma livslängdssamband som för det vänstra lagret enligt ekvation 8:

$$L_{10h} = \frac{10^6}{60n} \left(\frac{C}{P} \right)^3$$

Kravet på livslängd ger därför enligt ekvation 8

$$\left(\frac{C}{P} \right)^3 = \frac{L_{10h} \cdot 60n}{10^6}$$

$$\left(\frac{C}{P} \right)^3 = \frac{100000 \cdot 60 \cdot 1500}{10^6}$$

$$\left(\frac{C}{P} \right)^3 = 9000$$

$$\frac{C}{P} = 9000^{1/3}$$

$$\frac{C}{P} \approx 20.8$$

Den erforderliga dynamiska bärförmågan för det högra lagret blev därmed

$$C_{\text{krav},2} = 20.8 \cdot P_2$$

$$C_{\text{krav},2} = 20.8 \cdot 333.3$$

$$C_{\text{krav},2} \approx 6933 \text{ N}$$

$$C_{\text{krav},2} \approx 6.9 \text{ kN}$$

För SKF 6208 och det högra lagret beräknades den nominella livslängden enligt ekvation 8 till

$$L_{10h,2} = \frac{10^6}{60 \cdot 1500} \left(\frac{32500}{333.3} \right)^3$$

$$L_{10h,2} \approx 10300000 \text{ h}$$

Dessa kompletterande värden användes som underlag för jämförelse i diskussionsdelen.

5 Diskussion

I detta kapitel diskuteras de resultat och erfarenheter som framkommit under arbetets genomförande. Fokus ligger på att analysera metodernas tillförlitlighet, noggrannhet, resursanvändning och praktiska användbarhet i en CAD-baserad konstruktionsprocess. Vidare diskuteras säkerhetsaspekter och risker kopplade till AI-agentens åtkomst till lokala filer, externa resurser och CAD-modellen. Diskussionen behandlar även metodens begränsningar samt möjliga utvecklingsområden för framtida arbete.

5.1 Samhälle och etik

Den ökade användningen av artificiell intelligens inom ingenjörarbete och CAD-baserad konstruktion innebär betydande förändringar både tekniskt och samhälleligt. I detta arbete har AI använts som ett stödverktyg för att automatisera delar av konstruktionsprocessen. Detta aktualiserar flera etiska och samhälleliga frågeställningar. Bland annat berörs frågor om ansvar och tillförlitlighet, påverkan på arbetsmarknaden, hållbarhet, säkerhet samt människans roll i en allt mer automatiserad designprocess.

5.1.1 Ansvar och beslutsfattande

En central fråga vid användning av AI i konstruktion är vem som bär ansvaret för de beslut som fattas med hjälp av tekniken. I detta arbete har AI använts för att generera skript, skapa geometrier och till att föreslå dimensioneringar.

En grundläggande etisk fråga är hur ansvar ska fördelas när AI används i tekniska beslut. I traditionellt ingenjörarbete är ansvarsfördelningen tydlig då konstruktören ansvarar för dimensionering, val av komponenter och funktionalitet. När AI används som stöd blir situationen mer komplex.

I detta arbete har AI använts för att generera skript samt att stödja dimensioneringsprocesser, exempelvis vid val av kullager baserat på givna lastfall. Även om AI kan förstå lösningar saknar den förståelse för de verkliga konsekvenserna av dessa beslut. Detta innebär att AI inte kan hållas ansvarig i juridisk eller etisk mening.

Det slutgiltiga ansvaret ligger därför fortsatt hos ingenjören, vilket ställer krav på att användaren aktivt granskar och verifierar resultaten. Ett potentiellt problem uppstår om AI blir en implicit beslutsfattare snarare än stöd, exempelvis om användaren accepterar förslag utan granskning. Detta kan leda till en urholkning av det professionella ansvaret och kan påverka säkerheten i tekniska system.

5.1.2 Förändringar av ingenjörrollen

Införandet av AI i CAD-processer leder till att ingenjören roll förändras. Arbetet har historiskt sett bestått av manuella beräkningar, datorstödda simuleringar, modellering och iterativ problemlösning. Med hjälp av AI kan nu flera av dessa moment automatiseras. Detta leder till att arbetsfördelningen förändras.

Införandet av AI leder därmed till att ingenjörer behöver agera som en granskare och beslutsfattare som utvärderar AI-genererade förslag. Detta kräver både teknisk kompetens och kritiskt tänkande. Ingenjören måste både bedöma om ett resultat är rimligt och även upptäcka potentiella fel.

Samtidigt finns det en risk att grundläggande kunskaper inom exempelvis dimensionering och mekanik försvagas om användaren i alltför hög grad förlitar sig på AI. För att motverka detta så är det viktigt att AI används som ett komplement snarare än en ersättning av traditionell ingenjörskompetens.

5.1.3 Arbetsmarknad och kompetensutveckling

Automatisering av konstruktionsprocesser kan få konsekvenser för arbetsmarknaden. Uppgifter som tidigare varit tidskrävande och manuella, såsom dimensionering och generering av geometri kan i ökande grad utföras av AI-baserade system. Detta kan leda till minskat behov av vissa typer av arbetsuppgifter.

Samtidigt skapas nya behov av kompetens, särskilt inom områden som AI-integration, systemdesign och kvalitetssäkring. Ingenjörer kan i större utsträckning behöva förstå hur AI-system fungerar, vilka begränsningar de har och hur de kan användas effektivt i praktiken.

Denna utveckling innebär att utbildningssystemet behöver anpassas för att inkludera AI-relaterade kunskaper. Förmågan att arbeta tillsammans med AI kan bli en central kompetens inom framtidens ingenjörssyrke.

5.1.4 Hållbarhet och resurseffektivitet

Ur ett hållbarhetsperspektiv innebär användning av AI i konstruktionsprocesser både möjligheter och utmaningar. Å ena sidan kan AI bidra till mer optimerade konstruktioner, exempelvis genom mer precisa dimensioneringar och effektivare materialanvändning. I detta arbete, där AI används för att stödja dimensionering och komponentval, finns en potentiell möjlighet att minska överdimensionering och därmed resursförbrukning.

Samtidigt finns en tydlig negativ aspekt kopplad till AI-systemens energiförbrukning. Stora språkmodeller kräver omfattande beräkningsresurser, både vid träning och användning. Detta medför en indirekt miljöpåverkan genom ökad energianvändning i datacenter. Även om denna aspekt inte har analyserats kvantitativt i detta arbete så är det rimligt att anta att frekvent användning av AI i konstruktionsprocesser kan leda till en ökad total resursförbrukning.

Detta skapar en avvägning mellan effektivisering och hållbarhet. De tids- och resursvinster som uppnås genom automatisering måste vägas mot den ökade energiförbrukning som tekniken medför. Ur ett ingenjörsmässigt perspektiv innebär detta att AI bör användas selektivt, där den ger tydlig nytta, snarare än som en generell ersättning för alla arbetsmoment.

5.1.5 Sammanfattande reflektion

Sammanfattningsvis erbjuder AI stora möjligheter att effektivisera och förbättra konstruktionsprocesser inom CAD. Samtidigt medför tekniken ett antal etiska och samhällsliga utmaningar likt de nämnda ovan som måste hanteras.

I detta arbete har AI använts som ett stödverktyg, vilket möjliggör en balans där tekniken bidrar till ökad effektivitet utan att ersätta mänsklig kompetens. För att säkerställa en hållbar och etiskt försvarbar användning är det avgörande att ingenjören fortsatt har en central roll i beslutsfattandet, att resultaten granskas kritiskt och att tekniken används med medvetenhet om dess begränsningar.

5.2 Tillförlitlighet och noggrannhet

Tillförlitlighet och noggrannhet är centrala aspekter vid utvärdering av de AI-baserade metoder som undersökts i detta arbete, eftersom felaktiga resultat kan få betydande konsekvenser för efterföljande arbete. Det är därför viktigt att inte enbart bedöma om en metod kan generera ett resultat, utan även hur korrekt, konsekvent och reproducerbart detta resultat är. De resultat som framkommit i arbetet visar att skriptgenerering och Cursor-MCP-lösningen skiljer sig åt i fråga om tillförlitlighet, vilket gör det nödvändigt att analysera deras respektive styrkor, begränsningar och behov av validering.

5.2.1 Tillförlitlighet inom skriptgenerering av geometrier

AI-stödd skriptgenerering för geometrier visade sig fungera väl vid skapandet av enklare modeller. När geometrin var relativt enkel och tydligt definierad, exempelvis en kub, cylinder eller sfär, kunde AI:n generera skript som skapade den önskade modellen i CAD-miljön. Däremot krävdes ett flertal iterationer av den AI-genererade koden för att uppnå ett resultat som fungerade på det sätt som avsågs.

En tydlig fördel med skriptgenerering är att ett validerat skript inte behöver granskas på nytt vid varje användning, så länge inga ändringar görs i koden eller i de underliggande förutsättningarna. Detta gör metoden tidsmässigt effektiv, eftersom användaren inte behöver gå igenom samtliga skisser och modellsteg manuellt för att kontrollera att givna parametrar överensstämmer med den genererade modellen. När ett fungerande skript väl har tagits fram blir lösningen därmed både reproducerbar och mer förutsägbar än ett svar som genereras på nytt av AI:n vid varje enskilt tillfälle.

När ett fungerande skript väl hade tagits fram och validerats minskade även risken för manuella modelleringsfel. Vid manuell modellering finns alltid en möjlighet att användaren exempelvis väljer fel skissplan, anger ett felaktigt mått eller missar någon del av modelluppbyggnaden. Genom att istället låta geometrin genereras automatiskt från ett validerat skript säkerställs att samma operationsföljd och samma parametriska samband används vid varje körning. Detta bidrar till en högre reproducerbarhet och ökar tillförlitligheten i resultatet, särskilt vid upprepad generering av liknande geometrier.

Tillförlitligheten var samtidigt tydligt beroende av geometriernas komplexitet. För enklare geometrier blev resultaten ofta korrekta efter ett mindre antal iterationer, medan mer komplexa modeller ställde högre krav på både promptformulering och efterföljande justeringar i koden. De fel som uppstod var ofta kopplade till hur geometrin byggdes upp, exempelvis val av referenser, arbetsplan eller parametriska beroenden, snarare än till den grundläggande möjligheten att generera själva formen.

Noggrannheten i den genererade geometrin var också nära kopplad till hur tydligt modellen specificerades av användaren. Ju tydligare mått, relationer och parametrar som angavs, desto större var sannolikheten att AI:n genererade ett skript som motsvarade avsedd geometri. Detta innebär att metodens noggrannhet inte enbart beror på AI:ns förmåga, utan även på kvaliteten i den information som användaren tillhandahåller.

5.2.2 Tillförlitlighet med Cursor-MCP-lösningen

Tillförlitligheten för denna lösning bedöms som lägre än för ren skriptgenerering, eftersom AI:n genererar nya och i förväg icke-validerade lösningar för varje ny prompt. Till skillnad från ett tidigare verifierat skript, som kan återanvändas med relativt hög säkerhet så länge inga ändringar görs, innebär Cursor-MCP-lösningen att varje nytt svar i grunden behöver granskas på nytt. Detta medför att användaren ställs inför en avvägning mellan hur mycket tid som ska läggas på validering av den genererade geometrin och i vilken utsträckning man väljer att lita på AI:ns förslag.

Vid geometriskapande visade lösningen att AI:n i många fall kunde förstå den övergripande avsikten och generera modeller som låg nära det önskade resultatet. Samtidigt förekom fel i exempelvis måttsättning, referenser och ordningsföljd mellan operationer, vilket innebar att resultaten inte kunde betraktas som direkt tillförlitliga utan manuell kontroll. Tillförlitligheten var därmed högre på en konceptuell nivå än på detaljnivå. En fördel jämfört med manuell modellering var dock att AI:n kunde minska mängden repetitivt arbete och därmed även reducera vissa typer av manuella fel, förutsatt att användaren verifierade resultatet innan det användes vidare.

En viktig styrka med Cursor-MCP-lösningen är att den inte enbart används för att generera nya lösningar från grunden, utan även kan exekvera tidigare sparade skript. En geometri som väl genererats och validerats kan därför sparas och återanvändas vid senare tillfällen, utan att valideringsprocessen behöver upprepas. Detta är särskilt värdefullt i konstruktionssammanhang där liknande geometrier eller operationer återkommer regelbundet, eftersom ett växande bibliotek av verifierade skript successivt minskar beroendet av att generera nya lösningar vid varje tillfälle. Även om det inte rör sig om träning av AI-modellen i strikt teknisk mening, kan användaren genom att bygga upp ett sådant bibliotek skapa förutsättningar för mer konsekventa resultat. Där den kunskap som byggs upp genom tidigare valideringar stannar kvar i arbetsflödet snarare än att gå förlorad mellan sessioner.

Valet av AI-modell utgör också en källa till osäkerhet. Vid användning av den automatiska modellinställningen i Cursor är det inte transparent vilken modell som används vid ett givet tillfälle, vilket kan leda till variationer i resultatets kvalitet. Inom ramen för detta projekt genomfördes inga tester med mer avancerade modeller, främst på grund av kostnadsbegränsningar. Samtidigt är modellvalet sannolikt av stor betydelse för resultatets kvalitet, vilket innebär att tillförlitligheten i lösningen delvis påverkas av vilka ekonomiska resurser som finns tillgängliga. En potentiell förbättring av tillförlitligheten kan därmed vara möjlig, men till priset av ökade kostnader.

När lösningen används för dimensioneringsfunktioner blir kraven på tillförlitlighet ännu högre. Till skillnad från geometriskapande, där fel ofta kan upptäckas visuellt i CAD-miljön, kan fel i dimensionering vara svårare att identifiera direkt. AI:n behöver då inte bara tolka geometrin korrekt, utan även tillämpa rätt beräkningsmetoder, antaganden och dimensioneringskriterier. Ett resultat kan därför framstå som rimligt trots att det bygger på felaktiga antaganden eller ofullständig indata. Detta gör att användaren måste inta en mer kritisk roll och verifiera både beräkningsgång och slutresultat.

Sammantaget kan Cursor-MCP-lösningen främst betraktas som ett stödverktyg snarare än en fullt tillförlitlig automatiserad lösning. För geometriskapande kan verktyget effektivisera arbetet, men resultaten kräver fortfarande utomstående kontroll från användaren. Samtidigt finns en tydlig styrka i att validerade lösningar kan sparas som skript och återanvändas, vilket gör att metodens tillförlitlighet kan öka successivt över tid. För dimensioneringsfunktioner är behovet av validering ännu större, eftersom konsekvenserna av dolda fel kan bli mer allvariga. Det slutliga ansvaret för den genererade modellen kvarstår därför hos användaren, eftersom AI-agenten saknar förmåga att självständigt verifiera korrektheten utifrån en ingenjörsmässig kontext. Resultaten visar därmed

att AI-baserad geometrigeniering och dimensioneringsstöd har stor potential, men att användningen samtidigt medför risker som måste hanteras genom kritisk granskning och validering för att säkerställa tillförlitlighet och korrekthet i praktiska ingenjörssapplikationer.

5.2.3 Tillförlitlighet och noggrannhet i AI-baserad lagerdimensionering

Resultatet visar att AI:n i detta fall kunde genomföra en sammanhängande dimensioneringsgång för kullager, som redovisas i avsnitt 4.3 men med vissa begränsningar. AI:n tog hänsyn till att den radiella lasten inte var placerad symmetriskt mellan lagren och antog därför inte lika lastfördelning. I stället beräknades reaktionskrafterna med statisk jämvikt, vilket gav en radiell belastning på 666.7 N i det vänstra lagret och 333.3 N i det högra lagret.

AI:n hanterade även skillnaden mellan lokalisering och flytande lager på ett rimligt sätt. Eftersom det vänstra lagret definierades som lokalisering antogs detta lager ta upp den axiella lasten på 500 N, medan det högra flytande lagret antogs ha $F_{a2} = 0$ N. Detta gav olika ekvivalenta dynamiska belastningar för lagren, där det vänstra lagret blev dimensionerande med $P_1 \approx 1365$ N, medan det högra lagret fick $P_2 = 333.3$ N. AI:n använde därefter livslängdssambandet för kullager och kom fram till att SKF 6208 uppfyllde kravet på 100000 timmar vid 1500 rpm.

En styrka i AI:ns beräkning var att den använde en mer korrekt lastfördelning och tog hänsyn till den axiella belastningens inverkan på det lokalisering lagret. AI:n använde även SKF:s metod för ekvivalent dynamisk belastning genom att jämföra förhållandet F_a/F_r med gränsvärdet e . Eftersom $F_a/F_r = 0.75$ var större än $e \approx 0.222$, användes uttrycket $P = XF_r + YF_a$ för det vänstra lagret. Detta gav en mer realistisk dimensionering än om axiallasten hade försumrats.

Samtidigt framkom vissa begränsningar i AI:ns svar. Prompten angav inte uttryckligen att hela beräkningsgången skulle redovisas, vilket gjorde att delar av uträkningen presenterades utan fullständig motivering. Detta behövde följas upp med kompletterande frågor om hur AI:n hade tagit fram olika värden, exempelvis faktorerna X och Y i den dynamiska belastningsekvationen (3). AI:n antog även automatiskt att samma lagertyp skulle användas på båda sidor, trots att detta inte uttryckligen angavs i prompten. Det vänstra lagret var dimensionerande eftersom det tog upp både radiell och axiell belastning, medan det högra lagret endast tog upp en mindre radiell belastning. Den kompletterande handberäkningen visade att det högra lagret endast krävde en dynamisk bärförmåga på

$$C_{\text{krav},2} \approx 6.9 \text{ kN}$$

medan SKF 6208 har

$$C = 32.5 \text{ kN}$$

Detta innebär att det högra lagret blev kraftigt överdimensionerat ur ett rent livslängds- och lastkapacitetsperspektiv. AI:ns val av samma lager på båda sidor gav därmed en enkel och symmetrisk lösning, men inte en optimerad dimensionering. Detta visar att AI:n i detta fall prioriterade ett fungerande och konsekvent lagerarrangemang framför att minimera lagerstorlek eller bärförmåga.

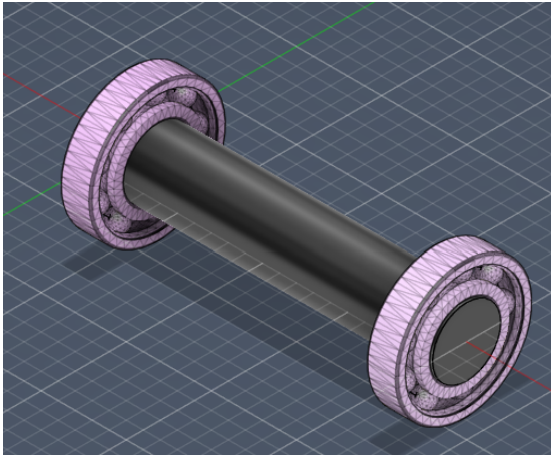
Om det högra lagret dimensioneras separat utifrån det beräknade värdet $C_{\text{krav},2}$, kan ett mindre lager väljas. Ett SKF 61908 enradigt spårkullager har en dynamisk bärförmåga på cirka

$$C = 13.8 \text{ kN}$$

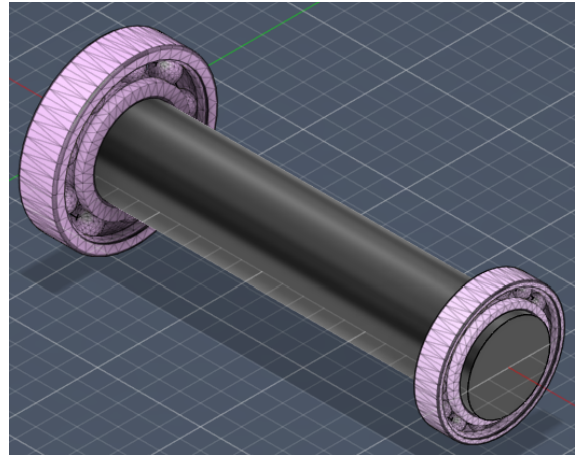
vilket överstiger det beräknade kravet

$$C_{\text{krav},2} \approx 6.9 \text{ kN}$$

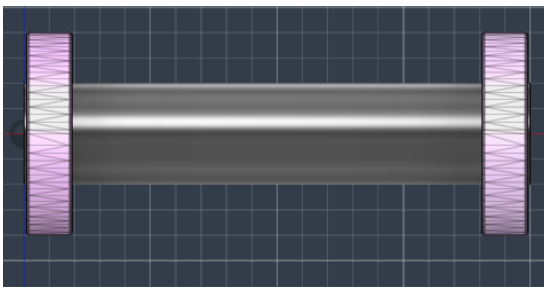
Därmed uppfyller även SKF 61908 livslängdskravet för det högra lagret. För att visualisera storlekskillnaden sammanfogades två olika lagerkombinationer i Fusion. Den första kombinationen bestod av två SKF 6208-lager, vilket visas i figur 7 och figur 9. Den andra kombinationen bestod av SKF 6208 som vänster lager och SKF 61908 som höger lager, vilket visas i figur 8 och figur 10.



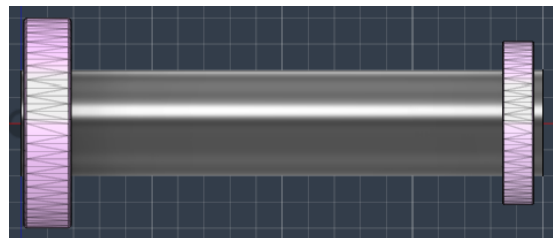
Figur 7: ISO vy med 2 stycken 6208 kullager



Figur 8: ISO vy med 6208 och 61908 kullager



Figur 9: Vy framifrån med 2 stycken 6208 kullager



Figur 10: Vy framifrån med 6208 och 61908 kullager

Sammanfogningen utgick från att lagrens centrum placerades 10 mm från respektive axelände. Detta var den tolkning som AI:n gjorde utifrån den givna prompten.

Det bör dock noteras att prompten inte uttryckligen angav att lagren skulle optimeras individuellt eller att det minsta möjliga lagret skulle väljas för respektive position. AI:ns svar kan därför ses som en rimlig tolkning av uppgiften, eftersom det ofta är praktiskt att använda samma lagerstorlek på båda sidor av en axel. Samtidigt visar resultatet att AI:n inte automatiskt undersökte alternativa lager för det mindre belastade flytlagret. För att få ett mer optimerat resultat hade prompten behövt specificera att varje lager skulle dimensioneras separat och att minsta möjliga SKF-lager som uppfyller kraven skulle identifieras.

En annan begränsning var att AI:n inte redovisade alla underliggande konstruktionsfaktorer som påverkar ett slutgiltigt lagerval. Beräkningen baserades på nominell L_{10} -livslängd och tog inte hänsyn till exempelvis smörjning, temperatur, föroreningar, passningar, tätningar eller justerad SKF-livslängd. Dessa faktorer kan ha stor betydelse i en verklig konstruktion, särskilt vid höga livslängdskrav som 100000 timmar. Resultatet bör därför betraktas som en grundläggande dimensioneringsberäkning snarare än ett fullständigt konstruktionsunderlag.

Sammanfattningsvis visar resultatet att AI:n kunde genomföra en tekniskt rimlig lagerdimensionering och identifiera det dimensionerande lagret i systemet. Beräkningen hanterade både osymmetrisk radiallyast och axiell belastning på ett relevant sätt. Samtidigt visade fallet att AI:n gjorde ett implicit antagande om att samma lager skulle användas på båda sidor, vilket ledde till att det högra lagret blev överdimensionerat. För att öka noggrannheten och få mer optimerade resultat krävs därför tydligare promptar samt manuell verifiering av AI:ns antaganden och lagerval.

En möjlig förbättring hade varit att använda en specialiserad AI-agent tränad för lagerdimensionering och andra dimensioneringsuppgifter. En sådan agent hade kunnat utformas för att tydligare identifiera vilka antaganden som krävs, dimensionera varje lagerposition separat och undvika implicita antaganden, exempelvis att samma lagertyp automatiskt ska användas på båda sidor av axeln.

En specialiserad agent hade även kunnat utvecklas för att hantera mer avancerade livslängdsberäkningar, exempelvis med justerad SKF-livslängd och faktorn a_{SKF} . I detta arbete begränsades beräkningen till nominell L_{10} -livslängd. En anledning till detta är att mer avancerade SKF-beräkningar kräver flera tabellvärden och driftdata, exempelvis smörjning, viskositet, föroreningsgrad och belastningsförhållanden. Dessa värden är svårare för en generell AI-modell att hantera på ett tillförlitligt sätt, särskilt när informationen finns i PDF-baserade handböcker eller på webbsidor som inte är strukturerade som maskinläsbara databaser.

SKF:s dokumentation och katalogdata är till stor del tillgänglig i form av handböcker, produktblad och webbsidor, vilket gör att informationen inte alltid är enkel att extrahera automatiskt. Om data i stället hade funnits i ett strukturerat format, exempelvis som Comma-Separated Values (CSV)-eller API-data, hade en AI-agent kunnat använda tabellvärden och lagerdata på ett mer konsekvent sätt. Dessutom framgick det inte fullt ut av prompten vilken noggrannhetsnivå som efterfrågades. Detta bidrog till att AI:n genomförde en grundläggande nominell livslängdsberäkning, men inte en mer detaljerad dimensionering med exempelvis a_{SKF} -faktorer. Detta utforskades dock inte vidare, eftersom det bedömdes ligga utanför arbetets ramar. Implementationen av en sådan lösning bedömdes dessutom vara relativt okomplicerad, men starkt beroende av det specifika dimensioneringsfallet och de aktuella indata som krävs.

5.3 Resursanvändning och modellvals påverkan på resultat

Resultaten visar att arbetssättet hade stor påverkan på resursanvändningen vid AI-baserad geometrigeniering. När AI-agenten genererade skript från grunden krävdes mer modellinteraktion, eftersom modellen behövde tolka prompten, skapa kod anpassad för Fusion och i vissa fall revidera skriptet efter felmeddelanden från CAD-miljön. Detta innebar att processen ofta blev mer resurskrävande, särskilt vid enklare fel som krävde flera iterationer innan ett fungerande skript kunde exekveras.

Vid användning av fördefinierade skript och add-ins minskade resursanvändningen tydligt. I detta arbetssätt behövde AI-agenten inte generera ny kod, utan endast tolka användarens prompt, konfigurera relevanta parametrar och initiera exekveringen av befintlig funktionalitet. I tester med

enklare geometrier observerades en ungefär halverad tokenanvändning jämfört med generering av skript från grunden, mätt via Cursors användningsstatistik. Det bör noteras att inga systematiska mätningar genomfördes och att observationen baseras på ett begränsat antal tester med enklare geometrier. Det är dock rimligt att anta att skillnaden skulle bli ännu mer påtaglig vid mer komplex geometri, där generering från grunden kräver betydligt mer kod och fler potentiella iterationer. Detta indikerar att ett arbetsflöde baserat på verifierade skript kan vara mer resurseffektivt när uppgiften är tydligt avgränsad.

Samtidigt innebär användning av fördefinierade skript en minskad flexibilitet. Om den önskade geometrin eller funktionen inte redan finns implementerad krävs vidareutveckling av skript eller add-ins. Generering från grunden är därför mer flexibel, men medför högre resursanvändning och större risk för iterationer. Detta visar på en avvägning mellan flexibilitet och effektivitet, där fördefinierade skript är lämpliga för återkommande och standardiserade uppgifter, medan generering från grunden lämpar sig bättre för mer varierande eller nya problem.

Modellvalet påverkade också resultatet. Vid användning av Cursors automatiska modellval kunde fungerande skript genereras, men kvaliteten varierade beroende på uppgiftens komplexitet och promptens tydlighet. I ett begränsat antal tester med Opus 4.6 gjordes den kvalitativa observationen att modellen i större utsträckning genererade korrekta skript utan behov av upprepade iterationer, jämfört med standardinställningen. Dessa observationer är inte systematiskt verifierade, men antyder att en mer avancerad modell kan generera funktionellt korrekta skript i högre utsträckning inom CAD-relaterade arbetsflöden.

Den förbättrade kvaliteten måste dock vägas mot ökad resursanvändning och kostnad. Mer avancerade modeller prissätts högre per anrop och kräver fler tokens, vilket sammantaget ökar kostnaden vid upprepade användning. Modellvalet bör därför anpassas efter uppgiftens karaktär. En enklare eller automatisk modellinställning kan vara tillräcklig för standardiserade operationer, medan mer avancerade modeller kan vara motiverade vid komplexare skriptgenerering eller när högre precision krävs.

Sammantaget visar resultaten att både arbetsflöde och modellval har betydelse för hur effektiv AI-integrationen blir i en CAD-miljö. Ett styrt arbetsflöde med fördefinierade skript minskar resursanvändningen, medan mer avancerade modeller kan förbättra kvaliteten på genererad kod. Valet mellan dessa angreppssätt innebär därmed en avvägning mellan kostnad, flexibilitet och precision, där uppgiftens karaktär bör vara avgörande.

5.4 Säkerhetsaspekter och risker

En särskild risk identifierades i samband med den metod som användes för automatiserat komponentval i Fas 3, se sektion 3.3. Genom att ge AI-agenten i Cursor tillgång till lokala filer samt internet möjliggjordes en dynamisk identifiering och nedladdning av komponenter, vilket introducerar flera säkerhetsaspekter kopplade till både systemarkitektur och användning. Denna typ av integration innebär att flera steg i processen sker automatiserat, vilket ökar effektiviteten men samtidigt minskar graden av direkt kontroll.

En central risk är kopplad till åtkomst till lokala resurser. Genom att AI-agenten ges tillgång till användarens filsystem finns en potentiell risk att känslig information exponeras eller används på ett oavsiktligt sätt. Vidare innebär nedladdning av komponenter från externa källor att ursprunget och kvaliteten på dessa filer inte nödvändigtvis är verifierad. Detta kan leda till att felaktiga eller olämpliga komponenter importeras i CAD-modellen, vilket i sin tur kan påverka den fortsatta konstruktionsprocessen.

För att minska dessa risker bör AI-agentens åtkomst begränsas och kontrolleras. Exempelvis kan filåtkomst begränsas till specifika kataloger och externa komponenter endast hämtade från verifierade källor. Vidare bör importerade filer och AI-genererade ändringar granskas manuellt innan de används i konstruktionsprocessen. En sådan kombination av tekniska begränsningar och mänsklig verifiering kan minska risken för att felaktiga eller skadliga filer integreras i CAD-miljön.

En ytterligare risk är kopplad till de beräkningar som ligger till grund för komponentvalet. Om de indata eller beräkningsmodeller som används av AI-agenten är felaktiga eller ofullständiga kan detta leda till att en olämplig komponent väljs. Eftersom valet av komponent i denna metod sker automatiserat och baserat på dessa beräkningar finns en risk att felaktiga antaganden inte upptäcks i ett tidigt skede, vilket kan påverka konstruktionens funktion och tillförlitlighet.

För att hantera dessa risker tillämpades ett arbetssätt där AI-agentens beräkningar inte användes direkt, utan först granskades av användaren. AI-agenten instruerades att redovisa beräkningssteg och antaganden, vilket möjliggjorde en manuell verifiering innan ett komponentval accepterades. Detta innebär att användaren kunde bedöma rimligheten i beräkningarna innan de låg till grund för vidare beslut i konstruktionen.

Det bör även noteras att vissa säkerhetsmekanismer redan finns implementerade i Cursor. Vid åtgärder som innebär åtkomst till lokala filer eller nedladdning av externa resurser krävs i regel ett aktivt godkännande från användaren. Detta innebär att användaren ges möjlighet att granska och bekräfta sådana operationer innan de genomförs. Samtidigt innebär detta att säkerheten i stor utsträckning blir beroende av användarens omdöme, då godkännande utan tillräcklig granskning kan medföra att de tidigare identifierade riskerna kvarstår.

Sammantaget visar detta att den ökade graden av automatisering medför potentiella säkerhets- och tillförlitlighetsrisker, men att dessa till viss del kan hanteras genom ett arbetssätt där användaren aktivt verifierar AI-agentens beräkningar och beslut innan de används i en ingenjörsmässig kontext.

5.5 Utvärdering av metod

Den metod som användes i detta arbete baserades på en iterativ och experimentell ansats där artificiell intelligens integrerades i en CAD-miljö genom en extern MCP-baserad lösning. Metoden delades upp i tre faser: geometrigenerering, automatiserad sammanställning och AI-baserad dimensionering. Denna uppdelning möjliggjorde en stegvis utvärdering av AI:s potential i olika delar av konstruktionsprocessen, vilket kan ses som en styrka då varje delmoment kunde analyseras separat.

En tydlig fördel med den valda metoden är dess praktiska genomförbarhet. Genom att använda befintliga verktyg såsom Cursor, Autodesk Fusion och en MCP-server kunde ett fungerande arbetsflöde etableras utan behov av att utveckla egna AI-modeller. Detta ligger i linje med arbetets avgränsningar, där fokus låg på tillämpning snarare än utveckling av AI-system. Metoden speglar därmed realistiska förutsättningar för hur AI kan implementeras i ingenjörsarbete idag, vilket stärker arbetets relevans.

Samtidigt medför den experimentella ansatsen vissa begränsningar. Metoden saknar en kvantitativ utvärdering av exempelvis tidsbesparing, noggrannhet eller effektivitet. Resultaten baseras istället främst på kvalitativa observationer, såsom antal iterationer som krävdes för att uppnå fungerande lösningar. Detta innebär att slutsatserna i viss mån blir beroende av subjektiva bedömningar, vilket kan påverka metodens generaliserbarhet.

Den iterativa arbetsprocessen, där AI-genererade lösningar successivt förbättrades genom justering av promptar, visade sig vara central för att uppnå fungerande resultat. Detta gäller särskilt i fas 1 och 2, se sektion 3.1 och 3.2, där flera iterationer ofta krävdes för att hantera fel relaterade till exempelvis referenser och operationsföljd. Metoden kan därför beskrivas som beroende av användarens förmåga att formulera effektiva promptar samt tolka och korrigera AI:s output. Detta innebär att metodens effektivitet inte enbart beror på AI-systemets kapacitet utan även på användarens kompetens.

En annan viktig aspekt är beroendet av den specifika implementationsmiljön. Arbetet genomfördes med Autodesk Fusion i kombination med en specifik MCP-server, vilket innebär att resultaten är starkt kopplade till denna tekniska lösning. Eftersom ingen jämförelse med andra CAD-system eller integrationsmetoder genomfördes är det svårt att avgöra hur väl resultaten kan generaliseras till andra plattformar.

Metoden visar dock tydligt att AI kan användas som ett stödverktyg i flera delar av konstruktionsprocessen. Särskilt framträdande är möjligheten att kombinera generering av nya lösningar med återanvändning av tidigare validerade skript. Detta skapar ett arbetsflöde där tillförlitligheten kan öka över tid, vilket kan ses som en styrka i metoden.

Samtidigt framgår det att metoden i sin nuvarande form inte är fullt robust. Behovet av iterationer, manuell verifiering och hantering av fel indikerar att metoden lämpar sig bättre för explorativt arbete och prototyputveckling än för direkt användning i säkerhetskritiska ingenjörsapplikationer. Detta ligger även i linje med arbetets avgränsning, där fokus varit på genomförbarhet snarare än färdig produkt.

Sammanfattningsvis kan metoden bedömas som lämplig för att uppfylla arbetets syfte, då den möjliggör en praktisk undersökning av hur AI kan integreras i CAD-baserat ingenjörsarbete. Metoden ger värdefulla insikter om både möjligheter och begränsningar, men skulle behöva kompletteras med mer systematiska och kvantitativa utvärderingar för att möjliggöra mer generella och robusta slutsatser.

5.6 Vidareutveckling och framtida arbete

Detta arbete har visat att artificiell intelligens kan användas som ett stödverktyg i flera delar av en CAD-baserad konstruktionsprocess. Genom integrationen mellan Cursor och Autodesk Fusion via en MCP-server möjliggjordes ett dynamiskt arbetsflöde där AI-agenten kunde generera, modifiera och exekvera skript direkt i CAD-miljön. Resultaten visar därmed att tekniken är praktisk genomförbar och kan bidra till att effektivisera delar av konstruktionsarbetet. Samtidigt identifierades ett antal begränsningar som pekar på möjliga utvecklingsområden.

En central utvecklingsmöjlighet är att utöka den AI-baserade geometrigenereringen till att hantera mer komplexa konstruktioner. I detta arbete var genereringen huvudsakligen begränsad till enklare geometrier vilket innebär att metodens tillämpbarhet i mer avancerade ingenjörsapplikationer är begränsad. Framtida arbete kan därför fokusera på att utveckla mer avancerade modeller eller arbetsflöden som kan hantera komplexa parametriska samband och mer realistiska komponenter.

Vidare finns potential att förbättra tillförlitligheten i den AI-baserade arbetsprocessen. Resultaten visar att flera iterationer ofta krävdes för att erhålla fungerande lösningar, särskilt vid skriptgenerering. Detta indikerar att metoden i sin nuvarande form inte är fullt robust. Ett möjligt utvecklingssteg är att integrera automatiserade valideringsmetoder, exempelvis genom att kombinera AI-genererade lösningar med analytiska beräkningar eller regelbaserade kontroller.

Detta skulle kunna minska behovet av manuell verifiering och öka tillförlitligheten i resultaten.

Trots att integrationen mellan AI-agenten och CAD-miljön fungerade i detta arbete, finns det även möjligheter att vidareutveckla den. Den nuvarande lösningen möjliggör interaktion och manipulation av modeller, men resultaten visar att precision och stabilitet kan variera beroende på uppgift. En vidareutveckling skulle kunna innebära mer robust kommunikation mellan AI och CAD-systemet, förbättrad felhantering samt tydligare koppling mellan AI-modellens output och CAD-systemets interna struktur.

Inom dimensioneringsdelen finns även möjligheter till vidareutveckling. I detta arbete behandlades främst grundläggande dimensionering av kullager baserat på givna lastfall. Framtida arbete kan inkludera mer avancerade beräkningsmodeller, fler komponenttyper samt integration av standarder och tillverkarbaserade databaser. Detta skulle öka metodens relevans och användbarhet i praktiska ingenjörssammanhang.

Ytterligare ett utvecklingsområde är att bygga upp strukturerade bibliotek av validerade skript och komponenter. Resultaten visade att tillförlitligheten ökade när tidigare verifierade lösningar kunde återanvändas. Genom att systematiskt lagra och återanvända sådana lösningar kan arbetsflödet successivt bli mer effektivt, samtidigt som variationen i resultat minskar.

En ytterligare aspekt att beakta är den snabba utvecklingen inom AI-området. De modeller och verktyg som använts i detta arbete representerar en momentanbild av teknikens nuvarande kapacitet snarare än en långsiktig begränsning i metoden. I takt med att mer avancerade språkmodeller och integrationslösningar utvecklas finns potential att flera av de begränsningar som identifierats, exempelvis behovet av iterationer och svårigheter i referenshantering kan minska. Detta innebär att framtida system kan komma att erbjuda mer tillförlitliga och automatiserade lösningar, vilket ytterligare kan förändra hur CAD-baserat ingenjörarbete bedrivs.

6 Slutsats

Detta arbete har undersökt hur artificiell intelligens kan användas som stödverktyg i en CAD-baserad konstruktionsprocess, med fokus på geometrigenerering, automatiserad sammanställning samt dimensionering och komponentval. Resultaten visar att AI-integration i CAD-miljöer är praktiskt genomförbar och att tekniken redan idag kan bidra till effektivisering av avgränsade konstruktionsmoment. Samtidigt identifierades ett antal begränsningar som i dagsläget hindrar en bredare och mer tillförlitlig tillämpning.

En genomgående iakttagelse är att AI fungerar bäst som stödverktyg när uppgiften är tydligt definierad och när verifierade lösningar kan återanvändas. Välformulerade promptar och fördefinierade skript reducerade både tokenförbrukning och antalet iterationer, medan generering från grunden gav större flexibilitet men krävde mer resurser. Inom dimensionering kunde AI genomföra tekniskt korrekta beräkningsgångar, men tenderade att välja konservativa lösningar och redovisade inte alltid sina antaganden explicit. Det slutgiltiga ansvaret för att granska och verifiera resultaten kvarstår därför hos ingenjören.

En central utmaning som framkommit i arbetet är förhållandet mellan automatisering och verifiering. I dagsläget är verifieringsprocessen i många fall mer tidskrävande än att bygga upp geometrin manuellt från grunden, vilket begränsar den praktiska nyttan särskilt vid mer komplexa uppgifter. För att AI-baserade arbetsflöden ska bli genuint konkurrenskraftiga i förhållande till traditionellt CAD-arbete krävs att modellernas tillförlitlighet når en nivå där användaren kan lita på genererade resultat utan att behöva kontrollera varje enskilt steg. Detta är i dagsläget den främsta faktorn som vi ser, som bromsar en bredare adoption.

Vi bedömer att text-till-CAD baserade arbetsflöden har stor potential att på sikt förändra hur konstruktionsarbete bedrivs. Den naturliga interaktionen mellan användare och CAD-miljö sänker tröskeln för att skapa och modifiera geometrier, och kan frigöra ingenjörstid från repetitiva moment till förmån för mer kvalificerad problemlösning. För att denna potential ska realiseras fullt ut krävs dock att både modellkvalitet och kostnad utvecklas i rätt riktning. Mer avancerade modeller visade i detta arbete på förbättrad kvalitet, men medför samtidigt högre kostnad per anrop. För praktisk tillämpning i industrin krävs att denna avvägning förskjuts, att kraftfullare modeller blir tillräckligt kostnadseffektiva för att motiveras även vid rutinemässig användning.

På längre sikt finns anledning att se optimistiskt på utvecklingen. Språkmodeller förbättras kontinuerligt och kostnaderna för avancerade modeller har historiskt sett sjunkit i takt med att tekniken mognat. I kombination med att strukturerade arbetsflöden och bibliotek av validerade skript byggs upp successivt finns goda förutsättningar för att tillförlitligheten ökar över tid. När modellerna når en punkt där de konsekvent genererar korrekta resultat utan omfattande manuell kontroll förändras förutsättningarna för CAD-baserat ingenjörarbete i grunden. Inte genom att ersätta ingenjören, utan genom att förskjuta dennes roll mot mer övergripande bedömningar och beslut.

Resultaten visar således att AI-integration i CAD-miljöer är ett område med betydande potential, men att tekniken ännu inte är mogen nog för att självständigt hantera konstruktionsuppgifter utan kontinuerlig mänsklig granskning. Den mest värdefulla tillämpningen i närtid ligger i kombinationen av verifierade skript för återkommande moment och mer avancerade AI-modeller för uppgifter som kräver flexibilitet. Ett val som alltid bör vägas mot kostnad och uppgiftens karaktär. Mer avancerade modeller visade i detta arbete på förbättrad förmåga att generera funktionellt korrekta skript med färre iterationer, men medför samtidigt högre kostnad per anrop. För praktisk tillämpning krävs därför att denna avvägning mellan modellkvalitet och kostnad förskjuts i takt med att tekniken utvecklas. Framtiden för text-till-CAD är lovande, men realiseras fullt ut först

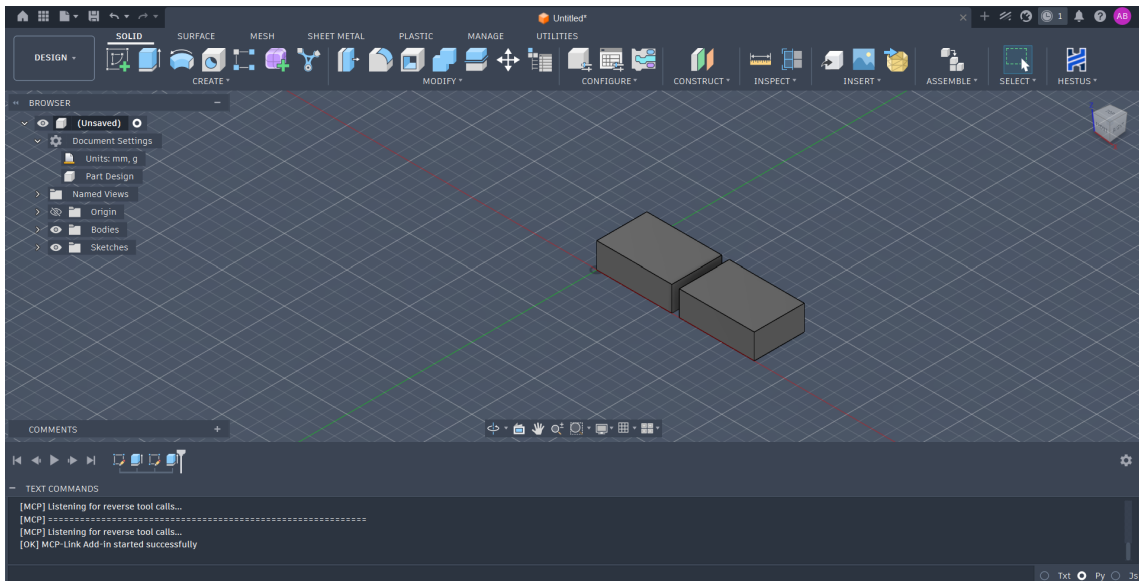
när förtroendet för AI:ns resultat kan motiveras av modellernas faktiska tillförlitlighet.

Därmed bedöms arbetets syfte och uppsatta mål ha uppnåtts. Genom de genomförda testerna har det visats hur AI kan användas för att stödja geometrigenerering, enkla sammanställningsoperationer samt dimensionering och komponentval inom CAD-baserat konstruktionsarbete. Resultaten ger därmed en konkret grund för att bedöma teknikens praktiska nytta i en produktutvecklingsprocess, samtidigt som de visar vilka begränsningar som behöver beaktas vid fortsatt användning och utveckling av AI-baserade arbetsflöden.

Referenser

- Allen, J., & Kouppas, P. (2012 januari). Computer Aided Design: Past, Present, Future. https://www.researchgate.net/publication/281455474_Computer_Aided_Design_Past_Present_Future
- Autodesk. (2026a). *Autodesk Assistant*. Autodesk. Hämtad 27 april 2026, från <https://www.autodesk.com/solutions/autodesk-ai/autodesk-assistant#tools>
- Autodesk. (2026b). *Autodesk Fusion: Frequently Asked Questions*. Autodesk. Hämtad 27 april 2026, från <https://www.autodesk.com/products/fusion-360/faq>
- Autodesk. (2026c). *Fusion for Personal Use*. Autodesk. Hämtad 27 april 2026, från <https://www.autodesk.com/products/fusion-360/personal>
- Buga, A., Borzan, M., & Trif, A. (2025). ARTIFICIAL INTELLIGENCE IN THE CAD PROCESS: MACHINE LEARNING MODELS, GENERATIVE OPTIMISATION, AND THEIR IMPACT ON DESIGN. *Academic Journal of Manufacturing Engineering*, 23, 28–43.
- Cursor. (2026). *Cursor Documentation*. Cursor. Hämtad 27 april 2026, från <https://docs.cursor.com/>
- Drake, C. (2026). Fusion 360 MCP Server. <https://github.com/AuraFriday/Fusion-360-MCP-Server>
- Enahoro, O. M., & Ekiugbo, A. E. (2025). Evolution of CAD in Mechanical Engineering. *Journal of Materials Engineering, Structures and Computation*, 4(3), 122–137. <https://doi.org/https://doi.org/10.5281/zenodo.16942884>
- Model Context Protocol. (2026). *Model Context Protocol Documentation*. Model Context Protocol. Hämtad 27 april 2026, från <https://modelcontextprotocol.io/>
- Mustapha, K. (2025). A survey of emerging applications of large language models for problems in mechanics, product design, and manufacturing. *Advanced Engineering Informatics*, 64, 103066. <https://doi.org/https://doi.org/10.1016/j.aei.2024.103066>
- Mägi, M., Melkersson, K., & Evertsson, M. (2017). *Maskinelement*. Studentlitteratur.
- Peckham, O., Raines, J., Bulsink, E., Goudswaard, M., Gopsill, J., Barton, D., Nassehi, A., & Hicks, B. (2025). Artificial Intelligence in Generative Design: A Structured Review of Trends and Opportunities in Techniques and Applications. *Designs*, 9(4). <https://doi.org/10.3390/designs9040079>
- Puttegowda, M., & Ballupete Nagaraju, S. (2025). Artificial intelligence and machine learning in mechanical engineering: Current trends and future prospects. *Engineering Applications of Artificial Intelligence*, 142, 109910. <https://doi.org/https://doi.org/10.1016/j.engappai.2024.109910>
- Regenwetter, L., Nobari, A. H., & Ahmed, F. (2022). Deep Generative Models in Engineering Design: A Review. *Journal of Mechanical Design*, 144(7), 071704. <https://doi.org/10.1115/1.4054135>
- SKF. (2019). *Rolling bearings catalogue* (tekn. rapport Nr PUB BU/P1 17000/1 SV) (November 2019). SKF Group.
- Tomiyama, T., Lutters, D., Stark, R., & Abramovici, M. (2019). Development capabilities for smart products. *CIRP Annals*, 68(2), 727–750. <https://doi.org/https://doi.org/10.1016/j.cirp.2019.05.010>
- Verganti, R., Vendraminelli, L., & Iansiti, M. (2020). Innovation and Design in the Age of Artificial Intelligence. *Journal of Product Innovation Management*, 37(3), 212–227. <https://doi.org/https://doi.org/10.1111/jpim.12523>
- Yüksel, N., Börklü, H. R., Sezer, H. K., & Canyurt, O. E. (2023). Review of Artificial Intelligence Applications in Engineering Design Perspective. *Engineering Applications of Artificial Intelligence*, 118, 105697. <https://doi.org/10.1016/j.engappai.2022.105697>

Appendix



Figur 11: Fusions standard-UI



Figur 12: Constraints i Fusion