

Hastighetsstyrsystem för Förarrobot

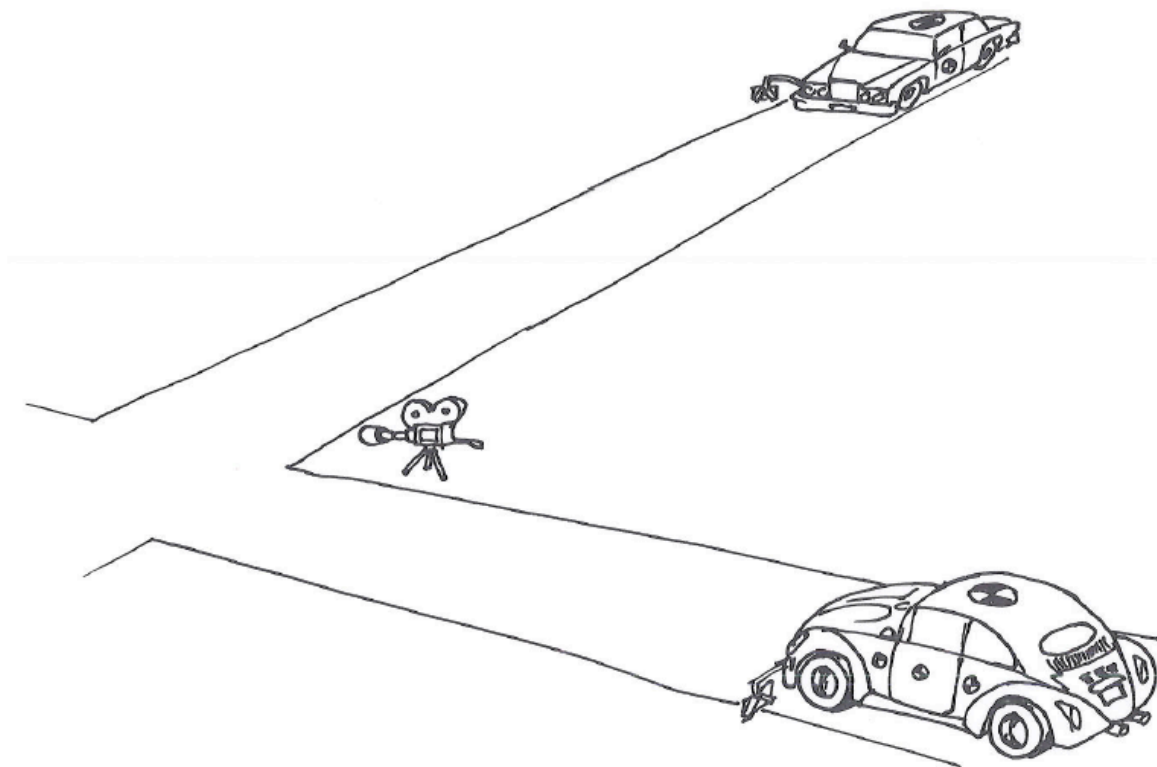
Konstruktion, Modell, Test och Utvärdering

Examensarbete utfört inom Fordonssystem och VTI

Av Daniel Axelsson

Reg nr: LiTH-ISY-EX--07/4090--SE

17 jan 2008



Sammanfattning

De krocktester som utförs idag är låsta till det system av styr och draganordningar som byggs upp. Att göra en krocktest där bilarna kan styras individuellt med hjälp av en förarrobot ökar friheten att designa tester.

Att krocka två bilar som på bilden på framsidan, med 90 graders vinkel mellan bilarnas färdriktning ställer det mest tidskritiska kravet när de ska träffa varandra. Detta kräver en bra hastighetsreglering vilket är huvuddelen i examensarbetet.

I denna rapport redovisas hur ett sådant styrsystem skulle kunna byggas samt tester i en simuleringsmodell för denna typ av krocktest.

Simuleringsmodellen är uppbyggd i Matlab/Simulink och eftersträvar att efterlikna ett verkligt sidokollisionstest tillräckligt mycket för att kunna testa styrsystemet.

Efter de tester som gjorts kan man konstatera att det är möjligt att konstruera ett sådant styrsystem som kan få två bilar att träffa varandra i en sådan sidokollision som är beskriven ovan.

Innehållsförteckning

1. Introduktion.....	5
1.1 Mål.....	5
1.2 Syfte.....	6
1.3 Förutsättning och avgränsning.....	6
1.4 Förstudie.....	6
1.5 Litteratur.....	7
2. Bakgrund.....	7
2.1 Företagspresentation.....	7
2.2 Fordonsbranschen.....	7
2.3 Metoder.....	8
2.3.1 Simuleringsmetod för att testa systemmodell.....	8
2.3.2 Reglertekniska metoder.....	8
2.3.3 Metod för att simulera ett fordon.....	8
2.3.4 Metod att mäta när ett fordon kommer att nå sin träffpunkt.....	9
2.3.5 Metoder att göra ett styrsystem.....	9
3. Modeller.....	9
3.1 Förord Modeller.....	9
3.2 Närmare inpå de olika.....	10
3.2.1 Egenstyrt system.....	10
3.2.2 Funktionsstyrt system.....	11
3.2.3 Hjälpsystem.....	13
3.2.4 Övervakningssystem.....	13
3.3 Krav på systemet.....	14
4. Simuleringsmodellen.....	14
4.1 Simuleringen i korthet.....	14
4.2 Initialfunktionen.....	15
4.3 Blocket Trottle Control.....	18
4.4 Bilmodellsblocket.....	19
4.4.1 M-filen i bilmodellsblocket.....	19
5. Teori.....	21
5.1 Inledning av teorin.....	21
5.2 MaxG kriterie.....	21
5.3 Kommunikationslänken.....	21
5.4 Avståndsmätning.....	22
5.4.1 Lasermätning.....	22
5.4.2 Pulsgivare.....	23
5.5 Vridmomentkurvan och växlingar.....	23
5.6 Andra typer av bilmodeller.....	23
5.7 Realtid vs hastighetsfunktion lagrad i array.....	24
5.8 Triggerpunkternas placering.....	24
6. Resultat.....	24
6.1 Inledning resultat.....	24
6.2 Typer av körningar och deras resultat.....	24
6.2.1 Test 1. 70 km/h vs 70 km/h.....	25
6.2.2 Test 2. 70 km/h vs 90 km/h.....	26
6.2.3 Test 3. 70 km/h vs 110 km/h.....	27
6.2.4 Test 4. 70 km/h vs 110 km/h.....	28
6.2.5 Test 5. 90 km/h vs 90 km/h.....	29

6.2.6 Test 6. 90 km/h vs 110 km/h.....	30
6.2.7 Test 7. 110 km/h vs 110 km/h.....	31
6.2.8 Test 8. 110 km/h vs 110 km/h med längre sträcka.....	32
6.3 Intressanta värden.....	32
6.4 PID val och varianter.....	33
6.4.1 Proportionell styrning.....	33
6.4.2 Proportionell och integrerande styrning.....	34
6.4.3 Proportionell, integrerande och deriverande styrning.....	35
6.5 Test av triggerpunkter.....	36
7. Slutsatser.....	37
8. Diskussion och felkällor.....	38
9. Litteraturförteckningar.....	39
10. Referenser.....	39

1. Introduktion

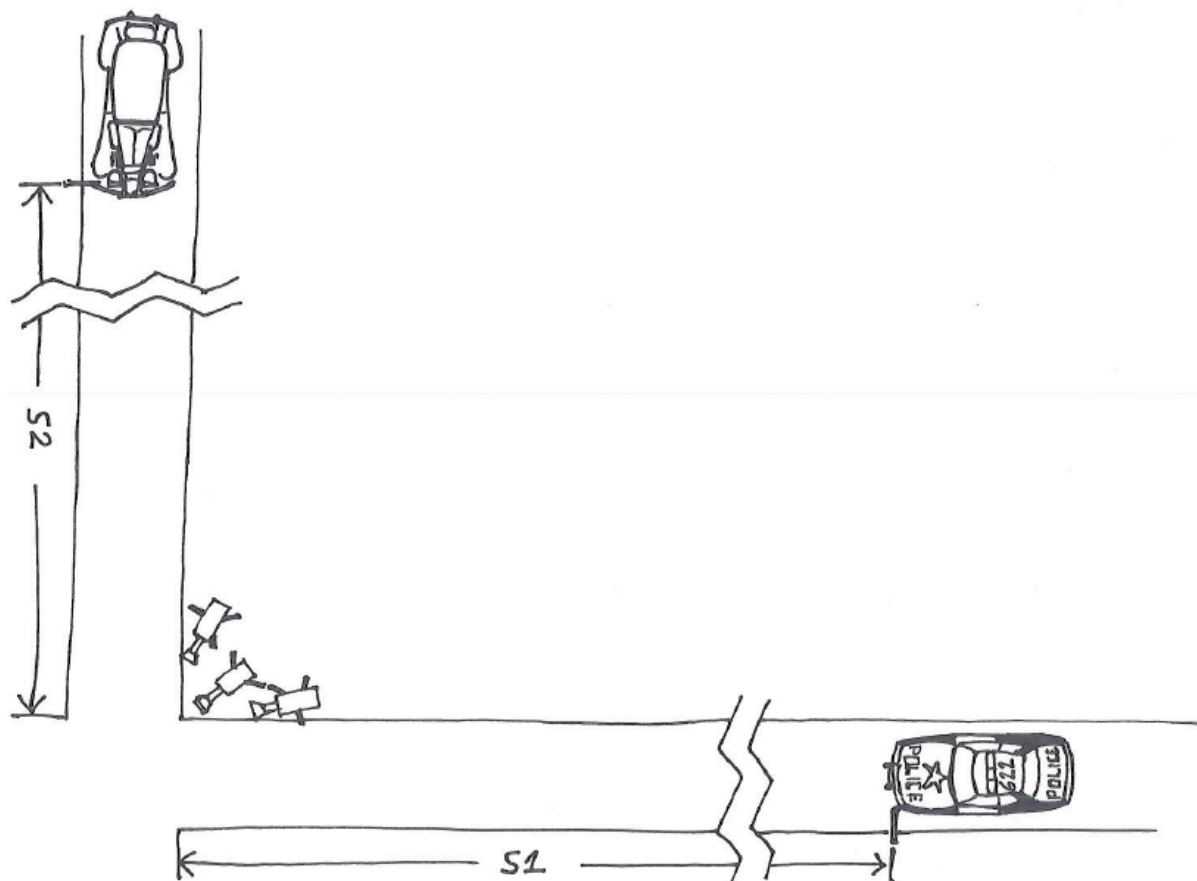
Ute på vägarna sker olyckor hela tiden. Trots fortlöpande förbättringar inom säkerhetstänkandet och konstruktionerna dör och skadas tyvärr människor fortfarande. För att kunna testa om en idé om förbättring faktiskt är en förbättring så behövs bra testutrustning. I takt med att bilbranschen gör förbättringar måste testbranschen utveckla sina testmetoder för att passa de nya förbättringarna.

För att kunna utföra kollisionstester på platser andra än en krockfacilitet vill man göra bilarna som fristående enheter som är oberoende av kringutrustning såsom vinschar, som drar fordonen till önskad hastighet, och styrskenor. Man vill ha en frihet i hur bilarna inkommer till krockpunkten.

Systemet får med denna frihet flera möjligheter däribland att testa många bilars elektroniska hjälpmedel som antisladd och antispinn.

Systemet är främst tänkt att användas för att ersätta föraren vid kollisionstester eller då det finns risk för kollision. Om man kunde styra bilens hastighet med bilens egen drivkälla, motorn, skulle det bli enklare att manövrera bilen fritt. Detta görs idag på försökstadiet med bilar som kan köra en konbana av sig själv, [Ref A]

Något som däremot inte gjorts är att få bilen att komma fram till ett mål på en bestämd sträcka och bestämd tid, gärna även med bestämd hastighet.



Figur 2.

Bilarna ankommer, i testet som examensarbetet beskriver, till kollisionspunkten i figur 2 med 90 graders vinkel mellan bilarnas färdriktning. S_1 och S_2 är de sträckor bilarna färdas från stillastående till kollisionögonblicket.

1.1 Mål

Målet med examensarbetet är att skapa ett system som styr två bilar som ska kollidera med varandra. Kollisionen (se figur 2) ska simulera korsningsolyckor med liten offset mellan bilarna. Bilarna ska kollidera med given hastighet efter given sträcka, på given tid. Bilarna är försedda med automatisk växellåda. Vinkeln mellan bilarnas färdriktning, 90 grader i kollisionspunkten, är vald då den vinkeln är den mest tidskritiska, men systemet går även att tillämpa för alla vinklar. Bilarnas drivkälla ska vara den egna motorn. Systemet ska kompensera för mätfel, segheter i systemet, samt eventuella störningar. Systemet ska vara anpassat till de krav som de olika komponenterna i systemet ställer. Till systemet ska göras en modell för att testa och utveckla systemet. Systemet har även andra användningsområden men det är ovan nämnda mål som eftersträvas i detta examensarbete.

1.2 Syfte

Syftet med examensarbetet är att tillverka en teoretisk modell för att utvärdera ett styrsystem som kan hantera två bilar som ska krocka. Samt att göra en rapportering, som underlag för praktisk tillämpning, över hur man bör gå tillväga och vad som bör betänkas inför ett sådant projekt.

Genom att, i rapporten, belysa, betrakta och studera eventuella scenarion vill rapportförfattaren ge ett ökat medvetande om denna typ av styrsystem och vilka möjligheter det innebär.

På sikt finns förhoppningen att detta styrsystem kommer att kunna vara en start till en framtida standardiseringsprocess i olika typer av testmetoder för utvärdering av nya hjälpsystem i fordon.

1.3 Förutsättning och avgränsning

Vissa begränsningar har gjorts. Dels är bilmodellerna förhållandevis enkla. Saker som berör all form av bilprestanda utom just longitudinell (se figur 3) har utelämnats då den inte anses relevant. Modellen måste beskriva fordonens prestanda tillräckligt bra för att det ska vara möjligt att studera uppgiften. Tillförlitligheten hos vald modell kan antas vara tillräcklig för utformningen av systemet.



*Figur 3.
Bilen betraktas bara i avseende att åka i longitudinell led.*

1.4 Förstudie

De första två veckorna ägnades åt att läsa på om vad som gjorts inom området tidigare, att läsa om liknande projekt. Särskild uppmärksamhet har riktats åt att utöka kunskapen om fordonsdynamik, främst longitudinell som examensarbetet enligt avgränsningarna avser. Men även krocktester i allmänhet har betraktats, då detta är en bransch som inte är så allmänt känd. Vidare har en del kunskaper i reglerteknik förnyats.

1.5 Litteratur

Att finna litteratur om tidigare gjorda arbeten var ytterst svårt då det inte funnits någon tillgänglig information om denna typ av projekt tidigare. Däremot har man gjort liknande projekt som har varit givande att läsa om och som bygger på det allmänna kunnandet inom styrsystem för fordon. För att få förståelse för krocktester i allmänhet har kunskap inhämtats dels från litteratur men även från praktisk erfarenhet under tiden på krockbanan på VTI.

2. Bakgrund

2.1 Företagspresentation

VTI, Väg och Transportforskningsinstitutet, är ett oberoende och internationellt framstående forskningsinstitut som verkar inom transportsektorn. Med omkring 190 medarbetare är VTI den största transportforskningsmiljön i Sverige. VTI utför tillämpad forsknings- och utvecklingsverksamhet som rör samtliga transportslag. Institutet har en bred kompetensprofil med kärnkompetens inom områdena säkerhet, ekonomi, miljö, trafik- och transportanalys, kollektivtrafik, beteende och samspel mellan människa-fordon-transportssystem samt inom vägkonstruktion, drift och underhåll. VTI:s samlade kunskap ger ökat beslutsunderlag hos aktörerna inom transportsektorn och får i många fall direkta tillämpningar i internationell och nationell transportpolitik.

Forskningsuppdrag utförs på uppdragsbasis i en tvärvetenskaplig organisation. VTI utför årligen forskningsuppdrag åt cirka 200 olika uppdragsgivare och kunder, där Vägverket och Vinnova är de största.

VTI:s huvudkontor finns i Linköping där också största delen av verksamheten är förlagd. Institutet har även lokalkontor i Borlänge och Stockholm, där tonvikten ligger på forskning inom transportekonomi och transportpolicy, samt i Göteborg med forskningsinriktning mot fordons- och farkostteknik.

Omkring hälften av forskarna har licentiat- eller doktorsexamen.

2.2 Fordonsbranschen

Fordonsbranschen utvecklas hela tiden. Som nämnts i inledningen behöver dessa utvecklingar kontrolleras och testas. Dels för att myndigheter har vissa krav på att fordon ska ha en viss säkerhetsnivå och inte kunna utgöra en onödig fara. Dels för att konsumenterna ska kunna se vad som är bra och dåligt på de fordon de eventuellt planerar att färdas i. Då en del av dessa tester utvecklas för att kunna bedöma bilars elektroniska stabilitetshjälpssystem och även kunna göra mera preciserade ankomster till träffpunkten i krocktester, väcktes frågan om det finns något styrsystem som kan driva bilen efter en ”exakt” körcykel trots yttre påverkan.

2.3 Metoder

2.3.1 Simuleringsmetod för att testa systemmodell

Huvudsakliga delen av arbetet har gjorts i Matlab och Simulink då detta är en utmärkt miljö att bygga upp, testa, utveckla och förbättra system med givna kriterier.

2.3.2 Reglertekniska metoder

Då systemet ska styras med hög precision behöver man en reglerteknisk del som kompenserar för fel i styrsignal och det egna systemet. De reglertekniska metoder som använts är Proportionell förstärkning av styrsignal, Integrerande förstärkning av styrsignal samt Deriverande förstärkning av styrsignal. Dessa kan behövas till att stabilisera styrsignalen till trotteln och kompensera för ojämnheter i bilarnas vridmomentskurvor, även för yttre påverkan som vind och ojämnheter i marken.

2.3.3 Metod för att simulera ett fordon

Till försöket har valts två olika bilar som försöksobjekt att modellera i simuleringen. Olika bilar för att kunna se skillnader som beror på bilarnas egenskaper. Fordonsdata har tagits från tyska tidningen ”Auto Motor & Sport” [Ref B] där allt från vridmomentskurva till utväxlingar på bilarna återfås.

Själva bilmodellen består av en kärna i form av en m-fil som simulerar på bilen inverkande krafter. Dessa krafter är motor och drivlina, växlar, varvtal och vridmoment, yttre krafter som luftmotstånd, rullmotstånd samt lutningen på underlaget. Motorn baseras på en vridmomentsmodell. Matematiskt sett är det bromsande krafter subtraherat från de drivande krafterna taget genom massan som ger bilarnas acceleration. Enligt ekvation 1.

$$a = \frac{F}{m} \quad (\text{Ekvation 1})$$

För luftmotstånd används ekvation 2.

$$F_a = 0,5\rho C_d A v^2 \quad (\text{Ekvation 2})$$

Där C_d är luftmotståndskoefficienten för bilen. ρ är luftens densitet. A är bilens area sett framifrån, kallad frontarea. v är hastigheten.

För rullmotstånd används ekvation 3.

$$F_r = mgC_r \quad (\text{Ekvation 3})$$

C_r är koefficienten för rullmotstånd.

Och för krafterna som uppkommer av lutningen används ekvation 4.

$$F_g = mg \sin(\theta) \quad (\text{Ekvation 4})$$

Där θ är markens lutning.

Bilens inre tröghet som drivaxlar, växellåda och inre friktion försummas i denna enkla modell av bilen. Då bilen står stilla sätts de bromsande krafterna lika med noll för att bilen inte ska rulla bakåt.

Utanför denna kärna finns kringutrustning som växelväljare, begränsare av indata och lagring av värden. Växelväljaren tittar på aktuell hastighet och beroende på vilken hastighet bilen har så väljs förutbestämd växel.

2.3.4 Metod att mäta när ett fordon kommer att nå sin träffpunkt.

När man vill veta hur långt ett fordon har kvar till kollision har man hastighet, tid och sträcka att laborera med. Det vanliga är att ange sträckan som är kvar men då måste man även ge hastigheten, alltså är det bättre att använda tiden då det blir mindre data att skicka.

2.3.5 Metoder att göra ett styrsystem

Det förslaget som gavs i exjobbetsförslaget var att ha en master och en slavebil som genom att masterbilen räknade ut sin tid till kollision eller TTC (Time To Collision) som räknas ut enligt ekvation 5.

$$TTC = \frac{s}{v} \quad (\text{Ekvation 5})$$

Där s står för den sträcka bilen färdats och v står för bilens aktuella hastighet. Denna tid skickades till slavebilen som har i uppgift att se till att sin egen TTC är densamma. Om de har samma TTC så träffar de varandra. För att räkna ut TTC behöver man mäta tid och sträcka. Systemet kan även styras av en hastighetsfunktion som bilarna ska följa. Hastighetsfunktionen kan dels räknas ut i realtid men även innan själva testet startar, detta har olika för- och nackdelar vilket berörs i det teoretiska kapitlet 5.7.

Till systemet kommer olika hjälpsystem beroende på vilken typ av system man har. Dels hjälpsystem som för att få bilarna att träffa varandra jämför TTC tiden och kompenserar för saker som gupp i vägen, vindhastighet, seghet i fordonet och fel i noggrannheten på uppmätt sträcka. Ett övervakningssystem är ett annat hjälpsystem som avbryter testet om något går fel eller om bilarna inte kommer att träffa varandra av någon anledning.

3. Modeller

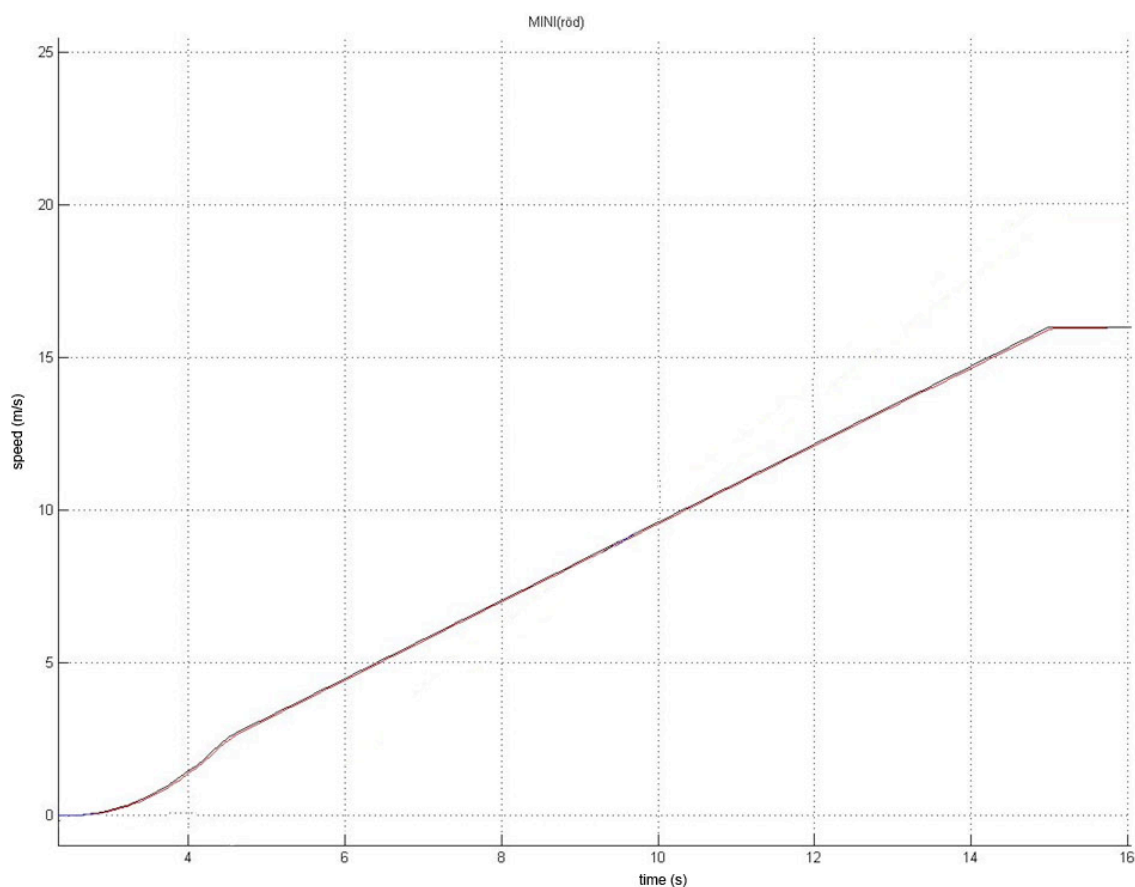
3.1 Förord Modeller

För att undvika att förbigå någon bra idé skapades flera system. Dels för att prova en viss teknik och dels för att ställas inför olika problem som vars olika lösningar i slutet kanske kunde hjälpas åt. En känd metod inom utvecklingsbranschen är just att testa ett större antal system och idéer för att sedan strypa åt för att slutligen ha ett system kvar. Då många modeller är snarlika på vissa punkter och vidareutvecklingen ofta gått från en modell till en annan har de delats in i två huvudkategorier, funktionsstyrda och egenstyrda system.

3.2 Närmare inpå de olika systemen

3.2.1 Egenstyrt system

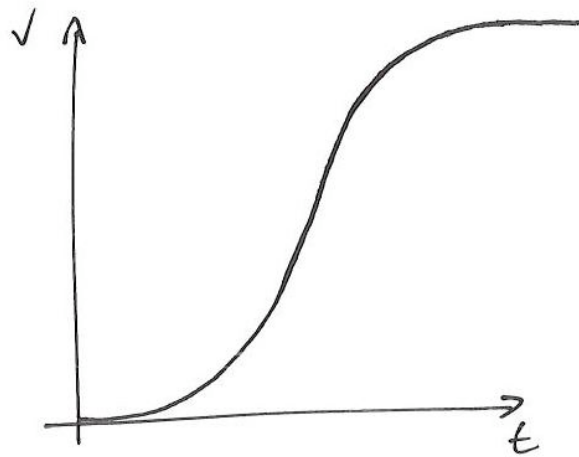
I det första styrsystemet som gjordes krockades bilarna så snabbt som möjligt. En master och en slave bil där masterbilen accelererar lite saktare för att slavebilen ska ha en chans att kompensera om den hamnar lite efter. Hamnar den före är det inte lika svårt att kompensera. De båda bilarna mäter sträckan de kört och systemet kan tack vare en klocka räkna ut sin egen hastighet. Masterbilen skickar sin tid till kollision (TTC- Time To Collision), enligt Ekvation 5, till slavebilen som har i uppgift att vara framme med rätt hastighet då TTC är noll. Att räkna ut denna TTC är krävande då man har ickelinjära kurvor och olika hastighet på bilarna. Det behövs en realtidssimulering för att räkna ut denna TTC samt att man måste skicka mycket data mellan bilarna hela tiden. Den modellen som testades här hade dock bara linjära accelerationskurvor (se figur 4) för att underlätta simuleringssmodellen.



Figur 4.

Här ses en linjär accelerationskurva, böjen i starten är för att undvika hjulspinn. Denna kurva är mycket enkel att räkna på.

Att göra en modell som vet precis hur bilen accelererar är både krävande och inte speciellt användarvänligt då varje enskild bil måste programmeras in i detalj i styrsystemet. Ett bättre alternativ är att göra en funktion som har en accelerationsprofil som de flesta fordon kan följa med ytterst små modifieringar. (se figur 5)

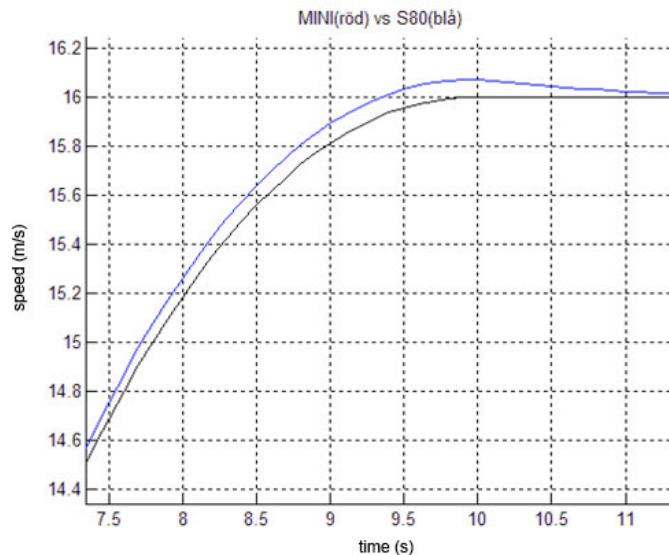


Figur 5.

Här ses en principskiss över hur man kan accelerera en bil mjukt.

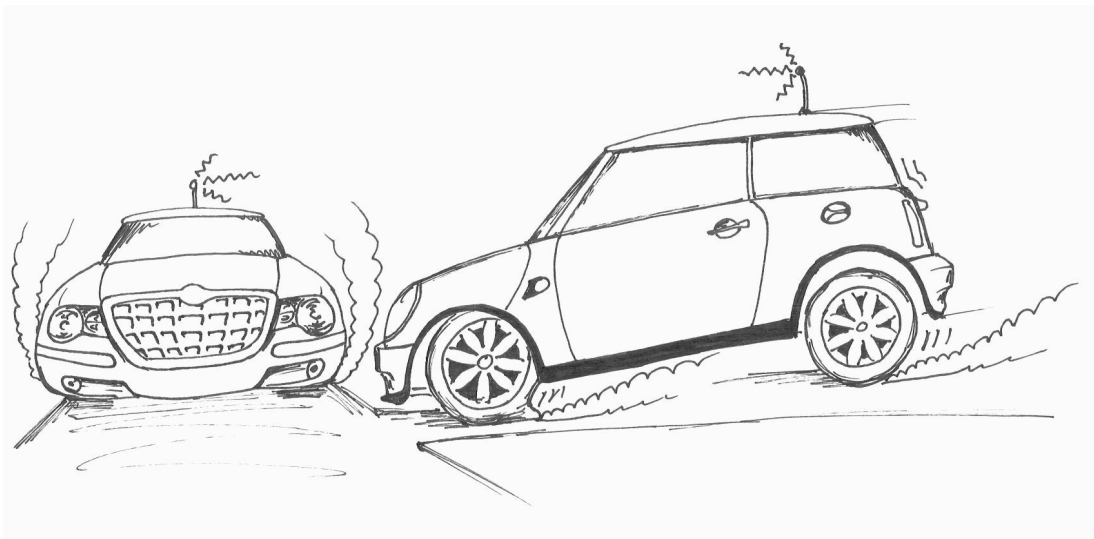
3.2.2 Funktionsstyrt system

Genom att göra en funktion som liknar den en bil normalt accelererar efter till en viss hastighet, på en viss tid, kan man få bilen att med styrsystemets hjälp följa den funktionen. Då är systemets uppgift bara att se till att den egna bilen följer sin egen funktion. (se figur 5) Detta system bör sedan kompletteras med ett system som under sista biten ser till att bilarna verkligen träffar varandra, då ingen sådan reglering finns i den grafstyrda regulatören. Ett bättre alternativ är att systemet själv skapar en funktion i testets början efter de förutsättningarna som finns. Fördelen här är att de båda funktionerna för de båda bilarna fungerar som en datalänk utan någon tidsfördröjning alls. Följer varje bil sin egen funktion som den ska så kommer den att vara på rätt plats vid rätt tid och med rätt hastighet. Det ställer stora krav på att funktionen räknas ut korrekt men detta behandlas i kapitel 4.2. Funktionen styrs utav två parametrar som avgör hur snabbt bilen ska accelerera och vad sluthastigheten ska vara efter den givna tiden. Tiden kan även anges i vilken den maximala g-kraftpåkänningen får vara, genom att man i själva initialfunktionen där grafen räknas ut kontrollerar att accelerations g-värdet inte överstiger angett värde (se kapitel 4.2). Varje bil måste även ha en kontrollfunktion som kontrollerar att den följer grafen som den ska. Gör den inte det kommer inte bilarna att träffa varandra som de ska och testet måste avbrytas. Den enda data som behöver skickas då, om en bil misslyckas med att följa funktionen (se figur 6), är den som avbryter testet. (se figur 7)



Figur 6.

Om bilen inte följer kurvan åker den inte med rätt hastighet och kommer med största sannolikhet att missa sitt mål.

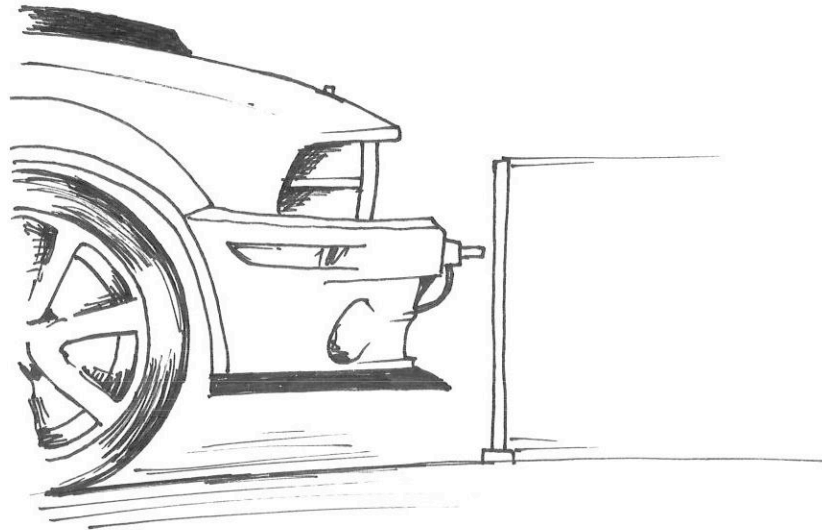


Figur 7.

Bilarna bör inte kunna träffa varandra om abortdelen i systemet är korrekt inställt.

Kontrollfunktionens uppgift är att avbryta den egna bilen om dess hastighet felar för mycket, vilket får anges i lämplig parameter, och att sedan skicka ett abortkommando till övriga berörda delar i testet. Olika tänkbara sätt att kontrollera att bilen följer kurvan är följande. Maximalt plus-minus-värde, där bilen som mest får avvika ett bestämt värde. Detta bör då sättas relativt snävt för att inte bygga på ett fel under hela accelerationsförloppet. Ett variabelt plus-minus värde som bilen måste hålla sig innanför. I början finns det utrymme för kompensering men ju längre man färdats och ju närmare målet man kommer, desto svårare är det att kompensera. Detta då trögheten ökar med hastigheten. För att kompensera för trögheten måste det tillåtna felvärdet smalt av till närmare noll fram till sista möjliga abortposition. En utmaning med att tillåta fel, om än ett litet, är att man måste kompensera för det på något vis. Det kan man göra på olika sätt och två av dem är triggerpunkter och virtuella triggerpunkter. Med en triggerpunkt menas en punkt som bilen passerar som antingen är fast eller virtuell. Den fasta är en plastpinne som bilen kör på och den, på bilen monterade,

strömställare som pinnen träffar skickar en signal till styrsystemet som registrerar att bilen nu befinner sig exakt på den sträcka som pinnen är från målet. (se figur 8)



Figur 8.

Längst fram på bilens kofångare ses en strömställare som när den träffas av pinnen ger kontakt och därmed skickar en signal om att då ha blivit träffad till styrsystemet.

En virtuell triggerpunkt fungerar på samma sätt, dock tar den inte hänsyn till mätfel då den bara får uppmätt sträcka från givaren som mäter sträckan. Givaren beskrivs i kapitel 5.4. Den virtuella triggerpunkten fungerar så att den avgör på bestämda ställen om det behövs en kompensering eller om felet anses tillräckligt litet för att ignoreras. Även den fasta triggerpunkten startar en liten analys hurvida kompensering behövs. Fördelen med en fast triggerpunkt är att man vet sträckan exakt.

3.2.3 Hjälpsystem

Då det alltid kommer med fel i ett verkligt system behövs ett hjälpsystem som kompenserar för dessa. Det kan vara ett gupp i vägen, vindhastighet, seghet i fordonet, fel i noggrannheten på uppmätt sträcka.

Det hjälpsystemet gör är att jämföra tiden till kollision (TTC) för de båda fordonen. Den bilen med mindre TTC än den andra är närmare kollisionspunkten. Därför justeras trottelsignalen på båda bilarna så de strävar efter samma TTC. Denna typ av hjälpsystem kräver just en uträkning av TTC

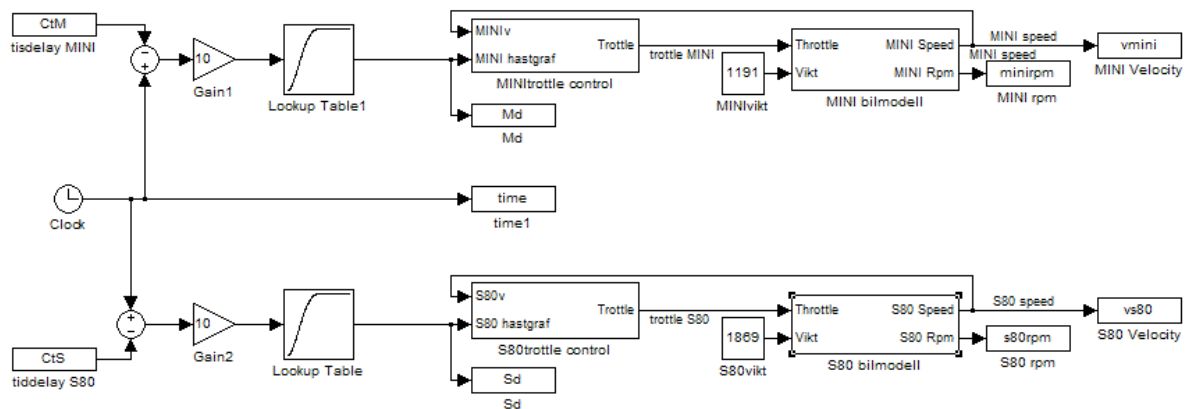
3.2.4 Övervakningssystem

Utifall att något skulle hända som gör att fordonen inte kommer att träffa varandra som de ska behövs ett övervakningssystem som avbryter förloppet på lämpligt sätt. Den främsta kontrollen görs av signalen från de båda bilarna som visar om de klarar att följa hastighetsfunktionen eller inte. Kommer signalen från en av bilarna avbryts båda på lämpligt sätt. Vad som avses som lämpligt behandlas nedan i detta kapitel. Nästa kritiska funktion övervakningssystemet kontrollerar är när bilarna nått fram till krockpunkten, då de ska bromsas för att begränsa området de kan rulla till efter själva krockögonblicket.

Det sätt som är tänkt att avbryta testet är att båda bilarna bromsas så hårt det går. Sista kontrollen bör vara placerad längre ifrån krockpunkten än någon av bilarnas stoppsträcka. Hur många kontrollpunkter som behövs tas upp i teorikapitel 5.8.

3.3 Krav på systemet

Förutom de funktionskrav som ställs på systemet så finns det vissa begränsningar i form av fysiska lagar, och tekniska begränsningar. Fordonen har inte hur mycket effekt som helst, överföringshastigheten på data som skickas mellan bilarna, hastigheten på de olika servomotorerna. Det är en del i examensarbetet att bestämma dessa begränsningar och göra ett system efter de spelvillkoren.



Figur 9. Här ses en översiktbild av styrsystemsmodellen modellerad i simulink.

4. Simuleringsmodellen

Själva simuleringsmodellen (se figur 9), som är modellerad i Simulink, består av flera block med sina olika uppgifter och en initialfunktion som körs innan testet genomförs. Först räknas hastighetsfunktionen ut i initialfunktionen varpå bilarnas hastighetsstyrssystem får bilarna att följa kurvan på bästa sätt. Bilarna i sin tur modelleras i varsina block där saker som vikt, prestanda med mera tas med i beräkningarna. Alla delar beskrivs ingående nedan.

4.1 Simuleringen i korthet

För att få en mjuk funktion för bilen att följa används ekvationen

$$g = \frac{n \cdot 2^{-\frac{n}{z}}}{k} \quad \text{Ekvation (6)}$$

Där g är hastigheten i varje tidpunkt n . z och k är parametrar för att ställa in kurvan. Ekvationen vilken är en exponentialfunktion används för att den ger en mjuk start och har en avtagande avrundad avslutning. Då kurvan vänder nedåt stoppar man den och bibehåller på så vis värdet vilket bör vara given sluthastighet. Man hittar max på kurvan. $g'(n) = 0$ då bilen uppnått rätt hastighet vid tiden n . Tiden n behöver dock inte vara den tiden som bilen har till

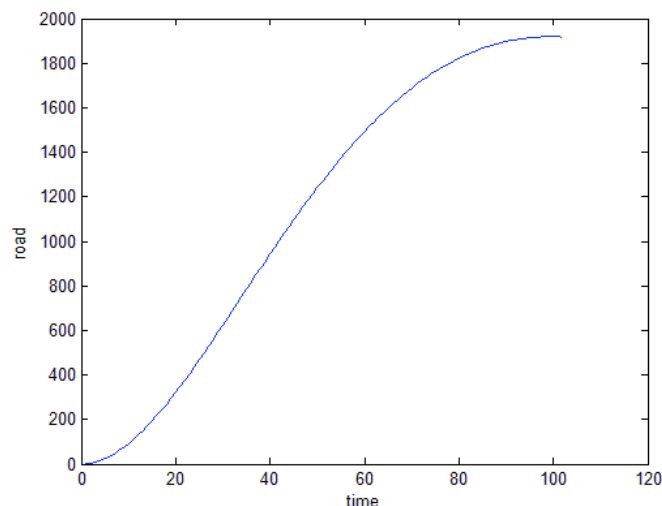
kollisionsögonblicket eftersom bilen kan behöva hålla sin hastighet för att färdas rätt sträcka för att vänta in den andra bilen beroende på vad som räknas fram i initialfunktionen. För att räkna ut hur lång tid det tar att accelerera upp till given hastighet på given sträcka behövs en egen räknare i initialfunktionen som räknar tiden tills det att hastigheten är uppnådd. I den linjära kurvan är det bara att ta medelhastigheten men i denna exponentialfunktion måste man integrera hastigheten över sträckan för att få fram tiden. Tiden behövs för att veta hur lång tid bilarna tar på sig upp till given sträcka. Med båda bilarnas tider kan man avgöra vilken som ska starta först och hur länge den bil som kommer fram först ska vänta innan även den startas. Detta är löst i initialfunktion som med givna indata räknar ut, samtidigt som funktionens hastighetsvärden lagras i en vektor, hur långt bilen färdats och hur lång tid som gått. Detta görs för båda bilarna samtidigt och tiden och sträckan används sedan, enligt tidigare beskrivning, för att få ut tiden det tar att färdas det avståndet man har till krockpunkten. Mera om hur tiden räknas ut beskriv senare i detta kapitel. Själva initialfunktionen skrivs som en m-fil som i en loop gör samma sak som modellen gör kontinuerligt, nämligen hastighetsfunktionen som bilen sen ska följa. För att få bilen att nå rätt hastighet på rätt tid görs först en graf av den givna ekvationen. Grafen jämförs sedan med givna kriterier, sluthastighet och tid. Parametrarna z och k anpassas för att ge kurvan den rätta formen.

Kurvan räknas på nytt ut nu med de erhållna parametrarna och lagras i den vektor som innehåller hastighetsvärdena som bilen sedan ska följa för att få rätt hastighetsprofil.

4.2 Initialfunktionen

Initialfunktionen som använts för i rapporten redovisade tester finns i Bilaga 1.

Startparametrar är de önskade hastigheterna, accelerationstider för de båda bilarna, samt deras accelerationssträckor. Ekvation (6) räknar ut en graf som har den önskade profilen och man hittar max i denna.



Figur 10.

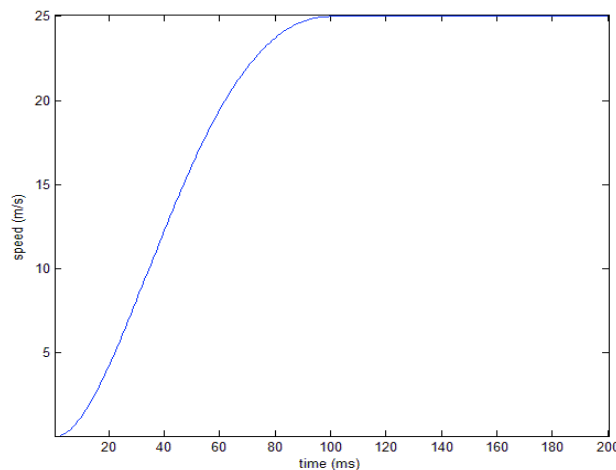
Innan man ställt in parametrarna z och k är namnet på variabeln för sträckan "road".

Detta max (se figur 10) stämmer troligen inte med givna indata utan man måste optimera den via två konstanter. Dessa två konstanter, z och k , räknas ut av att man jämför önskat och

aktuellt max på kurvan, den önskade hastigheten, och får då skillnaden mellan dessa, som då blir konstanten z . Sedan jämförs tiden med given accelerationstid och man får på samma sätt konstanten k för att få tiden att bli den önskade. Parametern k ställer in hastigheten och parametern z ställer in tiden för kurvan. Parametern k räknas ut enligt ekvationen

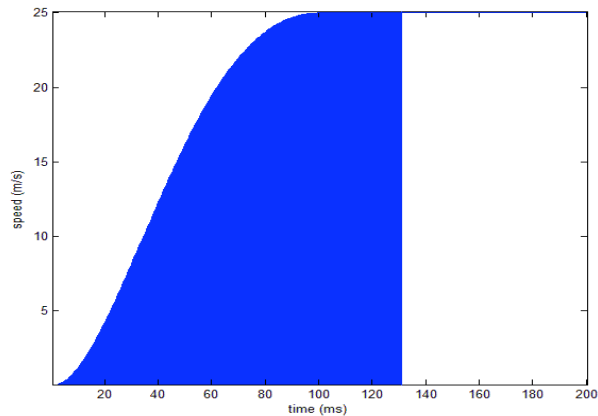
$$k = \max(\text{road})/v \quad (\text{Ekvation 7})$$

Där road är en variabel för sträckan i Ekvation 7 innan man ställt in parametrarna för z och k . Parametern z bestäms mera avancerat genom flera loopar där rätt värde fås då kriterierna för att ursprungskurvans max och accelerationstiden är uppfyllda. Nu kan man skapa en kurva som stämmer överrens med hur man vill att bilarna ska bete sig. Kurvan (se figur 11) skapas och lagras i en vektor med tidsintervallen en tiondels sekund.



Figur 11.

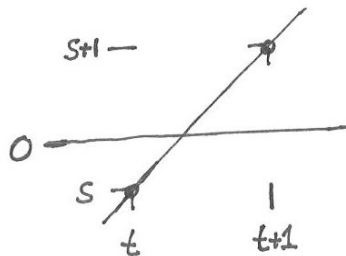
Mera noggrannhet än en tiondels sekund anses inte nödvändigt då man eftersträvar snabbhet och då kurvan är en mjuk kurva blir förändringarna små mellan tidpunkterna. Värden däremellan fås via interpolation via ett så kallat LookupTable i Simulink. För att få reda på hur lång tid det tar att transportera sig given sträcka, med den nu inte okända hastighetsprofilen, integreras kurvan och man får fram tiden det tar att färdas den givna sträckan. Integreringen görs i initialfunktionen där man adderar ihop alla hastighetsvärdena i hastighetsvektorn. (se figur 12)



Figur 12.

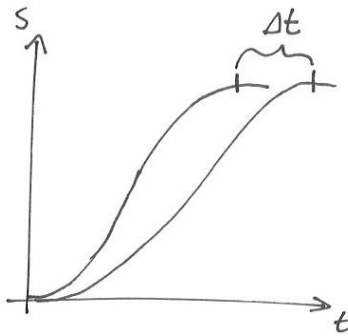
Då man adderar värden under integreringen av hastighetskurvan adderas även avrundningsfelet. Då man har stor siffernoggrannhet och mindre än 200 värden (ett värde per tiondels sekund och maximalt 20 sekunder under de tester som i rapporten gjorts) som ska adderas blir felet litet och försummas i dessa beräkningar.

Tiden får man ut genom att i en loop kontrollera när bilen färdats över given sträcka. Då själva initialfunktionen använder steglängden en tiondels sekund i alla uträkningar och loopar blir den upplösningen inte tillräcklig för att få reda på när bilarna färdats den givna sträckan. På en tiondel hinner en bil med hastigheten 20 meter per sekund två meter och det behövs större noggrannhet. Större noggrannhet fås genom att man ändrar upplösningen just mellan de värdena närmast övergången. Den lagrade sträckan i vektorvärdet innan sträckvärdet överskridits och det värde då sträckan överskridits används med deras tidsvärden till att göra en linjeekvation mellan dessa punkter. (se figur 13)

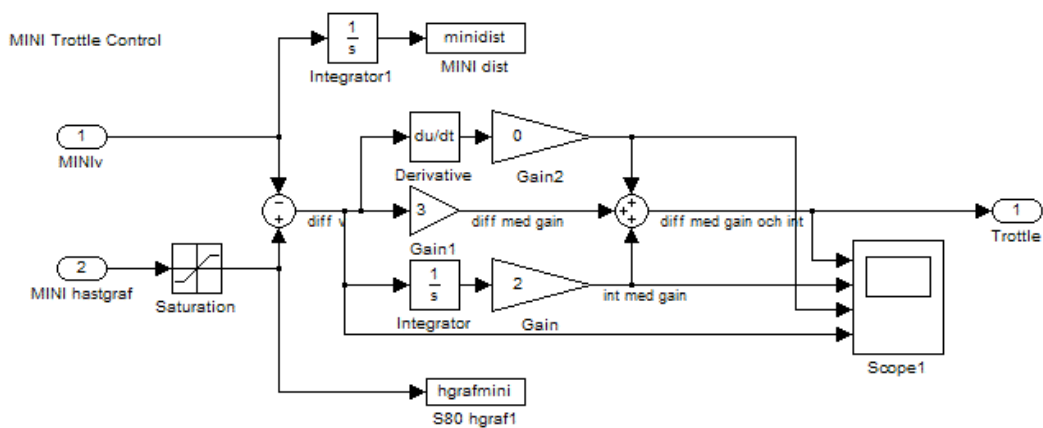


Figur 13.

Linjeekvationen används för att så när som på exakt räkna ut när bilen färdats angiven sträcka. Denna mera tillförlitliga tid jämförs med den andra bilens tid så man vet hur lång tidsdifferensen mellan de båda bilarna är. Tidsdifferensen avgör hur lång väntetid man ska ha på den snabbare bilen för att de ska vara framme samtidigt. Tidsdifferensens delta-t tecken avgör även vilken bil som ska starta först. (se figur 14)



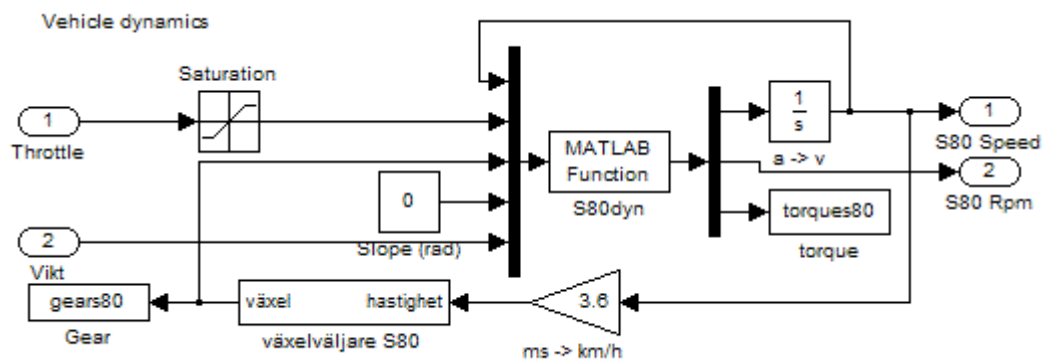
Figur 14.



Figur 15.

4.3 Blocket Trottle Control (figur 15)

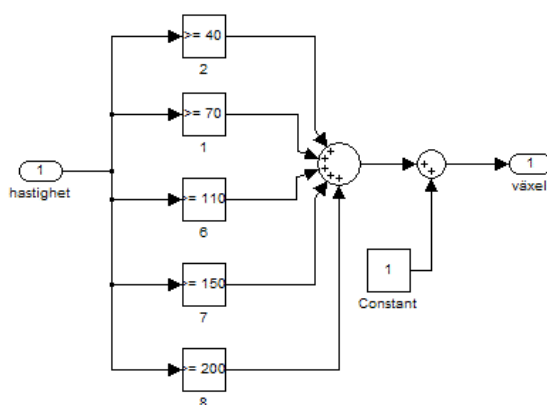
Indata är önskad hastighet och aktuell hastighet. Via en summator subtraheras aktuell hastighet från given hastighet vilket ger ett värde på trottelläget för att uppnå önskad hastighet. Är hastigheterna lika behöver man inte gasa, men blir det en skillnad blir inte differensen noll utan trotteln öppnas mera. Signalen satureras till ett maxvärde av ett för att inte ge ett trottelvärde större än ett. Värdet kan även PID regleras för att ta bort störningar och snabba svängar. Utgående värde bör hålla sig mellan noll och ett vilket kan ses som en procentuell siffra på trottelläget. 1 är 100 % och 0 är 0 % gaspådrag.



Figur 16.

4.4 Bilmodellsblocket

Själva kärnan i bilmodellsblocket (se figur 16) består av en m-fil som simulerar på bilen inverkan av krafter. Omkring denna kärna finns en växelväljare, begränsare av indata och lagring av värden. Den m-fil som simulerar det mesta av bilen har 5 olika indata vilka är: Aktuell hastighet, trottelläge, aktuell växel, vägens lutning samt bilens vikt. Vikten, och lutningen ställer man in på plats medan trottelläget kommer från blocket *trottle control*. Den aktuella hastigheten kommer från den egna m-filen som då kommer ut som accelerationen som integreras för att bli hastighet. Växelläget avgörs i ett växelväljblock i bilmodellblocket som adderar en etta för varje hastighetskriterium som är uppnått. (se figur 17) När bilen uppnår en given hastighet adderas en etta och m-filen får insignalen en växel högre än tidigare. Då bilens hastighet som ges av accelerationen är i enheten meter per sekund måste även denna omvandlas till kilometer per timme genom att multipliceras med konstanten 3,6 då hastigheten för att välja växel anges i just kilometer per timme.



Figur 17.

4.4.1 M-filen i bilmodellsblocket

Indata är aktuell hastighet, trottelläge, aktuell växel, vägens lutning samt bilens vikt. Först bestäms konstanter och parametrar för bilens dynamiska egenskaper. Det är konstanter för rullfriktion, luftens densitet, luftmotståndskoefficient, bilens frontarea samt gravitationskonstanten.

Vad dessa konstanter är satta till för de olika bilarna kan ses i Bilaga 2, ekvationerna hittas i Kapitel 2.3.3. Marginaler för indata sätts ifall de av misstag sätts till orealistiska värden. Även utväxlingsförhållandena matas in för de olika växlarna. Utväxlingsförhållandena kan ses i Bilaga 2. Dessa värden inkluderar all utväxling från motorn, genom växellåda och slutväxel till hjulets radie. För motormodellen bestäms först varvtalet genom att ta utväxlingen för hela drivlinan med aktuell växel gånger bilens nuvarande hastighet. Motorns varvantal behövs för att bestämma vridmomentkurvan. En enkel modell för bilens vridmomentkurva görs av räta linjer som markerar aktuellt vridmoment för varje varvantal. Dessa är beskrivna i Bilaga 2. Motorns kraft sätts lika med trottelläget multiplicerat med vridmomentet multiplicerat med aktuell växels utväxlingsförhållande.

De olika typerna av externa krafter räknas ut. Rullfriktionen är som bekant massan multiplicerat med gravitationen multiplicerat med rullmotståndskonstanten. Luftmotståndet är hastigheten i kvadrat multiplicerat med frontarean multiplicerat med luftmotståndskonstanten och dessa multipliceras med konstanten för luftens densitet. Den påverkande kraften som

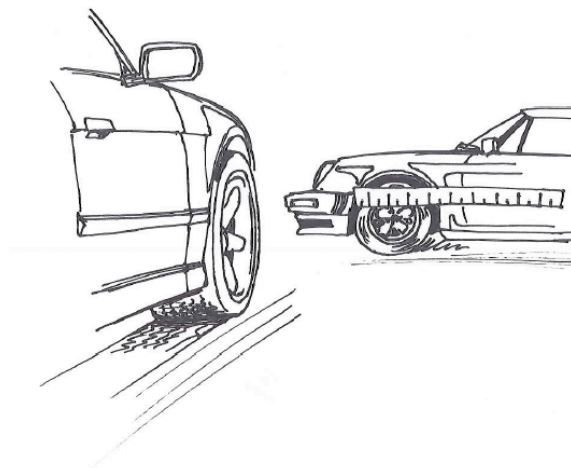
markens lutning resulterar i fås av massan multiplicerat med gravitationen multiplicerat med sinus för markens lutning. Dessa externa krafter summeras till ett. Är hastigheten noll är även krafterna noll, annars åker ju bilen bakåt i denna förenklade modell.

Själva accelerationen för bilen fås av formeln acceleration är lika med kraft dividerat med massa. Här är kraften de yttre krafterna subtraherat från motorkraften. Slutligen räknas varvtalet ut av motorvarvet i radianer gånger konstanten $60/6,28$ för att få det i varv per minut istället för radianer per sekund. Dessa formler återfinns i bilaga 2.

5. Teori

5.1 Inledning av teorin

För att kunna göra en i verkligheten fungerande applikation av denna simulering krävs tillförlitlig kringutrustning. Det är ett mycket snabbt system med ytterst små feltoleranser. Ett exempel på detta är att om en bil ligger efter sin graf med en tiondels sekund, vilket inte låter så mycket, så missar den sitt mål med 2 meter i hastigheten 20 meter per sekund. (se figur 18)



Figur 18.

Därför behöver man även system som övervakar händelseförloppet och som kan avbryta om det är för stora fel.

För att övervaka bilarna behöver man en länk mellan bilarna och en central (som skulle kunna befinna sig i en utav bilarna). Denna länk måste vara otroligt snabb beroende på vilka data som ska skickas. Exempel på olika länkar kan vara sladdar eller trådlös telekommunikation. Ett intressant område är även om man skulle kunna hitta en så snabb kommunikationslänk så man kan styra bilarna som en enhet. Eller om det kan lösas med en exakt tid mellan de båda bilarna.

5.2 MaxG kriterie

Då systemet ska användas på krocktester kan det användas krockdockor för mätning av krafter, accelerationer och skador. Dessa ställer speciella krav på själva accelerationsfasen. Krockdockor har givna värden för hur tröga dess leder får vara, och gör man en kraftig acceleration så kan dessa röra sig och på så vis hamna i fel position vilket ger felaktiga värden vid själva kollisionen. Därför vill man med krockdockor i bilen inte att den ska accelereras med mer än 0,2 till 0,3 g.

5.3 Kommunikationslänken

Typer av kommunikationslänkar kan vara radiolänk och olika typer av signalkabel. Signalkabel kan uteslutas direkt då det uppstår komplikationer med uppåt 400 meter kabel som uppnår höga hastigheter.

När man ska skicka data mellan bilar som avståndet varierar mellan 400 meter eller mer, ner till 0 meter så ställer det en del krav på kommunikationslänken.

Radiolänken trasslar inte men är en långsammare länk då data ska konverteras om till paket som skickas och sedan konverteras tillbaka då de tas emot av mottagarenheten. En beräkning i dator räknar man med att den tar 20 millisekunder och via radiolänken tar det 2 millisekunder att skicka ett paket data, sen tar det ytterligare 20 millisekunder att beräkna datan i den andra datorn. Data som är skickad mellan bilarna tar $20+20+2$ millisekunder det vill säga, den är 42 millisekunder försenad, vilket betyder att den egna bilen har färdats i 42 millisekunder innan den andra vet om det. I radiolänksystemet finns en funktion som skickar med aktuell tid med varje data och som synkroniserar de olika enheternas klockor med stor noggrannhet.

Radiolänken är det kommunikationssystem som valts då den egengrafstyrda regulatorn inte behöver en så snabb länk. Detta eftersom den information som ska skickas bara är informationen om det går bra eller om man ska avbryta.

Ett exempel på en radiolänk som uppfyller dessa krav är en LAN enhet som heter RM-240 från Westermo i Västerås. Se [Ref C].

5.4 Avståndsmätning

Avståndet bilarna färdas är en viktig del i projektet. Mätningen av avståndet vill jag påstå är den viktigaste punkten överhuvudtaget att få så exakt som möjlig. Är den ”exakt” så ska bilarna träffa varandra som det är tänkt om inget oförutsett inträffar. Mätningen av den sträcka bilarna färdats ger även hastigheten på bilen likaså acceleration då man mäter avverkad sträcka per tidsenhet.

Avståndet kan mätas på olika sätt och nedan presenteras de alternativ som varit uppe på förslag för detta projekt.

5.4.1 Lasermätning

En laser ger stora möjligheter på många områden men inom mätning är det främst två sätt man kan använda den. Dels avståndsmätning, som polisen, genom att skicka pulser och mäta tiden det tar innan de kommer tillbaka och dels som en optisk datormus, att scanna av underlaget och jämföra förflyttningsbilderna.

En ”polislasermätning” är mycket exakt på sitt område, vilket blir ett problem då man vill mäta över ett så brett område. Mätningen måste ju vara exakt från start till mål och den exakthet man vill ha på slutet kräver en lasermätare som inte klarar stora avstånd, och den med stora avstånd har i sin tur för stor felmarginal. Man skulle kunna ha flera lasrar och kolla medeltal från dem och placera dem fast på marken och mäta på den rörliga bilen. För förslag på laser av typen ”polislasar”, se [Ref D].

Den andra typen av laser som bara kollar på marken är då ett bättre alternativ. De tekniska svårigheterna är att få den att ligga på rätt höjd hela tiden då bilen kommer att röra sig upp och ner lite beroende på ojämnheter i vägbanan samt eventuella korrigeringar i färdriktningen, samt vad som händer om man kör över en vattenpöl.

En annan svårighet som gäller båda lasrarna är att en laser är en dyr och ömtålig apparatur vilken man inte vill ska vara med i själva krocken då den troligen skulle gå sönder.

5.4.2 Pulsgivare

På VTI används flitigt en typ av pulsgivare för att mäta avstånd på olika sätt.

[Ref E] Pulsgivare. I detta projekt är det lämpligaste att koppla denna till ett av de icke drivande hjulen och räkna pulserna den ger för varje varv hjulet snurrar. Anledningen till att man använder pulsgivaren på ett ickedrivande hjul är att man vill minimera risken för fel på grund av hjulspinn. Sträckan fås av att man vet omkretsen på hjulet vilket då ger den sträckan per varv hjulet snurrat.

Den pulsgivare som är tänkt till projektet ger 300 pulser per varv vilket blir ett stort antal pulser när en bil färdas i runt 20 meter per sekund. Ett bilhjul räknar man med har en omkrets på cirka två meter vilket ger 150 pulser per meter, och i 20 meter per sekund blir det 3000 pulser per sekund att ta hand om och beräkna. För så många pulser behövs en räknare som håller koll på pulserna eftersom styrdatorns hastighet inte är så snabb. Antingen kan man lagra pulserna på hög och alltså ha tillgång till gamla värden, eller så tar man bara värden på bestämda tider och lagrar för att spara datamängden. Förslagsvis det alternativ som sparar datamängd.

Det är viktigt att mäta hjulets omkrets noga då hela testet faller om detta värde felar. En millimeter fel hjulomkrets blir på 200 meter en decimeter i krockögonblicket.

5.5 Vridmomentkurvan och växlingar

Eftersom vridmomentkurvan är kraftigt förenklad förekommer viss kantighet på denne vilket resulterar i orealistiska egenskaper hos motorn. Dessa märks i hela systemet då kurvan ändrar sig efter varvtalet. Det är inga problem för styrsystemet att kompensera för detta och i verkligheten finns det inga sådana ”vassa kanter” och dessutom en tröghet i bilen som även den kompenseras för eventuella ojämnheter. Fenomenet syns i plottarna och behöver därför tas upp. Ett liknande ”fenomen” uppstår vid växlingarna eftersom de sker från en tidsenhet till en annan, det vill säga en tiondels sekund. Detta händer inte heller i verkligheten utan sker mycket mjukare då de bilar som är tänkta till denna typ av krocktester är försedda med automatlådor med koppling av momentomvandlartyp vilken är utmärkt i avseendet att filtrera bort ojämnheter. Mjukheten i momentomvandlaren beror på att det inte finns några fasta föremål i ingrepp utan en olja som överför momentet.

5.6 Andra typer av bilmodeller

Under arbetet testades olika avancerade typer av bilmodeller. Den främsta anledningen till den enkla modellen som använts är att det räcker gott med denna enkla modell, men även svårigheter som kommer med mera avancerade modeller. När man gör en modell av verkligheten måste man ofta beskriva alla inverkanse krafter. I en modell av en bil med motor, koppling, drivlina och hjul samt marken den kör på, finns delar som kommer att ha kriterier som talar emot varandra. Klassikern är att modellen inte kan veta vilken del som ska ge vika. Modellen ”vet” att denne har ett visst varvtal och att den har en viss hastighet. När då bilen växlar kvarstår ju bilens varvtal när man släpper upp kopplingen vilket antingen får kopplingen eller hjulen att slira. Direkt får hjulen en högre hastighet eftersom motorn har samma varvtal och nu en högre växel, då är det dags att växla igen och på några tiondelar har bilen uppnått sin teoretiska maxhastighet. Att avhjälpa detta ”fenomen” kräver ganska omfattande modeller och eftersom den valda enkla modellen fungerar tillfredställande lades inget extra arbete på att bygga en bättre modell.

5.7 Realtid vs hastighetsfunktion lagrad i array

Den funktionskurva som bilen ska följa under sin acceleration kan antingen räknas ut i en initialfunktion, och lagras i en array, före själva testet startas men även under testets gång. Den som räknas ut i förtid har fördelen att man under hela testet vet vad som kommer att hända i vilken tidpunkt. Nackdelen är att man bara kan påverka testet genom att avbryta det. Den realtidsbestämda funktionskurvan kan hela tiden anpassas om förutsättningarna ändras men man vet inte vad som händer framåt i testet. Det är den i initialfunktionen uträknade hastighetsfunktionen som har valts för nedan redovisade tester.

5.8 Triggerpunkternas placering

Triggerpunkternas placering och antal beror på testets utformning. Det effektivaste är att ha en triggerpunkt då bilen precis har sin aktuella stoppsträcka kvar till kollisionspunkten. Bilen stoppas i sista stund om den visar sig befinna sig fel jämte hastighetsfunktionen. Stoppas den ena bilen måste även den andra bilen stoppas. Detta förutsätter att man kör efter den i initialfunktionen uträknade hastighetsfunktionen. Kör bilen efter den i realtid uträknade hastighetsfunktionen kan man teoretiskt sett ha obegränsat med triggerpunkter. Det får bli en fråga för framtida utvärdering, beroende på test, hur många triggerpunkter som behöver användas om den i realtid uträknade hastighetsfunktionen väljs för tester.

6. Resultat

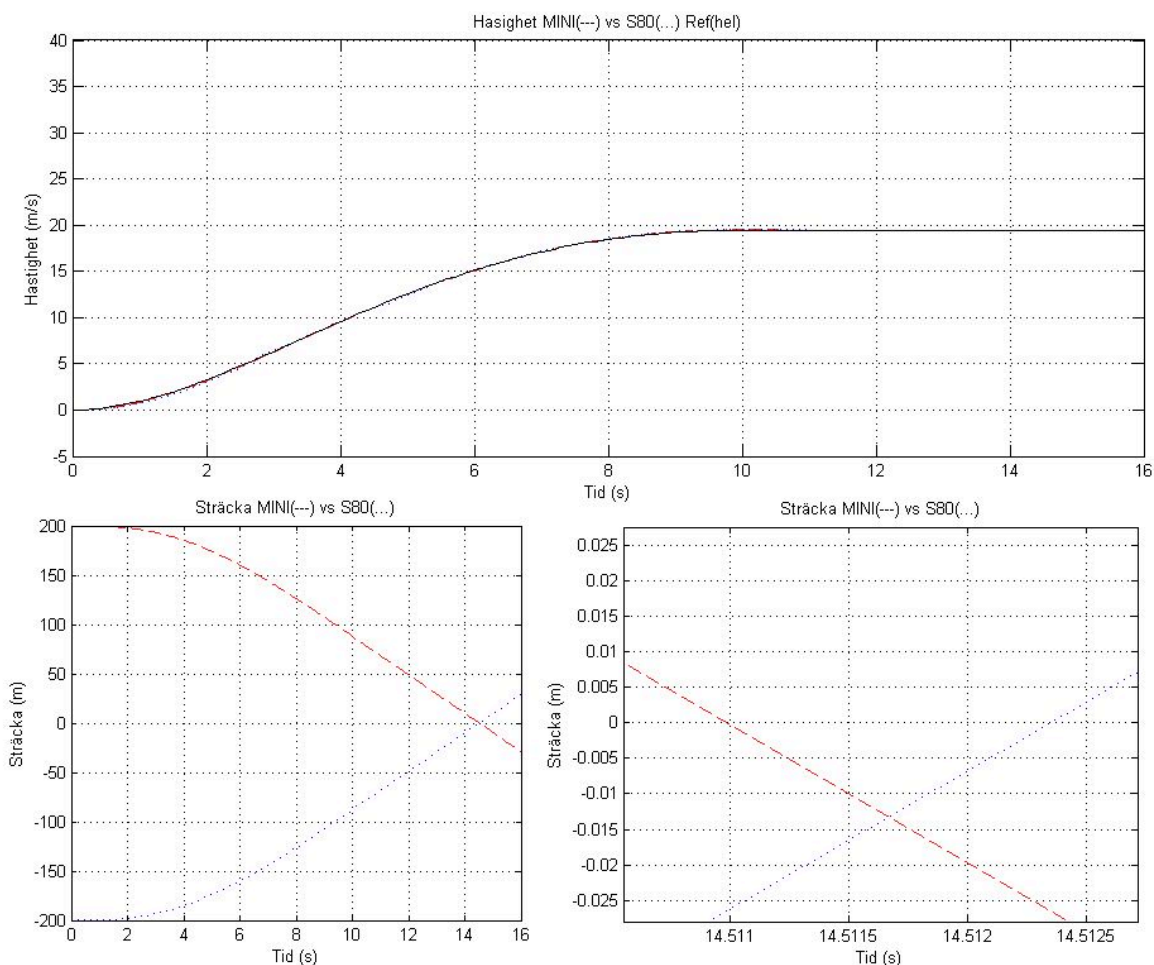
6.1 Inledning resultat

Körningen av testet går till så att man först kör initialfunktionen för att förse själva simulinkmodellen med data. Därefter simuleras styrsystem och bilmodeller varpå uträknade utdata kan analyseras i diverse plottar.

6.2 Typer av körningar och deras resultat

Testet som gjorts med nedan redovisade resultat avser en korsningskollision med vinkeln 90 grader mellan bilarnas färdriktning enligt Figur 2. Då testet ska vara flexibelt för olika hastigheter testas det i 3 olika hastigheter. De hastigheter som är testade i simuleringsmodellen är 70 kilometer i timmen, 90 och 110 kilometer i timmen. Valet har sin grund i att det är de hastigheter som troligen kommer att testas i verkligheten. Långsammare hastigheter görs mest inomhus och anses inte lika farliga för förare att till exempel testa ett antisladdsystem. Högre hastigheter testas vanligtvis inte då man inte får ut något användbart mätresultat, man ser inte heller något med hjälp av höghastighetskameror i så hög hastighet. De bilar som testet avser att simulera är en MINI Cooper och en Volvo S80, båda av årsmodell 2007.

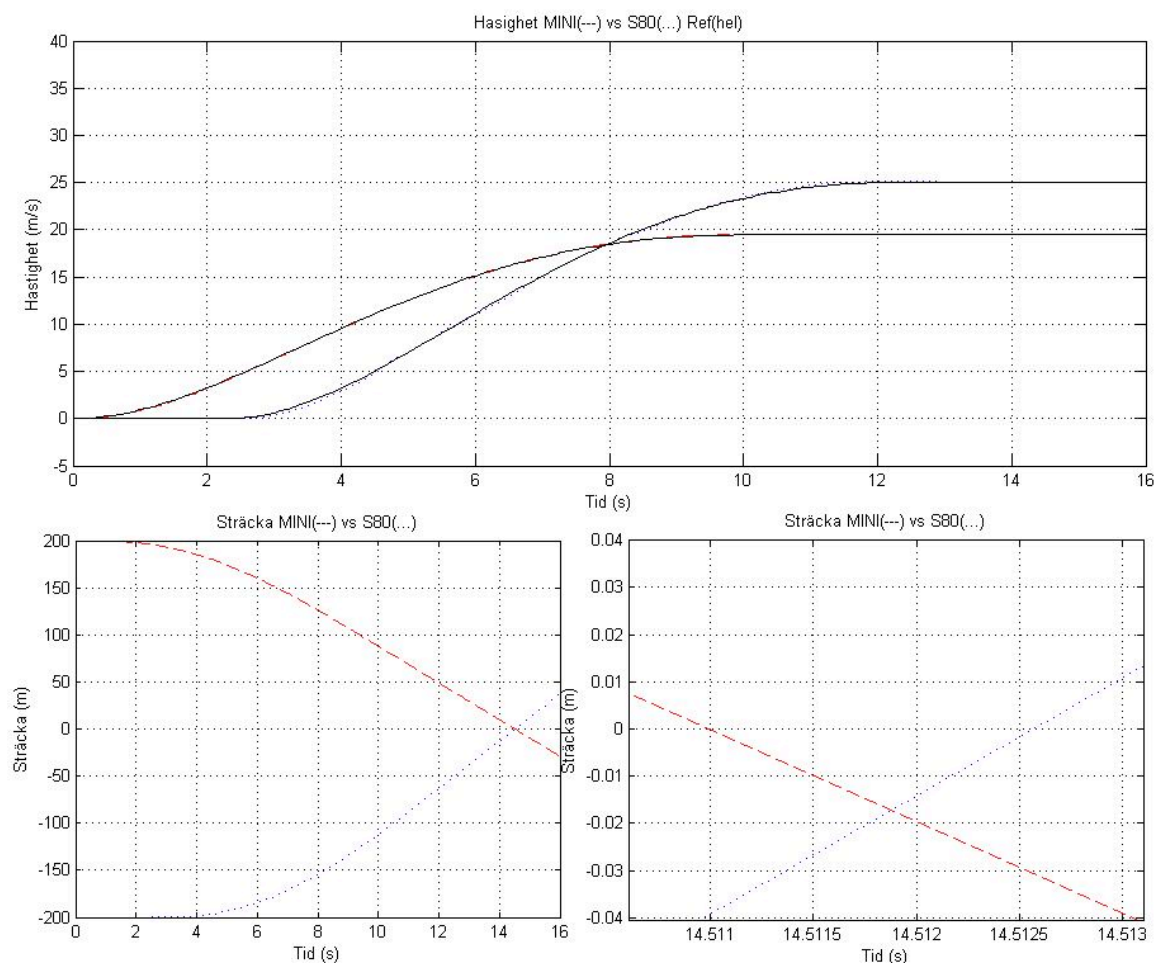
6.2.1 Test 1. 70 km/h vs 70 km/h



Figur: 70 km/h 200 meter 10 sekunder.

I figuren ovan ses då båda bilarna startar 200 meter från målet och ska accelerera upp till 70 km/h, det vill säga 19,45 m/s, på 10 sekunder. Eftersom båda bilarna får samma hastighetsfunktion att följa, eftersträvar de att köra lika fort i varje tidpunkt. Sträckan som börjar på 200 meter respektive -200 meter och går till 0 som är den punkten då bilarna träffar varandra. Tidpunkten då den enskilda bilen når sträckan 200 färdade meter är viktig då den är avgörande för om bilarna träffar varandra. Tidsdifferensen får inte bli större än några tusendelar av en sekund. Tag denna differens gånges den snabbaste bilens hastighet i m/s för att se hur långt den bilen hinner färdas på den tiden. Ett acceptabelt värde på denna sträcka är cirka 4 centimeter. I detta test blir det 0.03 meter, det vill säga 3 centimeter.

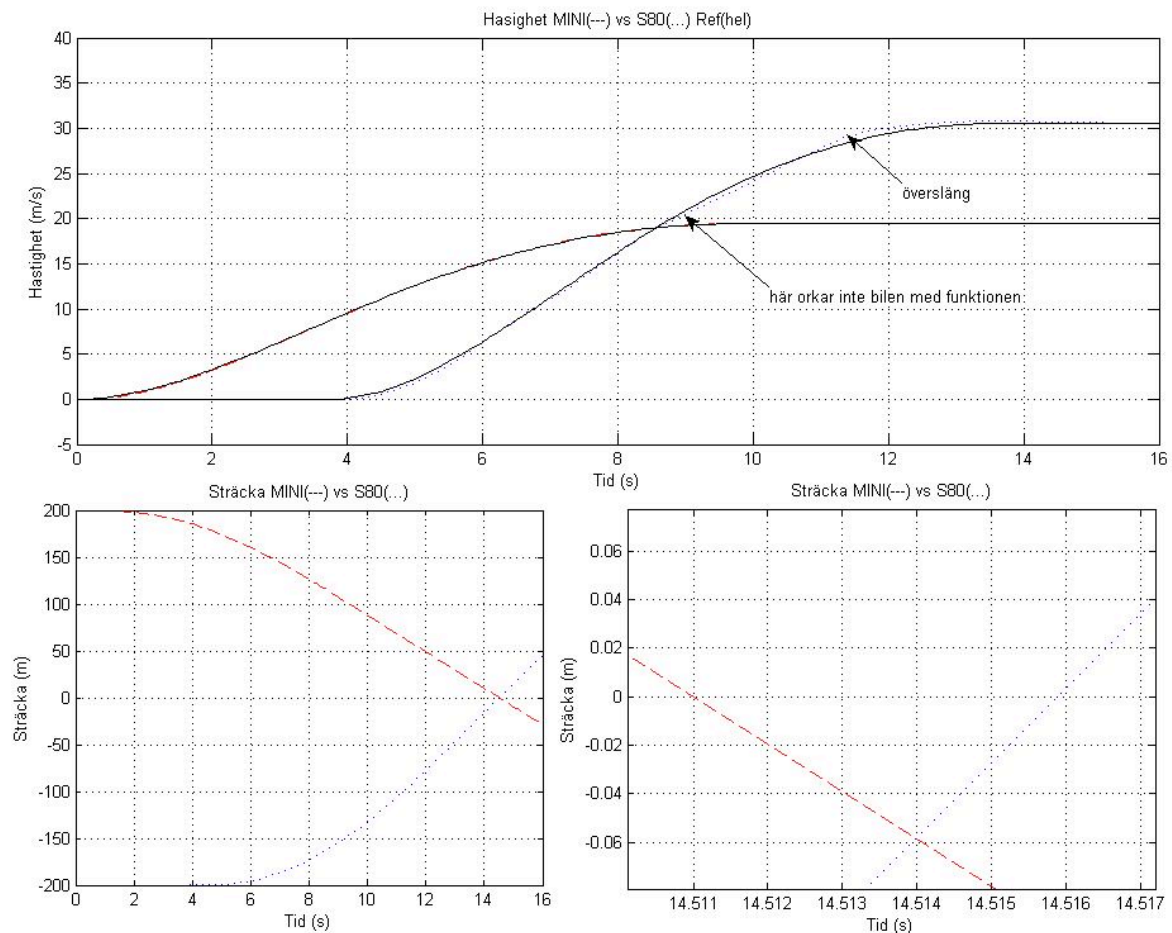
6.2.2 Test 2. 70 km/h vs 90 km/h



Figur: 70 km/h och 90 km/h 200 meter 10 sekunder.

På ovanstående bild körs den ena bilen i 70 km/h och den andra bilen i 90 km/h, fortfarande med 200 meters transportsträcka och 10 sekunders accelerationssträcka. Då S80'n ska accelerera till 90 km/h (det vill säga 25 m/s) kommer den att köra snabbare och alltså komma fram tidigare, därför väntar systemet med att starta denne enligt uträknad tid. Nu är accelerationen ganska kraftig och man ser i figuren att S80'ns styrsystem inte riktigt klarar att följa kurvan. Dels syns detta på grund av att S80'n är en tyngre bil men ojämnheter beror även på den kraftigt förenklade vridmomentkurvan som har en del vassa kanter, samt växlingarna som också gör vassa kanter som systemet inte klarar att reglera för. Nu är det relativt små fel och det märks inte i slutet. Dessutom kommer man inte att accelerera så kraftigt och man har inte de vassa partierna på en verklig bils vridmomentkurva eller växlingar. Träffsäkerheten här är 0,025 meter eller 2,5 centimeter.

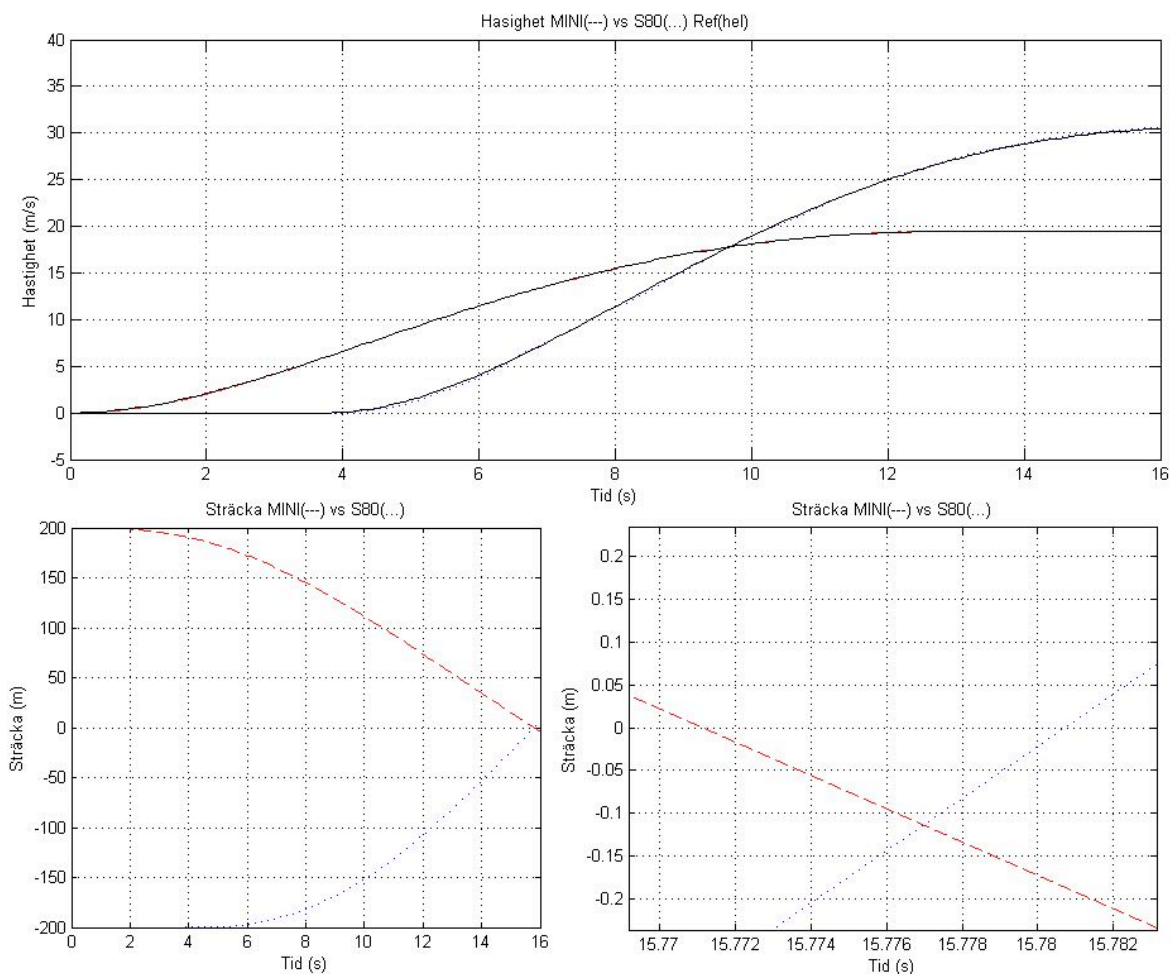
6.2.3 Test 3. 70 km/h vs 110 km/h



Figur: 70 km/h och 110 km/h sträckan 200 meter 10 sekunder.

Denna bild är intressant då den visar systemet under överbelastning. Man försöker accelerera S80'n till 110 km/h på 10 sekunder, det är visserligen möjligt men inte efter den valda accelerationen given ur hastighetsfunktionen. Man ser att vid 20 m/s släpper hastighetskurvan från hastighetsfunktionen. Dessutom har styrsystemet i denna kraftiga acceleration, med valda parametrar på regleringen, svårt att kompensera för, i förra testet (Test 2) förklarade, ojämnheter. Systemet blir slängt vilket främst ses då hastighetskurvan kommer ikapp hastighetsfunktionen. Det är dock inte till nackdel då denna översläng är anledningen till att bilarna ändå träffar varandra. Men det får anses som tur att bilarna här träffar varandra, cirka 15 centimeter ifrån tänkt träffpunkt. Som nämnt ovan är det inte tänkt att accelerera så kraftigt. Anledningen att detta test (Test 3) tagits med är att visa hur systemet beter sig på bristningsgränsen.

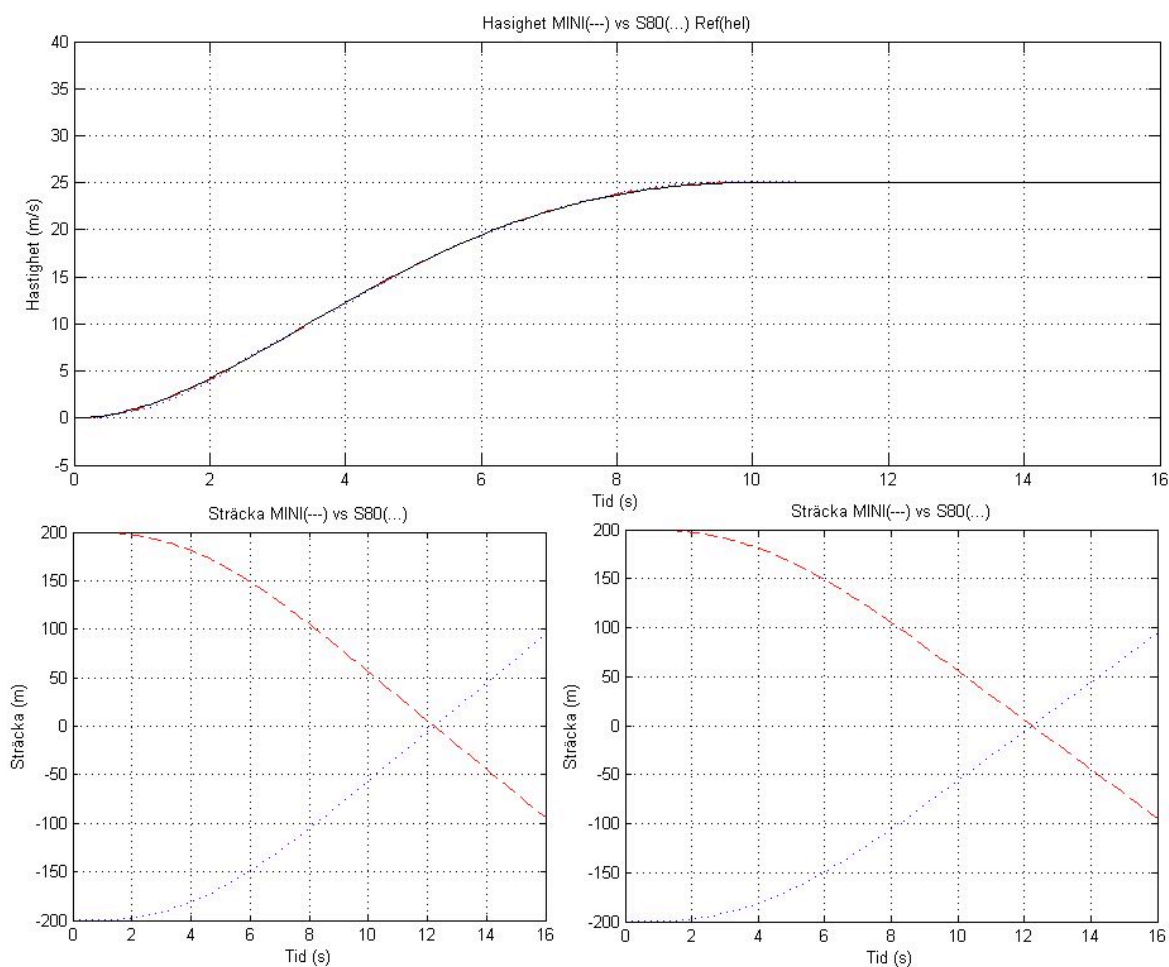
6.2.4 Test 4. 70 km/h vs 110 km/h



Figur: 70 km/h med sträckan 200 och tiden 10 sekunder mot 110 km/h med sträckan 200 meter och tiden 13 sekunder.

I detta test (Test 4) har tiden ändrats för S80'n för att ge den en mindre kraftig acceleration än i förra testet. I övrigt är det samma som i Test 3. Med tre sekunders längre accelerationstid fås en mycket mjukare kurva. Träffsäkerheten blir cirka 4 centimeter.

6.2.5 Test 5. 90 km/h vs 90 km/h



Figur: 90 km/h med sträckan 200 och tiden 10 sekunder.

I detta test (Test 5) körs bilarna, i likhet med test 1, med samma hastighet och som nu är höjd till 90 km/h istället för 70 km/h. För S80'n är det vid tiden 13,4914 sekunder och för MINI'n är det vid tiden 13,4907 sekunder. Detta ger en tidsdifferens på 0,0007 sekunder, vilket motsvarar 0,0175 meter i hastigheten 25 m/s. Träffsäkerheten kan alltså bestämmas till 2 centimeter för detta test.

6.2.6 Test 6. 90 km/h vs 110 km/h

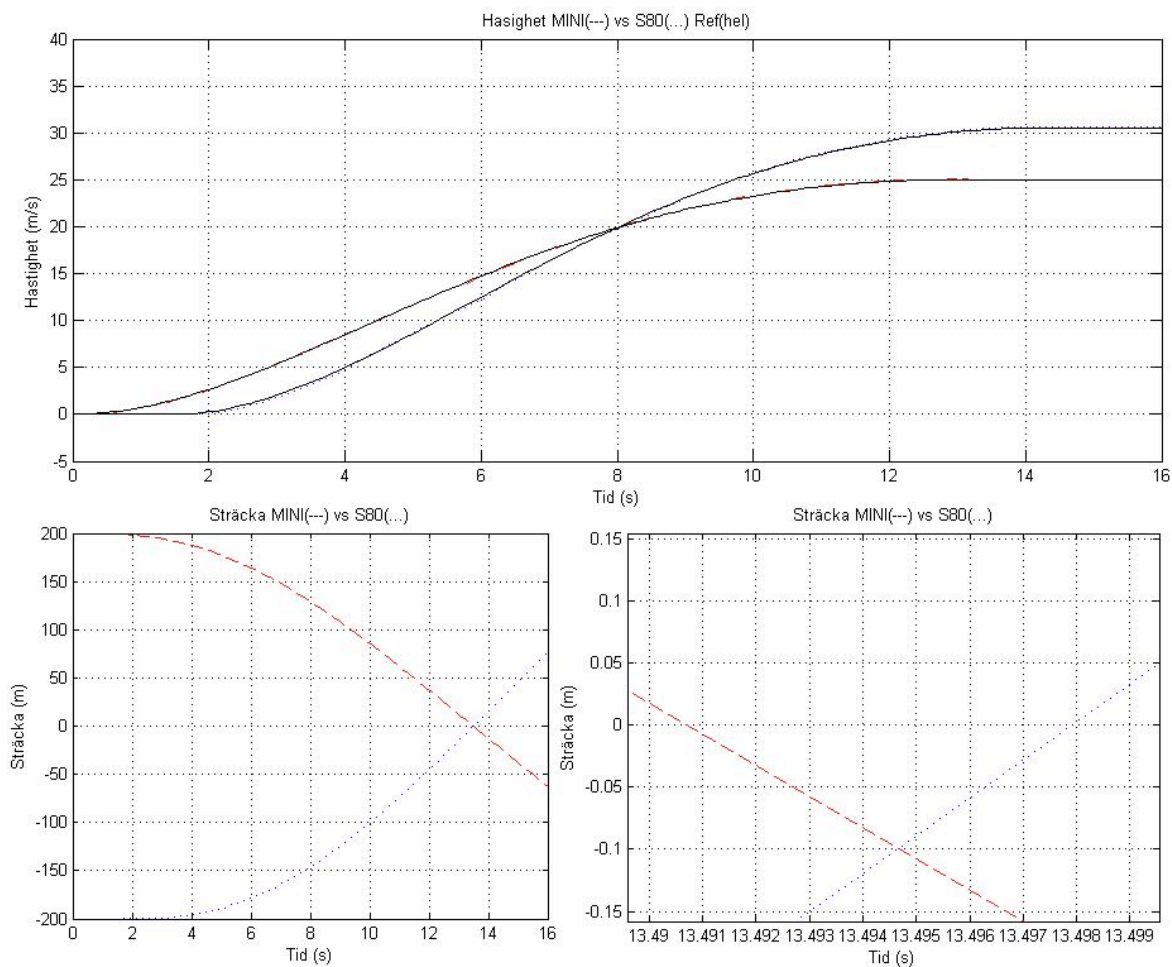
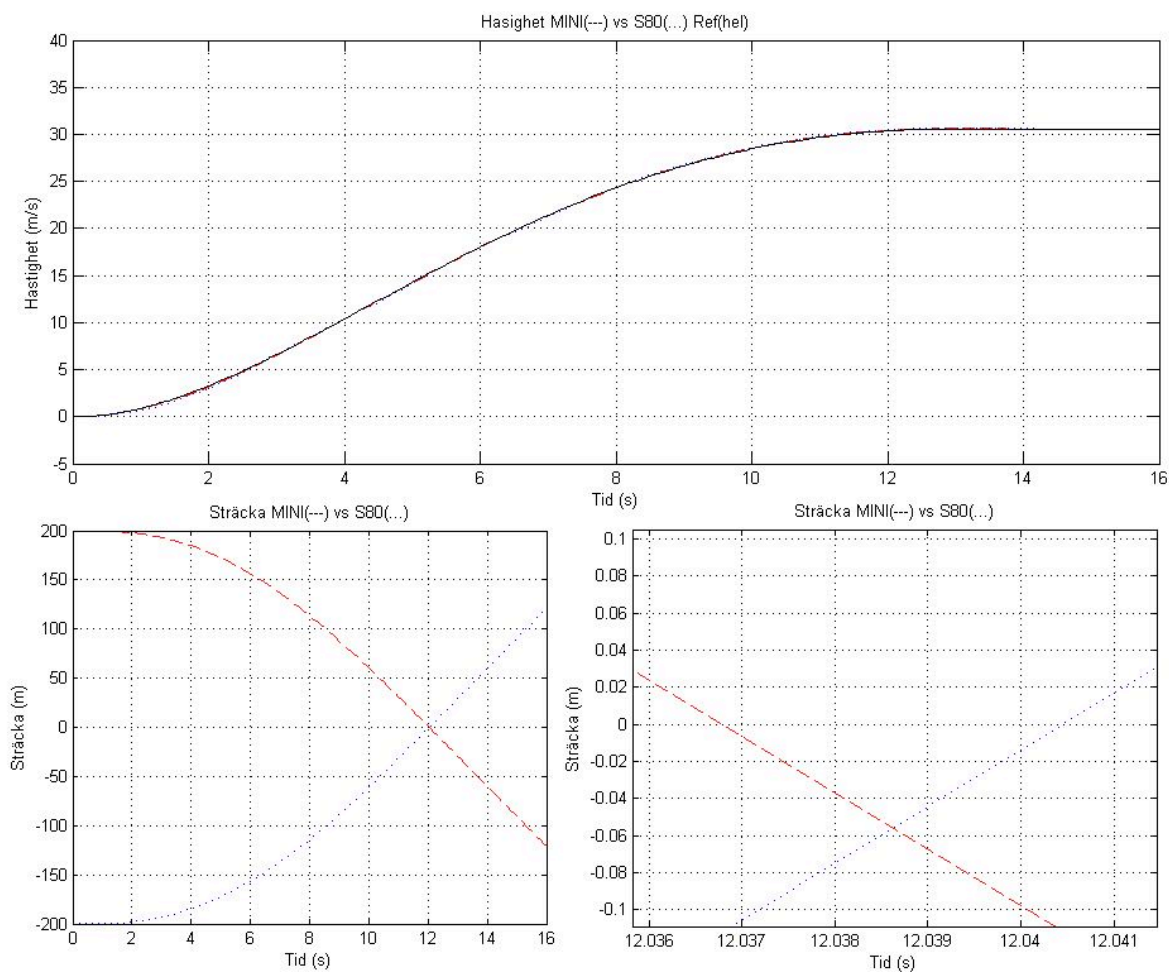


Bild: 90 km/h och 110km/h med sträckan 200 meter och tiden 13 sekunder.

I Test 6 som visas på ovanstående bild körs bilarna i 90 km/h och 110 km/h med 200 meters accelerationssträcka och tiden 13 sekunder. Man kan ana viss svajighet på S80'n som likt test 3 börjar närma sig gränsen för vad som är möjligt. Träffsäkerheten blir i detta test så dålig som 0,21 meter, det vill säga 21 centimeter. Detta beror på att kollisionen sker efter 13,495 sekunder och då har inte S80'n hunnit accelerera klart. Detta avhjälpas av att ge S80'n längre accelerationssträcka.

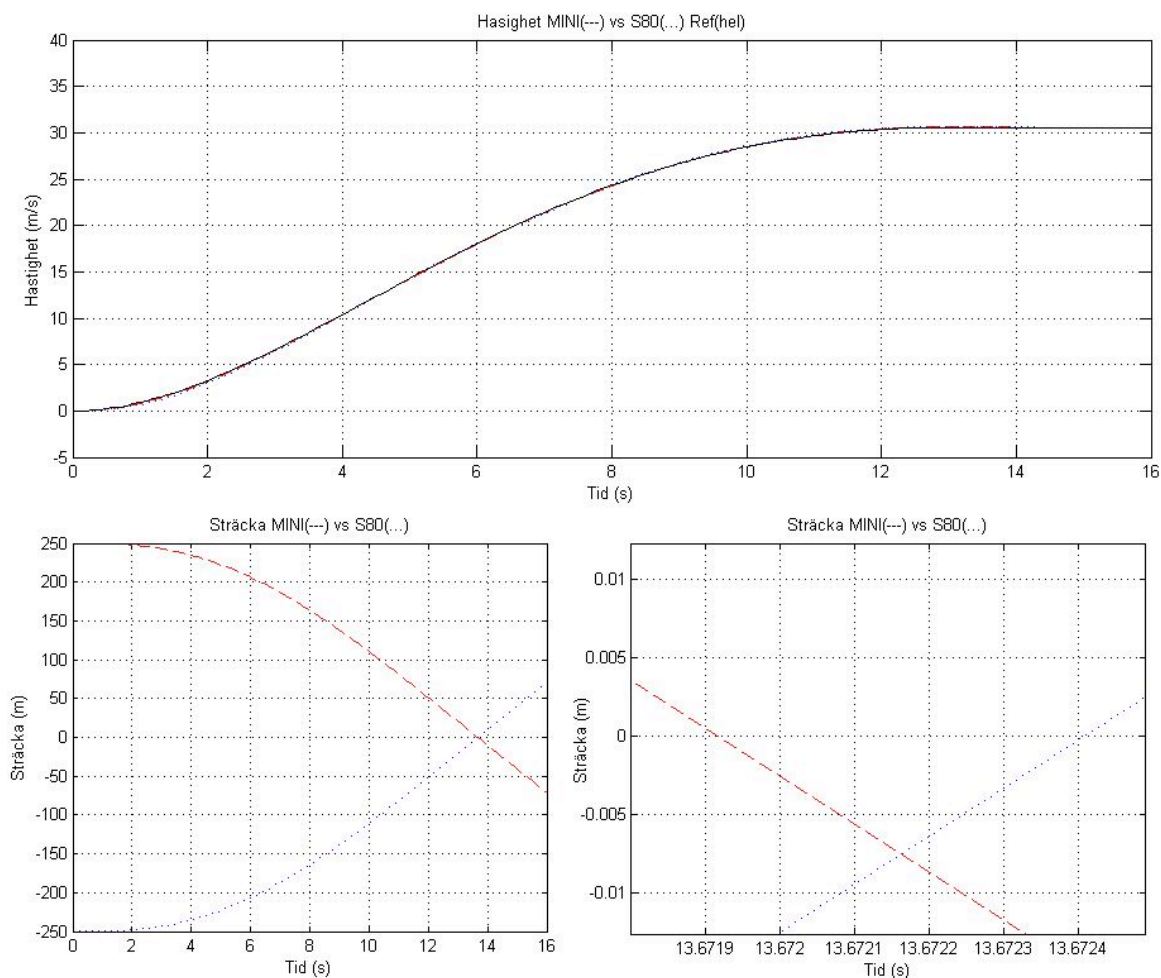
6.2.7 Test 7. 110 km/h vs 110 km/h



Figur: 110 km/h med sträckan 200 meter på 13 sekunder.

När båda bilarna i detta Test 7 accelererar till 110 km/h syns i figuren likt S80'n i Test 6 att man närmar sig det andra extremscenariot. Nämligen att sträckan blir för kort för att hinna upp i rätt hastighet. Här är det på gränsen att bilarna hinner upp i rätt hastighet. Eftersom de har samma hastighetsfunktion att följa blir träffsäkerheten inte lika dålig som i Test 6, dock har de ju inte uppnått den hastighet de ska hålla. Träffsäkerheten blir cirka 10 centimeter.

6.2.8 Test 8. 110 km/h vs 110 km/h med längre sträcka



Figur: 110 km/h med sträckan 250 meter på 13 sekunder.

När båda bilarna i detta Test 8 har 250 meter istället för 200 meter att accelerera till 110 km/h hinner de upp i rätt hastighet och träffsäkerheten blir låga 1,5 centimeter.

6.3 Intressanta värden

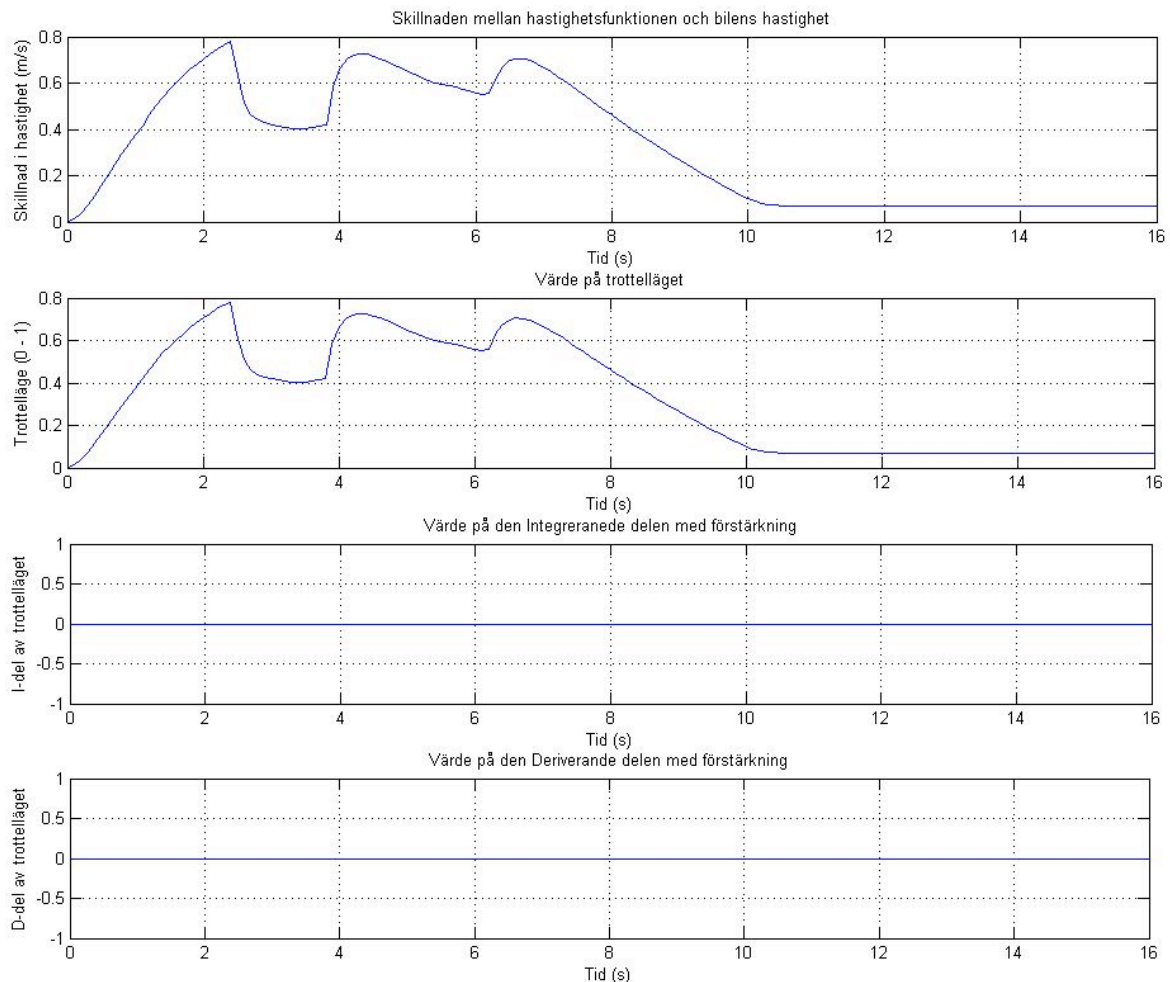
Efter körning plottas intressanta data för att studeras. Den intressantaste är tillryggalagd sträcka mot tid för de båda bilarna. Där denna kurva skär sträckan noll är då bilarna ska träffa varandra. Man eftersträvar att båda bilarnas kurvor skär varandra så nära sträckan noll som möjligt. I skärningen av sträckan noll tas tiden för de båda kurvorna och man läser av skillnaden och får då tidsdifferensen. Denna multipliceras med den hastighet i meter per sekund bilarna hade, eller den snabbaste bilens hastighet. Detta ger ett mått i meter på hur mycket bilarna missar varandras träffpunkter. I simuleringarna har detta avstånd blivit mellan 0,000 och 0,050 meter alltså max 5 centimeters differens. Vilket får anses som riktigt bra.

6.4 PID val och varianter

I styrningen till trottelläget finns möjlighet att använda PID reglering. Då bilen inte lyckas följa kurvan tillfredställande, med bara insignalen i den partiella delen, testas den integrerande och deriverande delen och lämpligt värde ställs in. Tillfredställande resultat får man genom att bara använda den integrerande delen och behöver då inte använda den deriverande delen då denna lätt ger upphov till svängningar i systemet. Speciellt i tester med högre hastigheter.

6.4.1 Proportionell styrning

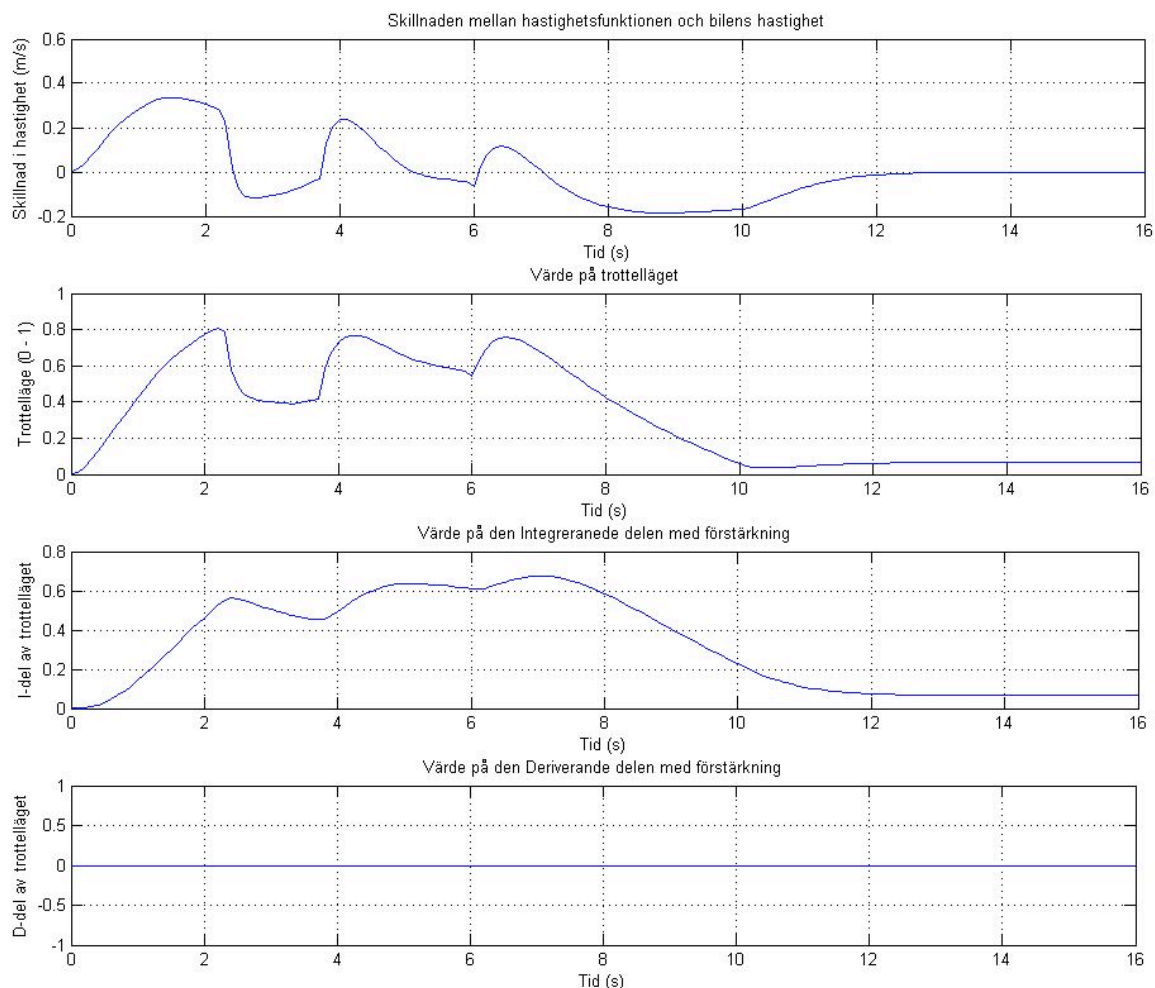
Då bara proportionell styrning av trotteln sker, är felet relativt stort. I figuren nedan visas först utsignal till trotteln, nedan det tillägg som den integrerande delen tillför. Nedan det visas den deriverande delens tillägg och underst visar felet mellan hastighet och önskad hastighet. Det vill säga skillnaden mellan hastighetsfunktionen och hastighetskurvan. Då bara proportionell styrning sker med värdet ett är både den integrerande och deriverande delen lika med noll. Styrningen till trotteln ska befinna sig mellan noll och ett vilket den gör. Därför kan man inte ha för stora värden på de olika reglerande delarna. Felet däremot vill man ha så litet som möjligt. Vid enbart proportionell styrning är felet uppe i 0.8 m/s enligt figuren.



Figur: Proportionell styrning.

6.4.2 Proportionell och integrerande styrning

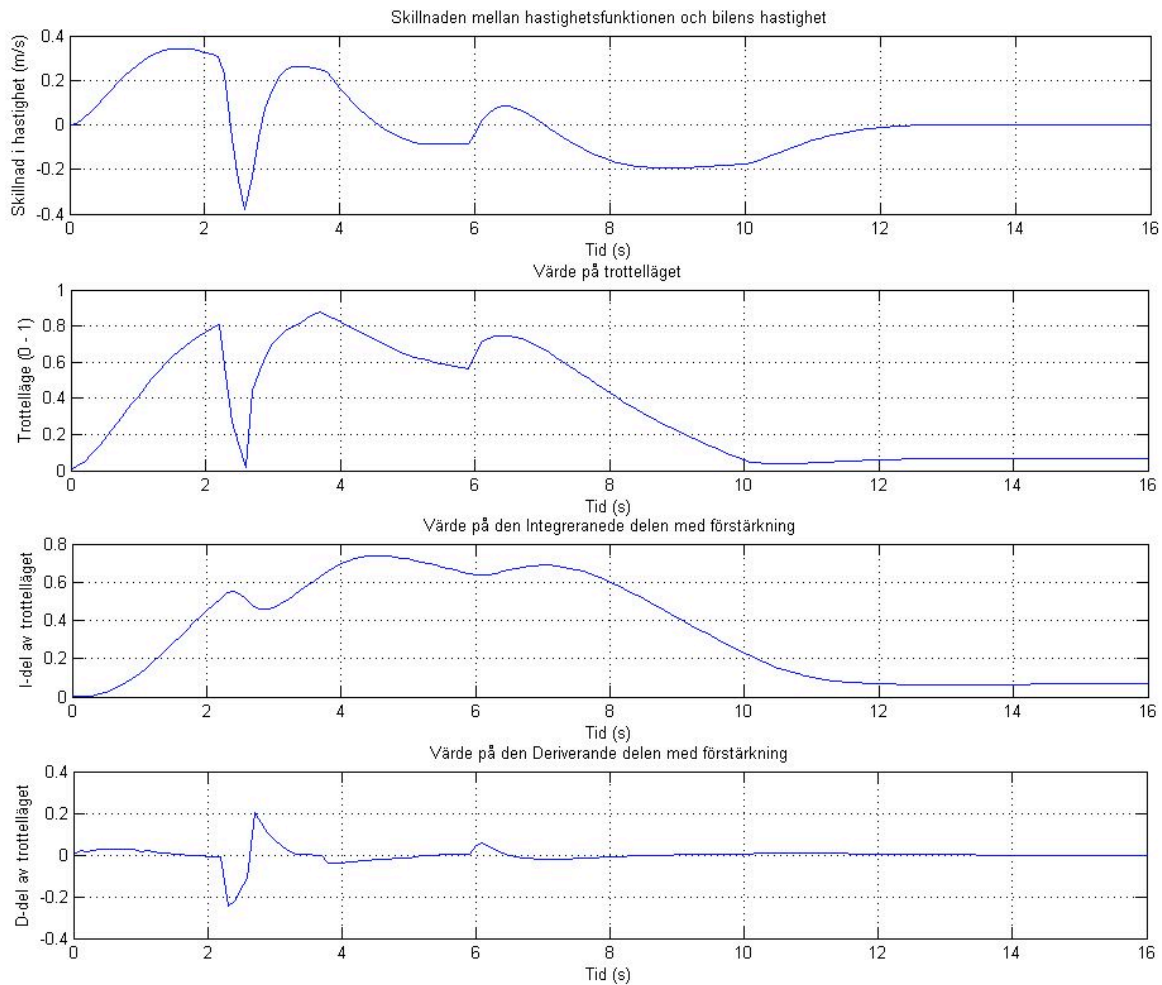
För att minska felet mellan hastighetskurvan och hastighetsfunktionen kan man lägga till den integrerade skillnaden på hastighetskurvan och hastighetsfunktionen och på så vis få en ändring i trottelsignalen som ger ett mindre fel då den integrerande delen motverkar den partiella delens överstyrning. Här ser man i figuren att felet blir bara så stort som 0.35 m/s och att kurvan är mjukare för trottelsignalen. Vilket den blir då det är ett mindre fel att kompensera för.



Figur med proportionell och integrerande styrning.

6.4.3 Proportionell, integrerande och deriverande styrning.

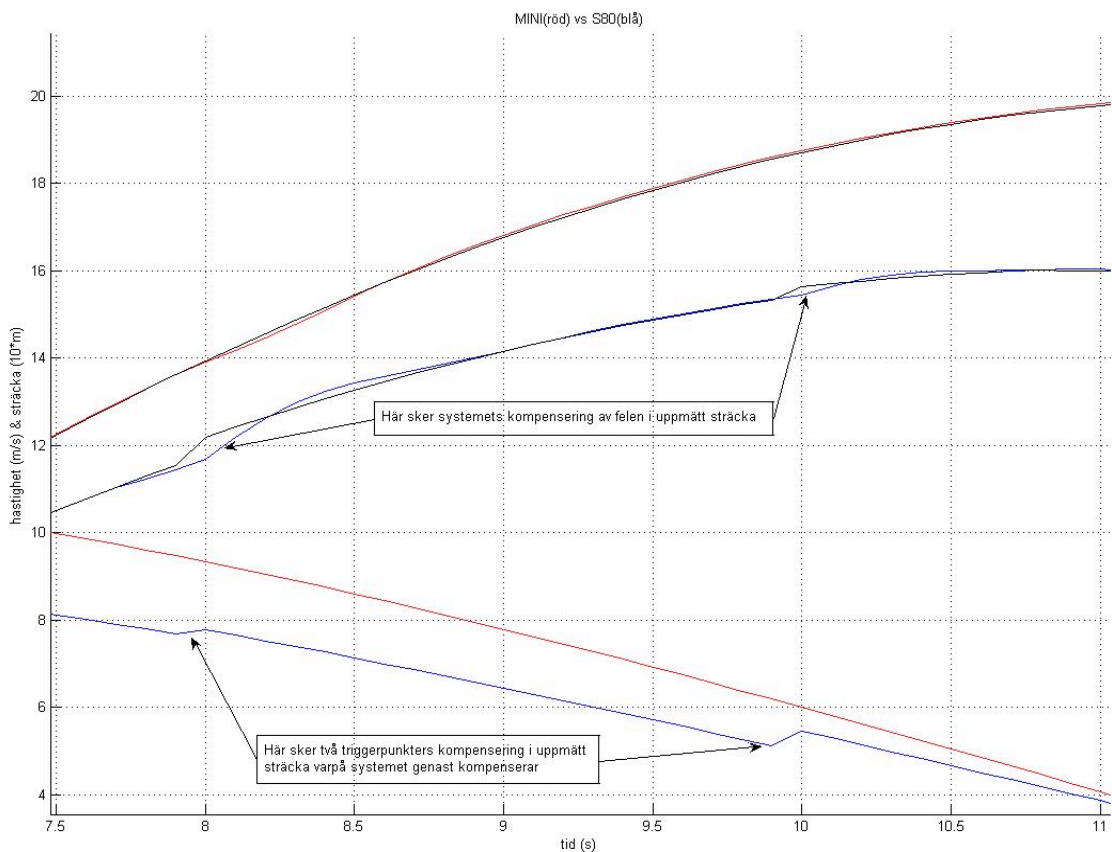
Lägger man på deriverande kompensation blir systemet som väntat slängigt och man får obetydlig minskning i felet. Dels beror svängigheten på att bilmodellernas vridmomentkurvor är spetsiga och att växlingarna sker snabbt. Detta ger de spikar man kan se i den deriverande delen i figuren. Taggigheten i början beror på de små värdena på hastighetskurvan och hastighetsfunktionen som ger de deriverade värdena större betydelse. Det kan man kompensera för genom att ha ett mindre eller inget värde alls på den deriverande delen i början av testet. Den deriverande delen i styrsystemet väljs bort då den utgör en risk och inte tillför eftersökta egenskaper, såsom snabbare felkompensering i systemet.



Figur: deriverande del inkluderad.

6.5 Test av triggerpunkter

Testet av triggerpunkter är konstruerat som sådant att man adderar ett värde till uppmätt sträcka vilket plötsligt ger ett steg i uppmätt sträcka. Detta följer systemet och kompenserar enligt figur relativt snabbt.



Figur: Triggerpunkter.

7. Slutsatser

Av ovanstående tester och resultat kan man konstatera att det är realistiskt att styra två bilar mot en gemensam tidsbestämd kollisionplats med ett i rapporten beskrivet styrsystem. Genom att vid varje test få tillfredställande värden på träffsäkerheten bevisas systemets flexibilitet och mångsidighet. Systemet tolererar ändringar i sträcka, hastighet och tid, förutsatt att det är inom rimliga gränser, utan att man behöver ändra annat än parametrar som är enkelt inställbara innan varje test.

8. Diskussion och felkällor

Ett upphov till fel i systemet är bilmodellernas enkelhet. Framför allt vridmomentkurvan som ger upphov till svårreglerade störningar. Även växlingarna som sker på en tiondels sekund utan rampning, vilket ger en vass kant, är svåra att reglera. De fel som är beroende av dessa felkällor är emellertid så små att de inte påverkar systemets resultat. Dessutom vet man av accelerationstester med personbil att bilens egen inre tröghet inte ger upphov till de vassa kanter som stör systemet. Se Bilaga 3, accelerationstest med personbil.

Det får anses som bra av systemet att klara av dessa svårigheter.

Ytterligare en felkälla som måste beaktas är avståndsmätningen. Denna måste vara mycket korrekt, annars fallerar hela systemet. Även då triggerpunktens placering måste vara mycket noggrant utmätt. Att tester som inte har stor träffsäkerhet har tagits med är för att visa systemet i dess Extremsituationer.

I beräkningarna finns det två källor till fel. Den första och med störst felpotential är där sträckan integreras fram i initialfunktionen. Numerisk integration i form av rektangeluppskattning, som används, får ett fel i varje steg. Detta ger ett fel på uppåt en meter vid hastigheten 90 km/h vilket är stort. Felet är dock i samma storlek för båda bilarna vilket gör att de ändå träffar varandra.

För felberäkning görs en integration med mycket liten steglängd. Genom att sätta in samma konstanter som initialfunktionen räknar fram, i Ekvation 6 med samma gränser som man valt i initialfunktionen. Dessa är, för hastigheten 90 km/h och tiden 100 millisekunder, $z=279$ och $k=98.6812$ vilket ger en sträcka på 112,289 meter.

$$\int_0^{100} \frac{n^{2-\frac{n}{279}}}{98.6012} dn = 112,289 \quad \text{Ekvation (8)}$$

Det som räknas fram i initialfunktionen blir 113,262 meter. Skillnaden blir med dessa förhållanden 0,974 meter. Detta avser bara accelerationsfasen eftersom Ekvation 6 bara är aktiv då och det inte blir ett fel i den numeriska integrationen om hastigheten är konstant. Därför görs integrationen för felberäkningen endast under de 10 sekunder bilarna accelererar. För att kunna räkna ut vilken steglängd som krävs för önskad felmarginal vid den numeriska integrationen jämförs över-och underintegrering enligt följande.

$$\left[h \sum_{i=1}^n v_i - h \sum_{i=0}^{n-1} v_i \right] \cdot h = \text{felet} \quad \text{Ekvation (9)}$$

Där h är steglängden och i och n är start respektive slutvärde. Felet vid den numeriska integrationen är alltså sista uträknade hastigheten, det vill säga sluthastigheten, gånger steglängden.

När hastigheterna är olika som i testet 70 km/h vs 110 km/h så ser man att träffsäkerheten är sämre och det beror just på att felet i den numeriska integrationen blir olika stora. Trots detta träffar bilarna varandra och satisfierar kraven på systemet. För framtida utvecklingar av systemet bör steglängden i den numeriska integrationen minskas eller så bör en annan typ av integration väljas.

Det andra felet är i matlabs ekvationslösare. I för rapporten gjorda tester har ekvationslösaren "ode45" använts med en relativ tolerans på 10^{-3} . För att testa vilket fel detta ger har en test gjorts med den relativa toleransen 10^{-4} vilket inte på femte decimalen, på tidpunkten för kollision, inte gjorde någon skillnad. Felet i "ode45" får anses försumbart för i rapporten redovisade tester.

9. Litteraturförteckningar

Glad T., Ljung L.: Reglerteknik. Grundläggande teori. Studentlitteratur 2006.
Användarhandledning för Matlab av Eva Pärt-Enander & Anders Sjögren 2001.
Course material Vehicular Systems av Lars Nielsen & Lars Eriksson 2003.
Avoding crashes artikel i Automotive Engineering aei-online.org
<http://www.ictct.org/workshops/93-Salzburg/Horst.pdf>
http://www.prevent-ip.org/en/about_preventive_safety/introduction/time_to_collision_02.htm

10. Referenser

[Ref A] <http://fifthgear.five.tv>
[Ref B] <http://www.auto-motor-und-sport.de>
[Ref C] <http://www.westermo.com> RM-240
[Ref D] <http://www.micro-epsilo.com> Laseroptical distance and position measuring.
[Ref E] <http://leinelinde.dk> RSI 503
[Ref F] <http://www.vericomcomputers.com/> VC-2000

Bilaga 1. Initialfunktionen

Nedan följer koden för initialfunktionen som körs innan testet startas.

Först en variabel och konstantförteckning, input för det specifika testet samt uträkning av hastighetsfunktion, tidsåtgång för bilarnas färd mot krockpunkten

```
% PROGRAM FÖR ATT SKAPA EN ACC - KURVA
% Variabel och konstantförteckning
% speed = hastighet (m/s) med prefix för de olika bilarna
% Cdist = avstånd (decimeter) med prefix för de olika bilarna
% acct = accelerationstid (0.1 sek) med prefix för de olika bilarna
% road = avstånd med odefinierad enhet
% time = tid (0.1 sek)
% k1 = konstant för att ställa in rätt tid för funktionen med prefix för
%     de olika bilarna
% k2 = konstant för att ställa in rätt hastighet för funktionen med
%     prefix för de olika bilarna
% x, z = variabler för loopar
% n = tidsvariebel för loop (0.1 sek)
% stopbit, qq = kontrollvariabler för att avbryta en loop.
% bb = slavvariabel för att lagra värden till nästa upprepning av
%     loop
% k, m = variabler för linjeekvation
% CC, Ct = tider (0.1 sek) för att bestämma bilarnas startfördröjning
close all
clear all
```



```

% S L U T H A S T I G H E T
    speedM=25; % Här matas önskad data in för testet
    speedS=25;
% S T R Ä C K A
    CdistM=2000;
    CdistS=2000;
% A C C T I D
    acctM=100;
    acctS=100;

% U T R Ä K N I N G   A V   " k 1 "   K O N S T A N T E N
stopbitM=1; % Stopbit för att avbryta testet avaktiveras
for k1M=1:1:10000 % Loop startas för att hitta max
    if stopbitM==0
        break
    end
    for nM=1:1:acctM+2 % Loop för att kolla om det finns ett max
        roadMq(nM)= (nM^(2-(nM/k1M))); %där man befinner sig i huvudloopen
        timeMq(nM)=nM;
    end

    for xM=1:1:length(timeMq)
        if max(roadMq)==roadMq(xM) && timeMq(xM)==acctM && stopbitM==1;
            setK1M=k1M; % Här sätts "setK1" till "k1"
            k1M=1000; % då kriterierna är uppfyllda
            stopbitM=0;
            break
        end
    end
    if stopbitM==0
        break
    end
end

stopbitS=1; % Samma sak görs för den andra bilen
for k1S=1:1:10000
    if stopbitS==0
        break
    end
    for nS=1:1:acctS+2
        roadS(nS)= (nS^(2-(nS/k1S)));
        timesS(nS)=nS;
    end

    for xS=1:1:length(timesS)
        if max(roadS)==roadS(xS) && timesS(xS)==acctS && stopbitS==1;
            setK1S=k1S;
            k1S=1000;
            stopbitS=0;
            break
        end
    end
    if stopbitS==0
        break
    end
end

% B E R Ä K N I N G   A V   " k 2 "   V A R I A B E L N
k2M=max(roadMq)/speedM; % Här setts "k2" till rätt värde
k2S=max(roadS)/speedS; % Här setts "k2" till rätt värde
% B E R Ä K N I N G   A V   S J Ä L V A   K U R V A N
% S A M T   B E R Ä K N I N G   A V   L I N J E E K V A T I O N E N

```

```

qq=1;
bb=0;
for zM=1:1:200
    roadM(zM) = (zM^(2-(zM/setK1M)))/k2M;
    timeM(zM)=zM;
    if zM>acctM
        roadM(zM)=speedM;
    end

    distM(zM)=bb+roadM(zM);
    if distM(zM)>=CdistM && qq==1
        qq=0;
        k6=distM(zM)-bb;
        m=bb-k6*(zM-1);
        CtM=(CdistM-m)/k6*0.1;
    end
    bb=distM(zM);
end

qq=1; % samma som ovan för den andra bilen
bb=0;
for zS=1:1:200
    roadS(zS) = (zS^(2-(zS/setK1S)))/k2S;
    timeS(zS)=zS;
    if zS>acctS
        roadS(zS)=speedS;
    end

    distS(zS)=bb+roadS(zS);

    if distS(zS)>=CdistS && qq==1
        qq=0;
        k=distS(zS)-bb;
        m=bb-k*(zS-1);
        CtS=(CdistS-m)/k*0.1;
    end
    bb=distS(zS);
end
CC=CtM-CtS; % Kontroll vilken bil som har längst transporttid
if CC==0 % samt uträkning hur länge de ska vänta i starten
    CtM=0;
    CtS=0;
end
if CC>0
    CtM=0;
    CtS=CC;
end
if CC<0
    CtM=-CC;
    CtS=0;
end

```

Bilaga 2. Bildynamik

I denna bilaga beskrivs den funktion som styr bilens dynamik

```
% FUNKTION FÖR BILDYNAMIK MINI COOPER
% Variabel och konstantförteckning
% v = bilens hastighet (m/s)
% u = trottelläge (0-1)
% gear = växel
% theta = marklutning (grader)
% m = bilens massa (kg)
% omega = motorns varvtal (rad/s)
% rpm = bilens motorvarv omräknat till (rpm)
% torque = motorns vridmoment (Nm)
% dv = bilens acceleration (m/s^2)

function x = minidyn(v, u, gear, theta, m)

% Definition av parameterar
alpha = [40, 25, 18, 14, 12]; % utväxlingar för de olika växlarna
Cr = 0.01; % Rullfriktionskoefficient
rho = 1.3; % luftdensitet, kg/m^3
Cd = 0.32; % drag koefficient
A = 2.4; % Bilens frontarea, m^2
g = 9.8; % gravitations konstant

% Begränsar trottelläget till mellan 0 och 1
u = min(u, 1); u = max(0, u);

% Räknar ut motorns varvtal från bilens hastighet
omega = alpha(gear) * v;

% En momentkurva ritas upp av linjeekvationer för aktuella varvtal
if omega>=1 && omega<157
    torque=1.52*omega;
elseif omega>=157 && omega<523
    torque=240;
elseif omega>=523
    torque=240+(523-omega);
else
    torque=40;
end

if v<5
    torque=200;
end
F = alpha(gear) * torque * u;

% Kontroll av varvtalet
if (omega > 900)
    fprintf(2, 'Motorvarvtalet för högt (%g)\n', omega*30/pi);
end
% Kontroll så varvtalet ej är negativt
if (torque < 0)
    fprintf(2, 'Motorvarvtalet är negativt (u=%g, omega=%g)\n', u, omega);
```

```

end

% Beräkning av Bromsande krafter
Fr = m * g * Cr;           % Rullmotstånd
Fa = 0.5 * rho * Cd * A * v^2; % Aerodynamiskt motstånd
Fg = m * g * sin(theta);  % Motstånd av vägens lutning
Fd = Fr + Fa + Fg;        % Summa bromsande krafter
if v<=1 % Styrning för att bilen inte ska rulla bakåt innan start
    Fd=0;
end

% Beräkning av accelerationen

dv = (F - Fd) / m;
rpm = omega*60/6.28;
x = [dv, rpm, torque]; % returvärden från funktionen

```

Bilaga 3. Accelerationstest med personbil

Accelerationstest som gjorts med bil med manuell växellåda. Accelerationen har mätts med systemet VC-2000 från Vericom [Ref F]. Pucklarna på hastighetskurvan är då man växlat, och den svängiga kurvan är accelerationen. Den lätt böjda kurvan är sträckan man färdats. Anledningen att denna bilaga finns med är för att få förståelse för hur det ser ut i verkligheten under en acceleration med personbil.

