



CHALMERS

Skalmodell av elvägssystem

Konstruktion och design av skalmodell
Kandidatarbete inom system- och reglerteknik

Emil Carlsson, Leo Eliasson, Omar Fawal,
Ebba Lindvall, August Ljungqvist, Daniel Rydberg

INSTITUTIONEN FÖR ELEKTROTEKNIK

CHALMERS TEKNISKA HÖGSKOLA
Göteborg, Sverige 2026
www.chalmers.se

Scale model of electric road system

Construction and design of scale model

Abstract

The transition to electric vehicles increases the demands on power distribution and charging infrastructure. One approach to enable extended driving range is dynamic charging through electrified roads. In this project, a physical scale model of an electric road system is developed to study charging during motion. The project includes the design and construction of vehicle models, modification and electrification of a slot-car track, and implementation of a modular energy storage system based on supercapacitors. In addition, a speed control system, sensors and real-time data visualization were implemented to enable continuous monitoring and analysis of the system behaviour. The physical model is complemented by a simulation model used to analyze and predict system behaviour under different design choices and operating conditions. Experimental data from the physical model is used to validate the simulation. The results show that the developed system successfully enables charging during motion and allows systematic studies of how speed, charging distance and energy storage capacity affect vehicle operation. Increasing the supercapacitor capacity resulted in significantly longer driving distances, while charging in curves proved to give less velocity in curves, which gave the vehicle more time to charge for the same length. The simulation model was able to reproduce the overall behaviour of the physical system and provided a useful tool for qualitative analysis of the system dynamics.

Skalmodell av elvägssystem

Konstruktion och design av skalmodell

Sammandrag

Elektrifieringen av transportsektorn ökar kraven på kraftdistribution och laddinfrastruktur. Ett sätt att möjliggöra längre räckvidd för elfordon är dynamisk laddning via elektrifierade vägar. I detta projekt utvecklas en fysisk skalmodell av ett elvägssystem för att studera laddning under färd. Projektet omfattar design och konstruktion av fordonsmodeller, modifiering och elektrifiering av en bilbana samt implementering av ett modulärt energilager baserat på superkondensatorer. Utöver detta implementerades ett hastighetsregleringssystem, sensorer och realtidsvisualisering av data för att möjliggöra kontinuerlig övervakning och analys av systemets beteende. Den fysiska modellen kompletteras med en simuleringsmodell som används för att analysera och förutsäga systemets beteende under olika konstruktionsval och driftfall. Experimentella mätdata från den fysiska modellen används för att validera simuleringen. Resultaten visar att det utvecklade systemet framgångsrikt möjliggör laddning under färd och skapar förutsättningar för systematiska studier av hur hastighet, laddningssträcka och energilagringsskapacitet påverkar fordonets drift. En ökning av superkondensatorns kapacitans resulterade i betydligt längre körsträckor, samtidigt som laddning i kurvor visade sig ge lägre hastigheter på laddsträckan, vilket i sin tur gav mer laddning per längdenhet. Simuleringsmodellen kunde återskapa det övergripande beteendet hos det fysiska systemet och visade sig vara ett användbart verktyg för kvalitativ analys av systemets dynamik.

Innehåll

1	Inledning	1
1.1	Syfte	2
1.2	Krav på systemet	2
1.2.1	Elektrifiering av vägbana	2
1.2.2	Laddning och lagring av energi	3
1.2.3	Hastighetsreglering	3
1.2.4	Design och konstruktion av fordon	3
1.2.5	Visualisera data	3
1.2.6	Simuleringsmodell	3
1.3	Avgränsningar	3
2	Systembeskrivning	5
2.1	Flödesmodell	5
2.2	Systembeskrivning mjukvara	6
2.2.1	Mjukvara i bilenheten	6
2.2.2	Mjukvara i centralenheten	7
2.3	Matematisk modell av systemet	7
2.3.1	Energitillförsel från buck-convertern	8
2.3.2	Strömfördelning	8
2.3.3	Kondensatordynamik	9
2.3.4	Motordynamik	10
2.3.5	Mekanisk kraftöverföring	10
2.3.6	Fordonsdynamik	10

3	Metod	12
3.1	Konstruktion och implementation av hårdvara	12
3.1.1	Mekaniska komponenter	12
3.1.2	Konstruktion och implementation av fordon	14
3.1.3	Implementering av energilagring	16
3.1.4	Reglering av fordonens hastighet	16
3.1.5	Elektrifiera vägen	17
3.1.6	Visualisering av data	18
3.2	Elektrisk koppling	19
3.2.1	Centralenheten och mikrokontroller	19
3.2.2	Bilenheten	20
3.3	Modifiering av simuleringsmodell	22
3.3.1	Strömupptagning	23
3.3.2	Energilager	23
3.3.3	Hastighetsreglering	24
3.3.4	Drivlinan	24
3.3.5	Vägdynamik	25
3.4	Validering	26
4	Resultat	27
4.1	Test 1 - Fordonens estimerade körsträcka - 5 F	27
4.2	Test 2 - Hastighet och körsträcka - 5 F	28
4.3	Test 3 - Varierande laddningslängd raksträcka - 5 F	29
4.4	Test 4 -Varierande laddningslängd kurva - 5 F	30

4.5	Test 5 - Hastighet och körsträcka - 2.35 F	31
4.6	Test 6 - Varierande laddningslängd raksträcka - 2.35 F	31
4.7	Test 7 - Varierande laddningslängd kurva - 2.35 F	32
4.8	Energilagring och drift	32
4.9	Simuleringsresultat	34
4.9.1	Simuleringstest 1 - ingen laddning från elväg	34
4.9.2	Simuleringstest 2 - laddningssträcka 35.2 cm	35
4.9.3	Simuleringstest 3 - laddningssträcka 52 cm	36
5	Diskussion	38
5.1	Analys av resultat	38
5.1.1	Estimerad körsträcka	38
5.1.2	Hastighetens påverkan på körsträcka	38
5.1.3	Laddningssträckans påverkan	40
5.1.4	Analys av superkondensatorns storlek	41
5.2	Simuleringsmodell	42
5.3	Kravuppfyllnad	43
5.3.1	Elektrifiering av vägbanan	43
5.3.2	Laddning och lagring av energi	43
5.3.3	Hastighetsreglering	44
5.3.4	Design och konstruktion av fordon	44
5.3.5	Visualisering av data	44
5.3.6	Simuleringsmodell	44
5.4	Förbättringar och framtida arbete	45

5.4.1	Segmentera elektrifieringen av vägbanan	45
5.4.2	Implementera höjdskillnader	45
5.4.3	Data för motor och friktion	45
5.4.4	Hastighetsreglering	46
6	Slutsats	47
	Referenser	48
A	Tester	49
A.1	Test 1	49
A.2	Test 2	49
A.3	Test 5	50
A.4	Test 6	50
B	Kodbas	50
B.1	Bilenheten	50
B.2	Centralenheten	56
B.3	Python skript för uträkning av avvikelse	61
B.4	MATLAB-kod för simuleringsmodell	62
B.5	Simuleringsmodellens parametrar	67
B.6	Simulink modell	67
C	Bill of materials	69

Nomenklatur

Symboler

ω	Vinkelhastighet [rad/s]
Ψ	Flödeskoppling [Wb]
C	Kapacitans [F]
e	Felvärde i regulatoren [m/s]
F	Kraft [N]
$F_{frikktion}$	Friktionskraft [N]
F_{netto}	Nettokraft som driver fordonet framåt [N]
g_r	Utväxlingsförhållande [-]
i_C	Ström till kondensatorn [A]
i_m	Motorström [A]
i_{buck}	Ström från buck-convertern [A]
i_{ers}	Ström från elvägen [A]
K_P, K_I, K_D	Förstärkningsparametrar för PID-regulatoren [-]
L_{dcm}	Motorns induktans [H]
M_v	Fordonets massa [kg]
Q	Laddning [C]
R_d	Resistans i motorkretsen [Ω]
R_w	Hjulradie [m]
R_{br}	Kontakmotstånd mellan borstar och bana [Ω]
R_{ch}	Laddningsresistans (kablage och anslutningar) [Ω]
R_{dcm}	Motorns inre resistans [Ω]
T	Vridmoment [Nm]
$u(t)$	Styrsignal från PID-regulatoren [V]
u_c	Spänning över kondensatorn [V]
v	Fordonets hastighet [m/s]

V_d	Diodspänningsfall [V]
V_{buck}	Utgångsspänning från buck-convertern [V]
V_{ers}	Matningsspänning från elvägssystemet [V]
v_{in}	Ingångsspänning till kondensatorn [V]
V_{PID}	Spänningsfall över PID-regulatorn och MOSFET [V]
W	Lagrad energi [J]
x	Fordonets position [m]

Termer

Anti-windup - Funktion i regulatorn som förhindrar att integraltermen växer okontrollerat

Back-EMF - Motspänning genererad i motorn, proportionell mot vinkelhastigheten

Borstar - Elektriska kontaktpunkter mellan fordon och bilbana som möjliggör strömöverföring

Buck-converter - Spänningsomvandlare som stegar ner spänningen till en lägre nivå

CAD - Datorstödd konstruktion (Computer-Aided Design), programvara för 3D-modellering

ESP-NOW - Trådlöst kommunikationsprotokoll för låg-latens datautbyte mellan mikrokontrollers

Hall-effektsensor - Sensor som registrerar magnetfält och används för att mäta hjulrotation

IR-sensor - Infraröd sensor som används för att detektera fordonets position på banan

Konduktiv överföring - Energiöverföring genom direkt elektrisk kontakt mellan bana och fordon

Laddningssträcka - Den del av banan där fordonet tillförs elektrisk energi

Mikrokontroller - Programmerbar enhet som används för styrning, reglering och datainsamling

MOSFET - Transistor som används för elektronisk växling och styrning av ström med hjälp av en spänning på gaten

OLED-display - Display som används för visualisering av mätdata i realtid

PID-regulator - Regulator som beräknar en styrsignal baserat på proportionell, integrerande och deriverande verkan av felvärdet

PWM - Pulsbreddsmodulering, metod för att styra motorns tillförda effekt

Referensvärde - Förinställt målvärde som regulatorn försöker uppnå

Reglerloop - Styrssystem där ett uppmätt värde jämförs med ett referensvärde

Relä - Elektromekanisk strömbrytare som styrs av en svagare signal för att koppla in eller bryta en starkare krets

Scalextric - System för elektriska bilbanor och tillhörande fordon

Simulink - Programvara som används för modellering och simulering av dynamiska system

Spökvärde - Ett mätvärde eller en observation som avviker markant från övriga värden i en datamängd och därför kan indikera mätfel, störning i systemet

Superkondensator - Energilager med hög effekttäthet som kan laddas och urladdas snabbt

Vin-stift - Stift på mikrokontroller där spänning matas in

1 Inledning

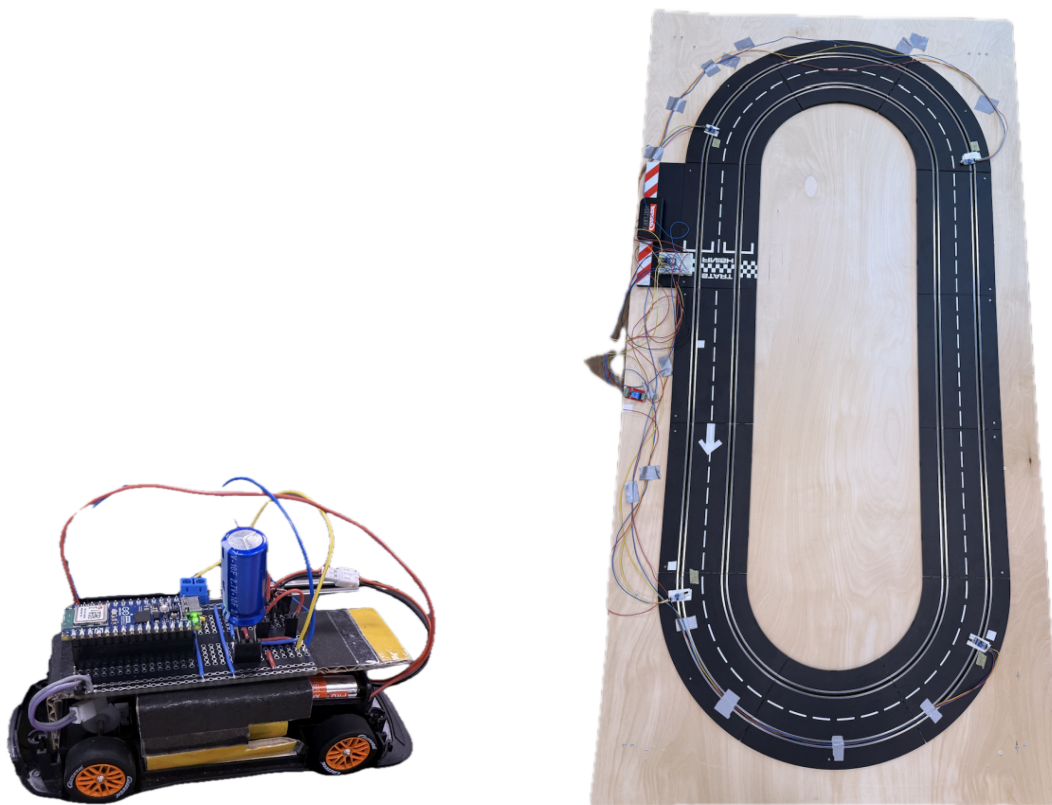
En av vår tids största utmaningar är klimatförändringarna, där transportsektorn står för en betydande andel av de globala utsläppen, uppskattningsvis omkring 23 % av de energirelaterade koldioxidutsläppen, enligt IPCC [1]. Till följd av detta har elektrifieringen av transportsektorn accelererat under de senaste åren. IEA säger att globalt utgjorde elbilar cirka 20 % av nybilsförsäljningen år 2024 [2], med tydliga regionala variationer. I Kina uppgick andelen till omkring 50 %, medan motsvarande siffror var cirka 20 % i Europa och 10 % i USA. Vissa länder har nått ännu högre nivåer av elektrifiering, där Norge utmärker sig med att elektriska fordon utgjorde cirka 88 % av nybilsförsäljningen. Trots denna snabba utveckling kvarstår flera utmaningar kopplade till elfordon. Räckvidden för elektriska fordon är begränsad och beror i stor utsträckning på batteriets kapacitet. Vidare kräver laddning av batterier relativt lång tid jämfört med tankning av konventionella bränslen. Därtill medför produktionen av batterier en betydande klimatpåverkan.

Ett område som har fått ökat intresse i arbetet med att möta dessa utmaningar är elvägssystem, där fordon kan laddas under färd. Elvägssystem är en typ av infrastruktur som möjliggör kontinuerlig energitillförsel till fordon under körning. I konduktiva system sker energiöverföringen via exempelvis en elektrifierad skena i vägbanan, vilket möjliggör kontinuerlig laddning av fordonet under färd. Detta kan minska behovet av stora batterier, vilket i sin tur reducerar både fordonets vikt och den miljöpåverkan som uppstår vid batteriproduktion. Dessutom minskar behovet av stillastående laddning, då energi kan tillföras under körning [3].

För att kunna utvärdera olika elvägskoncept utan att genomföra omfattande investeringar i fullskalig infrastruktur, kan experimentella studier i mindre skala utföras. Detta i kombination med matematiska simuleringar. Dessa metoder möjliggör analys av elvägstekniken och är särskilt värdefulla i utbildningssyfte. Vid ett tidigare projektarbete vid Chalmers tekniska högskola har en skalmodell av ett elvägssystem utvecklats, men denna har uppvisat flera begränsningar [4].

Exempelvis saknades kontinuerlig hastighetsmätning och därmed också reglering av fordonets hastighet. Detta trots att hastigheten har en betydande inverkan på laddningsdynamiken för fordonet, vilket begränsar möjligheten till en mer djupgående analys av systemets beteende. Även datainsamlingen var begränsad, vilket ytterligare försvårade analysen. Därtill saknades ett integrerat chassi, vilket innebär att komponenter var exponerade och mindre mekaniskt stabila.

Den tidigare konstruktionen visas i Figur 1, där bilmodellen och elvägsmodellen illustreras.



Figur 1: Föregående års bilmodell (vänster) och elvägsmodell (höger) [4]

1.1 Syfte

Syftet med projektet är att vidareutveckla en befintlig skalmodell av ett elvägs-system till en mer robust och flexibel testplattform. Plattformen ska möjliggöra systematiska studier av laddning under färd, där parametrar som hastighet, laddningssträcka och energilagringkapacitet kan varieras och analyseras.

1.2 Krav på systemet

För att säkerställa att systemet uppfyller projektets syfte formulerades krav som avgör om projektet anses lyckat.

1.2.1 Elektrifiering av vägbana

Laddningssträckans längd och placering ska kunna varieras för att möjliggöra studier av hur dessa parametrar påverkar systemets beteende. Systemet ska möjliggöra

effektöverföring till fordonet när det befinner sig inom en elektrifierad sektion.

1.2.2 Laddning och lagring av energi

Fordonet ska kunna laddas under färd. Energilagrets kapacitans ska kunna varieras mellan minst två nivåer för att möjliggöra studier av hur lagringskapaciteten påverkar fordonets räckvidd.

1.2.3 Hastighetsreglering

En hastighetsregulator ska implementeras för att fordonet ska kunna köras i en given referenshastighet. Detta med en genomsnittlig felmarginal som ej överstiger 15 %.

1.2.4 Design och konstruktion av fordon

Fordonen ska konstrueras i olika storlekar för att möjliggöra jämförelse av hur fordonsmassa påverkar energiförbrukning och körsträcka.

1.2.5 Visualisera data

Fordonets tillstånd ska kunna observeras i realtid via en display. På displayen ska det kunna utläsas energilagrets spänning samt fordonens hastighet.

1.2.6 Simuleringsmodell

Den föregående simuleringsmodellen [4] ska vidareutvecklas genom att en hastighetsregulator ska implementeras. Modellen ska kunna reproducera skalmodellens övergripande beteende.

1.3 Avgränsningar

För att begränsa arbetets omfattning och säkerställa projektets genomförbarhet införs ett antal avgränsningar.

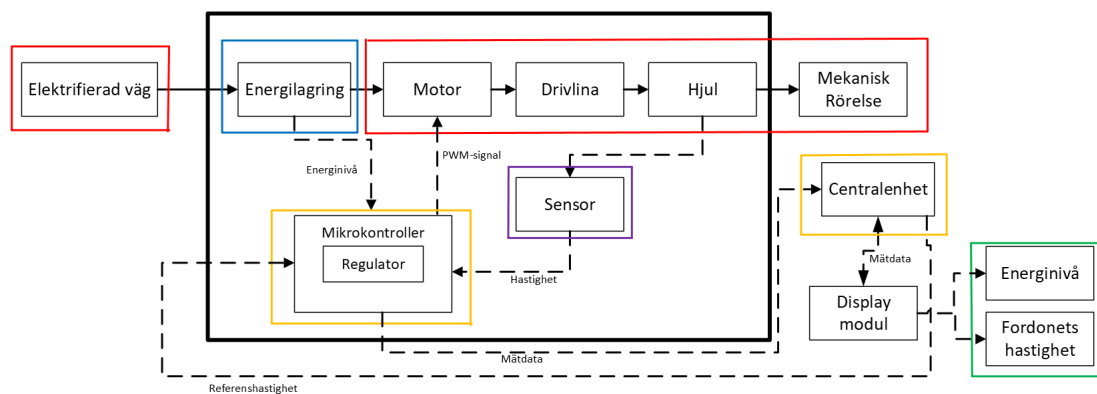
- Den väg som används antas vara plan, vilket innebär att höjdskillnader och lutningar inte beaktas i projektets analyser. Vidare tas ingen hänsyn till yttre miljöfaktorer såsom vind, regn eller andra väderförhållanden. Ingen hänsyn tas heller till luftmotstånd.
- Projektet utgår från ideal energitillgång, vilket innebär att begränsningar i energiförsörjning, effektöverföring eller elnätets kapacitet inte analyseras.
- Projektet kommer att utgå från en redan befintlig simuleringsmodell från föregående års kandidatarbete [4]. Ingen ny modell skapas utan den gamla anpassas till de nya förutsättningarna.

2 Systembeskrivning

Detta kapitel beskriver den vidareutvecklade skalmodellen av elvägssystemet samt hur dess olika delsystem samverkar. Syftet är att ge en sammanhängande bild av systemets uppbyggnad och funktion, inklusive energiflöde, signalflöde och mekanisk rörelse. Inledningsvis presenteras systemets övergripande i en flödesmodell, följt av systembeskrivning av mjukvara och slutligen matematiska beräkningar som beskriver sambanden i systemet.

2.1 Flödesmodell

Systemets ingående delar och deras inbördes samband illustreras i flödesmodellen i Figur 2. I flödesmodellen visualiseras energiflöden med helmarkerade pilar medan informationsflöden visualiseras med streckade pilar. De röda delarna representerar systemets mekaniska komponenter, medan blått illustrerar energilagringen. Gult illustrerar mikrokontrollern och centralenheten, lila representerar sensorerna och grönt är den data som visualiseras. Den heldragna svarta rutan representerar systemgränserna för fordonet.



Figur 2: Flödesmodell för elvägssystemet

Systemet baseras på en elektrifierad väg som tillför energi till fordonen under färd. Ström tas upp via kontakt mellan fordon och väg och leds till fordonets energilagring i form av en superkondensator. Beroende på driftfall kan energilagret laddas upp eller laddas ur. Om tillförd effekt från vägen överstiger motorns effektbehov lagras överskottsenergi i kondensatorn, medan kondensatorn istället laddas ur när motorns effektbehov överstiger tillförd effekt.

Den lagrade energin används för att driva en likströmsmotor som omvandlar elekt-

risk energi till mekanisk energi. Denna förs vidare via drivlinan till hjulen, vilket genererar fordonets rörelse.

Utöver energiflödet innehåller systemet funktioner för mätning, styrning och kommunikation. Hastigheten mäts med hjälp av en Hall-effektsensor. Den uppmätta hastigheten skickas till fordonets mikrokontroller, där en reglerfunktion justerar motorns styrsignal via en pulsbreddsmodulerings-signal (PWM-signal, vilket illustreras i Figur 2). Spänningen över superkondensatorn mäts kontinuerligt via en analog ingång på mikrokontrollern och används som ett mått på energinivån.

Mätdata från respektive fordon, bestående av aktuell hastighet och kondensatorspänning, skickas trådlöst till en centralenhet. Centralenheten vidarebefordrar informationen till en displaymodul, där systemets parametervärden visualiseras i realtid. Displaymodulen fungerar även som användargränssnitt, där referenshastigheten kan justeras via en potentiometer. Detta referensvärde skickas tillbaka via centralenheten till respektive fordon och används för dess reglering.

2.2 Systembeskrivning mjukvara

Mjukvaran utgör en central del av systemet och möjliggör samverkan mellan fordonet och centralenheten. I Figur 2 visas informationsflödet mellan enheterna. Bilenheten och centralenheten kommunicerar trådlöst via ESP_NOW. Fullständig källkod för samtliga mjukvarukomponenter återfinns i Bilaga B.

2.2.1 Mjukvara i bilenheten

Bilens mjukvara är uppbyggd kring en kontinuerlig loop där mätning, reglering och kommunikation utförs periodiskt. Hastigheten beräknas utifrån tiden mellan inkommande pulser från hastighetssensorn. Pulser som registreras för tätt samt orimliga hastighetsvärden filtreras bort, och ett lågpasfilter appliceras för att stabilisera hastighetssignalen.

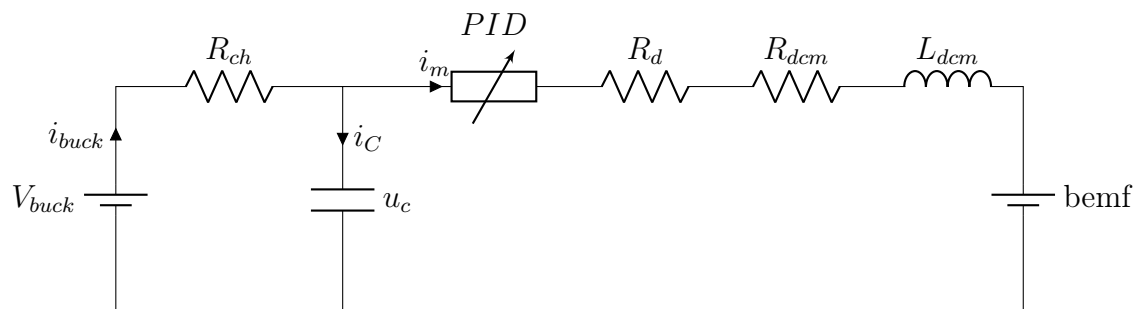
Den filtrerade hastigheten används i en PID-regulator som tar emot en referenshastighet från centralenheten och beräknar ett fel mellan denna och den uppmätta hastigheten. Utifrån felet genereras en styrsignal. Styrsignalen begränsas till PWM-signalens arbetsområde och anti-windup används för att förhindra okontrollerad tillväxt av integraltermen. Bilenheten beräknar även fordonets position genom integration av hastighet över tid. Mätdata bestående av hastighet, kondensatorspänning, PWM-signal och position paketeras och skickas regelbundet till centralenheten.

2.2.2 Mjukvara i centralenheten

Centralenhetens mjukvara hanterar tre huvudsakliga uppgifter: att skicka referenshastighet till bilenheten, att behandla och visualisera mottagen mätdata, samt att styra elektrifieringen av vägbanan. Referenshastigheten läses in och omvandlas till ett hastighetsvärde som kontinuerligt skickas till bilenheten. Mottagen mätdata från bilenheten vidarebefordras dels till en display för realtidsvisualisering, dels till en ansluten dator där ett Python-program loggar och plottar datan. Elektrifieringen av vägbanan styrs via mjukvaran baserat på insignaler från två sensorer, som detekterar när fordonet rör sig in respektive ut från laddningssträckan.

2.3 Matematisk modell av systemet

För att analysera elvägssystemets beteende har en matematisk modell tagits fram. Modellen beskriver sambanden mellan de elektriska och mekaniska delarna i systemet och utgör grunden för den simuleringsmodell som implementerats i Simulink. Systemet delas upp i fyra huvudsakliga delar: (i) energilagring, (ii) hastighetsreglering, (iii) motordynamik och drivlina samt (iv) mekanisk rörelse. I Figur 3 visas en förenklad kretsmodell av systemet. I den verkliga kretsen passerar spänningen från elvägen genom motståndet i borstarna och en skyddsdiode innan den stegas ner till en konstant spänning, V_{buck} , via en buck-converter. Eftersom buck-convertern håller utgångsspänningen konstant modelleras matningen förenklat som en konstant spänningskälla V_{buck} , vilket är den spänning som faktiskt matar resten av kretsen.



Figur 3: Krets för systemet

Strömmen passerar därefter genom laddningsresistansen R_{ch} [Ω], vilken modellerar resistiva förluster i kablage och anslutningar. Efter detta når strömmen kondensatorn med spänningen u_c [V], som fungerar som systemets energilager. Kondensatorn lagrar energi när tillförd effekt överstiger förbrukningen och avger energi när tillförseln är otillräcklig.

Den högra delen av kretsen beskriver PID-regulatorn samt motorn. PID regulatorn kan betraktas som en variabel resistans och modelleras förenklat som ett spänningsfall i serie med motorn, där den tillsammans med en MOSFET-transistor reglerar spänningen till motorn. Den detaljerade beskrivningen av PID-regulatorn finns i kapitel 3.2. Motorn modelleras som en elektrisk krets bestående av en resistiv del, representerad av R_d [Ω] och R_{dcm} [Ω], samt en induktiv del L_{dcm} [H]. Motorns elektromekaniska koppling representeras av en motspänning, så kallad back-EMF, som är proportionell mot vinkelhastigheten ω [rad/s] och flödeskopplingen Ψ [Vs].

Denna uppdelning möjliggör en systematisk analys av hur energi lagras och omvandlas till mekaniskt arbete i fordonet.

2.3.1 Energitillförsel från buck-convertern

Det enklaste sättet att förstå systemet är att börja från buck-convertern, där spänning tillförs och avsluta med att beskriva mekanisk rörelse. Genom att tillämpa Kirchhoffs spänningslag (KVL) på den vänstra delen av kretsen erhålls sambandet:

$$V_{buck} - i_{buck}R_{ch} - u_c = 0 \quad (1)$$

vilket kan lösas ut till:

$$i_{buck} = \frac{V_{buck} - u_c}{R_{ch}} \quad (2)$$

Uttrycket visar att strömmen från elvägen bestäms av spänningsskillnaden mellan buck-convertern och kondensatorn samt de resistiva förlusterna i systemet.

2.3.2 Strömfördelning

Den tillförda strömmen i_{buck} delas upp mellan motorn och kondensatorn enligt Kirchhoffs strömlag (KCL):

$$i_{buck} = i_m + i_C \quad (3)$$

där i_m är strömmen till motorn och i_C är strömmen till kondensatorn. Beroende på driftfall kan tre situationer uppstå: om $i_{buck} > i_m$ lagras överskottsenergi i kondensatorn ($i_C > 0$), om $i_{buck} = 0$ drivs motorn enbart av kondensatorn ($i_C < 0$), och om $i_{buck} = i_m$ är systemet i jämvikt ($i_C = 0$).

2.3.3 Kondensatordynamik

Kondensatorn beskrivs av sambandet:

$$i_C = C \frac{du_c}{dt} \quad (4)$$

där C är kapacitansen u_c är kondensatorspänningen t är tiden. Genom att kombinera detta med sambandet i (3) erhålls:

$$C \frac{du_c}{dt} = i_{buck} - i_m \quad (5)$$

vilket ger:

$$\frac{du_c}{dt} = \frac{1}{C} (i_{buck} - i_m) \quad (6)$$

Genom att substituera uttrycket för i_{buck} med hjälp av (2) erhålls:

$$\frac{du_c}{dt} = \frac{1}{C} \left(\frac{V_{buck} - u_c}{R_{ch}} - i_m \right) \quad (7)$$

Integrering ger:

$$u_c(t) = \int \frac{1}{C} \left(\frac{V_{buck} - u_c(t)}{R_{ch}} - i_m(t) \right) dt \quad (8)$$

Detta beskriver hur kondensatorspänningen förändras över tid beroende på tillförd och förbrukad ström. Detta visar tydligt att längre laddningstid ger högre spänning.

I projektet så används seriekoppling av kondensatorer för att variera energilagret. Vid seriekoppling av två identiska superkondensatorer förändras den totala kapacitansen enligt:

$$\frac{1}{C_{tot}} = \frac{1}{C_1} + \frac{1}{C_2} + \dots + \frac{1}{C_n} \quad (9)$$

För två identiska kondensatorer med kapacitansen C ger detta en total kapacitans på $\frac{C}{2}$

Vid seriekoppling av två likadana kondensatorer går det att mata med en högre spänning, eftersom kondensatorerna tar upp lika mycket var. Detta innebär i praktiken att det går att lagra mer energi eftersom den möjliga ökningen av matningsspänning kompenseras för minskningen i kapacitans. Den lagrade energin i kondensatorn ges av:

$$W = \frac{C}{2} u_c^2 = \frac{C u_c^2}{2} \quad (10)$$

2.3.4 Motordynamik

Motorn modelleras som en elektrisk krets bestående av resistans och induktans samt back-EMF. PID-regulatorn tillsammans med MOSFET-transistorn modelleras som ett spänningsfall V_{PID} [V] i serie med motorn. I modelleringen representerar R_d [Ω] den externa resistansen, R_{dcm} [Ω] motorns interna resistans, L_{dcm} [H] motorns induktans, ω [rad/s] vinkelhastigheten och Ψ [Vs] flödeskopplingen, där produkten $\Psi\omega$ motsvarar motorns back-EMF.

Med (KVL) över den högra delen av kretsen erhålls:

$$u_c - V_{PID} - i_m(R_d + R_{dcm}) - L_{dcm} \frac{di_m}{dt} - \omega\Psi = 0 \quad (11)$$

vilket kan skrivas om till:

$$\frac{di_m}{dt} = \frac{u_c - V_{PID} - i_m(R_d + R_{dcm}) - \omega\Psi}{L_{dcm}} \quad (12)$$

2.3.5 Mekanisk kraftöverföring

Motorströmmen i_m omvandlas till ett vridmoment T [Nm] enligt:

$$T = \Psi i_m \quad (13)$$

Vridmomentet överförs via drivlinan till hjulen med utväxlingen gr [-] och hjulradien R_w [m], vilket ger kraften:

$$F = \frac{T \cdot gr}{R_w} \quad (14)$$

Efter att friktionskraften F_{friktion} [N] beaktats erhålls den nettokraft som driver fordonet framåt:

$$F_{\text{netto}} = F - F_{\text{friktion}} \quad (15)$$

2.3.6 Fordonsdynamik

Fordonets rörelse beskrivs med Newtons andra lag, där M [kg] är fordonets massa och v [m/s] är hastigheten:

$$F_{\text{netto}} = M \frac{dv}{dt} \quad (16)$$

vilket ger accelerationen:

$$\frac{dv}{dt} = \frac{F_{\text{netto}}}{M} \quad (17)$$

Hastigheten och positionen x [m] längs banan erhålls genom upprepad integration:

$$v(t) = \int \frac{F_{\text{netto}}}{M} dt \quad (18)$$

$$x(t) = \int v(t) dt \quad (19)$$

3 Metod

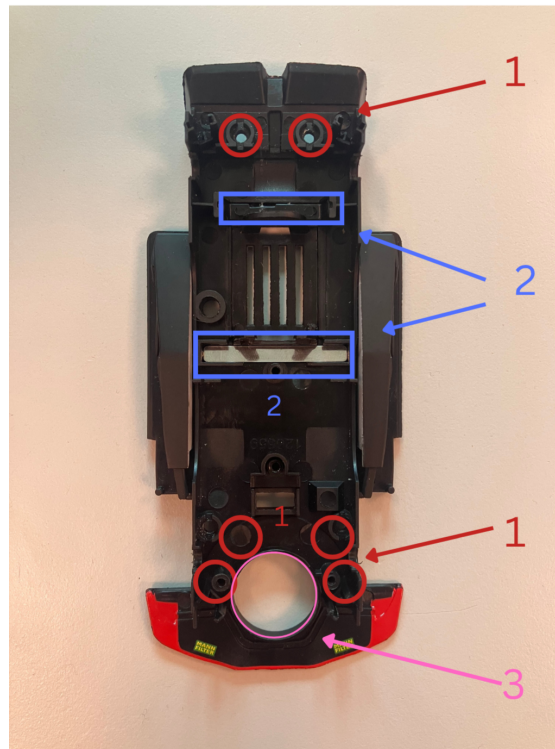
I metodavsnittet beskrivs den metodik som användes för att implementera projektets tekniska lösningar och därmed uppfylla de uppsatta kraven. Avsnittet är uppdelat i fyra centrala delområden: Konstruktion och implementation av hårdvara, elektrisk koppling, modifiering av simuleringsmodellen samt validering. Dessa delar utgör tillsammans projektets tekniska fundament samt de processteg som genomförts för att nå det slutgiltiga resultatet.

3.1 Konstruktion och implementation av hårdvara

Detta delkapitel beskriver den metodik som användes för att utveckla och implementera det fysiska systemet. Arbetet omfattade konstruktion av fordonen, implementering av energilagring, implementering av hastighetsreglering samt elektrifiering av vägbana. De olika delarna presenteras i separata avsnitt med fokus på hur systemet utformats och realiserats. För fullständig komponentlista se Bilaga C.

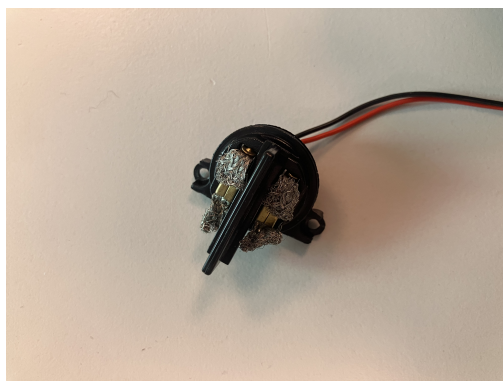
3.1.1 Mekaniska komponenter

Majoriteten av de mekaniska komponenterna återanvändes från ursprungskonstruktionen i Scalextric-systemet. De ursprungliga fordonen bestod av två huvudsakliga delar: ett underrede som utgjorde den bärande konstruktionen för de ingående komponenterna, samt ett chassi som kapslade in de elektriska och mekaniska delarna. Underredets design går att bryta ner i delar som fyller olika funktioner. De ihåliga cylinderformade komponenterna på underredet var till för att kunna fästa chassit på underredet. Dessa delar är i Figur 4 markerade med en röd cirkel och nummer ett. Nummer två pekar på två nedsänkta spår inringat i blått där magneter monterades, magneternas funktion var att hålla bilen mot bilbanans metallskenor. Nummer tre pekar på ett större hål inringat med rosa färg där en styrskena monterades.



Figur 4: Underrede

I det större hålet inneslöts en kombinerad styrskena och strömuttagare i form av två metallflätor. Se Figur 5a och Figur 5b.



(a) Undersidan

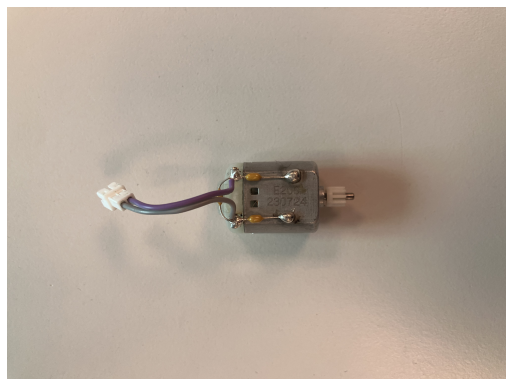


(b) Ovansidan

Figur 5: Styrskena med metallfläta

Styrskenan i Figur 5a och Figur 5b bestod av en skena och metallflätor som tog upp ström från vägbanan och förde den vidare via en tvåpinskontakt. Utöver detta innehöll styrskenan två plastcirkelskivor med en fjäder som återställde fordonets riktning efter kurvor. Styrskenan kopplades ihop med motorn via en tvåpinskontakt

för att ström skulle kunna färdas mellan de två komponenterna. Motorn, se Figur 6a, som användes i Scalextric bilarna var en DC-motor med ett kugghjul monterat på axeln. Detta kugghjul kopplades ihop med ett större kugghjul på hjulaxeln, se Figur 6b.



(a) DC-motor



(b) Hjulaxel

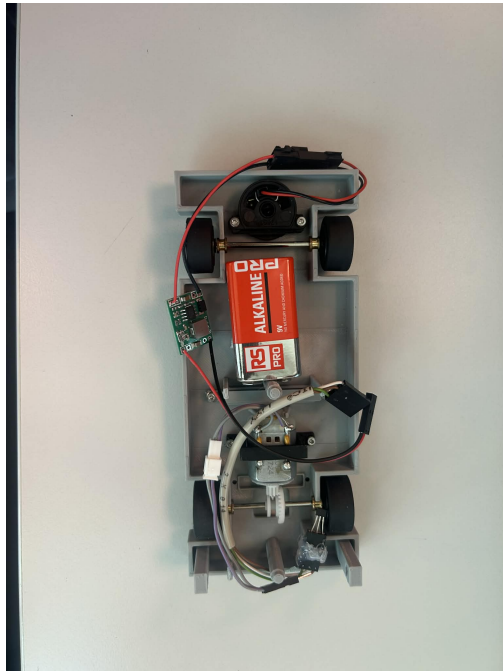
Figur 6: Motor och hjulaxel

Kugghjulet gav en utväxling på 3:1 från motorn till hjulaxeln. Kugghjulet och cirkelskivan på axeln fungerade tillsammans som en styrskena vilken fixerade och stabiliserade kring motoraxeln. Två glidlager som fästes i underredet i varsitt snäppfäste, möjliggjorde att axeln roterade fritt. På axeln var slutligen två gummihjul fastsatta.

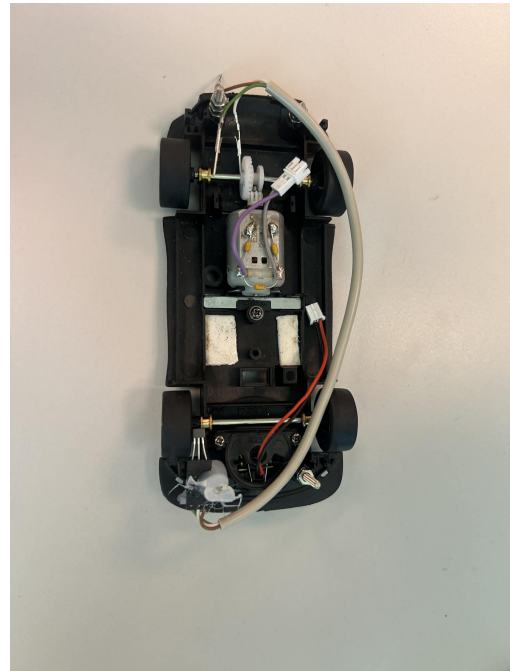
3.1.2 Konstruktion och implementation av fordon

Konstruktionen inleddes med utveckling av ett egenkonstruerat underrede för lastbilen, medan för det andra fordonet användes ett befintligt underrede från Scalextric-systemet. Som underlag för dimensioneringen av det egenkonstruerade underredet mättes de ingående komponenterna, såsom motor, hjulaxlar, magnetfästen samt styrskenan. Dessa mått användes sedan vid modellering i CAD-programmet Catia V5.

Efter modelleringen tillverkades underredet genom 3D-utskrift. De mekaniska komponenterna monterades därefter och passform samt funktion verifierades genom praktisk testning. Den praktiska testningen utfördes genom att komponenterna provmonterades samt provkördes ovanpå ett bansegment. Konstruktionen krävde flera iterationer, där design, tillverkning och testning upprepades tills samtliga komponenter passade och fungerade tillsammans. Det egenkonstruerade underredet visas i Figur 7a och det återanvända underredet ses i Figur 7b.



(a) Underrede egenkonstruerat



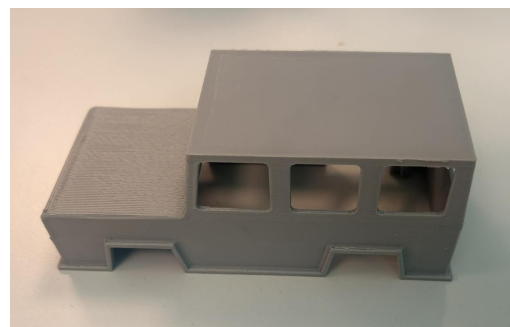
(b) Underrede ursprungsmodell

Figur 7: Underreden för det egenkonstruerade fordonet (a) och ursprungsmodellen (b)

När det egenkonstruerade underredet hade verifierats, påbörjades konstruktionen av chassin. Till det större egenkonstruerade underredet konstruerades ett chassi i form av en lastbil, se Figur 8a. Det återanvända Scalextric-underredet försågs med ett mindre bilchassi, se Figur 8b.



(a) Chassi lastbil



(b) Chassi bil

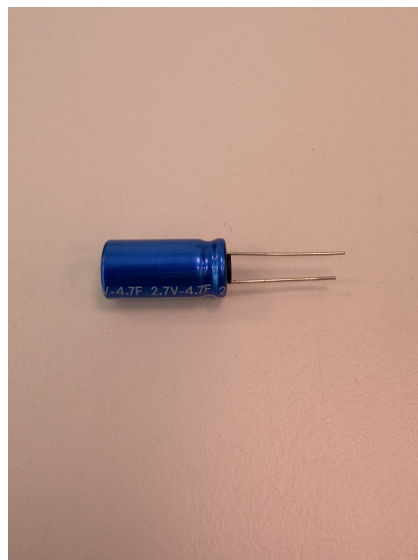
Figur 8: Chassi till bil och lastbil

För att möjliggöra montering mellan chassi och underrede genomfördes ytterligare en iteration av båda komponenterna, där infästningar för sammankoppling integrerades i konstruktionen.

3.1.3 Implementering av energilagring

Superkondensatorer valdes som energilager då de kombinerade hög effekttäthet med ett snabbt laddningsförlopp och hög tålighet mot upprepade laddningscykler [5]. Till skillnad från batterier kan superkondensatorer laddas och urladdas snabbt utan att degraderas, vilket var fördelaktigt i ett system där laddning skedde under färd. Superkondensatorn som användes i projektet visas i Figur 9.

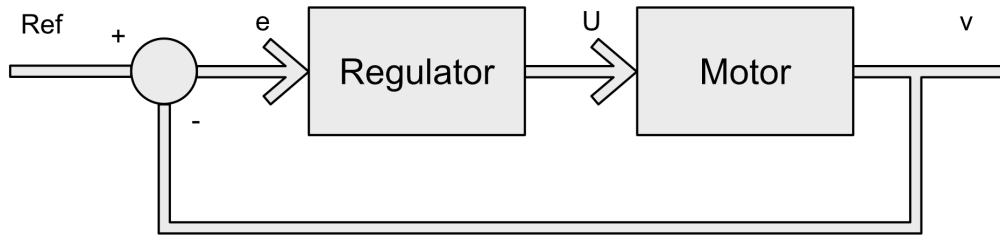
Två storlekar av superkondensatorer användes, med kapacitansen 4.7 F respektive 10 F per kondensator. Båda typerna hade en märkspänning på 2.7 V, vilket vid seriekoppling av de två kondensatorerna gav en total märkspänning på 5.4 V. Superkondensatorerna placerades längst bak i fordonet och kopplades in via en öppen sektion i chassit, vilket möjliggjorde variation av energilagrets storlek. Detta möjliggjorde studier av hur lagringskapaciteten påverkar fordonets räckvidd och prestanda.



Figur 9: Superkondensator

3.1.4 Reglering av fordonens hastighet

För att möjliggöra körning vid olika referenshastigheter implementerades en PID-regulator i mikrokontrollern. PID-regulatorer används ofta vid hastighetsreglering enligt [6]. Blockdiagrammet i Figur 10 illustrerar den återkopplade reglerloopen, där referenshastigheten jämfördes med den uppmätta hastigheten v [m/s] och gav upphov till ett felvärde e [m/s]. Regulatorn använde felvärdet för att generera en PWM-styrsignal som applicerades på motorn via MOSFET-transistorn.



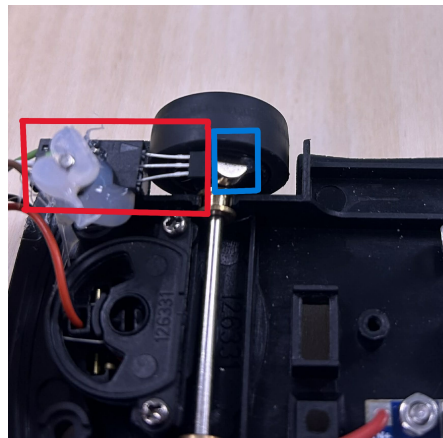
Figur 10: Blockdiagram för regulatorn

Styrsignalen beräknas enligt:

$$u(t) = K_P e(t) + K_I \int e(t) dt + K_D \frac{de(t)}{dt} \quad (20)$$

där $e(t)$ var felet mellan referenshastigheten och den uppmätta hastigheten. Styrsignalen begränsades till intervallet 0–255 för att motsvara PWM-signalens arbetsområde, och anti-windup implementerades för att begränsa integraltermen.

Hastigheten mättes med en Hall-effektsensor och en magnet fäst vid hjulaxeln, se Figur 11, där sensorn och magneten är markerade med rött respektive blått. Sensorn genererade en puls per hjulvarv, varifrån hastigheten beräknades via hjulradien. Regulatorparametrarna K_P , K_I och K_D bestämdes t genom experimentella tester tills stabil och jämn reglering uppnåddes. Se Bilaga B.1 för fullständig kod.



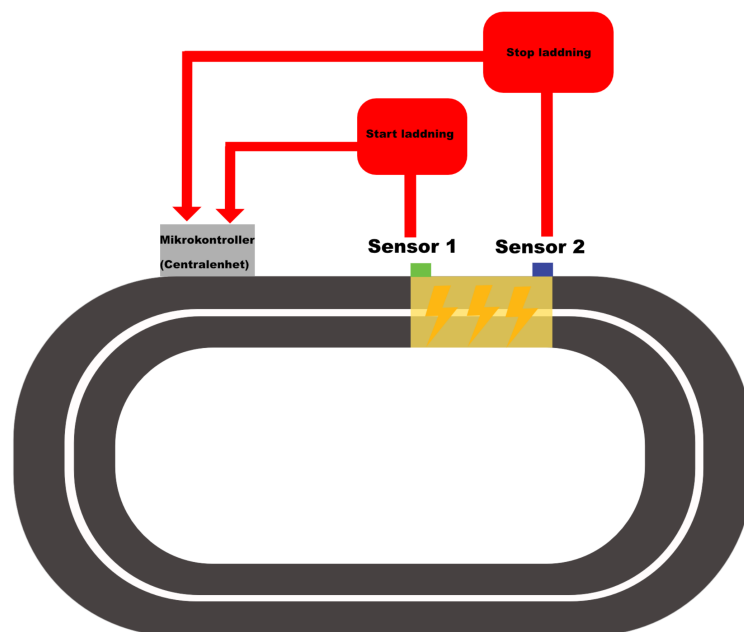
Figur 11: Hall-effektsensor

3.1.5 Elektrifiera vägen

Bilbanan återanvändes från tidigare kandidatarbete [4] och består av modulära vägdelar, både raka och svängda, som sätts samman till en sluten bana. Varje bansegment innehåller två spår med ledande metallskenor som fyller två funktioner:

att styra fordonets rörelse längs banan samt att överföra elektrisk energi till fordonet via direkt kontakt. När segmenten kopplas samman etableras elektrisk kontakt mellan delsegmenten, vilket möjliggör spänningsdistribution längs hela banan från en extern spänningskälla.

Elektrifieringen styrdes av två IR-sensorer placerade vid laddningssträckans start och slut, se Figur 12. När sensor 1 detekterade ett fordon skickade centralenheten en styrsignal till ett relä som aktiverade strömmen över hela banan. När fordonet nådde sensor 2 bröts styrsignalen och banans strömtillförsel deaktiverades. I praktiken spänningssattes hela banan, men fordonet upplevde detta som att endast det aktuella segmentet elektrifierades.



Figur 12: Elektrifiering av väg illustration

3.1.6 Visualisering av data

Fordonets hastighet och kondensatorspänning loggades kontinuerligt via mikrokontrollern och skickades trådlöst till centralenheten. För att visualisera mätdatan konstruerades en displaymodul, se Figur 13, där en OLED-display monterades för varje fordon. På displayerna visades aktuell hastighet, kondensatorspänning, referenshastighet och körsträcka i realtid. Centralenhetslådan försågs även med potentiometrar, vilket möjliggjorde justering av referenshastigheten under körning. Utöver detta skickades data till en ansluten dator där mätvärdena plottades med ett Python-program i realtid, vilket användes för att analysera systemets beteende och kalibrera PID-parametrarna.



Figur 13: Displaymodul

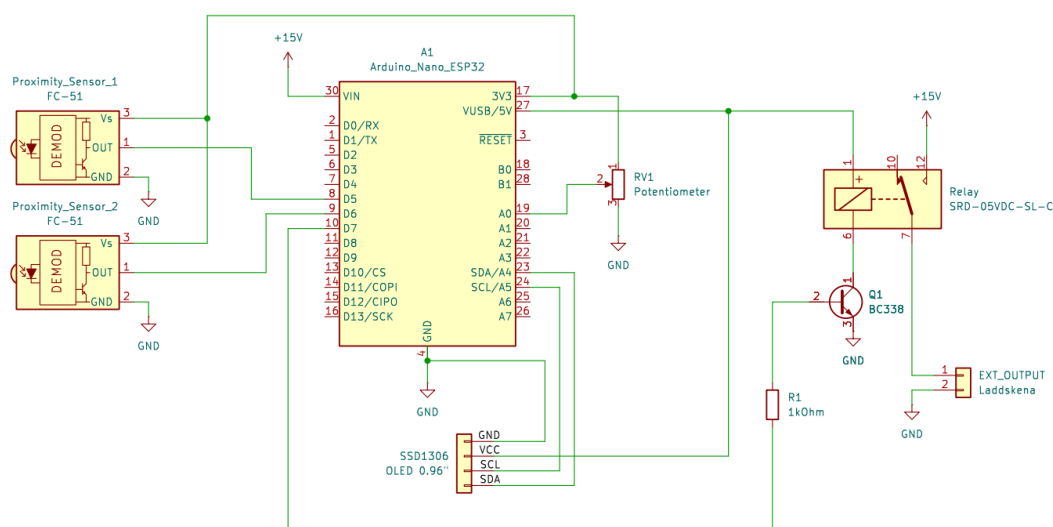
3.2 Elektrisk koppling

I projektet användes två olika elektriska system, en för centralenheten och en för fordonsenheten. Avsnittet beskriver de ingående elektriska detaljerna i dessa.

3.2.1 Centralenheten och mikrokontroller

Centralenheten fungerade som en knutpunkt mellan användare och system, vars kretsschema visas i Figur 14. Enhetens centrala komponent utgjordes av en Arduino Nano ESP32, som styr banans strömförsörjning och skickade referenshastighet samt tog emot mätdata från bilenheten. Referenshastigheten ställdes in av användaren via en potentiometer kopplad till en av mikrokontrollerns analoga ingångar. Potentiometern bildade en spänningsdelare vars utgångsspänning varierade linjärt mellan 0 och 3.3 V, vilket mikrokontrollern tolkade som ett hastighetsvärde i intervallet 0–1.6 m/s. Den inställda referenshastigheten samt övrig systemstatus presenterades i realtid på en OLED-display ansluten till mikrokontrollern. Längs banan var två IR-sensorer monterade som detekterade när ett fordon passerade. Sensorernas ut signaler leddes in till mikrokontrollerns digitala ingångar och användes för att avgöra när ett fordon befann sig i den förutbestämda laddningssträckan. Detta aktiverade respektive avaktiverade den digitala utgången D7, vilken styrde

det efterföljande relästeget. Eftersom mikrokontrollern arbetade med 3.3 V var utgångsströmmen otillräcklig för att direkt styra det relä som kontrollerade banans laddningsskena. Därför användes en bipolär transistor som förstärkarsteg där D7 signalen drev transistorns bas, varvid transistorn övergick i ledande tillstånd och slöt reläkretsen. Reläet var av typen normally open, vilket innebar att 15 V matningen till banans laddningsskena normalt var bruten och enbart slöts när fordonet detekterades.



Figur 14: Kretsschema centralenhet

3.2.2 Bilenheten

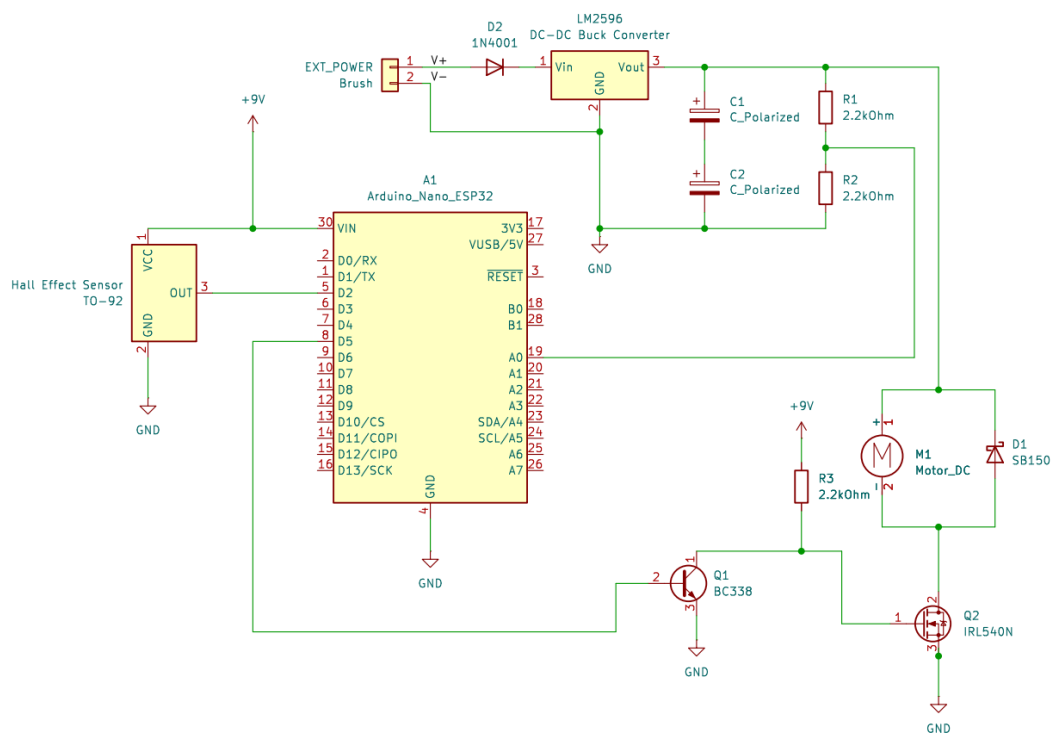
Fordonets styrenhet utgjordes även den av en Arduino Nano ESP32 som visas i mitten av Figur 15, vilken administrerade samtliga signaler från fordonets övriga komponenter. Mikrokontrollern strömförsörjdes via ett lokalt monterat 9V-batteri vars spänning anslöts till mikrokontrollerns VIN-stift. Spänningen transformerades därefter ned via en intern spänningsregulator till Arduinos arbetsspänning på 3.3 V. Samma batteri användes för att försörja fordonets Hall-effektsensor. Sensorns utgång var varken hög eller låg, så kallad flytande i sitt normaltillstånd, därför användes ett internt uppdragsmotstånd (pull-up) till 3.3 V definierat i mjukvaran. Vid detektering av ett yttre magnetfält drogs sensorns utsignal ner till jordpotential, vilket genererade en avläsbar fallande kant för mikrokontrollern.

Fordonets drivspänning på 15 V hämtades från banan via borstar. Den inkommande strömmen leddes genom en skyddsdiode för att skydda kretsen mot felpolarisering om bilen exempelvis sattes åt fel håll. Spänningen transformerades därefter

ned till 5.2 V via en buck-converter, vilket höll sig inom superkondensatorernas sammanlagda märkspänning på 5.4 V. Den totala spänningsnivån i kondensatorerna övervakades kontinuerligt via en spänningsdelare som halverade spänningen för att inte överstiga Arduinons arbetsspänning på 3.3 V. Den halverade spänningen matades i en av de analoga ingångarna och värdet kunde i mjukvaran dubblas igen för att få den verkliga spänningen.

Fordonets motorhastighet reglerades via PWM från mikrokontrollerns utgång D5. Signalen genererades med en frekvens på 5 kHz och pendlade mellan 0 och 3.3 V. För att styra strömtillförseln från energilagret till motorn användes en MOSFET-transistor. Eftersom den valda MOSFET-komponenten krävde en styrspänning (gate-spänning) på minst 5 V för effektiv styrning, var PWM-signalen på 3.3 V otillräcklig. Därför implementerades en drivkrets i form av en hjälptransistor.

Hjälptransistorn agerade som en nivåomvandlare som både förstärkte och logiskt inverterade signalen från D5. Kretsen var konstruerad så att ett motstånd kontinuerligt laddade upp MOSFET-transistorns gate-terminal till 9 V. När hjälptransistorn aktiverades av PWM-signalen laddades gate-kapacitansen ur till jord. Denna cykel repeterades 5 000 gånger per sekund. När MOSFET-transistorn var ledande slöts kretsen, varvid ström tilläts flyta från kondensatorerna, genom elmotorn, via MOSFET-transistorn och slutligen till jord. För att skydda drivkretsen från spänningsspikar som uppstod när strömmen genom motorlindningarna bröts, hade en skyddsdiode parallellkopplad över motorn.

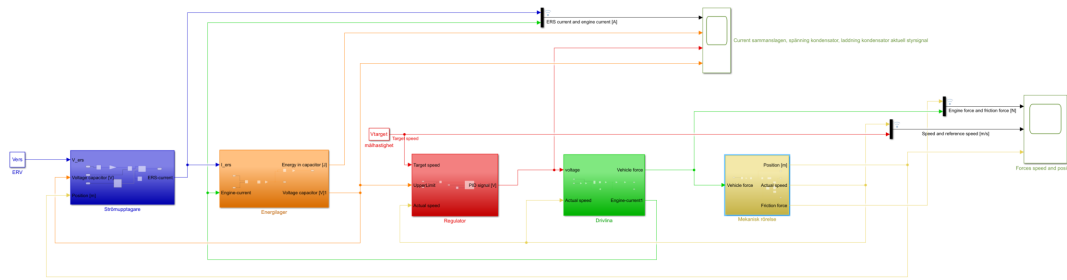


Figur 15: Kretsschema bilmodell

3.3 Modifiering av simuleringsmodell

Simuleringsmodellen implementerades i MATLAB/Simulink och baserades på matematiken i kapitel 2.3. Modellen utgick från en befintlig simuleringsmodell från föregående års projektarbete, vilken anpassades till projektet. Den stora skillnaden var att det implementerades en PID-regulator, vilket behandlades närmare under kapitel 3.1.4. Under arbetet identifierades även ett antal felaktigheter i originalmodellen som korrigerades, vilket diskuteras vidare i avsnitt 5.2.

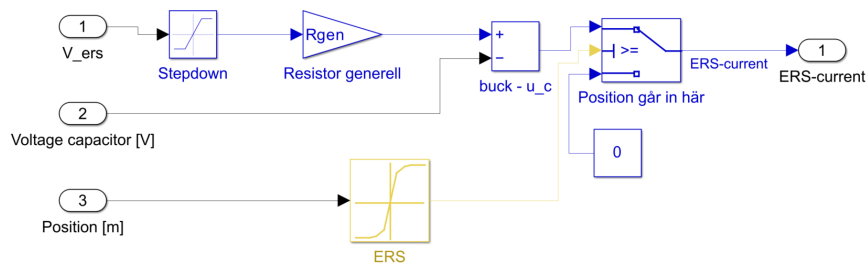
Simuleringen bestod av två delar: en MATLAB-kod som initierade parametervärden och en Simulink-modell. Denna bestod av standardblock där matematiken implementerades, således byggdes dynamiken i systemet upp. Simulink-modellen bestod av fem färgkodade delar som var uppdelade i strömupptagare, energilager, regulator, drivlina och mekanisk rörelse. Därutöver fanns två Scopes för att läsa av värden. I Figur 16 visas Simulink-modellen i sin helhet, se Bilaga B.6 för större bild av modellen. Därefter beskrivs delsystemen mer ingående.



Figur 16: Simulink-modell

3.3.1 Strömuttagning

Det första delsystemet av modellen simulerade strömuttagningen från vägbanan. Baserat på matematiken i kapitel 2.3.1. I Figur 17 visas det att efter V_{ers} är en buck-converter, i figuren benämnd stepdown, implementerad. Detta modellerades med hjälp av ett saturationblock. Därefter omvandlades nettospänningen till ström med hjälp av ett gainblock R_{gen} , vars värde kalibrerades mot uppmätt ström i den fysiska modellen. Därefter följde en switch som beroende på positionen kände av om bilen befann sig på en elektrifierad del av vägbanan. Detta beskrivs mer ingående i kapitel 3.3.5.

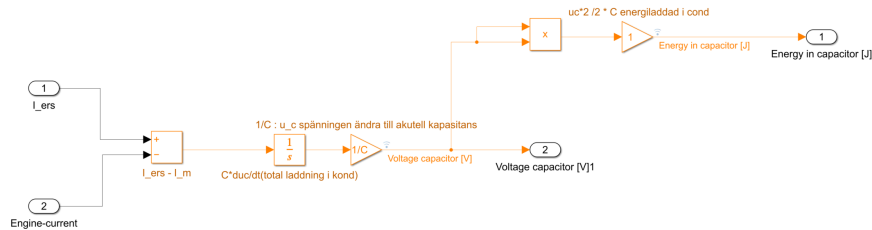


Figur 17: Strömuttagning

3.3.2 Energilager

Nästa delsystem modellerade energilagret på bilen. I Figur 18 visas hela implementeringen, vilket baserades på kapitel 2.3.3. Genom att subtrahera motorströmmen från banans ström simulerades den drivande spänningsskillnaden. Detta följdes av ett integratorblock i kombination med ett gainblock som integrerades och skalades för att erhålla kondensatorspänningen u_c .

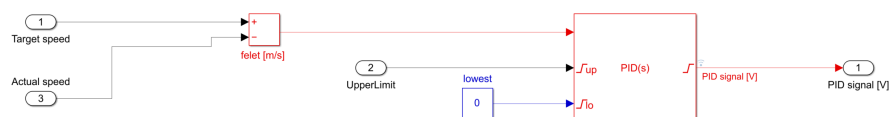
Därutöver fanns en avgrening som beräknade den totala energin i kondensatorn enligt ekvation (10). Kondensatorn initierades med spänningen 5.2 V, vilket innebar att fordonet startade med ett fulladdat energilager.



Figur 18: Energilager

3.3.3 Hastighetsreglering

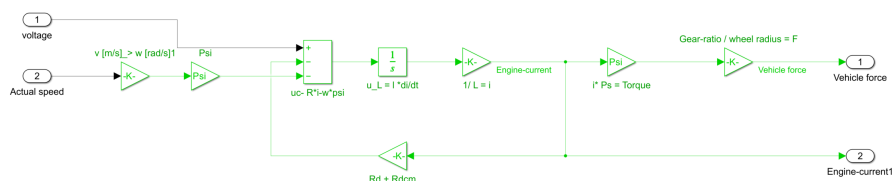
Detta delsystem implementerade hastighetsregleringen, se Figur 19. Regleringen implementerades med hjälp av ett PID-regulatorblock i Simulink. Värt att notera att till skillnad från den verkliga bilmodellen användes inte en PWM-signalgenerator, utan en förenklad ideal PID-regulator. Blocket tog emot felet mellan referenshastigheten V_{target} och den aktuella hastigheten som insignal. Styrsignalens övre begränsning sattes till kondensatorspänningen u_c , medan den undre begränsningen sattes till noll. Styrsignalen, som var utsignal från PID-regulatorn, blev insignalen till motorn. Regulatorparametrarna som användes togs från den verkliga regulatorn. Se Bilaga B.4 för exakta värden.



Figur 19: Regulator

3.3.4 Drivlinan

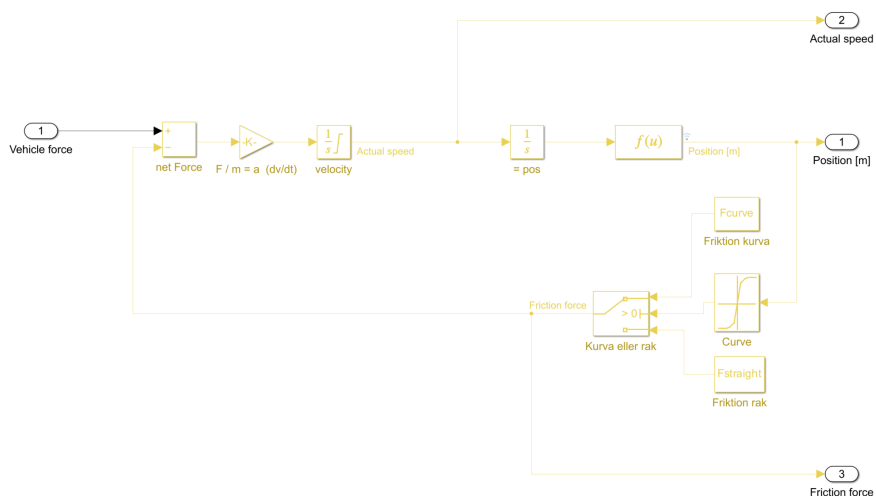
I nästa delsystem, illustrerad i Figur 20, simulerades hela drivlinan. Delsystemet implementerade matematiken från kapitel 2.3.4 och 2.3.5 med hjälp av sum-, gain- och integrator-block. Utsignalen från drivlinan blev därmed motorkraften.



Figur 20: Drivlina

3.3.5 Vägdynamik

Det sista delsystemet simulerade vägdynamiken och fordonets position på banan, se Figur 21. All matematik som implementerades var från kapitel 2.3.6, utöver de sista blocken till höger i bilden. Där användes ett funktionsblock som nyttjade moduloaritmetik. Detta innebär att när positionen nådde banans maxlängd på 4.27 m mappades den tillbaka till startpositionen och fordonet började om på banan. Maxlängden motsvarade 4270 positionssteg med steglängden 1 mm. Därefter följde två lookup-block som avgjorde om positionen motsvarade en raksträcka eller en kurva, samt om vägen var elektrifierad eller inte, vilket styrde friktionskraften och strömtillförseln. För implementeringen i koden, se Bilaga B.4.



Figur 21: Mekanisk rörelse

Centrala modellparametrar sammanfattas i Bilaga B.5. Det bör noteras att flera parametrar uppskattades snarare än uppmättes empiriskt. Induktansen L_{dcm} och friktionskrafterna estimerades genom försök till att likställa simuleringsmodellen till den verkliga modellen. Övriga parametervärden som är markerade som uppskattade i Bilaga B.5 återanvändes från föregående års kandidatarbete. Eftersom delar av parametrarna var uppskattade begränsas modellens kvantitativa preci-

sion. Modellen bör därför i första hand betraktas som ett verktyg för kvalitativ analys av systemets övergripande dynamiska beteende.

3.4 Validering

Systemet verifierades genom tester på den fysiska modellen samt simuleringsmodellen. Fordonen kördes på bilbanan under olika driftfall, där hastighet och kondensatorspänning observerades och loggades i realtid. Detta möjliggjorde en kvantitativ bedömning av systemets funktionalitet.

Vidare jämfördes resultaten från den fysiska modellens mätvärden med motsvarande simuleringar. Särskilt studerades förmågan att behålla referenshastigheten, kondensatorspänningen och urladdningen över tid.

4 Resultat

Detta avsnitt behandlar och visualiserar de resultat från de olika tester som genomförts, vilket visas i Tabell 1.

Tabell 1: Sammanställning av genomförda tester

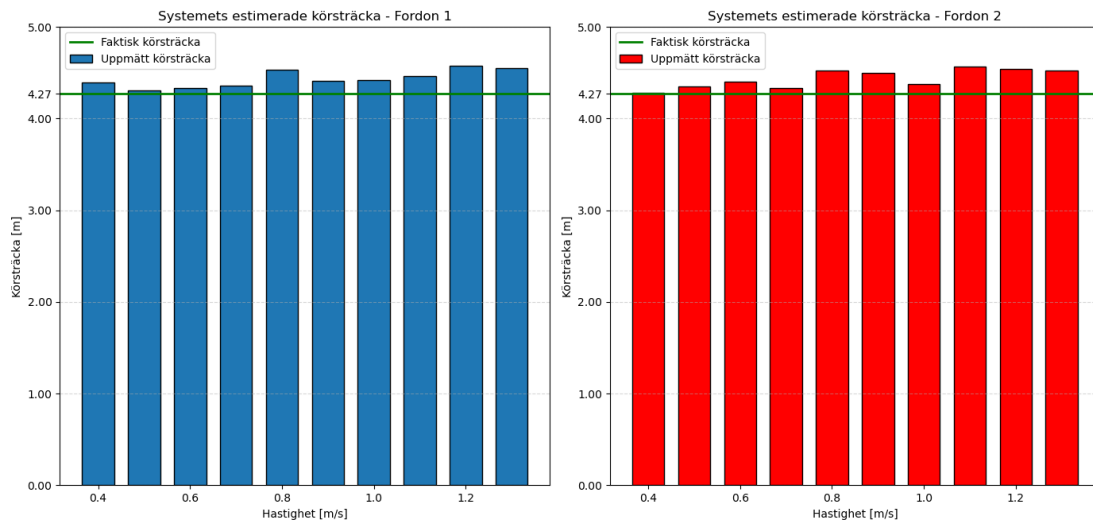
Test	Beskrivning	Kapacitans
1	Fordonens estimerade körsträcka	5 F
2	Hastighet och körsträcka	5 F
3	Varierande laddningslängd raksträcka	5 F
4	Varierande laddningslängd kurva	5 F
5	Hastighet och körsträcka	2.35 F
6	Laddningslängd på raksträcka	2.35 F
7	Laddningslängd i kurva	2.35 F
8	Simuleringstest 1 - ingen laddning från elväg	2.35 F
9	Simuleringstest 2 - laddningssträcka 35.2 cm	2.35 F
10	Simuleringstest 3 - laddningssträcka 52 cm	2.35 F

4.1 Test 1 - Fordonens estimerade körsträcka - 5 F

Inledningsvis genomfördes tester för att verifiera fordonens mätnoggrannhet. I Tabell 2 presenteras testförutsättningarna och i Figur 22 visas hur väl fordonets estimerade körsträcka överensstämde med den faktiskt körda sträckan. De exakta värdena för mätningarna återfinns i Bilaga A.1.

Tabell 2: Testförutsättningar test 1

Parameter	Värde
Fordon 1 vikt	220 g
Fordon 2 vikt	307 g
Kapacitans	5 F
Hastighet	Varierande
Laddning	Full laddning under färd



Figur 22: Fordonens estimerade körsträcka

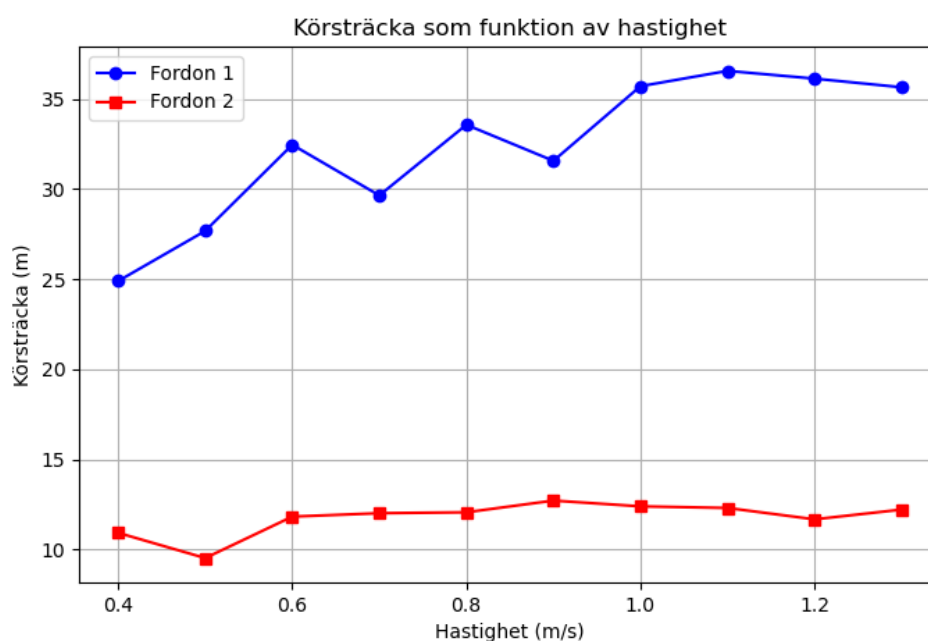
4.2 Test 2 - Hastighet och körsträcka - 5 F

Efter att fordonens mätnoggrannhet hade verifierats genomfördes test 2. Syftet med testet var att undersöka hur fordonens körsträcka påverkades vid olika hastigheter. Testet genomfördes genom att superkondensatorn laddades till full kapacitet samtidigt som fordonet hölls stilla för att förhindra att det körde iväg under uppladdningen. När superkondensatorn var fulladdad släpptes fordonet och den körda sträckan registrerades. Under testets gång loggades även regulatorns utsignal, superkondensatorns aktuella laddningsstatus, den uppmätta hastigheten samt referenshastigheten på datorn. Dessa värden loggades sedan på alla följande tester. Den körda sträckan visades dessutom i realtid på displayen i centralenheten.

När fordonet stannade avlästes och antecknades den totala körsträckan från displayen. Testet genomfördes för två fordon. Testförutsättningarna presenteras i Tabell 3 och resultaten från testet presenteras i Figur 23. De exakta mätningarna återfinns i Bilaga A.2.

Tabell 3: Testförutsättningar test 2

Parameter	Värde
Fordon 1 vikt	220 g
Fordon 2 vikt	307 g
Kapacitans	5 F
Hastighet	Varierande
Laddning	Ingen laddning under färd



Figur 23: Körsträcka som funktion av hastighet med kapacitans 5 F

Vid den sista mätningen överhettades motorn på fordon 2 och havererade. Därav genomfördes resterande mätningar endast med fordon 1.

4.3 Test 3 - Varierande laddningslängd raksträcka - 5 F

I test 3 testades hur körsträckan påverkades av laddningslängden. Fordonet kördes på den hastighet som resulterade i längst körsträcka, vilken från test 2 kunde bestämmas till 1.1 m/s. Testet genomfördes genom att superkondensatorn laddades upp till full kapacitet, därefter släpptes fordonet i väg och när fordonet stannade antecknades den aktuella körsträckan. Laddningssträckan ökades gradvis utefter de fästpunkter för sensorer som fanns på banan. Testförutsättningarna presenteras i

Tabell 4 och resultatet från testet presenteras i Tabell 5. Vid den sista mätningen observerades ingen förändring i laddningscykler och därav bedömdes fordonet kunna köras mycket långt och testet avbröts.

Tabell 4: Testförutsättningar test 3

Parameter	Värde
Fordonsvikt	220 g
Kapacitans	5 F
Hastighet	1.1 m/s
Laddning	Varierande på raksträcka

Tabell 5: Mätvärden varierad laddningslängd raksträcka

Mätning	Laddningssträcka [cm]	Körsträcka [m]
18	0	36.55
41	35.2	91.11
42	52.0	450.00+

4.4 Test 4 -Varierande laddningslängd kurva - 5 F

I test 4 testades olika laddningslängder i kurvor vid den optimala hastigheten 1.1 m/s. Testförutsättningarna presenteras i Tabell 6 och resultatet presenteras i Tabell 7. Vid den sista mätningen observerades ingen förändring i laddningscykler och därav bedömdes fordonet kunna köras mycket långt och testet avbröts.

Tabell 6: Testförutsättningar test 4

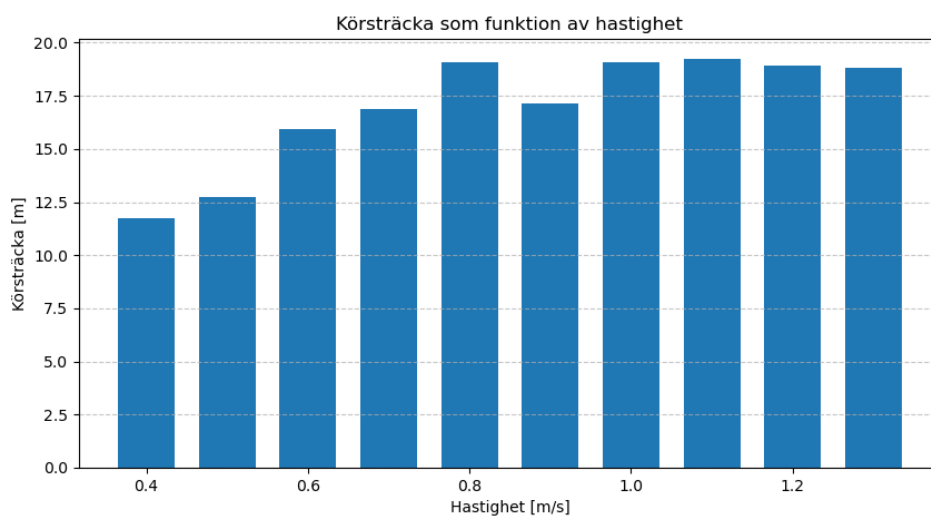
Parameter	Värde
Fordonsvikt	220 g
Kapacitans	5 F
Hastighet	1.1 m/s
Laddning	Varierande i kurva

Tabell 7: Tabellvärden vid de olika mätningarna

Mätning	Laddningssträcka [cm]	Körsträcka [m]
51	18.5	82.11
52	37.0	309.10
53	55.0	450.00+

4.5 Test 5 - Hastighet och körsträcka - 2.35 F

I test 5 genomfördes liknande test som test 2, det vill säga superkondensatorn laddades fullt och körsträckan testades vid olika hastigheter. Skillnaden från test 2 är att kapacitansen sänktes till 2.35 F samt att testet endast genomfördes på ett fordon. Resultatet av testet presenteras i Figur 24 och de exakta mätvärdena återfinns i Bilaga A.3.



Figur 24: Körsträcka som funktion av hastighet

4.6 Test 6 - Varierande laddningslängd raksträcka - 2.35 F

I test 6 genomfördes liknande tester som i test 3 med den enda skillnaden att kapacitansen sänktes till 2.35 F i test 6. Resultatet presenteras i Figur 25 och de exakta mätningarna återfinns i Bilaga A.4.



Figur 25: Körsträcka som funktion av hastighet

4.7 Test 7 - Varierande laddningslängd kurva - 2.35 F

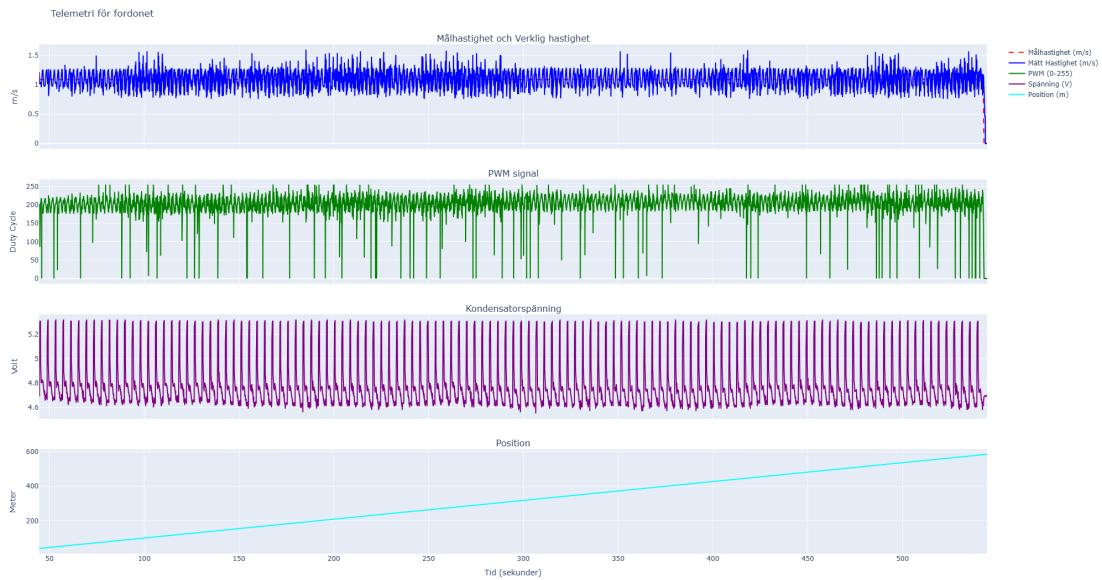
I test 7 genomfördes tester med varierande laddningslängd i kurvorna. Detta är liknande test som test 4 med enda skillnaden att kapacitansen i test 7 sänktes till 2.35 F. Resultaten för testet presenteras i Tabell 8.

Tabell 8: Tabellvärden vid de olika mätningarna

Mätning	Laddningssträcka [cm]	Körsträcka [m]
31	37.0	25.50
32	55.0	456.44

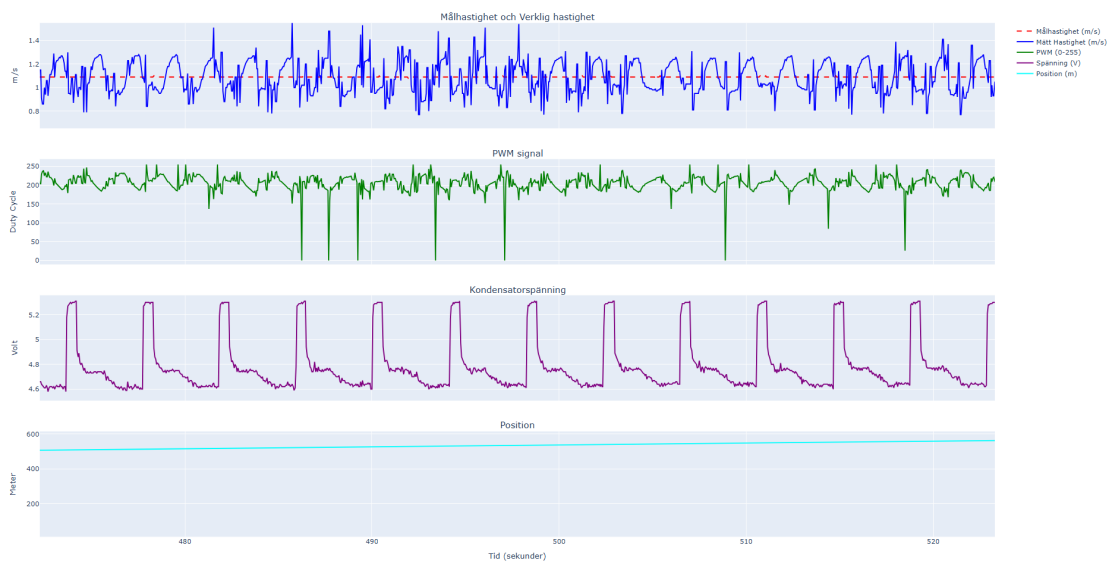
4.8 Energilagring och drift

I grafen i Figur 26 visas hur hastigheten, PWM-signalen, kondensatorspänningen och positionen för fordonet förändras över tid under en testkörning med 55 cm laddningssträcka. Kondensatorspänningen sjunker under körningen och ökar vid varje laddningssträcka till samma ursprungsnivå. Minsta spänningen sjunker aldrig under 4.5 V under drift och därför antas oändlig drift vid given laddningssträcka. Regulatorn justerar kontinuerligt PWM-signalen för att hålla hastigheten nära referensvärdet. Absolutbeloppet av medelavvikelsen räknades ut med ett Python skript i Bilaga B.3. Det gav att genomsnittliga absoluta avvikelsen från referenshastigheten uppmättes till 11.1% med en standardavvikelse på $\pm 12.8\%$.



Figur 26: Graf för mätvärden vid laddning i kurva för 55 cm laddningssträcka

I samma graf, inzoomad i Figur 27, kan kondensatorns spänningsnivå ses tydligare. Varje laddningspuls syns som en spänningshöjning följt av en gradvis urladdning, i takt med att fordonet förbrukar energi under varvet. PWM-signalen varierar där värdet kan utfalla mellan 0 och 255 och motsvarar fordonets gaspådrag.



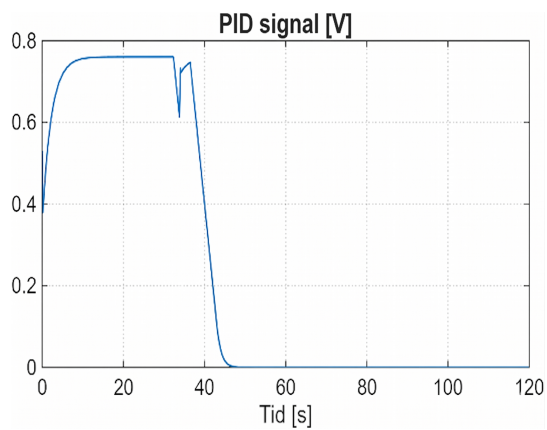
Figur 27: Inzoomad graf för mätvärden vid laddning i kurva för 55cm laddningssträcka

4.9 Simuleringsresultat

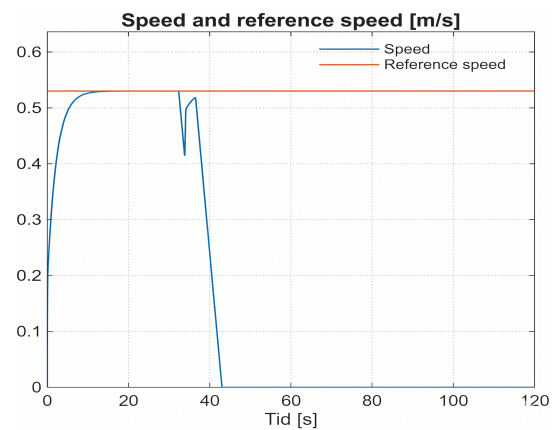
I detta delkapitel presenteras simuleringsresultat från tre tester som genomfördes för att utvärdera hur väl simuleringsmodellen återger det fysiska systemets beteende. Testerna varierar laddningssträckans längd och studerar hur detta påverkar kondensatorspänningen, fordonets hastighet och körsträcka.

4.9.1 Simuleringstest 1 - ingen laddning från elväg

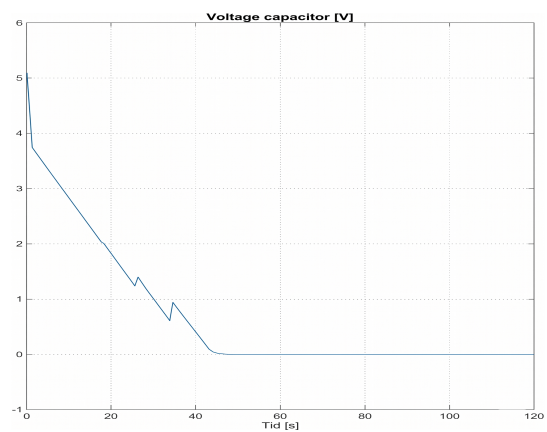
I det första testet tillförs ingen laddning från elvägen, vilket innebär att fordonet enbart drivs av den initialt fulladdade kondensatorn. Testet syftar till att undersöka regulatorns funktion samt hur kondensatorspänningen och körsträckan utvecklas över tid. Av Figur 28b framgår att fordonet håller referenshastigheten väl under de första 35 sekunderna. Därefter sjunker hastigheten snabbt då kondensatorn laddas ur, vilket även syns i Figur 28c där spänningen faller till noll efter ungefär 40 sekunder. Av positionsgrafan i Figur 28d framgår att fordonet kunde köra drygt tre varv, motsvarande cirka 13 m, innan det stannade.



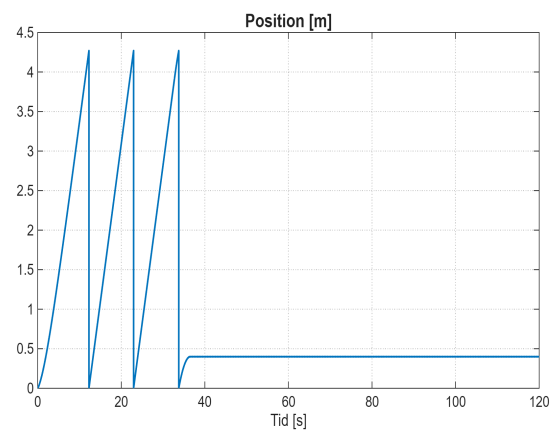
(a) PID-signal från regulatorn



(b) Simulerad hastighet och referenshastighet



(c) Spänning i kondensatorn

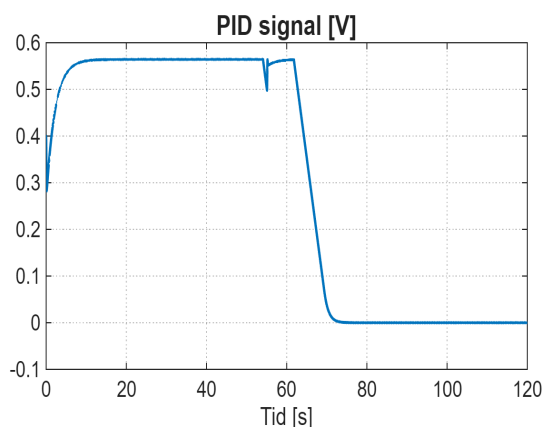


(d) Bilens position på banan

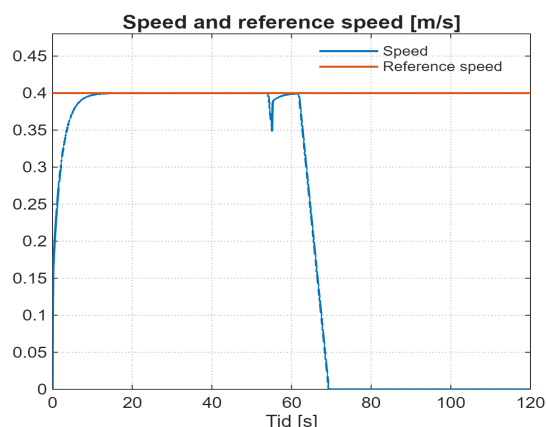
Figur 28: Resultat simuleringstest ett

4.9.2 Simuleringstest 2 - laddningssträcka 35.2 cm

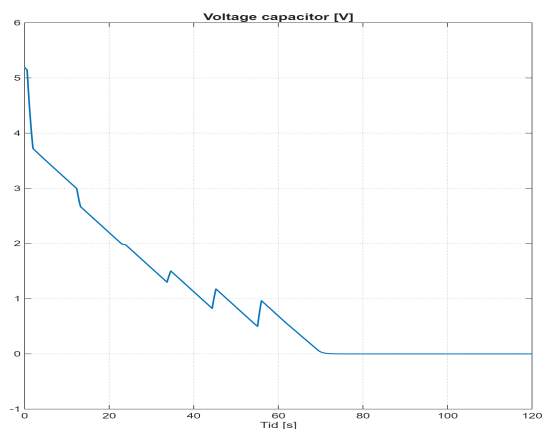
I det andra testet elektrifierades en sträcka på 35.2 cm, med start efter att fordonet kört 10 cm. Av Figur 29b framgår att fordonet höll referenshastigheten väl under de första 55 sekunderna, varefter hastigheten sjönk snabbt. Kondensatorspänningen, se Figur 29c, minskade kontinuerligt under testets gång och nådde noll efter ungefär 70 sekunder. Av positionsgrafan i Figur 29d framgår att fordonet körde nästan sex varv, motsvarande cirka 25.6 m, innan det stannade.



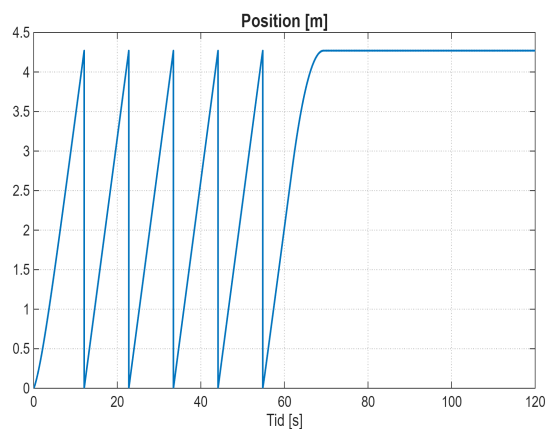
(a) PID-signal från regulatorn



(b) Simulerad hastighet och referenshastighet



(c) Spänning i kondensatorn

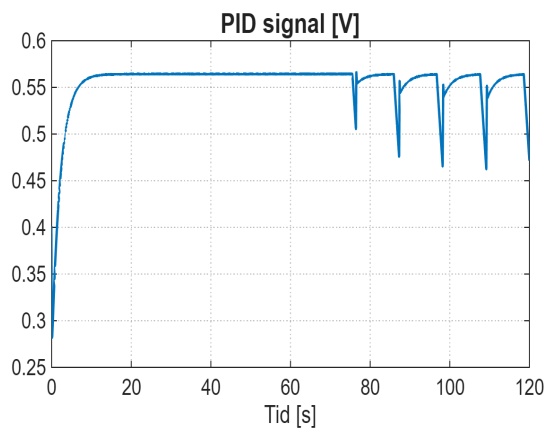


(d) Bilens position på banan

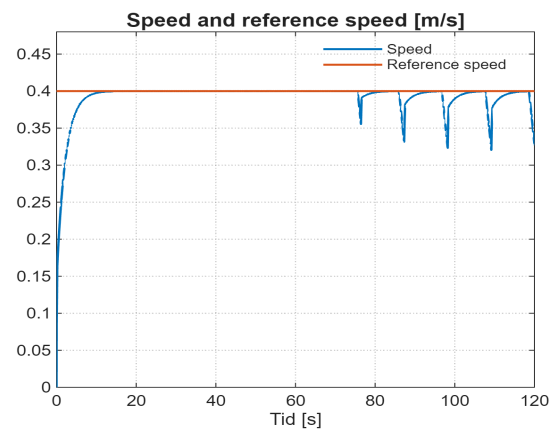
Figur 29: Resultat simuleringstest två.

4.9.3 Simuleringstest 3 - laddningssträcka 52 cm

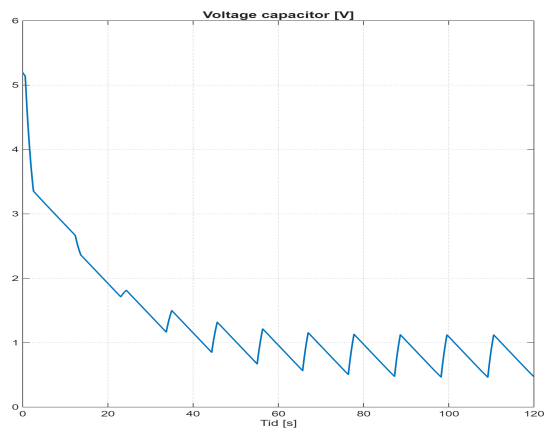
I det tredje testet ökades laddningssträckan till 52 cm, medan övriga förutsättningar var identiska med föregående test. Av positionsgrafan i Figur 30d framgår att fordonet fortsatte köra under hela simuleringens 120 sekunder. Kondensatorspänningen i Figur 30c uppvisar ett oscillerande mönster där spänningen sjunker mellan laddningstillfällena men återhämtar sig vid varje passering av den elektrifierade sträckan. Av Figur 30b framgår att fordonets hastighet periodvis sjunker under referenshastigheten, vilket även återspeglas i PID-signalen i Figur 30a.



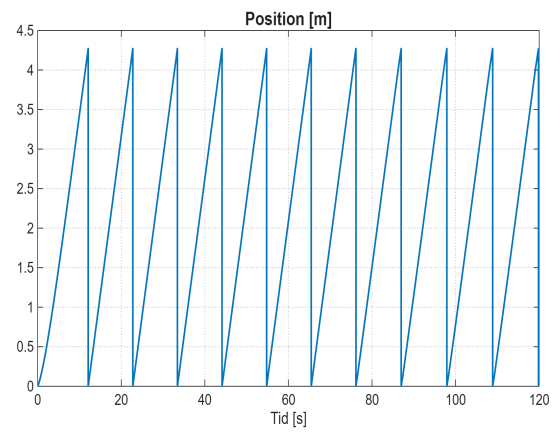
(a) PID-signal från regulatorn



(b) Simulerad hastighet och referenshastighet



(c) Spänning i kondensatorn



(d) Bilens position på banan

Figur 30: Resultat simuleringstest tre.

5 Diskussion

I denna del diskuteras och analyseras resultaten av de olika mätningar som genomförts med stöd av den tekniska teorin och simulering. Projektets arbetsprocess utvärderas även, där möjliga felkällor och brister identifieras och analyseras i syfte att förbättra framtida arbeten.

5.1 Analys av resultat

I detta avsnitt analyseras och tolkas resultaten från de genomförda testerna. Syftet är att identifiera orsaker till observerade beteenden samt koppla resultaten till den underliggande teorin.

5.1.1 Estimerad körsträcka

I test 1 undersöktes hur väl fordonen kunde estimera sin körda sträcka. Tidigt i projektet observerades att hastighetsmätningen genererade ett antal spökvärden. Eftersom körsträckan beräknas genom integration av hastigheten ledde detta till att den uppmätta körsträckan avvek kraftigt från den faktiska körsträckan. Detta hade även påverkat tillförlitligheten i efterföljande tester, då resultaten blivit starkt beroende av förekomsten av dessa slumpmässiga spökvärden.

För att minska problemet implementerades ett filter som eliminerade en stor del av spökvärdena. Detta förbättrade mätnoggrannheten avsevärt, vilket visualiseras i Figur 22. Resultaten visar att fordonen genomgående överskattade den körda sträckan med några centimeter jämfört med den faktiska körsträckan. Det genomsnittliga felet för fordon 1 var 3.84% och för fordon 2 var den 3.96% enligt Bilaga A.1. Anledningen till att fordonen estimerade att de hade kört något längre än den faktiska sträckan berodde troligtvis på testets utformning. För att stoppa fordonen hölls en hand vid stoppunkten. När fordonen körde in i handen åkte fordonen tillbaka några centimeter, vilket också registrerades av systemet och inkluderades i den uppmätta körsträckan.

5.1.2 Hastighetens påverkan på körsträcka

I test 2 undersöktes hur fordonets hastighet påverkade den maximala körsträckan med en kapacitans på 5 F. I test 5 genomfördes liknande test fast med en

kapacitans på 2.35 F. Den initiala hypotesen var att lägre hastigheter skulle resultera i längre körsträckor, vilket innebar att en negativ trend mellan hastighet och körsträcka förväntades. Resultaten för test 2 och 5 visar däremot att körsträckan initialt ökade med hastigheten. En möjlig förklaring till detta är att hastighetsmätningen genererade fler spökvärden vid låga hastigheter. Dessa spökvärden gjorde att regulatorn tillfälligt uppfattade hastigheten som betydligt högre än den verkliga hastigheten, vilket ledde till kraftiga inbromsningar. När fordonet därefter skulle accelerera igen krävdes ett högt moment för att åter nå referenshastigheten. Dessa upprepade accelerationer och inbromsningar kan ha resulterat i en högre energiförbrukning jämfört med vid en jämn hastighet.

En annan möjlig förklaring till att körsträckan ökade med hastigheten är att DC-motorn arbetar närmare sitt optimala verkningsgradsområde vid högre varvtal. Enligt Monolithic Power Systems [7] har en DC-motor en specifik punkt där verkningsgraden är som högst, kallad max efficiency, samt en rekommenderad driftpunkt, kallad rated operating point. Vid driftpunkter långt under dessa, det vill säga vid låga hastigheter och lågt belastningsmoment, är verkningsgraden lägre vilket innebär att en större andel av den tillförda energin går förlorad som värme snarare än att omvandlas till rörelse. Detta stämmer väl överens med resultaten i test 2 och 5, där körsträckan ökade med hastigheten upp till cirka 1.1 m/s. Vid denna hastighet bedöms fordonet ha arbetat nära sitt optimala verkningsgradsområde, varefter ytterligare hastighetsökning inte gav någon märkbar förbättring av körsträckan.

Det går även att utläsa från Figur 23 att hastigheten hade en mindre påverkan på det större fordonet. En möjlig förklaring är att det större fordonets högre massa medför större mekaniska förluster och ett högre belastningsmoment på motorn. Detta innebär att motorströmmen hålls relativt hög även vid förändrade varvtal. Därmed får den minskade strömförbrukning som kan uppstå vid högre vinkelhastigheter, till följd av ökad back-EMF, mindre påverkan på den totala energiförbrukningen. Resultatet blir att körsträckan påverkas mindre av hastighetsförändringar för det större fordonet.

Vid den sista mätningen havererade motorn på fordon 2. En möjlig förklaring är att motorn utsattes för hög belastning under en längre tid, främst på grund av fordonets höga vikt. Samtidigt användes ett egenkonstruerat underrede som saknade ventilation för kylning av motorn. Detta begränsade värmeavledningen och medförde sannolikt att temperaturen i motorn steg successivt under testets gång. Kombinationen av hög arbetsbelastning och otillräcklig kylning bedöms därför ha orsakat att motorn överhettades och slutligen havererade.

5.1.3 Laddningssträckans påverkan

I test 3 och 6 undersöktes hur laddningssträckans längd på raksträckor påverkade fordonets körsträcka. Skillnaden mellan testerna var att test 3 genomfördes med kapacitansen 5 F medan test 6 genomfördes med 2.35 F. Hypotesen var att längre laddningssträckor skulle ge längre körsträckor, vilket resultaten också visar.

För båda kapacitanserna ökade körsträckan successivt med ökande laddningssträcka. Vid kortare laddningssträckor var ökningen relativt begränsad, men när laddningssträckan blev tillräckligt lång ökade körsträckan kraftigt. För 2.35 F uppnåddes detta vid ungefär 106 cm laddningssträcka, medan motsvarande beteende för 5 F uppstod redan vid cirka 52 cm. I båda fallen stabiliserades superkondensatorns laddningscykler över tid, vilket indikerar att energitillförseln närmade sig fordonets energiförbrukning. Testerna avbröts vid 450 m trots att fordonen sannolikt hade kunnat fortsätta längre. Detta visualiseras i Figur 26 där spänningstrenden kan ses över tid och det kan observeras att den aldrig sjunker under 4.5 V. Detta observerades i samtliga tester där körsträcka registrerades till 450+m.

Test 4 och 7 genomfördes på motsvarande sätt som test 3 och 6, men med laddningssegmenten placerade i kurvor istället för på raksträckor. Resultaten visar en tydlig trend: laddning i kurvor ger betydligt längre körsträcka än laddning på raksträckor vid jämförbara laddningslängder. För kapacitansen 5 F gav laddning på raksträcka (test 3, laddningslängd 35.2 cm) en körsträcka på 91.11 m, medan motsvarande laddning i kurva (test 4, laddningslängd 37.0 cm) gav en körsträcka på 309.10 m, en nästan trefaldig ökning med marginell skillnad i laddningslängd. Samma trend observerades för kapacitansen 2.35 F, där laddning på raksträcka (test 6, laddningslängd 52 cm) resulterade i en körsträcka på 55.23 m, jämfört med 456.44 m vid laddning i kurva (test 7, laddningslängd 55 cm).

Ett undantag observerades vid en av de kortare körningarna i test 6 och 7, där trenden var omvänd. En laddningslängd på 35.2 cm i raksträcka med kapacitansen 2.35 F gav en körsträcka på 34.14 m, medan en laddningslängd på 37 cm i kurva gav 25.50 m. Denna mätning bedöms dock vara en avvikelse snarare än ett representativt resultat eftersom alla andra tester pekar på att laddning i kurva ger en längre körsträcka.

En trolig förklaring till att laddning i kurvor är mer effektiv är att fordonet färdas långsammare i kurvor till följd av högre friktionskrafter. Detta innebär att fordonet uppehåller sig under längre tid på det elektrifierade segmentet, vilket möjliggör större energiöverföring per längdenhet enligt ekvation (8). Sammantaget visar resultaten att laddningssegmentets placering har en betydande påverkan på systemets energidynamik och körsträcka, som är en konsekvens av tiden spenderad på

laddningssegmentet.

5.1.4 Analys av superkondensatorns storlek

I test 5, 6 och 7 genomfördes motsvarande experiment som i test 2, 3 och 4, med den enda skillnaden att kapacitansen minskades från 5 F till 2.35 F. I test 2 och test 5 undersöktes hur långt fordonet kunde köra vid olika hastigheter när superkondensatorn initialt var fulladdad och ingen laddning skedde under färd.

I kapitel 2.3.3 visas att den lagrade energin i en kondensator är proportionell mot kapacitansen vid konstant spänning (10). En ökning i kapacitansen från 2.35 F till 5 F medför därför en teoretisk energiökning enligt:

$$\frac{5 - 2.35}{2.35} \cdot 100 \approx 113\% \quad (21)$$

Detta innebär att den lagrade energin, och därmed även den förväntade körsträckan, teoretiskt bör öka med ungefär 113 %. Utifrån detta beräknades ett estimerat värde för körsträckan genom att multiplicera resultaten från test 2 med faktorn 2.13.

I Tabell 9 jämförs de estimerade värdena med de uppmätta resultaten från test 5. Resultaten visar att körsträckan ökade ungefär proportionellt med kapacitansen, vilket överensstämmer väl med teorin. Samtidigt kan det observeras att körsträckan inte ökar exakt med den teoretiska faktorn 2.13. Detta är dock förväntat eftersom systemet påverkas av flera störningar och felkällor. Den största felkällan bedöms vara hastighetsmätningen, som tidigare visats generera spökvärden vilket leder till oscillationer vid gaspådrag. Eftersom förekomsten av dessa störningar varierar mellan olika körningar blir även energiförbrukningen något varierande mellan testerna.

Trots dessa variationer visar resultaten tydligt att en ökning av energilagrets kapacitans leder till en ökad körsträcka, vilket stärker sambandet mellan lagrad energi och fordonets räckvidd.

Tabell 9: Jämförelse mellan uppmätt och estimerad körsträcka vid olika kapacitanser

Hastighet [m/s]	Test 5 (2.35 F) [m]	Estimerad (+113 %) [m]	Test 2 (5 F) [m]	Fel [%]
0.4	11.74	25.01	24.89	0.48
0.5	12.74	27.14	27.67	1.92
0.6	15.95	33.97	32.45	4.68
0.7	16.90	35.99	29.64	21.42
0.8	19.07	40.61	33.56	21.01
0.9	17.12	36.47	31.56	15.56
1.0	19.09	40.65	35.70	13.87
1.1	19.22	40.93	36.55	11.98
1.2	18.94	40.33	36.12	11.66
1.3	18.81	40.05	35.64	12.37

5.2 Simuleringsmodell

Simuleringsmodellen har i detta projekt fyllt en viktig roll som ett kvalitativt verktyg för att förstå systemets dynamik. En av de mest betydelsefulla förbättringarna jämfört med föregående års modell var implementeringen av en PID-regulator. Utöver detta har simuleringsmodellen uppdaterats med kurvsegment med estimerad friktion, vilket fanns i den tidigare modellen men inte fungerade som avsett.

Resultaten från simuleringstest 3 visar att modellen kan identifiera det tillstånd där energitillförseln ungefär matchar förbrukningen, vilket resulterar i att fordonet fortsätter köra under hela simuleringstiden. Detta beteende observerades även i de fysiska testerna, dock krävdes en laddningssträcka på 106 cm i verkligheten jämfört med 52 cm i simuleringen. Skillnaden förklaras dels av att flera parametrar i modellen behövt uppskattas, och dels av att den fysiska motorn inte kan drivas vid spänningar under 2 V. Detta är ett område där simuleringsmodellen avviker från verkligheten. I Figur 28c syns att kondensatorspänningen i simuleringen oscillerar mellan 0.5–1 V utan att fordonet stannar, vilket inte stämmer överens med det fysiska systemets beteende då det kräver 2V för att drivas framåt.

Simuleringstest 2 och 3, där laddningssträckan varieras med en kondensatorkapacitans på 2.35 F, kan jämföras med de fysiska testerna i avsnitt 4.6, där laddningslängden på raksträckan varierades under motsvarande förhållanden. I Tabell 10 presenteras en kvantitativ jämförelse mellan simulering och testresultaten, där det framgår att simuleringen konsekvent underskattar den erforderliga laddningssträckan men återger det kvalitativa beteendet väl.

Tabell 10: Jämförelse mellan simulering och fysiska tester vid varierande laddningssträcka (kapacitans 2.35 F)

Laddningssträcka [cm]	Simulering [m]	Fysiskt test [m]
0.0	13.0	19.22
35.2	25.6	34.14
52.0	Kontinuerlig	55.23
106.0	–	450+

Sammantaget återger simuleringsmodellen systemets kvalitativa beteende väl, men den kvantitativa precisionen begränsas av uppskattade parametervärden, främst motorparametrar och friktionskrafter. Modellen bör därför i första hand betraktas som ett verktyg för kvalitativ analys snarare än exakt kvantitativ prediktion.

5.3 Kravuppfyllnad

I detta avsnitt diskuteras i vilken utsträckning projektets uppställda krav uppfylls.

5.3.1 Elektrifiering av vägbanan

Kravet på elektrifiering av vägbanan anses vara uppfyllt. Laddningssträckans längd och placering kan enkelt varieras, även om hela banan elektrifieras så upplever fordonet det som att endast en specifik sektion är elektrifierad. Justerbara fästen för sensorerna som styr elektrifieringen av banan konstruerades, vilket möjliggör den enkla justeringen av laddningssträckan. Konstruktionen ger stor flexibilitet för användaren och möjliggör enkla studier av hur olika laddningssträckor påverkar systemets beteende och prestanda.

5.3.2 Laddning och lagring av energi

Kravet var att fordonet ska kunna ladda upp och lagra energi under drift. Energi-lagret skulle även vara enkelt utbytbar. Detta är uppfyllt, resultaten visar tydligt att fordonet kan lagra energi under drift. Energilagret är också enkelt att byta ut och det fungerar med olika storlekar vilket skapar flexibilitet hos användaren.

5.3.3 Hastighetsreglering

Kravet kopplat till hastighetsreglering anses vara uppfyllt. Kravet var en felmarginal från referensvärdet på 15%. Medelvärdet av felet landade på 11% enligt kapitel 4.8, vilket innebär att kravet överträffades. Däremot var standardavvikelsen $\pm 12.8\%$ vilket indikerar på att felet varierar kraftigt. Detta syns också tydligt i Figur 26. Därmed finns fortfarande stora förbättringsområden kopplat till hastighetsregleringen vilket analyseras i senare kapitel.

5.3.4 Design och konstruktion av fordon

Kravet kopplat till fordonens konstruktion anses vara delvis uppfyllt. Två fordon av olika storlekar konstruerades och båda fungerade vid körning på den elektrifierade vägbanan. Ena fordonets motor överhettades dock, troligtvis på grund av avsaknad av ventilation i underredet och fordonets större massa. Elektroniken i båda fordonen är delvis inkapslad, bortsett från kondensatorerna. Detta gjordes dock med avsikt då utbytbarheten bedömdes väga tyngre än inkapslingen, just för kondensatorerna.

5.3.5 Visualisering av data

Systemet visualiserar data via centralenheten i realtid och detta krav är således uppfyllt. Displayerna som är fästa på centralenheten visar aktuell hastighet, laddning i superkondensatorn, referenshastigheten samt hur långt fordonet har kört, vilket innebär att kravet överträffades.

5.3.6 Simuleringsmodell

Kravet kopplat till simuleringsmodellen anses vara uppfyllt. Simuleringsmodellen har byggts upp från grunden med stora delar från den ursprungliga modellen återanvända. I den nya simuleringsmodellen har en PID-regulator integrerats vilket gör det möjligt att testa olika hastigheter. Utöver detta så har den nya simuleringsmodellen anpassats för att bättre efterlikna systemet, trots att det inte speglar verkligheten perfekt.

5.4 Förbättringar och framtida arbete

Detta avsnitt behandlar möjliga vidareutvecklingar av arbetet, potentiella förbättringar samt övriga reflektioner.

5.4.1 Segmentera elektrifieringen av vägbanan

I framtida projekt vore det fördelaktigt att dela upp banan i flera elektrifierade segment. På så sätt kan olika delar av banan aktiveras oberoende av varandra. Detta är särskilt intressant eftersom elektrifiering av olika segment kan påverka fordonets laddningsdynamik. Det kan också vara relevant att undersöka andra elektrifieringskombinationer, exempelvis effekterna av att elektrifiera båda kurvorna eller vartannat segment. Med en segmenterad elektrifiering skapas betydligt större möjligheter att enkelt variera och optimera tester.

5.4.2 Implementera höjdskillnader

I den nuvarande modellen antas vägbanan vara helt plan, vilket innebär att variationer i höjddled inte beaktas. Detta är en förenkling jämfört med verkliga vägar, där lutningar påverkar fordonets energiförbrukning och dynamik. För att öka modellens realism skulle bilbanan kunna modifieras med höjdvariationer. Detta möjliggör studier av hur gravitation och förändrad belastning påverkar hastighet och energiflöde. Även simuleringsmodellen i Simulink kan utökas med dessa effekter genom att inkludera lutningar i systemets dynamik, vilket ger en mer verklighetsnära analys.

5.4.3 Data för motor och friktion

De parametrar som använts i simuleringsmodellen för att beskriva DC-motorn och friktionen har uppskattats. För framtida projekt hade det därför varit fördelaktigt att använda en motor med väl dokumenterade specifikationer. Om motorns elektriska parametrar samt kretsens resistanser kan bestämmas med hög noggrannhet blir det lättare att identifiera vilka parametrar som främst påverkar avvikelser mellan simulering och verklighet. Friktionen kan då optimeras genom att justera friktionsparametrarna tills simuleringsmodellen överensstämmer bättre med det fysiska systemets beteende.

5.4.4 Hastighetsreglering

Att implementera en hastighetsregulator var en central målsättning i projektet, eftersom det möjliggör analyser av hur systemets beteende påverkas av olika hastigheter. Den implementerade regleringen fungerar övergripande väl, men det finns flera områden med tydlig förbättringspotential.

Ett viktigt utvecklingsområde är hastighetsmätningen. I projektet användes en Hall-effektsensor, men denna genererade periodvis slumpmässiga spökvärden vilket påverkade regleringen genom oönskade accelerationer och inbromsningar. För framtida projekt kan det därför vara relevant att undersöka alternativa sensortekniker eller mer robusta mätmetoder.

Även regulatorns mjukvara kan vidareutvecklas. PID-parametrarna bestämdes experimentellt genom a tester, men en mer djupgående teoretisk analys och systematisk trimning hade sannolikt kunnat förbättra reglerprestandan ytterligare. Under testerna observerades dessutom att regulatorn fungerade olika bra vid olika hastigheter. Detta indikerar att det kan vara fördelaktigt att anpassa regulatorparametrarna beroende på fordonets aktuella hastighet.

Ytterligare ett förbättringsområde är filtreringen av felaktiga mätvärden. I den nuvarande implementationen användes filter för att eliminera orimliga hastighetsvärden, men vissa spökvärden passerar fortfarande genom systemet. Mer avancerade filtreringsmetoder hade därför kunnat bidra till en stabilare och mer tillförlitlig reglering.

6 Slutsats

I detta projekt har en modulär skalmodell av ett elvägssystem framgångsrikt utvecklats och vidareutvecklats till en fungerande testplattform för studier av dynamisk laddning under färd. Två fordon konstruerades delvis från grunden i form av en personbil och en lastbil med eget underrede och chassi. Båda fordonen uppfyller delvis de uppställda kraven med inkapslad elektronik och utbytbara energilager baserade på superkondensatorer. Resultaten visar att systemet möjliggör laddning under färd och att laddningssträckans längd och placering har stor inverkan på fordonets räckvidd. En längre laddningssträcka gav en markant ökad körsträcka, och vid tillräcklig laddningslängd uppnåddes ett drifttillstånd där spänningen aldrig sjönk under lägsta nivån för kontinuerlig drift. Det framgick även att laddning i kurvor är betydligt mer effektivt än laddning på raksträckor, vilket förklaras av att fordonets lägre hastighet i kurvor ger en längre kontakttid med den elektrifierade sektionen och därmed större energiöverföring. En ökad kapacitans hos superkondensatorn resulterade i en körsträcka som ökade i linje med teorin om att lagrad energi är proportionell mot kapacitansen. Simuleringsmodellen kunde återge systemets övergripande beteende och fungerade som ett användbart verktyg för kvalitativ analys. Modellen innehåller uppskattade parametrar, bland annat för motorns induktans och friktionskrafter, vilket begränsar dess kvantitativa noggrannhet. De identifierade begränsningarna, framför allt spökvärden i hastighetsmätningen, påverkade regulatorns prestanda och ledde till viss variation i de experimentella resultaten. Trots detta uppfyller systemet majoriteten av de uppställda kraven och utgör en väl fungerande plattform för fortsatta studier av elvägssystem i mindre skala.

Referenser

- [1] IPCC, “Climate change 2022: Mitigation of climate change. chapter 10: Transport,” 2022, available at: <https://www.ipcc.ch/report/ar6/wg3/chapter/chapter-10/>.
- [2] International Energy Agency, “Global ev outlook 2025: Trends in electric car markets,” <https://www.iea.org/reports/global-ev-outlook-2025/trends-in-electric-car-markets-2>, 2025, hämtad: 2026-04-21.
- [3] D. Wenander, F. J. Márquez-Fernández, and M. Alaküla, “Modelling electric transients in a conductive electric road system,” in *2023 AEIT International Annual Conference (AEIT)*, 2023, pp. 1–6. [Online]. Available: <https://ieeexplore.ieee.org/document/10330352>
- [4] C. Rosin, F. Djokic, F. Johansson, H. Karlsson, L. Bengtsson, and R. Bradosti, “Skalmodell av elvägssystem,” 2025.
- [5] J.-S. Zheng, N. Qin, X. Guo, L.-M. Jin, and J. P. Zheng, “High energy density supercapacitors: electrode material, electrolyte and energy density limitation principle; [;],” *Cailiao Gongcheng/Journal of Materials Engineering*, vol. 48, no. 9, p. 47 – 58, 2020, cited by: 6. [Online]. Available: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85091557188&doi=10.11868%2fj.issn.1001-4381.2019.000721&partnerID=40&md5=ae7b9275e1ba524aca35d0e3cc05cfc3>
- [6] B. Lennartsson, *Reglerteknikens grunder*, 4th ed. Lund: Studentlitteratur, 2002.
- [7] Monolithic Power Systems, “Fundamentals of DC motors,” <https://www.monolithicpower.com/en/learning/mpscholar/electric-motors/dc-motors/fundamentals>, hämtad: 2026-05-11.

A Tester

A.1 Test 1

Mätning	Hastighet [m/s]	Körsträcka F1 [m]	Fel F1 [%]	Körsträcka F2 [m]	Fel F2 [%]
1	0.4	4.39	2.81	4.28	0.23
2	0.5	4.31	0.94	4.35	1.87
3	0.6	4.33	1.41	4.40	3.04
4	0.7	4.36	2.11	4.33	1.41
5	0.8	4.53	6.09	4.52	5.85
6	0.9	4.41	3.28	4.50	5.39
7	1.0	4.42	3.51	4.38	2.58
8	1.1	4.46	4.45	4.57	7.03
9	1.2	4.58	7.26	4.54	6.32
10	1.3	4.55	6.56	4.52	5.85
Medel			3.84		3.96

Mätvärden test 1

A.2 Test 2

Mätning	Hastighet [m/s]	Körsträcka fordon 1 [m]	Körsträcka fordon 2 [m]
11	0.4	24.89	10.92
12	0.5	27.67	9.52
13	0.6	32.45	11.82
14	0.7	29.64	12.01
15	0.8	33.56	12.06
16	0.9	31.56	12.71
17	1.0	35.70	12.39
18	1.1	36.55	12.30
19	1.2	36.12	11.67
20	1.3	35.64	12.21

Mätvärden test 2

A.3 Test 5

Mätning	Hastighet [m/s]	Körsträcka fordon 1 [m]
11	0.4	11.74
12	0.5	12.74
13	0.6	15.95
14	0.7	16.90
15	0.8	19.07
16	0.9	17.12
17	1.0	19.09
18	1.1	19.22
19	1.2	18.94
20	1.3	18.81

Mätvärden test 5

A.4 Test 6

Mätning	Laddningssträcka [cm]	Körsträcka [m]
18	0	19.22
21	35.2	34.14
22	52	55.23
23	71.2	102.68
24	106	450+

Mätvärden test 6

B Kodbas

B.1 Bilenheten

```
1 #include <esp_now.h>
2 #include <WiFi.h>
3
4 // Macadress för centralenheten, behövs för handshake mellan
   våra Esp32
5 uint8_t centralAddress[] = {0xE4, 0xB0, 0x63, 0xAF, 0x9F, 0
   xF8};
6
```

```

7 // Tidsenheter för Hall-effektsensor
8 volatile unsigned long lastTime = 0;
9 volatile unsigned long period = 0;
10
11 // PID Variabler
12 float Proportional = 600.0;
13 float integral = 200.0;
14 float Derivative = 20.0;
15
16 float integralSum = 0.0;
17 float previousError = 0.0;
18 unsigned long lastPITime = 0;
19
20 // Variabler för hjuluträkning
21 const float wheelDiameter = 0.022;
22 const float pi = 3.14159265;
23 float circumference = 0;
24 float newSpeed = 0;
25 float prevSpeed = 0;
26
27 float position = 0.0;
28
29 // Definiera Pins
30 const int pwmPin = D5;
31 const int hallPin = D2;
32 const int voltPin = A0;
33
34 // PWM Inställningar
35 const int pwmChannel = 0;
36 const int pwmFreq = 5000;
37 const int pwmResolution = 8;
38 int currentPWM = 0;
39
40 // Målhastigheten från centralen
41 float targetSpeedMPS = 0.0;
42
43 // Centralenhetens medskickade datastruktur
44 typedef struct data_from_central {
45     float targetSpeed;
46 } data_from_central;
47
48 data_from_central incomingCommand;
49
50 // Datastruktur för datan till centralenheten

```

```

51 typedef struct struct_message {
52     float speed;
53     float voltage;
54     int currentPWM;
55     float position;
56 } struct_message;
57
58 struct_message outgoingData;
59
60 // Timers
61 unsigned long lastSendTime = 0;
62 const int sendInterval = 10;
63
64
65 void OnDataRecv(const uint8_t * mac, const uint8_t *
66     incomingData, int len) {
67     if (len == sizeof(incomingCommand)) {
68         memcpy(&incomingCommand, incomingData, sizeof(
69             incomingCommand));
70
71         // Uppdaterar målhastigheten
72         targetSpeedMPS = incomingCommand.targetSpeed;
73     }
74 }
75
76 void IRAM_ATTR pulse() { // Filter för att få bort spökvärden
77     (debounce)
78     unsigned long now = micros();
79     unsigned long p = now - lastTime;
80     if (p > 25000) { // Pulser registreras endast om det
81         minst är 25000 micro sekunder mellan dem
82         period = p;
83         lastTime = now;
84     }
85 }
86
87 void setup() {
88     Serial.begin(115200);
89
90     ledcSetup(pwmChannel, pwmFreq, pwmResolution); //
91         Initiering av PWM styrning, med dess inställningar
92     ledcAttachPin(pwmPin, pwmChannel);
93     ledcWrite(pwmChannel, 0);
94 }

```

```

90 circumference = pi * wheelDiameter;
91
92 pinMode(hallPin, INPUT_PULLUP); // Initiering av
    halleffect sensorn
93 attachInterrupt(digitalPinToInterrupt(hallPin), pulse,
    FALLING); // Trigger vid falling edge, då körs pulse
    funktionen
94
95 analogReadResolution(12); // Gör att vi läser analogRead vä
    rden mellan 0 och 4095
96
97 WiFi.mode(WIFI_STA);
98
99 if (esp_now_init() != ESP_OK) {
100     Serial.println("esp_now_init() failed"); // För felsö
        kning
101     return;
102 }
103
104 esp_now_register_recv_cb(OnDataRecv);
105
106 esp_now_peer_info_t peerInfo;
107 memset(&peerInfo, 0, sizeof(peerInfo));
108 memcpy(peerInfo.peer_addr, centralAddress, 6);
109 peerInfo.channel = 1;
110 peerInfo.encrypt = false;
111
112 if (esp_now_add_peer(&peerInfo) != ESP_OK){
113     Serial.println("Connection failure"); // För felsökning
114     return;
115 }
116
117 lastPITime = millis();
118 }
119
120 void loop() {
121     unsigned long currentTime = micros();
122     unsigned long localPeriod;
123     unsigned long localLastTime;
124
125     // Läser inga interrupts när tiden fastställs
126     noInterrupts();
127     localPeriod = period;
128     localLastTime = lastTime;

```

```

129 interrupts();
130
131 // Hastighetsberäkning
132 if (currentTime - localLastTime < 300000 && localPeriod >
    0) { //Om 0.3 sekunder gått sedan senaste avläsningen
        antas hastigheten vara 0.
133     newSpeed = circumference / (localPeriod / 1000000.0);
134     // Filter, där hastigheter över 2.0m/s antas vara
        överkliga
135     if (newSpeed < 2.0){
136         newSpeed = 0.8* newSpeed + 0.2 *prevSpeed; // Lågpas
            filter som tar 80% av nya och 20% av gamla värdet,
            behövs för D verkan i regulatorn.
137         outgoingData.speed = newSpeed;
138         prevSpeed = newSpeed;
139     } else {
140         outgoingData.speed = prevSpeed;
141     }
142 } else {
143     outgoingData.speed = 0.0;
144 }
145
146
147
148 // PID Reglering
149 unsigned long currentMillis = millis();
150 float dt = (currentMillis - lastPITime) / 1000.0; // Räkna
    ut vårt tidssteg
151
152 if (dt > 0) {
153     if (targetSpeedMPS <= 0.0) { // Nollställer alla värden
        om målhastigheten är 0, så ingen gammal integralsumma
        påverkar
154         integralSum = 0;
155         previousError = 0;
156         currentPWM = 0;
157         prevSpeed = 0;
158         ledcWrite(pwmChannel, 255); // OBS pum signalen är
            inverterad, då vi har en inverterande Gate drivare.
            Därav innebär 255 egentligen 0
159     } else {
160         float error = targetSpeedMPS - outgoingData.speed; // R
            äknar ut P delen av regulatorn
161

```

```

162     integralSum += error * dt;           // Räknar ut I delen av
        regulatorn
163     float maxintegral = 255.0 / integral; // Clamping based
        antiwindup, som begränsar I delens bidrag till 255
164     if (integralSum > maxintegral) integralSum =
        maxintegral;
165     if (integralSum < -maxintegral) integralSum = -
        maxintegral;
166
167     float derivative = (error - previousError) / dt; // Räk-
        nar ut D delen av regulatorn
168
169     float output = (Proportional * error) + (integral *
        integralSum) + (Derivative * derivative); // Räknar
        en output av ingående P I D delar
170
171     currentPWM = (int)output; // Gör om decimaltalet till
        heltal
172     if (currentPWM > 255) currentPWM = 255;
173     if (currentPWM < 0) currentPWM = 0;
174
175     ledcWrite(pwmChannel, 255 - currentPWM); // Skickar ut
        signalen
176     // OBS pwm signalen är även här inverterad, då vi har
        en inverterande Gate drivare
177
178     previousError = error;
179 }
180     lastPITime = currentMillis;
181 }
182
183 // Positionsuträkning där s=s+v*t
184 position = position + dt * outgoingData.speed;
185 outgoingData.position = position;
186
187 // Spänningsmätning
188 int rawADC = analogRead(voltPin);
189 float pinVoltage = 0.0017 * rawADC; // Funktion för att ber-
        äkna spänning från ADC värdet 0 - 4095
190 outgoingData.voltage = pinVoltage;
191
192 // Skickar med PWM signalvärdet till centralenheten
193 outgoingData.currentPWM = currentPWM;
194

```

```

195 // Skickar all outgoing data med vårt sendInterval för att
      inte överbelasta enheten
196 if (currentMillis - lastSendTime >= sendInterval) {
197     lastSendTime = currentMillis;
198     esp_now_send(centralAddress, (uint8_t *) &outgoingData,
      sizeof(outgoingData));
199 }
200 }

```

B.2 Centralenheten

```

1 #include <esp_now.h>
2 #include <WiFi.h>
3 #include <Wire.h>
4 #include <Adafruit_GFX.h>
5 #include <Adafruit_SSD1306.h>
6
7 // OLED display inställningar
8 #define SCREEN_WIDTH 128
9 #define SCREEN_HEIGHT 64
10 #define OLED_RESET -1
11 #define SCREEN_ADDRESS 0x3C
12
13 Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire,
      OLED_RESET);
14
15 // Macadress till bilen
16 uint8_t carMacAddress[] = {0x20, 0x6E, 0xF1, 0x33, 0x42, 0x28
      };
17
18 // Data till bilen
19 typedef struct data_to_car {
20     float targetSpeed;
21 } data_to_car;
22
23 data_to_car outgoingCommand;
24
25 // Telemtridatan från bilen
26 typedef struct data_from_car {
27     float speed;
28     float voltage;
29     int currentPWM;

```

```

30     float position;
31 } data_from_car;
32
33 data_from_car incomingCarData;
34
35 // Variabel initiering
36 float inputSpeedMPS = 0.0;
37 float targetSpeedMPS = 0.0;
38 float carSpeedMPS = 0.0;
39 float carVoltage = 0.0;
40 int currentPWM = 0;
41 float position = 0;
42
43
44 // PIN initiering
45 const int signalOnPin = D5; // Signal in för att slå PÅ relä
    , från IR sensorn
46 const int signalOffPin = D6; // Signal in för att slå AV relä
    , från IR sensorn
47 const int relayPin = D7; // Pinne ut till reläet
48
49
50 // Timers
51 unsigned long lastPrintTime = 0;
52 const int printInterval = 50; // i ms hur ofta vi får
    Serialprints och display update
53 unsigned long lastSendTime = 0;
54 const int sendInterval = 50; // Skicka data var 50:e ms
55
56 esp_now_peer_info_t peerInfo;
57
58 // Funktioner
59 void OnDataSent(const uint8_t *mac_addr,
    esp_now_send_status_t status) { // Om sändningen lyckats kö
    rs funktionen
60     if(status == ESP_NOW_SEND_SUCCESS) {
61         targetSpeedMPS = inputSpeedMPS; // Endast om bilens enhet
            tar emot data bekräftar vi för användaren att
            referenshastigheten har ändrats
62     }
63
64 void OnDataRecv(const uint8_t * mac, const uint8_t *
    incomingData, int len) { // När datan tagits emot
    uppdaterar vi våra variabler

```

```

65  if (len == sizeof(incomingCarData)) {
66      memcpy(&incomingCarData, incomingData, sizeof(
        incomingCarData));
67      carSpeedMPS = incomingCarData.speed;
68      carVoltage = incomingCarData.voltage;
69      currentPWM = incomingCarData.currentPWM;
70      position = incomingCarData.position;
71  }
72  }
73
74  void setup() {
75      Serial.begin(115200);
76
77
78      pinMode(signalOnPin, INPUT_PULLDOWN); // Initiering av IR-
        sensor Pins
79      pinMode(signalOffPin, INPUT_PULLDOWN);
80
81      pinMode(relayPin, OUTPUT); // Initiering av relä pin
82      digitalWrite(relayPin, HIGH); // Reläet sätts på vid start
        för enklare möjlighet till laddning vid startsekvens
83
84      if(!display.begin(SSD1306_SWITCHCAPVCC, SCREEN_ADDRESS)) {
85          Serial.println(F("display.begin fail"));
86      } else {
87          display.clearDisplay();
88          display.setTextSize(2);
89          display.setTextColor(SSD1306_WHITE);
90          display.setCursor(0,20);
91          display.println("Loading...");
92          display.display();
93          delay(2000); // delay behövs för att systemet ska hinna
        starta upp olika del enheter innan WIFI enheten kan
        startas upp
94      }
95
96      WiFi.mode(WIFI_STA);
97
98      if (esp_now_init() != ESP_OK) {
99          Serial.println("esp_now_init()");
100         return;
101     }
102
103     esp_now_register_send_cb(OnDataSent);

```

```

104  esp_now_register_recv_cb(OnDataRecv); // Funktionerna
      OnDataSent resp OnDataRecv körs endast om sändingen har
      lyckats
105
106  memset(&peerInfo, 0, sizeof(peerInfo)); // Garbage
      collector funktion för att städa upp gamla värden i
      peerInfo
107  memcpy(peerInfo.peer_addr, carMacAddress, 6); //
      Informationen om mottagaren laddas in i peerInfo
      funktionen
108  peerInfo.channel = 0;
109  peerInfo.encrypt = false;
110
111  if (esp_now_add_peer(&peerInfo) != ESP_OK){
112      Serial.println("esp_now_add_peer fail");
113      return;
114  }
115 }
116
117 void loop() {
118
119     // Läser av proximity sensorerna och slår på och av reläet
      därefter
120     if (digitalRead(signalOnPin) == LOW) {
121         digitalWrite(relayPin, HIGH); // Slå PÅ reläet
122     }
123
124     if (digitalRead(signalOffPin) == LOW) {
125         digitalWrite(relayPin, LOW); // Slå AV reläet
126     }
127
128     unsigned long currentMillis = millis();
129     if (currentMillis - lastSendTime >= sendInterval) { //
      Timerfunktion för att sändingen inte ska behövas ta
      processorkraft i varje loop varu
130     lastSendTime = currentMillis; // Data skickas endast
      en gång inom varje sendinterval
131
132     float input = analogRead(A0); // Läser
      Referenshastighetsvärdet från potentiometern
133     inputSpeedMPS = (input / 4095.0) * 1.6; // Gör om
      analoga värdet från 0-4095 till 0-1.6 för att motsvara
      en referenshastighet

```

```

134     if (inputSpeedMPS < 0.1) inputSpeedMPS = 0.0; // Filter f
        ör att värdet inte ska fladdra kring 0
135
136     outgoingCommand.targetSpeed = inputSpeedMPS; // Datan som
        skickas till bilen
137
138     esp_now_send(carMacAddress, (uint8_t *) &outgoingCommand,
        sizeof(outgoingCommand)); // Send kommandot som
        skickar paketet med data
139 }
140
141 if (currentMillis - lastPrintTime >= printInterval) { // Ä
        ven här ett printintervall för att spara provessorkraft
142     lastPrintTime = currentMillis;
143
144     // Serial prints för felsökning och plottning
145     Serial.print("Mål:");
146     Serial.print(targetSpeedMPS, 2);
147     Serial.print(",");
148     Serial.print("Fart:");
149     Serial.print(carSpeedMPS, 2);
150     Serial.print(",");
151     Serial.print("Spänning:");
152     Serial.print(carVoltage, 2);
153     Serial.print(",");
154     Serial.print("PWM:");
155     Serial.print(currentPWM);
156     Serial.print(",");
157     Serial.print("Position:");
158     Serial.println(position);
159
160     // Presentation av informationen i OLED displayen
161
162     display.clearDisplay();
163
164     display.setTextSize(1);
165     display.setCursor(20, 0);
166     display.println("Telemetridata");
167
168     // Visar målhastigheten
169     display.setCursor(0, 18);
170     display.print("Target: ");
171     display.print(targetSpeedMPS, 2);
172     display.println(" m/s");

```

```

173
174 // Visar aktuell hastighet
175 display.setCursor(0, 30);
176 display.print("Speed: ");
177 display.print(carSpeedMPS, 1);
178 display.println(" m/s");
179
180 // Visar spänningen i kondensatorn
181 display.setCursor(0, 42);
182 display.print("Volt: ");
183 display.print(carVoltage, 2);
184 display.println(" V");
185
186 // Visar PWM signalvärdet
187 display.setCursor(0, 54);
188 display.print("PWM: ");
189 display.print(currentPWM);
190
191 // Visar Positionen i displayen
192 display.print(" Pos");
193 display.print(position, 2);
194
195 display.display();
196 }
197
198 }
199 }

```

B.3 Python skript för uträkning av avvikelser

```

import pandas as pd
import numpy as np

df = pd.read_csv('bil_data53.csv')

# Filtrera bort rader där referensen är noll eller nära noll
df = df[df['Mal_ms'] > 0.2].copy()

# Beräkna avvikelse
df['diff'] = df['Fart_ms'] - df['Mal_ms']
df['diff_pct'] = (df['diff'] / df['Mal_ms']) * 100

```

```

print(f"Antal mätpunkter:      {len(df)}")
print(f"Medel avvikelse:      {df['diff_pct'].mean():.2f}%")
print(f"Medel |avvikelse|
→ (MAE):{df['diff_pct'].abs().mean():.2f}%")
print(f"Standardavvikelse:     {df['diff_pct'].std():.2f}%")
print(f"Median avvikelse:      {df['diff_pct'].median():.2f}%")
print(f"Max avvikelse:         {df['diff_pct'].max():.2f}%")
print(f"Min avvikelse:         {df['diff_pct'].min():.2f}%")

```

B.4 MATLAB-kod för simuleringsmodell

Listing 1: Huvudskript: ElectricRoad20260423

```

%
*****

% ERS small-scale model. Parameter investigation
% Initieringsscript för ElectricRoad260421
%
*****

%% 1. Gemensamma parametrar
Vers = 10;           % Electric road voltage (5 - 10 V)
Psi = 4.5e-3;       % Motor flux linkage
Ldcm = 1e-5;        % Completely guessed inductance!!
gr = 3;             % Gear ratio motorspeed/wheel speed
                    confirmed
Rw = 0.011;         % Wheel radius, it should be measured
Rdcm = 0.35;        % Motor internal resistance 10
                    original uppmätt
Fcurve = 0.25;      % guessed friction force in the curves
Fstraight = 0.2;    % guessed friction force in the
                    straight parts
Rgen = 2/5.2;       % generell resistor to get the right
                    current
Vtarget = 0.4;      % target velocity

%% 2. Fordon (bil)
Rch_veh1 = 0.1;     % [ohm] Charge resistance vehicle 1

```

```

Rd_veh1 = 0.1;      % [ohm] Drive resistance vehicle 1
Mv_veh1 = 0.22;    % [kg] Vehicle mass vehicle 1
C_veh1  = 2.35;    % [F] Capacitance vehicle 1

%% 3. Bana fordon 1 (yttervarv)
Pos_veh1 = (1:1:4270)';
Ers_veh1 = zeros(length(Pos_veh1),1);
ERSend_veh1 = 452;      % 4270/12
                        356
Ers_veh1(100:ERSend_veh1, 1) = 1;

Curve_veh1 = zeros(length(Pos_veh1),1);
Curve1Start_veh1 = 1036;
Curve1End_veh1   = 2135;      % 1036 + 1100 -
                        1
Curve_veh1(Curve1Start_veh1:Curve1End_veh1, 1) = 1;

Curve2Start_veh1 = 3171;      % 2136 + 1035
Curve2End_veh1   = 4270;      % 3171 + 1100 -
                        1
Curve_veh1(Curve2Start_veh1:Curve2End_veh1, 1) = 1;

%% 4. Kompatibilitetsvariabler
Rch  = Rch_veh1;
Rd   = Rd_veh1;
Mv   = Mv_veh1;
C    = C_veh1;
Pos  = Pos_veh1;
Ers  = Ers_veh1;
Curve = Curve_veh1;

%% 5. Kör simuleringen
disp('Startar simulering av ElectricRoad260421 ... ');
simOut = sim('ElectricRoad260421');
disp('Simuleringen är klar! Skapar snygga grafer ... ');

%% 6. Plotter
logNames = simOut.logout.getElementNames;
numLogs = length(logNames);

if numLogs == 0

```

```

warning('Inga_sIGNALER_hittades!_Kontrollera_Log_
Selected_Signals"_i_Simulink. ');
else
if ~exist('Enskilda_Grafer', 'dir'), mkdir('
Enskilda_Grafer'); end

fig = figure('Color', 'w', 'Name', 'ERS_Analys', '
Position', [50 50 1200 850]);
tlo = tiledlayout('flow', 'Padding', 'compact', '
TileSpacing', 'compact');

colors = [
    0.00, 0.45, 0.74; % Deep Blue
    0.85, 0.33, 0.10; % Rust Orange
    0.93, 0.69, 0.13; % Golden Yellow
    0.49, 0.18, 0.56; % Purple
    0.47, 0.67, 0.19; % Apple Green
    0.30, 0.75, 0.93; % Sky Blue
    0.64, 0.08, 0.18 % Dark Red
];

for i = 1:numLogs
    element = simOut.logout.get(i);
    while isa(element, 'Simulink.SimulationData.Dataset
        '), element = element.get(1); end

    try
        dataObj = element.Values;
        t = dataObj.Time;
        y = dataObj.Data;
        signalName = element.Name;

        ax = nexttile;

        p = plot(t, y, 'LineWidth', 1.2);

        if size(y, 2) > 1
            for k = 1:size(y, 2)

```

```

        p(k).Color = colors(mod(k-1, size(
            colors,1))+1, :);
    end
else
    p.Color = colors(1, :); % Enkel-linjer blir
        alltid Deep Blue
end

axtoolbar(ax, {});
ax.Interactions = [];

% Styling
grid on; box on;
set(ax, 'GridLineStyle', ':', 'GridAlpha', 0.3,
    'FontSize', 9, 'FontName', 'Helvetica');
title(signalName, 'Interpreter', 'none', '
    FontSize', 11, 'FontWeight', 'bold');
xlabel('Tid_[s]');

if ~isempty(y) && all(isfinite(y), 'all')
    minY = min(y, [], 'all');
    maxY = max(y, [], 'all');
    if minY == maxY % Om signalen är helt plan
        (t.ex. referens)
        ylim(ax, [minY-0.5, maxY+0.5]);
    elseif contains(signalName, 'speed', '
        IgnoreCase', true)
        ylim(ax, [0, max(maxY*1.2, 0.1)]);
    end
end

% Legend-hantering
if size(y, 2) > 1
    if contains(signalName, 'speed', '
        IgnoreCase', true)
        labels = {'Speed', 'Reference_speed'};
    elseif contains(signalName, 'force', '
        IgnoreCase', true) || contains(
        signalName, 'friction', 'IgnoreCase',

```

```

        true)
        labels = {'Friction', 'Engine_force'};
    elseif contains(signalName, 'current', '
IgnoreCase', true)
        labels = {'ERS-current', 'Engine_
current'};
    else
        labels = arrayfun(@(x) sprintf('Signal_
%d', x), 1:size(y, 2), '
UniformOutput', false);
    end
    legend(labels, 'Location', 'best', '
FontSize', 8);
    legend boxoff;
end

% Spara enskild graf
filnamn = sprintf('Enskilda_Grafer/%s.png',
    regexp(signalName, '[^a-zA-Z0-9]', '_'));
exportgraphics(ax, filnamn, 'Resolution', 300);

catch ME
    fprintf('Kunde_inte_rita_signal_"%s":_%s\n',
        logNames{i}, ME.message);
end
end

title(tlo, 'Electric_Road_System:_Parameter_
Investigation', 'FontSize', 16, 'FontWeight', 'bold'
);
exportgraphics(tlo, 'ERS_Total_Samling.png', '
Resolution', 300);
disp('Klart!_Graferna_är_nu_tunnare_och_har_en_ny_fä
rgpalett. ');
end

```

B.5 Simuleringsmodellens parametrar

Tabell 1: Simuleringsmodellens parametrar

Parameter	Värde	Enhet	Källa
<i>Gemensamma parametrar</i>			
V_{ers}	10	V	Uppmätt
Ψ	4.5×10^{-3}	Wb	Uppskattad
L_{dcm}	1×10^{-5}	H	Uppskattad
g_r	3	–	Bekräftad
R_w	0.011	m	Bekräftad
R_{dcm}	0.35	Ω	Uppmätt
F_{curve}	0.25	N	Uppskattad
$F_{straight}$	0.20	N	Uppskattad
R_{gen}	$2/5.2 \approx 0.385$	Ω	Kalibrerad
V_{target}	0.4	m/s	Vald
<i>Fordonsparametrar</i>			
R_{ch}	0.1	Ω	Uppskattad
R_d	0.1	Ω	Uppskattad
M_v	0.22	kg	Uppmätt
C	2.35	F	Uppmätt
<i>Banparametrar</i>			
Banlängd	4270	mm	Uppmätt
ERS-start	100	mm	Uppmätt
ERS-slut	452	mm	Uppmätt
Kurva 1 start	1036	mm	Uppmätt
Kurva 1 slut	2135	mm	Uppmätt
Kurva 2 start	3171	mm	Uppmätt
Kurva 2 slut	4270	mm	Uppmätt

B.6 Simulink modell

C Bill of materials

Tabell 2: Bill of materials – Skalmmodell av elvägssystem

#	Komponent	Beskrivning / Specifikation	Antal
Fordon & Mekanik			
1	Scalextric bilbana	Modulär bilbana med raka och svängda segment, ledande metallskenor	1
2	Scalextric underrede (bil)	Befintligt underrede från Scalextric-system	1
3	Underrede lastbil (3D-utsk.)	Egenkonstruerat underrede designat i Catia V5	1
4	Chassi – lastbil	3D-utskrivet chassi i form av lastbil	1
5	Chassi – bil	3D-utskrivet bilchassi	1
6	DC-motor (Scalextric)	DC-motor med kugghjul på axeln, utväxling 3:1	2
7	Hjulaxel med kugghjul	Hjulaxel med större kugghjul och 2 gummi-hjul	2
8	Styrskena med metallfläta	Kombinerad styrskena och strömupptagare, tvåpinskontakt	2
9	Magneter	Håller fordon mot metallskenorna	4
10	Glidlager	Glidlager i snäppfäste för hjulaxel	4
Energilagring			
11	Superkondensator 4.7 F	4.7 F / 2.7 V (ger 2.35 F vid seriekoppling)	4
12	Superkondensator 10 F	10 F / 2.7 V (ger 5 F vid seriekoppling)	4
Elektronik – Fordonsenhet			
13	Arduino Nano ESP32	Mikrokontroller, dubbla kärnor, WiFi/BT/ESP-NOW	2
14	9 V batteri	Strömförsörjning till mikrokontrollern	2
15	Buck-converter LM2596	DC-DC omvandlare, stegar ner 15 V till 5.2 V	2
16	MOSFET IRL540N	N-kanal MOSFET för motorstyrning	2
17	Transistor BC338	Hjälptransistor / level shifter för gate-drivning	2
18	Diod 1N4001	Skyddsdiode mot felpolarisering	2
19	Diod SB150 (flyback)	Frihjulsdiod parallellkopplad över motor	2
20	Hall-effektsensor TO-92	Hastighetsmätning via magnet på hjulaxel	2
21	Magnet (Hall-sensor)	Fäst vid hjulaxeln för pulsgivning	2
22	Motstånd 2.2 k Ω	Används i drivkrets och spänningsdelare	6

Fortsätter på nästa sida

Tabell 2 – fortsättning

#	Komponent	Beskrivning / Specifikation	Antal
23	Spänningsdelare	Skalar ner kondensatorspänning till <3.3 V för ADC	2
Elektronik – Centralenhet			
24	Arduino Nano ESP32 (central)	Mikrokontroller för centralenheten	1
25	OLED-display SSD1306	0.96" display för realtidsvisualisering	2
26	Potentiometer	Justering av referenshastighet under körning	2
27	Relä SRD-05VDC-SL-C	Styr elektrifieringen av laddningssektionen	1
28	Transistor BC338 (relä)	Styr reläet via mikrokontrollern	1
29	Motstånd 1 k Ω	I serie med relästyrning	1
30	IR-sensor FC-51	Detekterar fordonets position för laddningsstyrning	2
31	Centralenhetslåda	Kapslad låda med OLED och potentiometrar	1
Mjukvara & Verktyg			
32	Arduino IDE	Utvecklingsmiljö för programmering av ESP32	1
33	MATLAB/Simulink	Simuleringsmodell implementerad i Simulink	1
34	Python	Realtidsplottning av mätdata via seriell port	1
35	Catia V5	CAD-program för 3D-modellering av underrede och chassi	1