

Real-Time Cadansaanpassing in een Automatische fietstransmissie

Arno Cools

Promotor:
Prof. M. Moens

Begeleider:
Ir. Tomas Keppens
Ir. Jorrit Heidbuchel
Ir. Rugen Heidbuchel

Proefschrift ingediend tot het
behalen van de graad van
Master of Toegepaste Informatica

Academiejaar 2018-2019

Voorwoord

A preface

Abstract

An abstract.

Inhoudstafel

Voorwoord	
Abstract	i
Lijst van figuren	iii
Lijst van afkortingen en symbolen	iv
1 Probleemstelling	1
1.1 Mobiliteitsvraagstuk	1
1.2 Online machine learning voor geïndividualiseerde cadanscontrole	2
1.3 Huidige systeem	3
2 Methode	5
2.1 De fietssimulatie	5
2.2 Modelleren van het fietserkoppel	5
2.3 Het fietsersmodel	9
2.4 Het lastmodel	11
2.5 Snelheidsvergelijking	11
3 Resultaten	14
4 Discussie	15

Lijst van figuren

1	Snelheid-veiligheid trade-off (bron: IntuEdrive)	1
2	Grootte van het voertuigenpark 2014-2018 (bron: statbel.fgov.be)	2
3	Blokdiagram van het fiets-fietser-controller systeem	4
4	Het koppel-toerentalkarakteristiek	7
5	Het koppelverloop van een mens (linksboven), de simulatie (rechtsboven) en een gesimuleerd dominant been (onderaan)	8
6	Planeetwielmechanisme (bron: wikipedia)	8
7	Verwacht cadansverloop in functie van de snelheid. Er wordt verwacht dat de eigenlijke cadans tussen het minimum en maximum ligt.	10
8	Voorbeeld helling verloop	12

Lijst van afkortingen en symbolen

Afkortingen

CVT continu variabele transmissie. 3

FCC freely chosen cadence. 3

Symbolen

FCC_{est} Schatting van FCC geleverd door de cadanscontroller. 4

K Agressiviteits parameter voor proportionele regelaar. 6

S Ondersteuningsniveau. 8

T_{MG2} Koppel geleverd door motor op het voorwiel. 8

$T_{cy,m}$ Gemeten koppel geleverd door de fietser. 4

T_{cy} Koppel geleverd door de fietser. 3

T_{rw} Koppel op het achterwiel. 7

α Helling. 4

ω_{cr} Cadans in rpm. 6

θ_{cr} Hoek van de trapas. 4

$k_{cr,r}$ Verhouding overbrenging trapas-ringwiel. 7

nr Aantal tanden op het ringwiel. 7

ns Aantal tanden op het zonnewiel. 7

u_c Knop die bepaald of de cadans veranderd. 3

v_{bike} Snelheid van de fiets. 4

v_{ref} Referentie snelheid van de fietser. 3

y Output vector fiets. 4

Hoofdstuk 1

Probleemstelling

1.1 Mobiliteitsvraagstuk

De auto is het slachtoffer geworden van zijn eigen succes: we staan meer dan ooit in de file en de CO₂ van personenverkeer stijgt jaar na jaar. De Belg neemt al snel de auto voor korte afstanden (< 25 km). In deze auto zit meestal maar 1 persoon. Het Belgische wagenpark blijft groeien (figuur 2). Hier zien we wel een trend ontstaan. Er worden steeds meer elektrische en hybride wagens verkocht, maar die staan natuurlijk net zo goed in de file. Mobiliteit op twee wielen kan hier een oplossing bieden.

Mobiliteit op twee wielen kennen we al lang: fietsen bestaan al sinds de 19de eeuw. Elektrische fietsen hebben het potentieel van deze tweewielers enorm verhoogd: fietsen wordt moeiteloos en stukken sneller. Spijtig genoeg neemt het risico op ongevallen ook toe bij hogere snelheid. Dat komt omdat e-bikes en speed e-bikes precies dezelfde technologie gebruiken als normale fietsen – grote wielen met smalle banden, kettingaandrijving met manuele versnellingen, mechanische handremmen – bij veel hogere snelheden. IntuEdrive noemt dit de snelheid-veiligheid trade-off. De veiligheid kan beperkt worden verhoogd door componenten toe te voegen (bv. Bosch e-bike ABS), maar de functionaliteit van deze systemen blijft beperkt. Er is een meer holistische aanpak nodig. Bovendien bieden elektrische fietsen vandaag nog niet het gebruiksgemak en de betrouwbaarheid die de consument gewend is van zijn wagen.

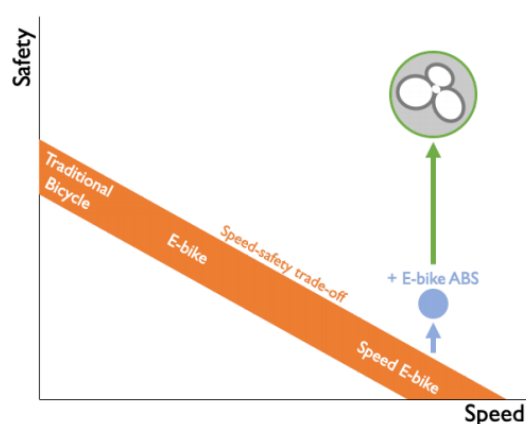


Figure 1: Snelheid-veiligheid trade-off (bron: IntuEdrive)

IntuEdrive's *CoSaR* is een snelle elektrische fiets die veiliger is dan de klassieke mechanische fiets, dankzij hun innovatie tweewielaandrijving en elektrische remfunctie. Dit systeem reduceert de stopafstand met 60% en maakt schakelen overbodig (automatische versnellingen). Het stapt ook af van de onderhoudsintensieve fietscomponenten (ketting, tandwielen, mechanische remmen). Dit maakt CoSaR de perfecte e-bike voor woon-werkverkeer: makkelijk, veilig en betrouwbaar.

Op 1 augustus van het jaar	2014	2015	2016	2017	2018
Personenwagens	5.555.499	5.623.579	5.712.061	5.785.447	5.853.782
- rijdend op benzine	2.029.688	2.091.327	2.199.038	2.335.349	2.518.942
- rijdend op diesel	3.458.424	3.457.526	3.424.592	3.339.034	3.193.658
- rijdend op gas	22.051	18.967	17.238	15.965	15.500
- met elektrische motor	1.792	2.871	4.368	6.552	9.244
- hybride	23.444	32.151	44.364	63.740	87.012
- niet nader bepaald	20.100	20.737	22.461	24.807	29.426
Inwoners per personenauto op 1 augustus (d)	2,01	1,99	1,97	1,96	1,94

Figure 2: Grootte van het voertuigenpark 2014-2018 (bron: statbel.fgov.be)

Door automatisch te schakelen zorgt CoSaR ervoor dat de fietser in elke situatie precies zo snel trapt als hij of zij wil. Deze gewenste trapsnelheid – of beter trapcadans – varieert van persoon tot persoon en hangt af van omstandigheden zoals helling, tegenwind en rijsnelheid. Omdat deze gewenste cadans niet op voorhand gekend is, schakelt de transmissie momenteel op basis van een vaste wetmatigheid die tijdens testen getuned is om voor zoveel mogelijk gebruikers comfortabel aan te voelen. Wijkt deze wetmatigheid af van de gewenste cadans van een specifieke gebruiker, dan kan deze gebruiker via knoppen op het stuur tijdens het fietsen zijn of haar cadans manueel aanpassen.

1.2 Online machine learning voor geïndividualiseerde cadanscontrole

Deze thesis werkt verder op het prototype geleverd door IntuEdrive. Zoals reeds aangehaald schakelt de fiets automatisch. De trapcadans wordt hierdoor stabiel gehouden, ook wanneer de fietser harder of zachter trapt. Het doel is om deze instelling te veranderen door de cadans van de e-bike in real time te voorspellen aan de hand van de toestand van de fiets, zodat de trapsnelheid zich aanpast aan de huidige omstandigheden en de individuele gebruiker. Deze implementatie zal ervoor zorgen dat de fietser meer aandacht kan besteden aan de weg, waardoor gevaarlijke situaties kunnen vermeden worden. Om de cadans te personaliseren en dynamisch te maken, zal een machine learning algoritme ontwikkeld worden dat de toestand van de fiets als input binnenkrijgt en hiermee de trapsnelheid berekent. Wanneer de fietser besluit om de cadans manueel aan te passen, interpreteert het algoritme dit als een signaal om bij te leren.

Om de performantie van het machine learning algoritme te testen zal het volledig systeem fiets-fietser-cadanscontrole gesimuleerd worden. Het fietsmodel wordt geleverd door IntuEdrive en zal geïmplementeerd worden in python. Vervolgens worden een aantal machine learning algoritmes vergeleken op basis van een aantal vooraf gedefinieerde performance indicatoren. De machine learning algoritmes zijn afkomstig uit scikit-learn, een machine learning library.

De cadanscontrole moet aan verschillende eisen voldoen. Het algoritme moet draaien op een Raspberry Pi, samen met het controleprogramma van de fiets. Door deze beperkte resources moet het algoritme zo efficiënt mogelijk zijn. De voorspellingen moeten bijna in real time berekend worden. Het doel is om aan 10Hz de cadans aan te passen, maar hoe meer voorspellingen per seconde, hoe beter. Tragere voorspellingen kunnen hinderlijk zijn voor het rijgedrag. Ten slotte moet er ook rekening gehouden worden met de veiligheid van de fietser. Opeenvolgende voorspellingen mogen niet te veel van elkaar verschillen, anders zou de fietser erdoor gestoord kunnen worden en zijn concentratie verliezen. Bovendien mag de cadans nooit hoger dan een bepaalde maximum limiet ingesteld worden.

De algoritmes worden geëvalueerd op basis van de mean squared error tussen de ingestelde cadans - afkomstig van het machine learning algoritme - en de zogenaamde *freely chosen cadence (FCC)* van de fietser. Die freely chosen cadence is niet precies gekend en wordt in de simulatie bepaald aan de hand van een fietsersmodel. Dit is een functie die de toestand van de fiets en de fietser (rijsnelheid, helling,...) afbeeldt op de trapsnelheid die de fietser in dat geval het meest comfortabel vindt. Simpel gezegd is het fietsersmodel een functie met als input de toestand van de fiets en als output een “optimale cadans”. Deze functie is speculatief en kan makkelijk aangepast worden. Op welke basis de fietser precies zijn freely chosen cadence bepaalt is voor dit onderzoek weinig relevant. Het gaat er hier vooral om dat het machine learning algoritme het fietsersmodel kan achterhalen.

$$\text{Fietsersmodel:} \quad fcc = f(\text{snellheid, koppel, vermogen, helling, ...})$$

Het algoritme moet kunnen bijleren met een kleine hoeveelheid data. De gebruiker zal immers niet vaak manuele aanpassingen doen aan de cadans. Te veel data gebruiken kan een negatieve invloed hebben op reeds correcte voorspellingen. Het algoritme moet ook snel bijleren. Elke verandering moet zo snel mogelijk doorgevoerd worden en moeten een betekenisvolle impact hebben.

1.3 Huidige systeem

De fiets van intuEdrive gebruikt een elektrische continu variabele transmissie (CVT) die ervoor zorgt dat er naadloos geschakeld kan worden tussen versnellingen, in tegenstelling tot het traditionele ketting-en-tandwiel systeem. Dit oude systeem schakelt in discrete trappen, waardoor de fietser tijdens het schakelen een discontinuïteit voelt. Het CVT-systeem gebruikt 2 motoren en schakelt traploos. Eén van de motoren regelt de trapcadans, de andere motor regelt het ondersteuningsniveau. Het ondersteuningsniveau bepaald hoeveel extra elektrisch vermogen er geleverd wordt, bovenop wat de fietser zelf levert.

Figuur 3 toont een blokdiagram van het systeem fiets-fietser-controller. We gaan ervan uit dat de fietser op elk moment een bepaalde referentiesnelheid (v_{ref}) probeert te halen, hier aangeduid met r . Deze kan variëren naar gelang de situatie, maar is voor elke gebruiker anders. Tijdens het fietsen geeft de fietser input aan de fiets. Zo kan hij of zij het geleverde koppel variëren (T_{cy}), i.e. meer of minder kracht op de pedalen zetten of de cadans aanpassen met de knoppen (u_c). Inputs en fysische toestand van de fiets

worden gemeten door sensoren op de fiets: het koppel ($T_{cy,m}$), de hoek van de trapas (θ_{cr}), snelheid (v_{bike}), helling (α), etc. T_{cy} en $T_{cy,m}$ zijn niet hetzelfde, want er kunnen fouten gebeuren tijdens het meten. De vector van meetwaarden (y) is input voor de fietscontroller. De fietscontroller stuurt de motoren in de E-bike aan op basis van de metingen y en de ingestelde referentiecaden. De cadanscontroller die in deze thesis uitgewerkt zal worden zal op basis van dezelfde metingen een gepersonaliseerde referentiecaden (FCC_{est}) voorspellen die als input dient voor de controller.

$$r = [v_{ref}] \quad u_{cy} = \begin{bmatrix} T_{cy} \\ u_c \end{bmatrix} \quad cc = [FCC_{est}] \quad y = \begin{bmatrix} \theta_{cr} \\ T_{cy,m} \\ v_{bike} \\ \alpha \end{bmatrix}$$

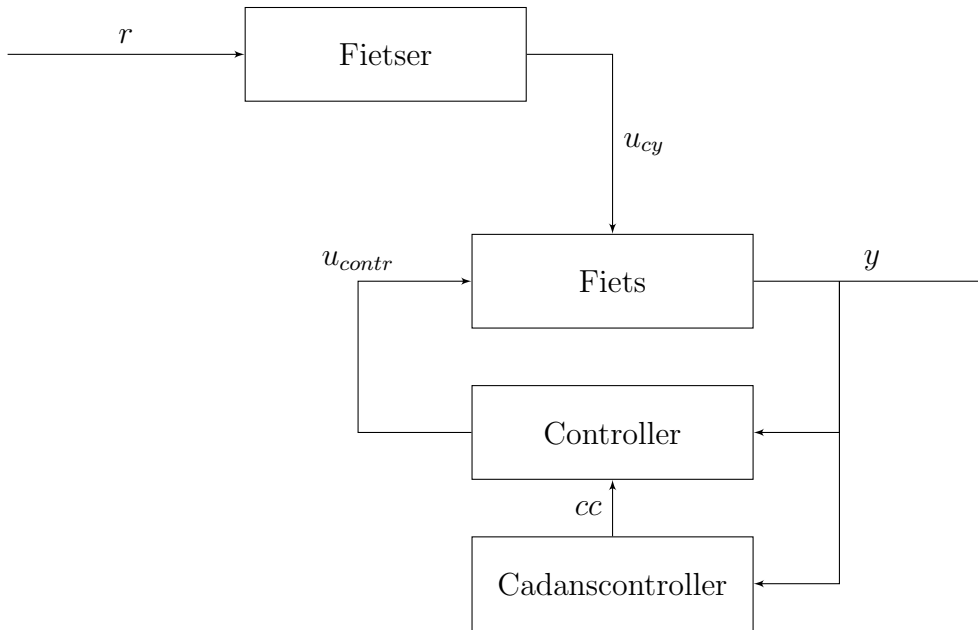


Figure 3: Blokdiagram van het fiets-fietscontroller systeem

Hoofdstuk 2

Methode

2.1 De fietssimulatie

Er wordt een simulatie gemaakt die de toestand van de fiets zo goed mogelijk probeert te benaderen. Er zal geen rekening gehouden worden met het manoeuvreren van de fiets of van tegenwind. Enkel de relevante meetwaarden worden bijgehouden. De simulatie is geparametriseerd om eenvoudig verschillende scenario's te testen.

Het voordeel van de fietssimulatie is de enorme flexibiliteit. Uren aan data kunnen in een moment tijd gegenereerd worden, waardoor het makkelijk is om verschillende tests uit te voeren. Hiervoor moeten slechts enkele instellingen aangepast worden. Het is ook mogelijk om slechts een enkele parameter aan te passen tijdens tests terwijl de rest constant blijft (*ceteris paribus*), wat praktisch onmogelijk is in een veldtest. Omdat de simulatie bovendien een duidelijke referentie genereert voor de FCC (output van het fietsersmodel), kan de performantie van de cadanscontroller op een kwantitatieve manier worden geëvalueerd. Tijdens een veldtest zou de fietser alleen kwalitatief kunnen aangeven of hij of zij de voorspelde cadans goed vindt.

2.2 Modelleren van het fietserkoppel

Het fietserkoppel wordt gemodelleerd als een sinusfunctie met twee pieken per omwenteling van de trapas (2 benen), met het DC koppel van de fietser als parameter.

$$T_{cy} = T_{dc}(1 + \sin(2\theta_{cr} - \frac{\pi}{6}))$$

Uit deze formule is het ook meteen duidelijk dat het DC koppel ook het vermogen-equivalent koppel is. Dat wil zeggen dat het DC koppel gedurende een volledige omwenteling van de trapas evenveel arbeid levert als het fietserkoppel.

$$\begin{aligned}
\int_0^{2\pi} T_{cy}(1 + \sin(2x - \frac{\pi}{6}))dx &= T_{cy} \int_0^{2\pi} (1 + \sin(2x - \frac{\pi}{6}))dx \\
&= T_{cy} \left[x - \frac{1}{2} \sin(\frac{1}{3}(6x + \pi)) \right]_0^{2\pi} \\
&= 2\pi T_{cy}
\end{aligned}$$

Het gemiddelde koppel geleverd door de fietser wordt gemodelleerd als een proportionele regelaar. Het doel is om een bepaalde snelheid, v_{ref} , te behalen. Hoe groter het verschil is tussen de referentie snelheid en de eigenlijke snelheid, hoe meer kracht er geleverd zal worden. Als deze referentie snelheid overschreden wordt, dan zal er geen koppel meer geleverd worden. Dit wordt ook wel freewheelen genoemd. Om de kracht van de actor te limiteren, wordt er een maximum koppel ingesteld naar gelang de huidige cadans (ω_{cr}). Zo wordt er meer kracht geleverd wanneer de cadans laag is, net zoals in de werkelijkheid. K bepaalt de agressiviteit van de regelaar. De formules zien er als volgt uit:

$$\begin{aligned}
T_{dc,max} &= \frac{-\omega_{cr}}{2} + 60 \quad (Figuur 4) \\
T_{dc} &= \min(T_{dc,max}, \max(0, -K * (v_{bike} - v_{ref})))
\end{aligned}$$

Figuren 5a en 5b tonen een menselijk koppelverloop en gesimuleerd koppelverloop, gesampled aan 10Hz. Zoals te zien is het gesimuleerde koppel heel consistent. Het menselijk koppel volgt duidelijk een cyclische functie, maar toont vormen van inconsistentie. Merk wel op dat er telkens een afwisseling is van een hoge en een lage piek. Dit wijst op een dominant been. Figuur 5c toont een gesimuleerd koppelverloop van een fietser met een dominant been.

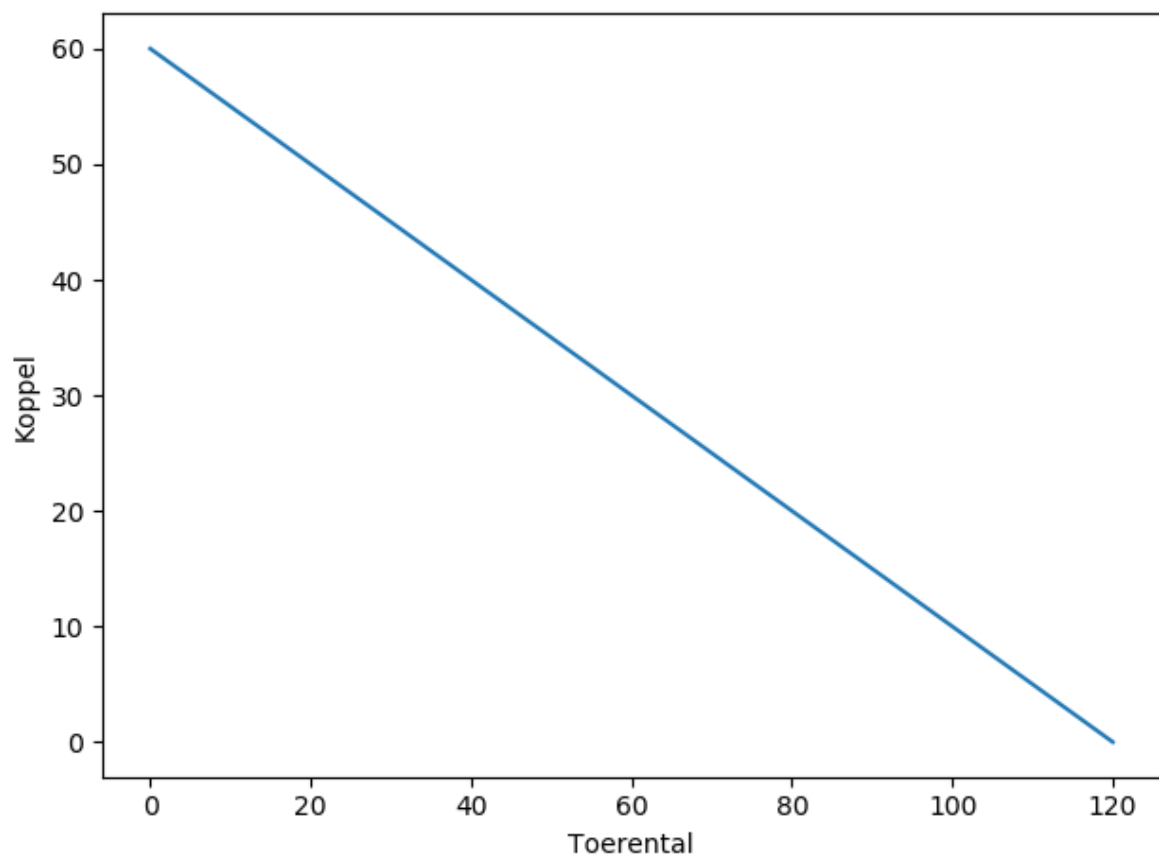


Figure 4: Het koppel-toerentalkarakteristiek

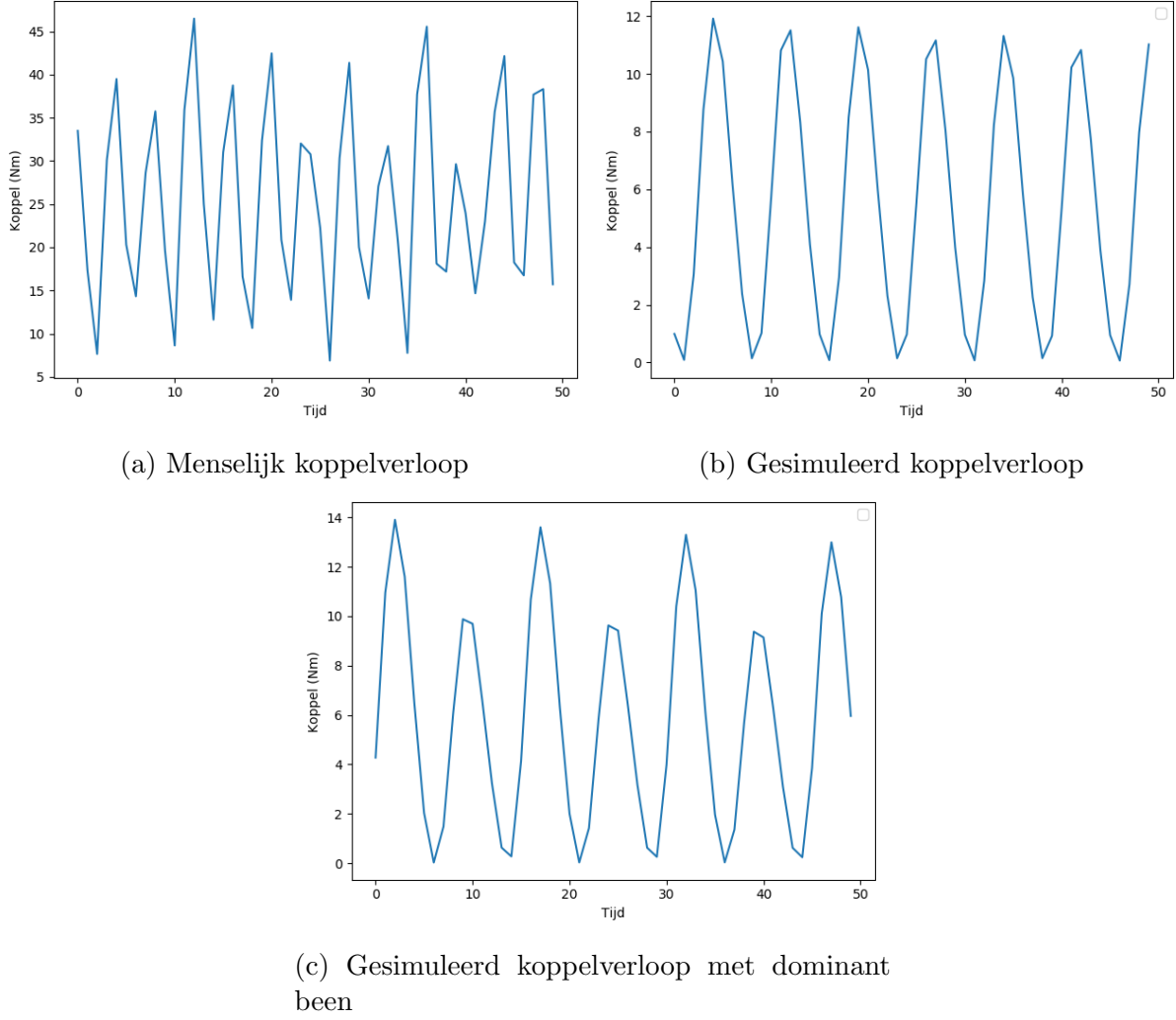


Figure 5: Het koppelverloop van een mens (linksboven), de simulatie (rechtsboven) en een gesimuleerd dominant been (onderaan)

T_{cy} is het koppel op de trapas. Dit moet nog overgebracht worden op het achterwiel. CoSaR maakt gebruik van een planeetwielmechanisme (figuur 6). Dit mechanisme laat toe om een grote overbrengingsverhouding te voorzien in een kleine ruimte. Het achterwiel-koppel wordt beïnvloed door het aantal tanden op het zonnewiel (1; ns) en het ringwiel (2; nr) en de overbrengingsverhouding tussen de trapas en het ringwiel ($k_{cr,r}$). Het koppel op het achterwiel (T_{rw}) ziet er als volgt uit:

$$T_{rw} = T_{cy} * k_{cr,r} * \frac{nr + ns}{nr}$$

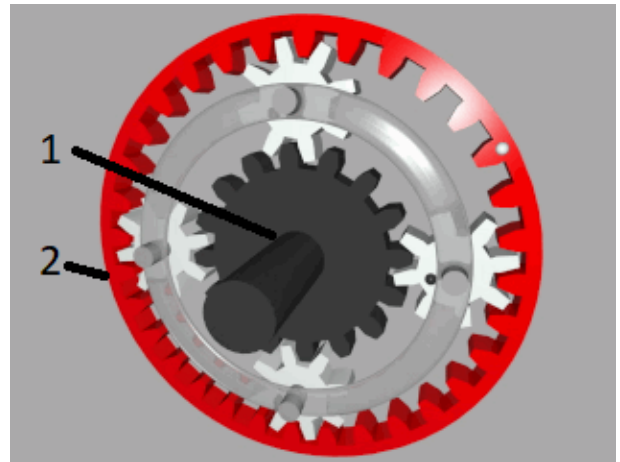


Figure 6: Planeetwielmechanisme (bron: wikipedia)

Bovenop het vermogen geproduceerd door de fietser, levert CoSaR extra ondersteuning a.d.h.v. een motor (T_{MG2}) gekoppeld aan het voorwiel. De fietser kan zelf een ondersteuningsniveau (S) instellen tussen 0 en 5. Hoe hoger dit ondersteuningsniveau, hoe minder inspanning de fietser moet leveren.

$$T_{MG2} = \min(35, S \cdot T_{cy})$$

2.3 Het fietsersmodel

Hoe kiest een fietser zijn cadans? Dit is voor elke fietser verschillend en er is nog nauwelijks onderzoek naar gebeurd. Wielrenners trainen om sneller te kunnen trappen omdat dit efficiënter is.. Ze kunnen een gemiddeld vermogen leveren van 300 Watt. De doorsnee fietser levert gemiddeld ongeveer 75 Watt tijdens een normale fietstocht. Het fietsersmodel zal hierop worden afgesteld, aangezien wielrenners niet de voornaamste doelgroep zijn voor CoSaR.

Het fietsersmodel is een functie die op verschillende manieren uitgedrukt kan worden: op basis van de helling, gemiddeld koppel, of een stabiel vermogen. Wat het correcte model is wordt in deze thesis niet uitgewerkt. Het is vooral van belang dat de cadanscontroller het model zo snel en zo nauwkeurig mogelijk kan achterhalen, ongeacht wat het model precies is. Hier wordt de volgende aanname gemaakt: hoe hoger het koppel geleverd door de fietser, hoe hoger de gewenste cadans. Wanneer de fietser bijvoorbeeld een helling oprijdt schakelt hij of zij een versnelling omlaag zodat de kracht die op de pedalen gezet moet worden aangenaam blijft. We stellen hier volgende eenvoudige modellen voor:

$$\begin{array}{ll} \text{Gemiddeld koppel :} & f_{cc} = f_k \cdot T_{dc} \\ \text{Helling :} & f_{cc} = f_h \cdot \alpha \\ \text{Snelheid :} & f_{cc} = f_v \cdot v_{bike} \end{array}$$

Er wordt verder aangenomen dat de fietser ook een zeker lineariteit verwacht bij lage snelheden. Dat wil zeggen dat een fietser het niet comfortabel vindt wanneer hij of zij snel moet trappen wanneer de fiets nog stilstaat of heel traag rijdt, ook al moet er op dat moment veel koppel geleverd worden om te kunnen vertrekken. Daarom wordt bij lage snelheden de FCC begrensd door een lineair oplopend maximum, te vergelijken met een mechanische fietsversnelling. Omdat de doorsnee fietser niet heel traag of heel snel trapt wordt de FCC begrensd tussen de 40 en 120 rpm.

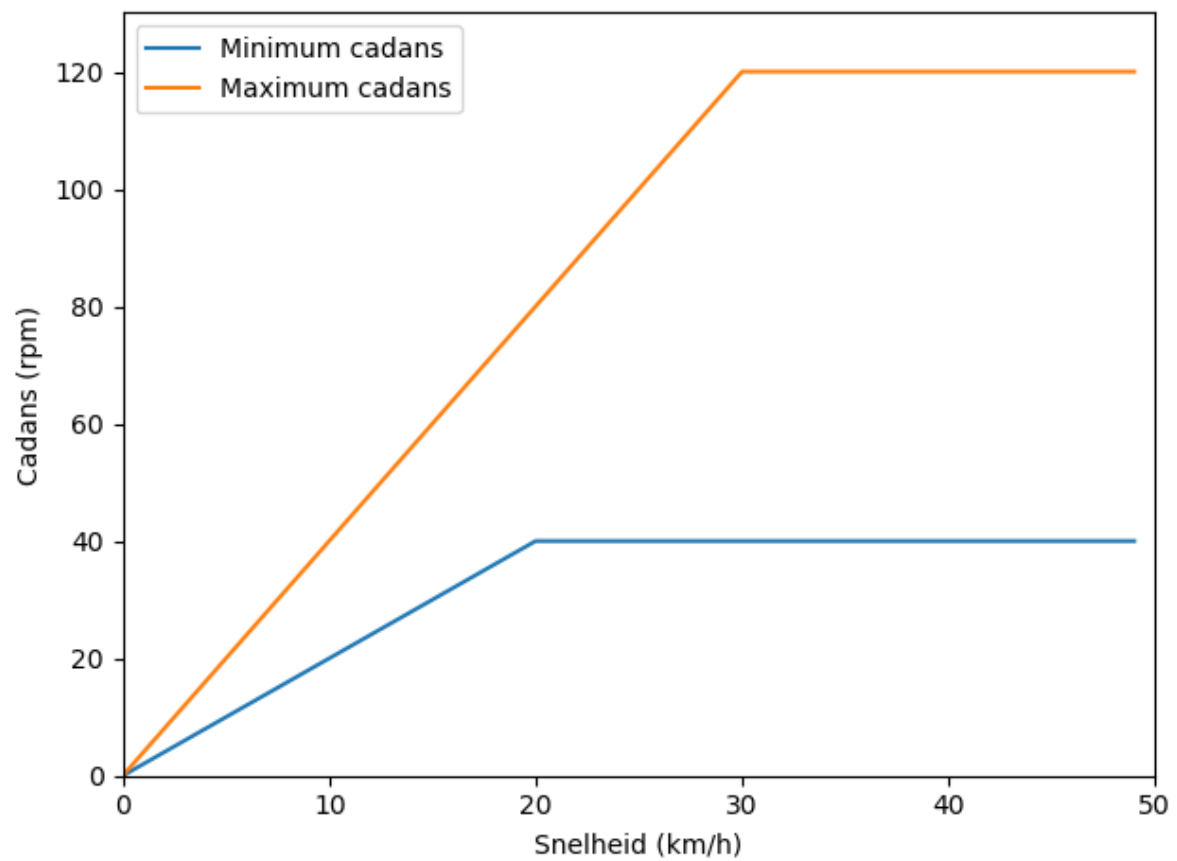


Figure 7: Verwacht cadansverloop in functie van de snelheid. Er wordt verwacht dat de eigenlijke cadans tussen het minimum en maximum ligt.

2.4 Het lastmodel

De simulatie is voorzien van een lastmodel. Zoals in realiteit, werken lasten in op de fiets. Zwaartekracht, wrijving met de weg en luchtweerstand zijn gemodelleerd als volgt:

$$\begin{aligned} F_{grav} &= m \cdot g \cdot \sin \alpha \\ F_{friction} &= m \cdot g \cdot c_r \cdot \cos \alpha \\ F_{aero} &= \frac{c_d \cdot \rho_{aero} \cdot A_{aero} \cdot v_{bike}^2}{2} \end{aligned}$$

Samen vormen ze de totale belasting op de fiets, die kwadratisch is in de snelheid.

$$F_{load} = F_{grav} + F_{friction} + F_{aero}$$

Deze lasten zorgen ervoor dat de simulatie een realistische hoeveelheid vermogen nodig heeft om een bepaalde snelheid te halen. Er wordt hier geen rekening gehouden met de wind. Ten eerste zou dit extra complexiteit toevoegen aan de simulatie. En ten tweede vermoeden we volgende hypothese:

De freely chosen cadence hangt af van de hoeveelheid last, van welke bron dan ook, die de gebruiker ondervindt en de gebruiker zelf.

Het voorgestelde lastmodel omvat deze vereiste. Door de helling en referentie snelheid te variëren ondergaat de fietser een veranderende last. Zoals in de realiteit zoeken mensen een bepaalde snelheid te halen. Wanneer de fietser een te hoge last ondervindt, bijvoorbeeld door een berg op te rijden, moet hij of zij meer vermogen genereren om zijn of haar gewenste snelheid te behouden. Hiervoor zijn 2 mogelijkheden: het verhogen van het koppel of de trapsnelheid. Mensen zijn meer geneigd om hun gewenste cadans te behouden, ongeacht het koppel (binnen bepaalde grenzen). De formule voor mechanisch vermogen gaat als volgt:

$$P = T_{cy} \cdot \omega_{cr}$$

Om het lastmodel correct te laten werken, moet er nog een helling gegenereerd worden. Om veel werk uit te sparen met het uitstippelen van parcours, wordt dit dynamisch gegenereerd met behulp van perlin noise. Perlin noise kan gebruikt worden om willekeurige getallen te genereren waarbij opeenvolgende getallen weinig van elkaar verschillen. Een perfecte kandidaat dus om terrein te simuleren. Om verschillende trajecten te creëren, kan de seed variabele aangepast worden. Een hellingsgraad wordt in de fietswereld vaak percentueel voorgesteld. Deze implementatie heeft echter radialen nodig. De helling zal beperkt worden tussen ≈ 0 en 10% (0 en 0.1 radialen). Ter vergelijking, de Koppenberg heeft een gemiddeld stijgingspercentage van 11.6%. Het minimum stijgingspercentage is zo gekozen dat de simulatie zo weinig mogelijk gaat freewheelen.

2.5 Snelheidsvergelijking

De snelheid wordt berekend met een standaardformule: vorige snelheid plus acceleratie met respect tot de genomen tijdsprong. De acceleratie is in functie van de last, het totaalgewicht (m), het vermogen geleverd door de fietser op het achterwiel (T_{rw}) en het

vermogen van een motor bevestigd op het voorwiel (T_{MG2}).

De bewegingsvergelijking van de fiets is, met inbegrip van het lastmodel en het fietsersmodel:

$$F = m \cdot a$$

Deze vergelijking wordt elke tijdsstap geïntegreerd met behulp van een voorwaartse Euler methode:

$$v_{bike}[h] = v_{bike}[h-1] + \Delta t \cdot \frac{1}{m} \cdot \left(\frac{T_{MG2} + T_{rw}}{r_w} - F_{load} \right)$$

De volledige simulatie ziet er als volgt uit:

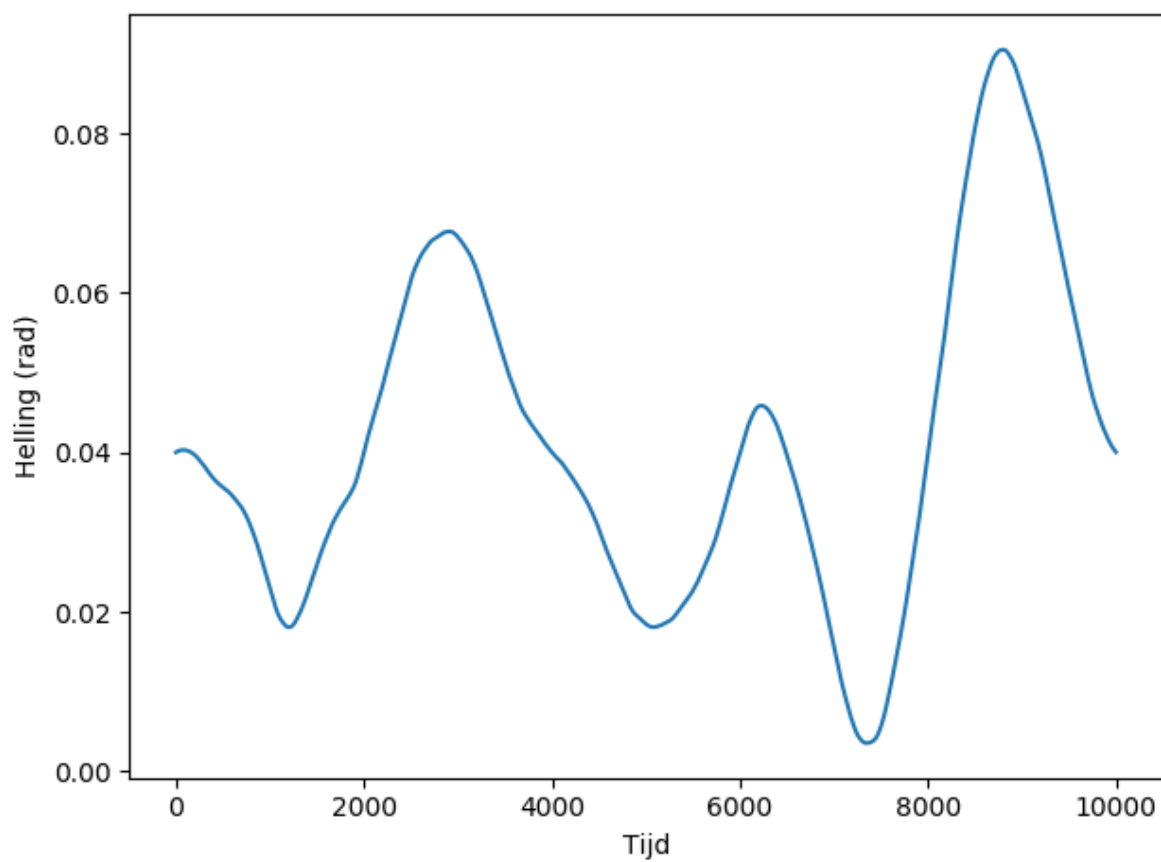


Figure 8: Voorbeeld helling verloop

```

for h in 1..#tijdssprongen
   $T_{dc,max} = \frac{-\omega_{cr}[h-1]}{2} + 60$ 
   $T_{dc} = \min(T_{dc,max}, \max(0, -K * (v_{bike}[h-1] - v_{ref})))$ 
   $f_{cc} = f(T_{dc})$ 
   $\omega_{cr} = \text{cadans}(v_{bike}[h-1], T_{dc}, f_{cc})$ 
   $\theta_{cr} = \theta_{cr}[h-1] + \Delta t \cdot \omega_{cr}$ 
   $T_{cy} = T_{dc}(1 + \sin(2\theta_{cr} - \frac{\pi}{6}))$ 
   $T_{rw} = T_{cy} * k_{cr,r} * \frac{nr+ns}{nr}$ 
   $T_{MG2} = \min(35, S \cdot T_{cy})$ 
   $F_{grav} = m \cdot g \cdot \sin \alpha$ 
   $F_{friction} = m \cdot g \cdot c_r \cdot \cos \alpha$ 
   $F_{aero} = \frac{c_d \cdot \rho_{aero} \cdot A_{aero} \cdot v_{bike}[h-1]^2}{2}$ 
   $F_{load} = F_{grav} + F_{friction} + F_{aero}$ 
   $v_{bike} = v_{bike}[h-1] + \Delta t \cdot \frac{1}{m} \cdot (\frac{T_{MG2} + T_{rw}}{r_w} - F_{load})$ 

```

Hoofdstuk 3

Resultaten

$$FCC_{est,t} = sf.FCC_{pred,t} + (1 - sf).FCC_{est,t-1}$$
$$FCC_{est,t} = \frac{\sum_{i=0}^n FCC_{pred,t-i}}{n}$$

$$x_t = \begin{bmatrix} \theta_{cr} \\ T_{cy,m} \\ v_{bike} \\ \alpha \end{bmatrix} \quad input = \begin{bmatrix} x_t \\ x_{t-1} \\ \dots \\ x_{t-n} \end{bmatrix}$$

$$y_t = w_t \cdot x_t$$
$$loss(w_{t+1}; (x_t, y_t)) \leq \epsilon$$

Hoofdstuk 4

Discussie

Computerwetenschappen
Celestijnenlaan 200 A bus 2402
3000 LEUVEN, BELGIË
tel. + 32 16 32 77 00
fax + 32 16 32 79 96
www.kuleuven.be

