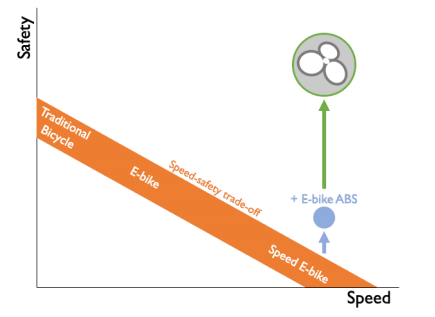
**Breder Kader: mobiliteitsvraagstuk**

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 (tabel ##). 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. Hierdoor nemen de risico’s toe. 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.

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 CoSaRde perfecte e-bike voor woon-werkverkeer: makkelijk, veilig en betrouwbaar.

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.

**Online machine learning voor geïndividualiseerde cadanscontrole.**

Maar om meer mensen op de elektrische fiets te krijgen, moeten deze aantrekkelijker worden. Zo zou een e-bike snel, veilig en intuïtief moeten zijn. Hier ontstaat echter wel een een probleem. Snelheid en veiligheid spreken elkaar tegen. Hoe sneller de fiets, hoe minder reactietijd de fietser heeft om een ongeval te ontwijken. **(hoe is het gebruiksgemak/betrouwbaarheid van de speed e-bike slecht?)**

# IntuEdrive

Voor het ontstaan van intuEdrive werkte Ir. Tomas al meerdere keren samen met de Universiteit KU Leuven om zijn idee stap per stap uit te werken. Na jaren van samenwerking met de Universiteit werpte dit vruchten af. Uit Jorrit zijn masterproef, in het schooljaar 2016-2017, ontstond dan het eerste prototype. Dit werkte beter dan verwacht. Vanaf dan besloten ze samen intuEdrive op te richten in December 2017.

visie mobiliteit elektrisch op 2 wielen. Missie = beter 2 wieler mobiliteit creëren door speed e-bike veiliger, makkelijker in gebruik en betrouwbaarder maken. Komen in 2020 op de markt. Cosar in productie in najaar 2019 en intuedrive zal in 2020 1000 cosars bouwen. (appendix) ir tomas was directeur. In 2010 starte hij met cosar proejct

Het is misschien al duidelijk, maar intuEdrive maakt elektrische fietsen. Het is niet zomaar een standaard e-bike. De elektrische fiets geproduceerd door intuEdrive is een speed e-bike. Dit houdt in dat deze fiets een topsnelheid kan behalen van 45km/h, tegenover 25km/h van de normale pedelec. Maar met deze verhoging in snelheid komt natuurlijk een groter risico. Dankzij de implementatie van hun e-CVT systeem, gecombineerd met een stevig frame, kleinere wielen en dikkere banden maakt CoSaR: the Convenient, Safe and Reliable speed E-bike. Naast veiligheid, is CoSaR ook intuïtief te gebruiken. De fietser kan remmen door de pedalen stil te houden in een bepaalde positie. Dit is geïnspireerd door one-pedal drive.

Het e-CVT systeem, ofwel elektrische continue variabele transmissie, heeft vele voordelen. Ten eerste verlaagt het systeem de remafstand met meer dan 60%. Hier bovenop heeft de fiets ook nog elektrische remmen en kan deze automatisch schakelen. Hierdoor kan de fietser meer aandacht besteden aan de weg dan aan randzaken. Dit resulteert in twee keer meer ruimte om een ongeval te ontwijken. De fiets heeft meer tractie in vergelijking met de traditionele elektrische fietsen. Een veilige en intuïtieve rijervaring maakt CoSaR: the Convenient, Safe and Reliable speed E-bike. Bovendien is het onderhoudsvrij en makkelijk te gebruiken. We kunnen dus zeggen dat CoSaR een goed alternatief levert voor de auto.

# Doel van de thesis

Deze thesis werkt verder op het prototype geleverd door intuEdrive, met als doel het aantrekkelijker maken van hun e-bike. Zoals reeds aangehaald blijft de cadans stabiel in het huidige systeem. Hoe hard of zacht de fietser ook trapt, de cadans zal steeds hetzelfde blijven. 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. Met deze oplossing zal de fietser meer aandacht kunnen besteden aan de weg. Waardoor gevaarlijke situaties ontweken kunnen worden.

Het voornaamste probleem hier is dat de optimale cadans van een fietser varieert van persoon tot persoon. Voor dit probleem zal een machine learning algoritme ontwikkeld worden dat de toestand die de fiets meet als input binnenkrijgt en hiermee een zo goed mogelijke voorspelling maakt. Deze voorspelling zal dan doorgegeven worden aan de fietscontroller die dan de nodige aanpassingen zal maken.

Naast het voorspellen van de cadans, moet de fietser ook kunnen aangeven dat de voorspelde cadans incorrect is. Zo zou de fietser op een knop moeten drukken om aan te geven dat het algoritme foute voorspellingen maakt. Er zullen twee knoppen beschikbaar zijn. 1 voor aan te geven dat de cadans te hoog is en 1 voor een te lage cadans. Bijvoorbeeld: de gebruiker fietst een berg op en trapt 70rpm. Dit vindt hij niet goed omdat hij liever sneller trapt, in plaats van veel kracht op de pedalen te zetten. Dus drukt hij op de knop om zijn cadans te verhogen. Achter de schermen leert het machine learning algoritme bij dat de fietser in de huidige situatie een hogere cadans wil. Even later zou het algoritme een hogere cadans moeten voorspellen. De voorspellingen in andere situaties zouden quasi hetzelfde moeten blijven.

Om deze doelstellingen te bereiken zal er eerst een simulatie opgezet worden die de fiets zo goed mogelijk benadert. De berekeningen hiervoor zijn geleverd door Jorrit en zullen gecodeerd worden in python. Vervolgens worden een aantal machine learning algoritmes getest om te zien of deze het probleem aankunnen. De algoritmes maken deel uit van 2 libraries: Keras en scikit learn.

Er zijn verschillende eisen waar dit programma aan moet voldoen. Het algoritme moet draaien op een Raspberry Pi, samen met het controleprogramma van de fiets. De voorspellingen moeten in real time op de fiets berekend worden.Hoe meer voorspellingen per seconde, hoe beter. Het doel is om aan 10Hz de cadans aan te passen. Trager dan dit zou merkbaar kunnen zijn door de gebruiker, wat kan leiden tot een onaangename rijervaring. Er moet natuurlijk ook rekening gehouden worden met de veiligheid van de fietser. Zo mogen opeenvolgende voorspellingen niet veel van elkaar verschillen en moet er een limiet staan op het maximum aantal toeren dat voorspeld kan worden.

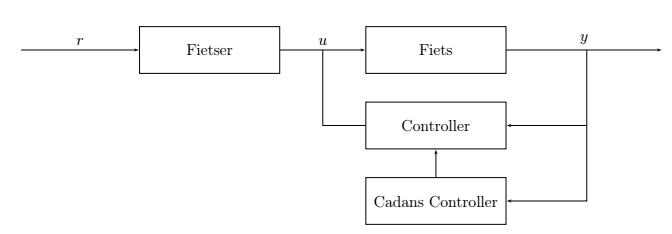
Als performantie indicator voor de verschillende algoritmes wordt er de Mean Squared Error gebruikt. Om deze indicator te gebruiken moet er ook een notie zijn van een fietsersmodel. Een soort van actor die naar gelang van de gesimuleerde toestand van de fiets gaat beslissen wat de optimale cadans is. Simpel gezegd is het fietsersmodel een functie met als input de toestand van de fiets en output een “optimale cadans”. Deze functie is speculatief en kan makkelijk vervangen worden. Ten tweede moet het algoritme kunnen bijleren met een kleine hoeveelheid data. Aanpassingen gemaakt door de fietser zijn van toepassing op de recente toestand van de fiets. Er kan dus maar een beperkte hoeveelheid aan data gebruikt worden van de laatste X aantal seconden. Ten slotte wordt er vereist dat het algoritme snel bijleert, zowel op vlak van performantie als het aantal learning cycles. Elke verandering van de gebruiker gebeurt liefst in real time. Hier zit echter wel wat speling op. En de veranderingen moeten een betekenisvolle impact hebben.

VUL HIER AAN (uittesten op echte data, op fiets zetten...)

# Huidige systeem

Het is als sinds 2011 de droom van Tomas om een perfecte E-bike te maken, wanneer Tomas nog werkte bij Toyota. Het idee hier was om een E-bike te bouwen gemengd met enkele karakteristieken van een auto. Zo gebruikt de e-bike een transmissie gelijkaardig aan die van de Toyota Prius hybrid. De implementatie gebruikt een Continue Variabele Transmissie, of kortweg CVT. Er zijn verschillende soorten CVT’s: een mechanische, een hydraulische en een elektrische. De fiets gebruikt de elektrische variant hiervan. Deze gebruikt 2 motoren elk met hun eigen functie. De ene motor stuurt de trapcadans aan. De andere motor reguleert de hoeveelheid kracht dat op de pedalen gezet wordt. Samen zorgen deze motoren ervoor dat de fiets natuurlijk aanvoelt. De motoren worden aangestuurd door de controller. Deze is geprogrammeerd in C.

Het huidige prototype, is al enkele iteraties ouder dan het prototype van Jorrit zijn masterproef, maar in essentie is het nog steeds dezelfde fiets. Figuur ## toont een blokdiagram van het huidige systeem. De fietser zoekt op elk moment een bepaalde snelheid te halen, hier aangeduid met r. Deze kan variëren naar gelang de situatie waarin de fietser zich bevindt. Bijvoorbeeld wanneer de fietser bergop rijdt kan de gewilde snelheid trager zijn dan bergaf. De fietser oefent ook acties uit op de fiets. Deze acties, hier u, zijn: de hoeveelheid kracht uitgeoefend op de pedalen, of er al dan niet aangegeven wordt dat de voorspellingen aangepast moeten worden, en tot slotte het besturen van de fiets. Met het laatste zal geen rekening gehouden worden. De fiets heeft als output verschillende metingen. De voornaamste metingen zijn: de hoek van de trapas, de gemeten kracht op de pedalen, de snelheid van de fiets en de helling waarop de fiets zich bevindt. Deze output wordt gebruikt door de controller om de fiets verder aan te sturen. Deze controller draait op een Raspberry Pi. Voorafgaand deze thesis zorgde de controller ervoor dat de cadans op een stabiel niveau blijft van 70 rpm. De controller doet dit door continu te motoren aan te spreken en indien nodig te schakelen. Als er geen versnellingen meer overblijven, wanneer de fietser bijvoorbeeld heel hard aan het trappen is, dan zal de cadans uiteindelijk boven de 70 rpm gaan. De cadans controller, wat in deze thesis uitgewerkt zal worden, moet deze instelling overpakken door de output van de fiets te analyseren.



R=[v\_ref] u=[Tcy,uc] y=[θcr,Tcy,m,vbike,α] (TODO: deftig afbeelden)

Hoofdstuk 1: Simulatie

**Waarom een simulatie  
 voor en nadelen**

Zoals reeds aangehaald wordt er een simulatie gemaakt die de toestand van de fiets zo goed mogelijk probeert te benaderen. Om de werkdruk reëel te houden is de simulatie beperkt tot hetgeen wat nodig is. Zo zal er geen rekening gehouden worden met het manoeuvreren van de fiets, wind tegen of mee. Er zal ook enkel de toestanden berekend worden die gemeten worden door de fiets. Het is immer onnodig om data te verzamelen waar de fiets geen weet van heeft.

Het voordeel van de fietssimulatie is de enorme flexibiliteit. Uren aan data kan in een moment tijd gegenereerd worden, waardoor het makkelijk is om verschillende tests uit te voeren. Er is wel een hoop gelogde data beschikbaar. Maar deze heeft veel nul waarden, aangezien er niet constant door gefietst kan worden. Hierdoor is het moeilijk om een lange reeks van opeenvolgende data punten te verkrijgen. Bovendien kan deze data een bias hebben voor een bepaalde route. Waardoor we niet genoeg data hebben in verschillende omstandigheden. Met de fietssimulatie kunnen makkelijk verschillende situaties gegenereerd worden, waardoor we over enorm gevarieerde data kunnen beschikken.   
Het voornaamste nadeel is dan toch wel de onnatuurlijke consistentie van de fietssimulatie. De geprogrammeerde actor is zeer consistent met de hoeveelheid vermogen dat hij levert. Er kan wel wat ruis toegevoegd worden, of zelfs een dominant been. Maar er is toch nog wel een verschil met een echte fietser. Zo zullen opeenvolgende cycli van de simulatie niet ver uit elkaar liggen en een trend vormen. Deze trend is bijvoorbeeld het versnellen van de cadans. Een menselijke fietser daarentegen trapt inconsistent. Er zal bijvoorbeeld wel een gelijkaardige trend ontstaan, maar opeenvolgende cycli tonen een eerder inconsistente versnelling of vertraging.

**blokdiagram van formules, cadans in functie van snelheid**

**modelleren van trapkracht = sinus (foto sinus tov echte cadans), reëel?**

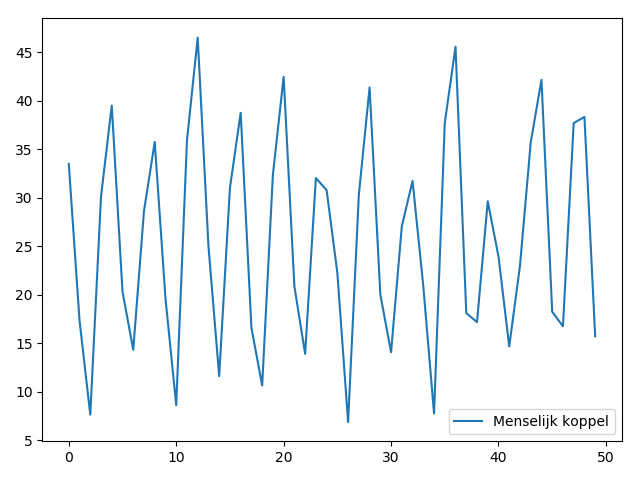
De modellering van het koppel is in functie van het gemiddelde koppel en een sinus functie. We gaan ervan uit dat 1 cyclus een volledige omwenteling is van de trapas. Het gegenereerde koppel vormt 2 pieken, 1 piek voor elk been. De functie ziet er als volgt uit:

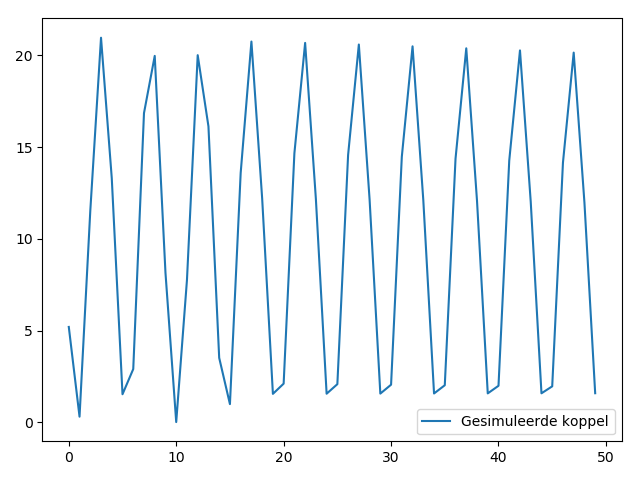
t\_cy=t\_dc\*(1+sin(2θ-pi/6))

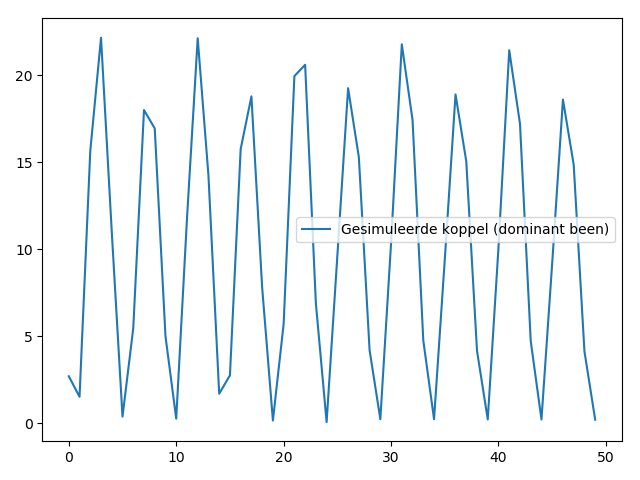
Het gemiddelde koppel geleverd door de fietser wordt gemodelleerd als een proportionele regelaar. Het doel is om een bepaalde snelheid, v\_ref, te behalen. Hoe verder de simulatie af is van deze doelstelling, hoe meer kracht er zal geleverd worden. Als deze snelheid overschreden is, dan wordt er gefreewheeled. Er wordt dus geen koppel meer geleverd. Om de kracht van de actor de limiteren, wordt er ook een maximum koppel ingesteld naar gelang de huidige cadans. De agressiviteit dat de actor doorvoert om de snelheid te halen wordt hier bepaald door K. De formules zien er als volgt uit:

t\_dc\_max = (-omega\_crank[h - 1]) / 2 + 60  
 t\_dc = min(t\_dc\_max, max(0, -K \* (v\_fiets[h - 1] - v\_fiets\_ref)))

Hoe realistisch is deze modellering juist? Figuren ## en ## tonen het gesimuleerde en menselijk koppel respectievelijk beide aan 10Hz. Zoals te zien valt is het gesimuleerde koppel enorm 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 ## toont een gesimuleerde dominant been. We stellen hier vast dat de gekozen modellering van het koppel realistisch genoeg is.





**waaro****m een goed lastmodel nodig? Geen wind tegen? Hypothese: cadans in functie van last**

**Terrein generatie**

**fietsmodel,fcc  
 verschillende mogelijkheden, plug and play**

Referenties

(<https://www.milieucentraal.nl/duurzaam-vervoer/fiets-ov-of-auto/>: consumptie/km)

(tabel van het voertuigenpark in belgië 2014-2018)

[https://statbel.fgov.be/nl/themas/mobiliteit/verkeer/voertuigenpark#figures](https://statbel.fgov.be/nl/themas/mobiliteit/verkeer/voertuigenpark" \l "figures)

(evolutie van de file zwaarte in Vlaanderen 2012-2018. (Filezwaarte = file lengte \* duur))

https://www.statistiekvlaanderen.be/filezwaarte