

Marko Jurić; Luka Vukelić 0036527393; 0036530728	SVEUČILIŠTE U ZAGREBU FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA	27.06.2024.
	Modeliranje i upravljanje dinamikom vozila	
	Upravljanje uzdužnom i bočnom dinamikom vozila	

1. Uvod

Cilj projekta je implementirati sustave estimacije i upravljanja uzdužnom i bočnom dinamikom vozila na realističnom primjeru. U prvom dijelu potrebno je estimirati vertikalnu silu na kotačima i uzdužnu brzinu vozila koristeći standardne senzore (IMU i enkodere na kotačima). Zatim treba razviti sustav za praćenje uzdužne brzine (cruise control) i sustav kontrole proklizavanja (traction control). Na kraju, potrebno je dodati sustav aktivnog skretanja (upravljanje zakretom volana) i testirati praćenje zadatog manevra. Ispitivanje algoritama provest će se koristeći MATLAB/Simulink model vozila s Vehicle Dynamics BlockSet-om, pri čemu se svim signalima pristupa u bloku Sensors, a kontrolni kod je u bloku Controller. 3D prikaz vozila moguće je uključiti putem bloka Visualization -> 3D Engine.

1.1. Parametri modela

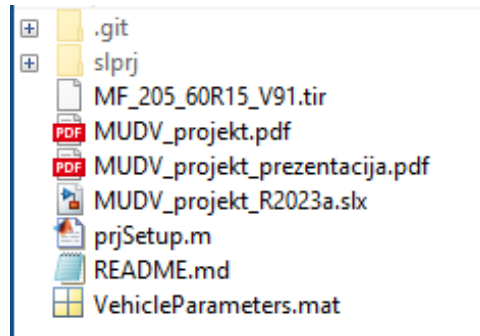
- Masa: 1341 kg
- Udaljenost prednje osovine od centra mase: 1.732 m
- Udaljenost stražnje osovine od centra mase: 1.343 m
- Vertikalna udaljenost centra mase i prednje osovine: 0.134 m
- Aerodinamička površina: 2.11 m²
- Koeficijent otpora zraka: 0.33
- Moment inercije oko vertikalne osi: 2066 kg·m²
- Prijenosni omjer zakreta volana: 18:1
- Širina traga vozila: 1.922 m
- Polumjer kotača: 0.3 m

1.2. Arhitektura sustava

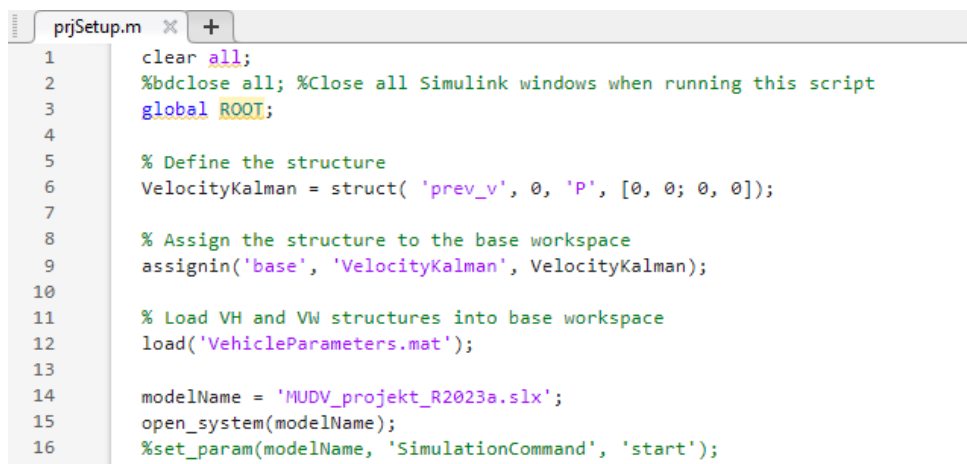
Rješenje projekta je implementirano korištenjem MATLAB-a/Simulink-a verzije 2023a uz pomoć Vehicle Dynamics Blockset-a. Arhitektura sustava sastoji se od ulazne skripte **prjSetup.m** koja učitava parametre vozila **VehicleParameters.mat** u *Base Workspace* kako bi mogli biti korišteni u Simulink modelu. Nakon toga se kreira struktura *VelocityKalman* koja će se koristiti u estimaciji longitudinalne brzine, a sadrži trenutna stanja brzine i matrice kovarijanci. Nakon toga se otvara Simulink model i, po potrebi, pokreće simulacija. Ovim jedinstvenim ulazom elegantano se rješava problem ulazne točke u rješenje problema, učitavanje globalnih parametara te definiranje početnih uvjeta pojedinih algoritama.

Model je zadan u obliku velike povratne petlje s podmodelima. Model koji je bio očekivan za našu implementaciju je *Controler*. Sastoji se od ulaza, upravljačkih algoritama, estimatora, evaluacije i izlaza svaki odvojen u svoje područje radi preglednosti. Radi lakšeg snalaženja u modelu korišteni su *Goto*

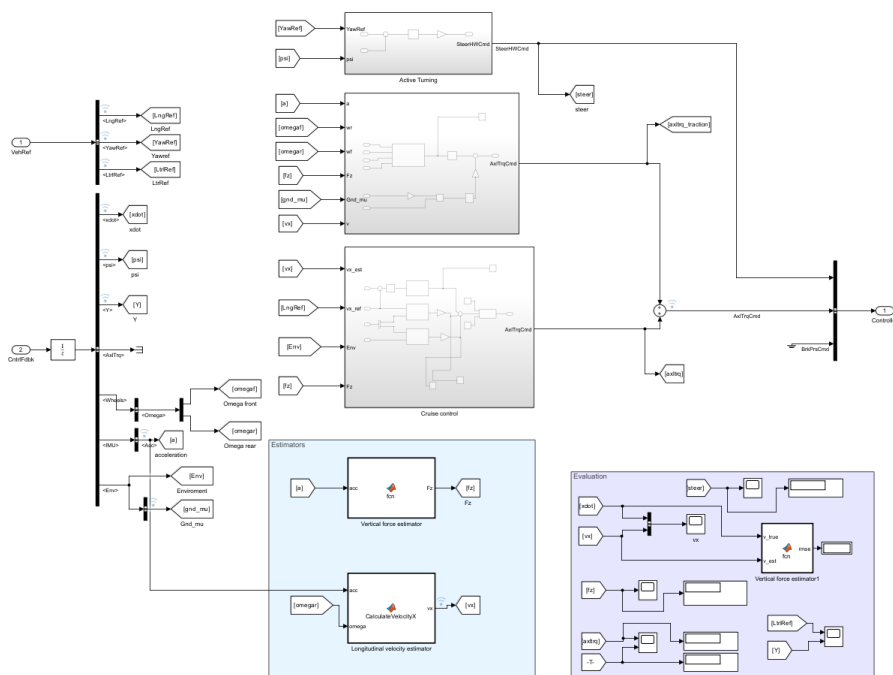
i *From* blokovi čime je izbjegnuta "šuma" veza među blokovima. Svaki pojedini zadatak je odvojen u svome podmodelu.



Slika 1: MATLAB Simulink sučelje



Slika 2: MATLAB Simulink sučelje



Slika 3: MATLAB Simulink sučelje

2. Estimacija potrebnih veličina

2.1. Vertikalne sile na kotačima

Vertikalne sile na kotačima su ključne za dinamiku vozila, jer određuju raspodjelu težine i prijanjanje kotača uz podlogu. Koristili smo poprečnu raspodjelu za izračun sila između lijeve i desne strane vozila, dok su u stacionarnom stanju normalne sile na prednjim i stražnjim kotačima izračunate prema specifičnim izrazima, što smo potom integrirali s uzdužnim silama dodajući stacionarno stanje.

$$Fzfl_{Long} = \frac{m \cdot g}{2} \cdot \left(\frac{lr}{L} \right) + \left(\frac{-m \cdot a_x \cdot h}{2 \cdot L} \right)$$

$$Fzfr_{Long} = \frac{m \cdot g}{2} \cdot \left(\frac{lr}{L} \right) + \left(\frac{-m \cdot a_x \cdot h}{2 \cdot L} \right)$$

$$Fzrl_{Long} = \frac{m \cdot g}{2} \cdot \left(\frac{lf}{L} \right) + \left(\frac{m \cdot a_x \cdot h}{2 \cdot L} \right)$$

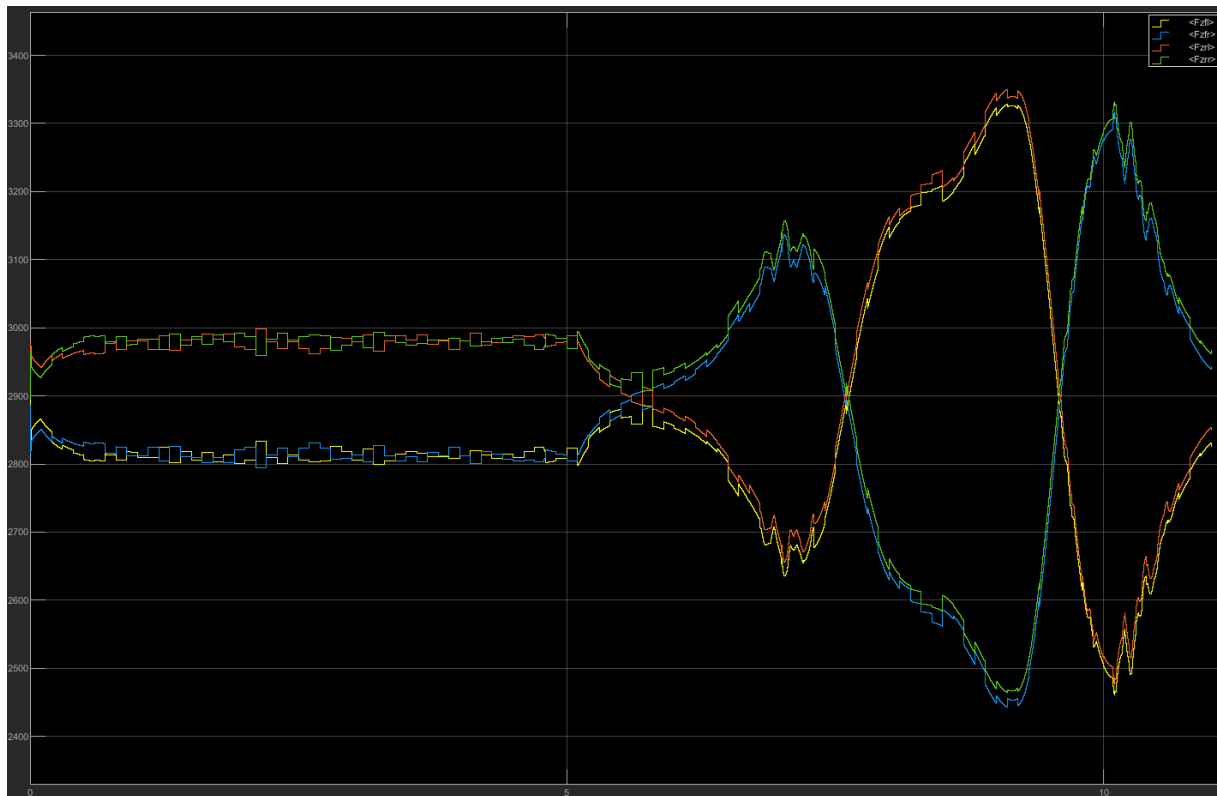
$$Fzrr_{Long} = \frac{m \cdot g}{2} \cdot \left(\frac{lf}{L} \right) + \left(\frac{m \cdot a_x \cdot h}{2 \cdot L} \right)$$

$$Fzfl_{Lat} = \frac{-lr \cdot m \cdot a_y \cdot h}{L \cdot w}$$

$$Fzfr_{Lat} = \frac{lr \cdot m \cdot a_y \cdot h}{L \cdot w}$$

$$Fzrl_{Lat} = \frac{-lr \cdot m \cdot a_y \cdot h}{L \cdot w}$$

$$Fzrr_{Lat} = \frac{lr \cdot m \cdot a_y \cdot h}{L \cdot w}$$



Slika 4: Estimacija vertikalnih sila

2.2. Uzdužna brzina

Uzdužna brzina vozila igra ključnu ulogu u njegovoj dinamici i kontroli. Estimirali smo je korištenjem podataka s enkodera na kotačima, što omogućuje izračun brzine kao umnožak rotacije kotača i radijusa kotača. Taj pristup se naziva "Jednostavnim algoritmom". Kako bismo dodatno poboljšali preciznost estimacije, razmatrali smo implementaciju Kalmanovog filtra. Međutim, zbog zahtjeva za jednostavnijim pristupom, odlučili smo se za direktno izračunavanje uz pomoć enkodera kao prihvatljivije rješenje u ovom kontekstu.

$$\omega_{\text{avg}} = \frac{\omega_{\text{rl}} + \omega_{\text{rr}}}{2},$$

$$v_x = r \cdot \omega_{\text{avg}},$$

```
% % Measurement noise covariance
% R = 0.1;
%
% % Prediction step
% predicted_vx = A * [prev_vx; ax];
% predicted_P = A * prev_P * A' + Q;
%
% % Measurement step
% measurement = r * omega_avg;
%
% % Update step
% K = predicted_P * H' / (H * predicted_P * H' + R);
% vx = predicted_vx + K * (measurement - H * predicted_vx);
% P = (eye(size(K * H)) - K * H) * predicted_P;
%
% % Update
% VelocityKalman.prev_v = vx(1);
% VelocityKalman.P = P;
% vx = vx(1);
```

Slika 5: Zakomentiran programski kod

2.3. Ideja s Kalman filtrom

Kalmanov filter funkcionira tako što kombinira informacije iz dva glavna izvora: mjerenja (kao što su podaci s dobiveni iz satelitne lokacije vozila ili s enkodera) i modela dinamike sustava (model koji opisuje kako se brzina vozila mijenja s vremenom, koristeći akcelometar). Filter iterativno ažurira svoju procjenu uzdužne brzine vozila koristeći sljedeće korake:

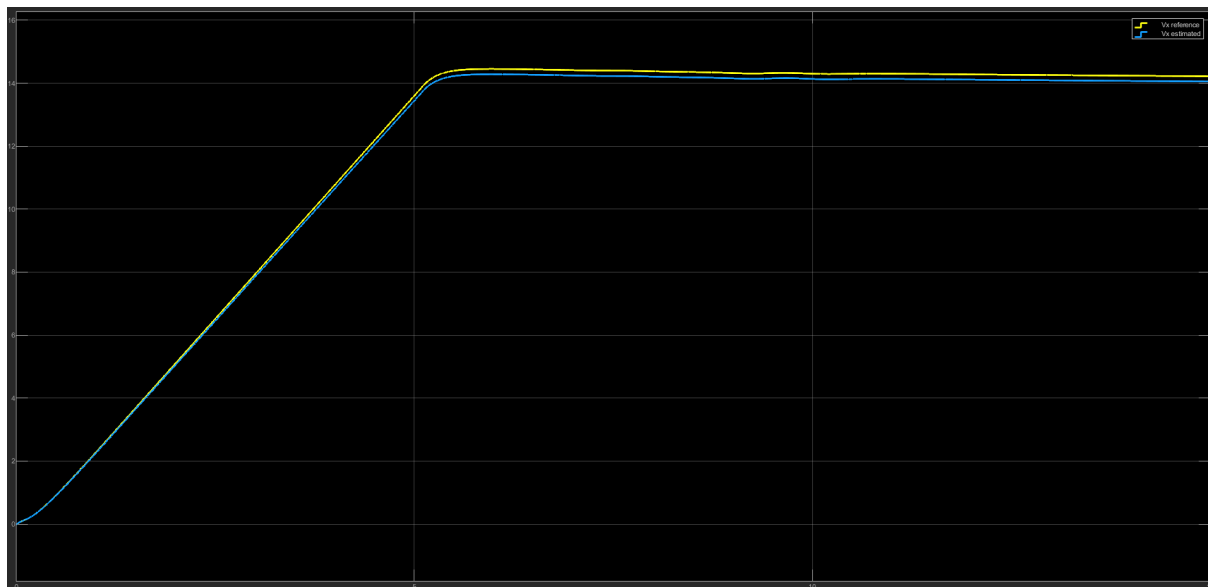
Predikcija: Na temelju prethodne procjene i poznatih dinamičkih svojstava vozila, Kalmanov filter predviđa kako će se brzina vozila mijenjati u sljedećem vremenskom koraku.

Korekcija: Kada stignu nova mjerenja (npr. nove vrijednosti enkodera kotača), filter uspoređuje predikciju sa stvarnim mjerenjem. Koristi matematičke tehnike za procjenu koliko je pouzdana svaka od tih informacija i kombinira ih kako bi stvorio najbolju moguću procjenu trenutne uzdužne brzine vozila.

Ovaj proces iterativno poboljšava procjenu brzine vozila, minimizirajući utjecaj šuma iz mjerenja i poboljšavajući stabilnost i preciznost estimacije u stvarnom vremenu.

U končanom rješenju Kalman filter nije korišten za estimaciju longitudinalne brzine jer smo zaključili da u *Double lane change* manevru i nema potrebe za mjerenjima s IMU-a jer su vrlo neprecizna na ovome konkretnome primjeru: akcelerometar ima veliki šum pri velikim i malim pozitivnim ubrzanjima. IMU bi bio koristan pri nekim drugim manevrima poput testiranja kočenja ili zaustavljanja na desnoj strani gdje se koriste kočnice.

2.4. Usporedba referentne i estimirane brzine



Slika 6: Estimacija uzdužne brzine (žuto-referenca; plavo-estimacija) [m/s]

Također je provedena procijena kvalitete estimacije RMSE (root mean square error) pokazateljem. RMSE nakon simulacije iznosi 0.3147 m/s.

3. Upravljanje uzdužnom dinamikom vozila

Upravljanje uzdužnom dinamikom vozila uključuje niz tehnika i sustava koji kontroliraju brzinu i ubrzanje vozila kako bi se osigurala sigurnost i udobnost vožnje. Jedan od ključnih sustava u ovom kontekstu je tempomat, koji omogućuje vozaču postavljanje i održavanje konstantne brzine vozila bez stalnog pritiskanja papučice gasa. Tempomat koristi senzore za nadzor brzine i sustav za upravljanje motorom kako bi održavao željenu brzinu.

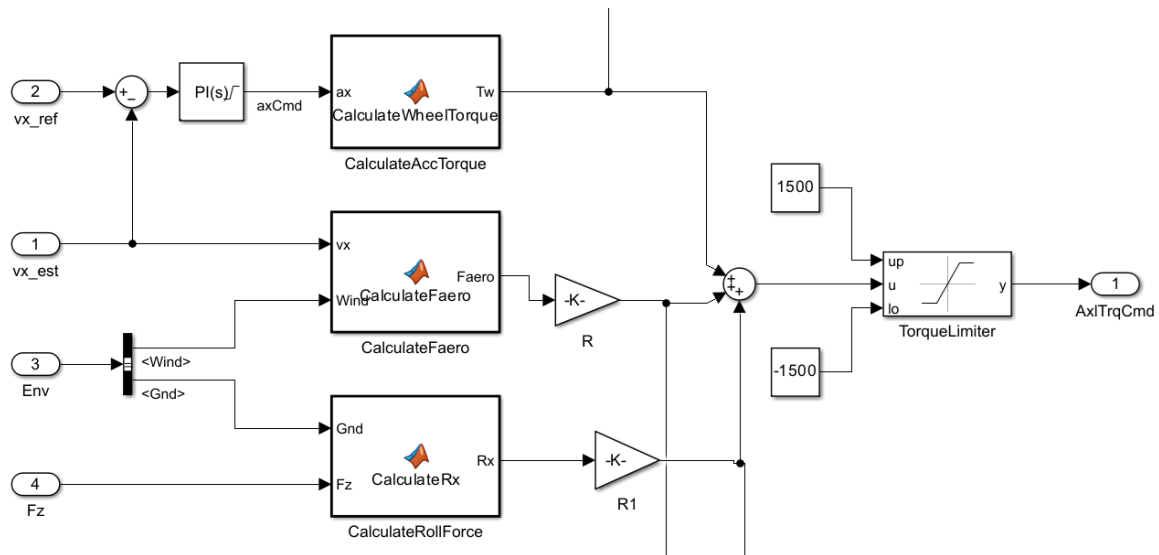
Upravljanje proklizavanjem kotača je još jedan važan aspekt dinamike vozila, posebice u uvjetima niske trakcije. Sustavi kontrole proklizavanja (Traction Control Systems, TCS) nadziru brzinu vrtnje kotača i, u slučaju detekcije proklizavanja, automatski prilagođavaju snagu motora ili primjenjuju kočenje kako bi se održala stabilnost i kontrola vozila.

Ovi sustavi zajedno doprinose poboljšanoj sigurnosti i performansama vozila, omogućujući vozaču ugodniju vožnju uz optimiziranu kontrolu brzine i trakcije, što je ključno za različite uvjete vožnje i okolnosti na cesti.

3.1. Tempomat

Koristeći estimiranu uzdužnu brzinu, implementiran je tempomat te su pokazani rezultati s realističnim profilom referentne brzine. Pogonski moment ograničen je na ± 1500 Nm.

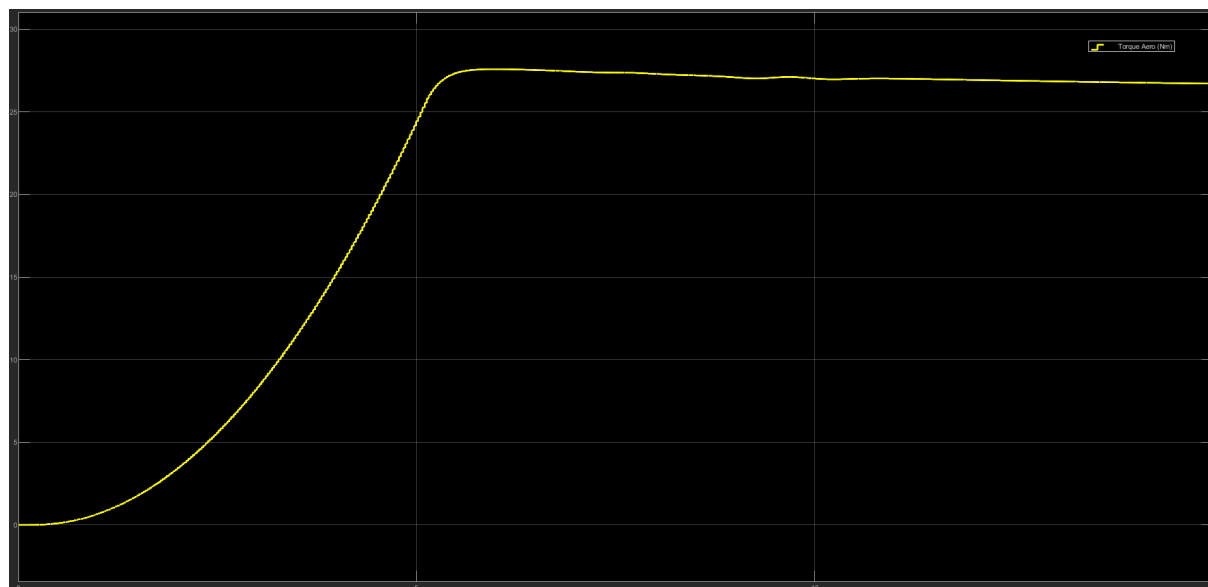
Na ulazu u tempomat prima se trenutna i referentna brzina u longitudinalnom smjeru. Razliku prima PI kontroler koji izbacuje upravljački signal kojega interpretiramo kao željenu akceleraciju u x smjeru.



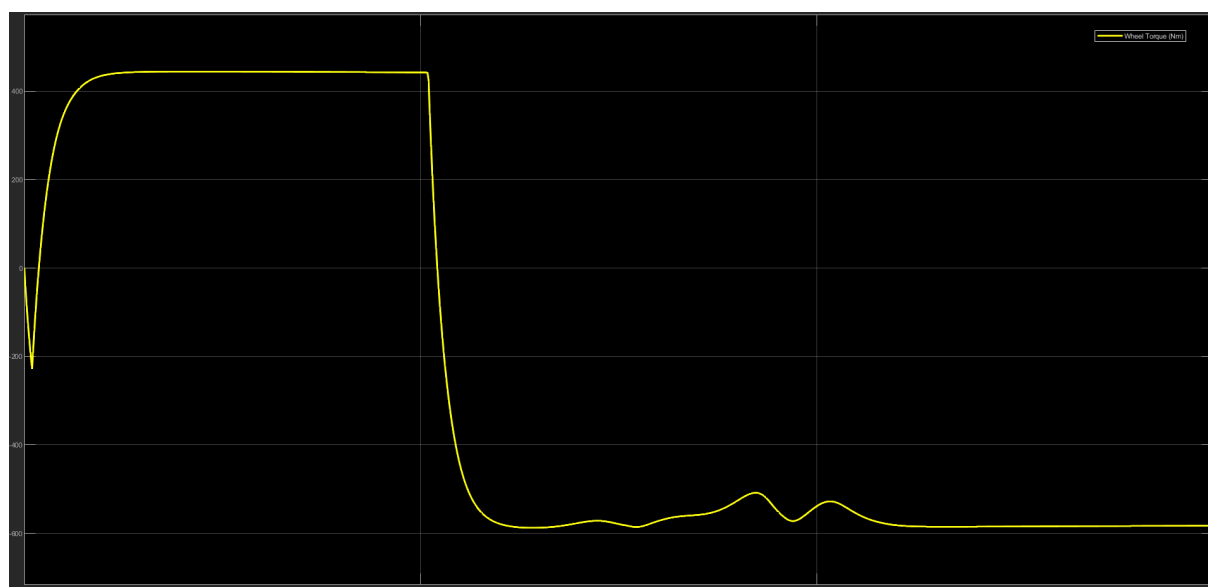
Slika 7: Prikaz tempomata u Simulinku

Akceleracija se potom koristi u izračunu željenog momenta kojeg želimo ostvariti na pogonskim kotačima.

Moment na pogonskim kotačima je ostvaren pomoću $T_w = m * a_x * r$ gdje su m masa vozila, a_x linearna akceleracija vozila u x smjeru te r polumjer kotača. Kako bi tempomat dosegno referencu potrebno je uključiti i sile otpora, a to su sila otpora zraka i otpora kotrljanja svakog kotača. Te sile su pretvorene u momente na kotačima i sumirane sa upravljačkim signalom momenta. U ukupnu sumu koja se šalje dalje u sustav, također je uključen moment koji proizlazi iz otpora zraka i momentom otpora kotrljanja. Kako bi se ograničio ovaj moment, koristi se blok *SaturationDynamics*.

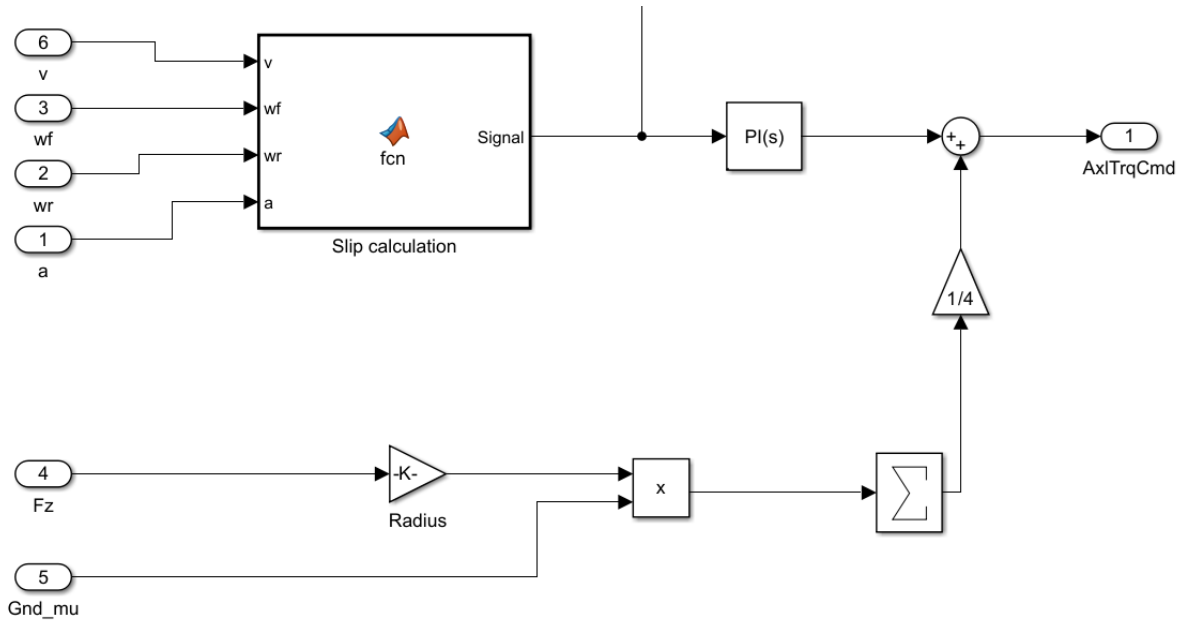


Slika 8: Moment otpora zraka [Nm]



Slika 9: Pogonski moment na kotačima [Nm]

3.2. Upravljanje proklizavanjem kotača



Slika 10: Kontrola klizanja

Prvo se računao signal razlike klizanja koji se propušta kroz PI regulator.

Algorithm 1 Izračun Slip-a, wf_ref i Signal

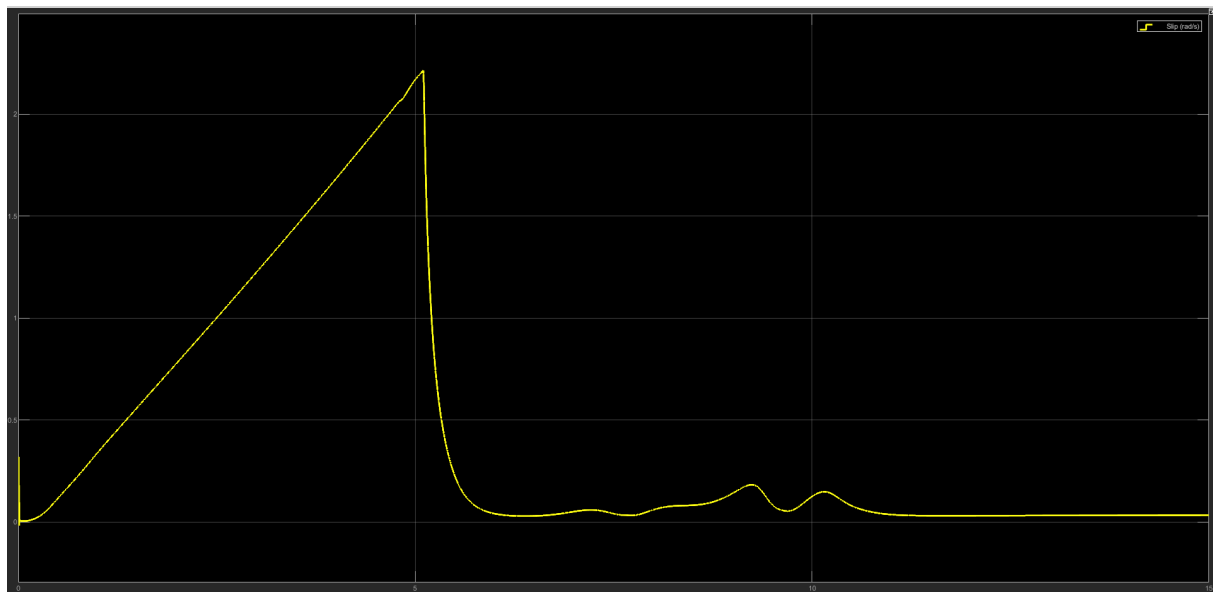
```

1: Input:  $ax, radius, wf_{-}, v, wr_{-}$ 
2:
3: Output:  $Slip, wf_{ref}, Signal$ 
4: if  $ax \leq 0$  then
5:    $Slip \leftarrow \frac{radius \cdot wf_{-} - v}{v}$ 
6: else
7:    $Slip \leftarrow \frac{radius \cdot wf_{-} - v}{radius \cdot wf_{-}}$ 
8: end if
9:  $wf_{ref} \leftarrow \frac{wr_{-}}{1 - Slip}$ 
10:  $Signal \leftarrow wf_{ref} - wf_{-}$ 

```

Nakon toga, moment koji se dobije množenjem normalne sile s radijusom kotača i faktorom trenja zbraja se s prethodnim signalom koji je prošao kroz regulator. Na taj način se formira signal za upravljanje momentom vozila.

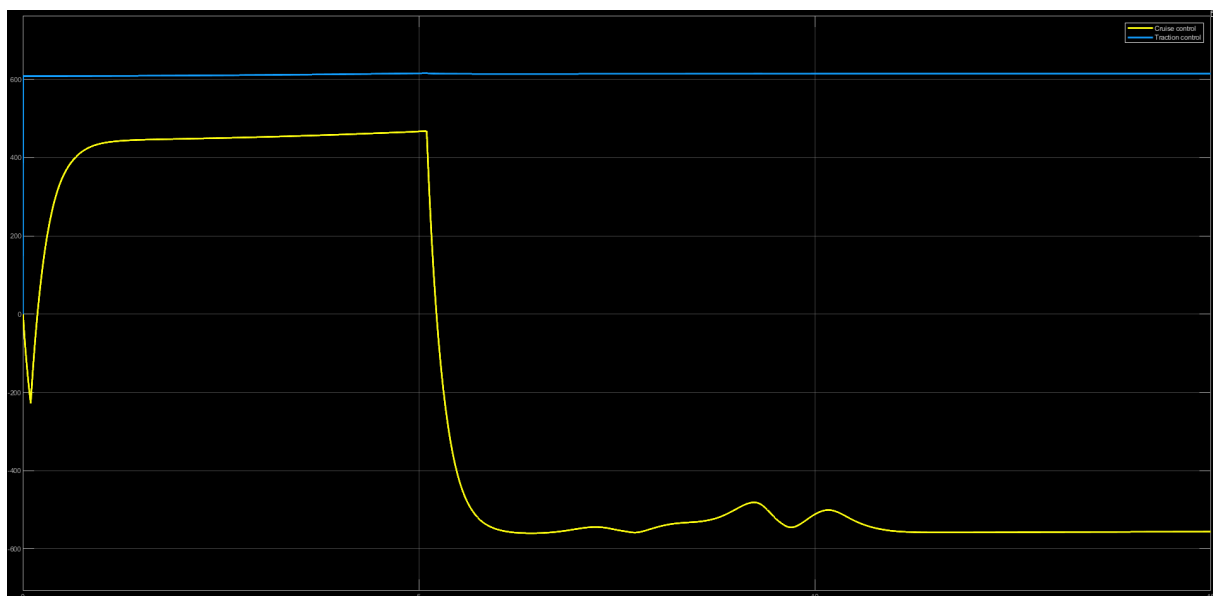
Komanda količine momenta na kotačima iz bloka tempomata i kontrole klizanja sumiraju se te se predaju bloku vozila u upravljačkoj petlji.



Slika 11: Klizanje

3.3. Usporedba konkurentnih momenata

Tempomat je sustav koji nastoji postići i održavati referentnu longitudinalnu brzinu vozila, osiguravajući konstantno kretanje bez potrebe za kontinuiranim pritiskom na papučicu gasa. S druge strane, kontrola proklizavanja (slip control) nadzire brzinu rotacije kotača vozila te sprječava prekomjerno proklizavanje koje bi moglo narušiti stabilnost ili sigurnost vozila, djelujući tako da ne dopušta vozilu da klizi više od predefinirane granice.



Slika 12: Usporedba momenata (žuto-odziv tempomata; plavo-odziv kontrole proklizavanja [Nm])

4. Upravljanje bočnom dinamikom vozila

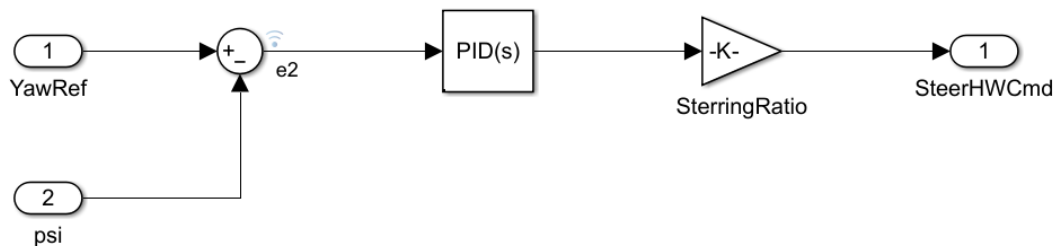
Implementiran je sustav aktivnog skretanja za praćenje željene putanje. Rezultati usporedbe s referencom na *double lane change* manevru su se pokazali uspješnim.

Signal aktivnog skretanja kreće u 9-toj sekundi te se odrazuje na odzive upravljanja. Signal je dobio iz Reference bloka te pohranjen u singalu *YawRef*. Isti signal je korišten i kao referenca zakreta vozila, računata je razlike između signala i trenutnog zakreta kotača ψ . Razlika je zatim slana PID regulatoru.

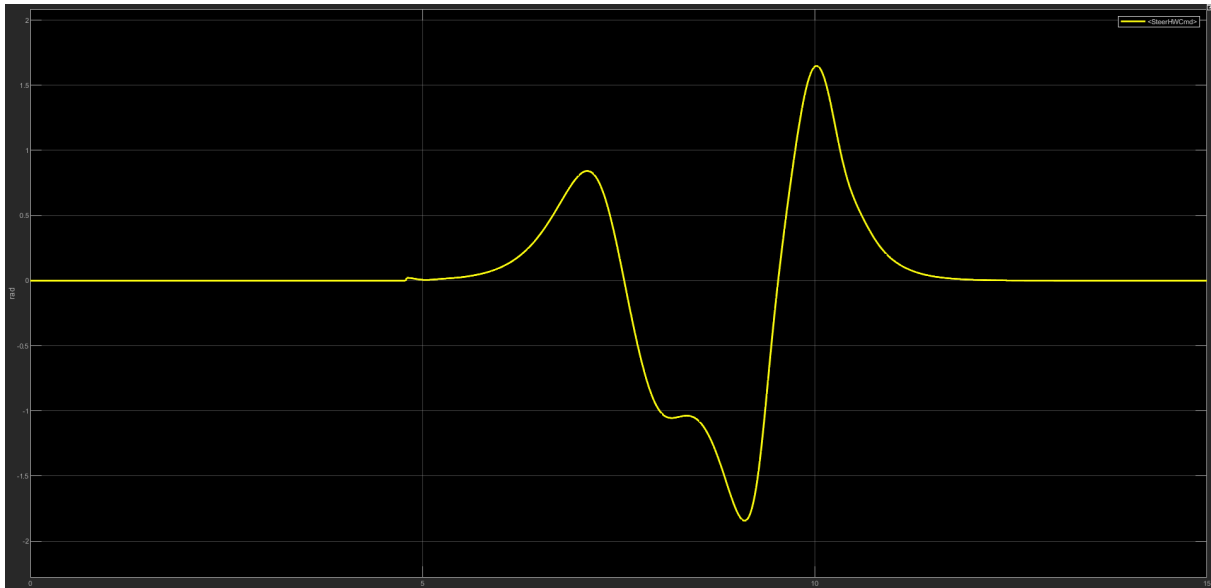
Za reguliranje signala skretanja koristi se PID regulator s povećanim proporcionalnim članom postavljenim na vrijednost 3. Povećanje proporcionalnog člana obavljeno je kako bi se brže pratila referentna vrijednost. Integralni član je dosta smanjen jer je dinamika upravljanja drukčija nego li u tempomatu gdje integralnim članom pokušavamo pokriti sve gubitke koje smo sumirali dolaskom do reference. U ovome slučaju to ne koristimo jer nemamo tih gubitaka poput otpora koje bismo sumirali, a u manevru trebamo čvrst i brz odziv koji mijenja reference u pozitivnom i u negativnom smjeru. Zbog potencijalnog *overshoota* razmatrano je i korištenje derivativne konstante, no nije bilo potrebe.

Zatim referentni zakret kotača množimo s globalnim parametrom prijenosog omjera zakreta volana te krajnju vrijednost šalje signalom *SteerHWCmd* dalje bloku vozila u upravljačkoj petlji.

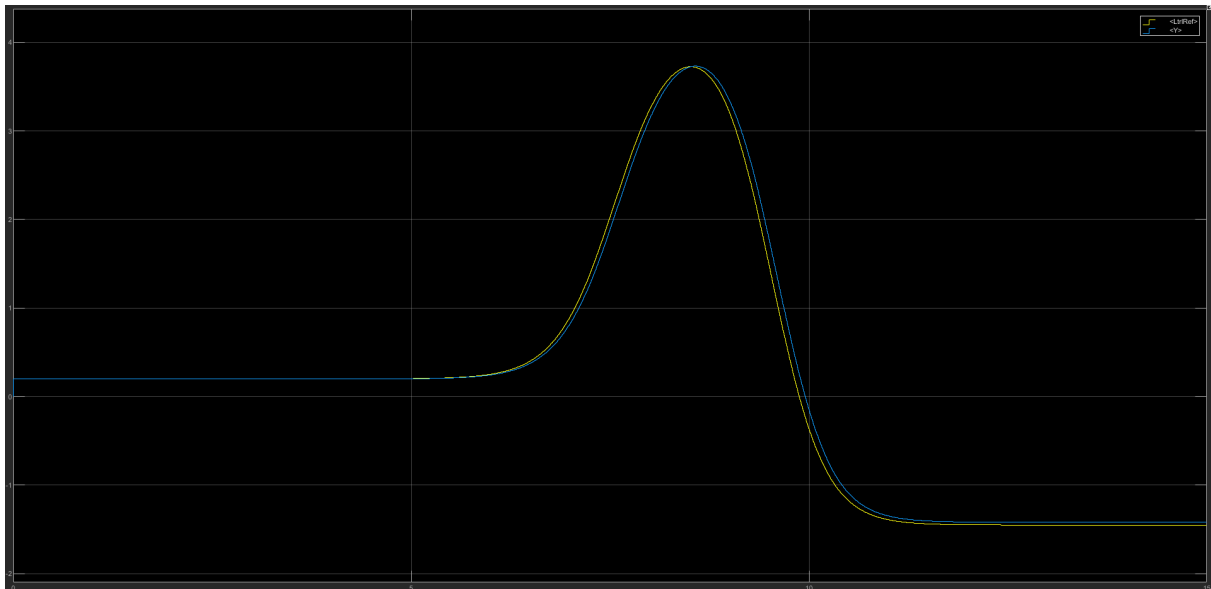
Iako je rješenje izuzetno jednostavno, odzivi su dobri i precizni. Naime, pokušano je i rješenje ne koristeći se jedino razlikom željenog kuta zakreta kotača i stvarnoga, već i udaljenosti vozila od zamišljene kružnice po kojoj bi ono trebalo skretati. Umjesto velikog modela s 4 estimirana stanja, pokušano je koristiti dva koja se odnose na već napisane pogreške. Obje pogreške bi se slale svojim PID kontrolerima koji bi slali konkurentne signale zakreta kotača. Problem je nastao pri davanju prednosti - kome dati prednost i u kojoj situaciji. *tuneiranje* dva paralelna kontrolera se učinilo previše dugotrajnim i nismo uspjeli doći do rješenja. Vjerujem da implementacijom optimizatora varijabli bi se moglo doći do optimalnog rješenja no nismo imali vremena za istraživanje.



Slika 13: Blok aktivnog skretanja



Slika 14: Signal bočnog upravljanja [rad]



Slika 15: Usporedba stvarne pozicije u y-smjeru i očekivane [m]

5. Zaključak

Ovaj projekt nam je dao mali uvid u korištenje alata i razvoj rješenja na jednostavnome problemu modeliranja i upravljanje dinamikom vozila. Bilo je korisno implementirati svoja rješenja i imati slobodu prilagoditi kompleksnost rješenja ovisno o zahtjevnosti manevra. Osim toga, projekt je zahtijevao određeno poznavanje MATLAB-a i Simulink-a pa je bilo nužno pojačati i to znanje.

Sam algoritam upravljanja i estimacije radi vrlo dobro na ovom jednostavnome manevru u simuliranome okruženju, no za zahtjevnije primjene je potrebno robusnije i temeljitije rješenje. Od korištenja svih mogućih senzora te njihove fuzije, do definicije konkurentnosti i komplementarnosti algoritama te potrebe za njihovim pravovremenim uključivanjem u petlju sustava upravljanja.

Literatura

1. Rajamani, Rajesh. *Vehicle Dynamics and Control*. Springer Science & Business Media, 2011.
2. Modeliranje i upravljanje dinamikom vozila, prezentacije, FER.