

Sistemi bazirani na znanju

Car assist

Članovi tima:

- Stefan Pejinović SV13/2021
- Veljko Ivanišević SV44/2021

Motivacija

Smatramo da je bezbednost u saobraćaju veoma bitna jer milioni ljudi svakodnevno učestvuju u njemu. Trenutak nepažnje i umor mogu rezultovati tragičnim ishodom. U cilju umanjenja takvih slučajeva sve više se radi na sistemima asistencije vozaču tokom vožnje. Ovi sistemi bazirani su na prikupljanju podataka sa određenih senzora koji se unutar aplikacije procesuiraju. Ova aplikacija ima cilj da pomogne vozaču, ali i mehaničaru kroz olakšan sistem dijagnostike problema.

Pregled problema

Sistem bi imao prisup senzorima automobila, a zadatak aplikacije bio bi da uz pomoć unapred definisanih pravila obezbedi sledeće funkcionalnosti:

- *Line assist* - na osnovu pozicije vozila u odnosu na traku, stanja žmigavaca, vidljivost trake i brzine vozila se aktivira upozorenje (LDW - *line departure warning*) ili se usmerava volan (LKA - *line keeping assist*)
- *Brake assist* - na osnovu rastojanja od vozila ispred i brzine kretanja se aktivira kočenje ili upozorenje
- *Air condition device* - na osnovu merene temperature i željene temperature se pali/gasi grejanje/hlađenje
- *ESP* - na osnovu ugla volana i ugaone brzine vozila sistem koči
- *Tire pressure* - na osnovu pritiska u svakoj gumi i minimalnih i maksimalnih pragova upozori vozača
- *Revolutions per minute - RPM* - na osnovu trenutnog broja obrtaja i maksimalnih pragova sistem izdaje obaveštenje za promenu stepena prenosa

U današnjim automobilima svaka od ovih funkcionalnosti obično predstavlja zaseban podsistem dok je kod nas sve centralizovano. Na ovaj način imamo lakšu sinhronizaciju među komponentama i nemamo dupliranog procesuiranja podataka.

Takođe, za većinu ovih funkcionalnosti je korišćen ML, dok mi smatramo da je Rule Base bolji pristup zbog 100%-ne tačnosti i determinističkih izlaza.

Metodologija rada

Ulazi u sistem vezani za pravila:

- pozicija vozila u odnosu na traku (*lineDistance*) - par brojnih vrednost razdaljine od trake (leva i desna)
- stanje žmigavca (*isLeftTurnSignal*) - levo, desno, ugašen
- vidljivost trake (*lineVisible*) - dobra, loša
- brzina vozila (*speed*) - brojna vrednost
- rastojanje od vozila ispred (*frontVehicleDistance*) - brojna vrednost
- temperatura u vozilu (*measuredTemp*) - brojna vrednost
- željena temperatura (*desiredTemp*) - brojna vrednost
- ugao volana (*steering_angle*) - brojna vrednost
- ugaona brzina vozila (*yaw_rate*) - brojna vrednost
- pritisak u svakoj gumi (*leftFrontTirePressure*) - 4 brojne vrednosti
- trenutni broj obrtaja motora (*currentRPM*) - brojna vrednost
- zaključan (*keyPressed*) - bool, da ne
- vrataZatvorena (*doorsClosed*) - bool, da ne
- rikverc (*reverseGear*) - bool, da ne

Ulazi u sistem vezani za dijagnostiku (backward chaining):

- senzor pritiska goriva - bool, radi ne radi
- regulator pritiska - bool, radi ne radi
- relej pumpe - bool, radi ne radi
- MAF-protok vazduha - bool, radi ne radi
- MAP-pritisak u Usisu - bool, radi ne radi
- EGR ventil - bool, radi ne radi
- CKP-radilica - bool, radi ne radi
- senzor detonacija - bool, radi ne radi
- Senzor pritiska freona - bool, radi ne radi
- senzor temp. spolja/unutra - bool, radi ne radi
- ventilator kondenzatora - bool, radi ne radi
- Aktuatori klapni - bool, radi ne radi
- regulator ventilatora kabine - bool, radi ne radi
- motor ventilatora kabine - bool, radi ne radi

- HVAC kontrolni modul - bool, radi ne radi
- sun-load senzor (solarni) - bool, radi ne radi

Uz ovaj set ulaza ide i infomarcija za koji su sistem vezani, npr. Napajanje gorivom, Paljenje...

Izlazi iz sistema:

- LDW - *line departure warning*
- LKA - *line keeping assist*
- Break assist - with all 4 tires
- Break warning
- Raise/Lower temperature of Air Condition
- Tire pressure warning
- ESP break - with specific tire
- Gear shift notice
- LockCar warning
- TurnLightsOn warning
- TurnCameraOn

Konfiguracija

Kako bismo ostavili prostora za dalja unapređenja i primene, određene parametre smo smestili u konfiguracioni fajl. Tu spadaju informacije o graničnoj brzini, minimalnim i maksimalnim pragovima pritiska u gumama i razlika između ugaone brzine vozila i ugla volana.

Baza znanja

Trigger predstavlja pozivanje eventa.

Pravila

```
when measuredTemp < desiredTemp  
then heatOn()
```

```
when measuredTemp > desiredTemp  
then coolOn()
```

```
when measuredTemp > desiredTemp  
then heatAndCoolOff()
```

```
when currentRPM > maxRPM:  
then shiftGearUp()
```

```
when currentRPM < minRPM:  
then shiftGearDown()
```

```
when leftFrontTirePressure < minFrontTirePressure  
then inflateTireNotice(leftFront)
```

```
when leftFrontTirePressure > maxFrontTirePressure  
then deflateTireNotice(leftFront)
```

```
when keyPressed and not doorsClosed  
then turnLightsOn()
```

```
when keyPressed and doorsClosed  
then lockCar()
```

```
when reverseGear  
then turnCameraOn()
```

```
when steering_angle != 0 and checkYawRate(steeringAngle, yaw_rate, speed) ==  
UNDERSTEER  
then breakInsideRear()
```

```
when steering_angle != 0 and checkYawRate(steeringAngle, yaw_rate, speed) ==  
OVERSTEER  
then breakOutsideRear()
```

Forward chaining

Primer 1: Line assist logic:

```
when lineDistance < minDistance and lineVisible  
then trigger(OnLeftLine)
```

```
when OnLeftLine and not isLeftTurnSignal  
then trigger(HelpLineAssist)
```

```
when HelpLineAssist and speed < minLineSpeed  
then printLineWarning()
```

```
when HelpLineAssist and speed >= minLineSpeed  
then steerWheel(side)
```

Primer 2: Break assist logic:

```
when frontVehicleDistance < minFrontVehicleDistance and speed >  
minBreakingSpeed  
then trigger(VehicleMoving)
```

```
when VehicleMoving and calculateTTC(frontVehicleDistance, speed) < dangerTTC  
then trigger(TTCDanger)
```

```
when TTCDanger over window:time(intervalTTC) >= minDangerTTC  
then trigger(HelpBreaking)
```

```
when HelpBreaking and pedalPressed  
then printBreakingWarning()
```

```
when HelpBreaking and not pedalPressed  
then break()
```

Template

Primer 1:

Pozivanje eventa za obavestjenje vozača o potrebi naduvavanja ili izduvavanja guma bi napravili kao template gde bi kroz spreadsheet definisali minimalne i maksimalne vrednosti za svaku od guma (prednja i zadnja, leva i desna).

Primer 2:

Pozivanje eventa za obaveštenje vozača o promeni stepena prenosa bi napravili kao template gde bi kroz spreadsheet definisali intervale u kojima obrtaji treba da budu za odgovarajuće brzine. U spreadsheetu bi postojale i informacije u koji stepen prenosa vozač treba da promeni. (Npr. Ukoliko su obrtaji 3000, a nalazimo se u 2. brzini, pređi u 3. s tim da postoji definisan opseg preporučenih obrtaja za svaku brzinu.).

*Ovde se može primeniti i CEP tako da se posmatra **moving window** za poslednjih 2-3 sekunde kako vozač ne bi bio obaveštavan tokom preticanja i slično.*

CEP

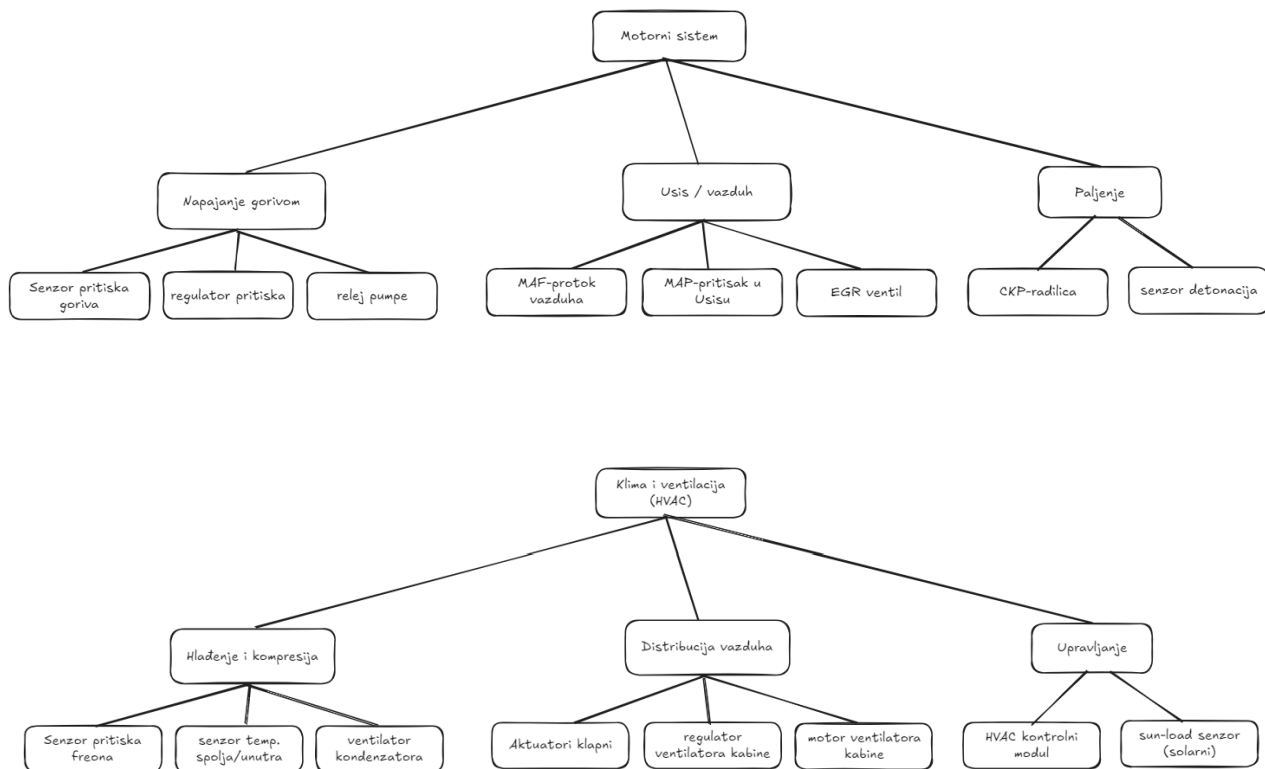
U primeru 2 za Forward chaining smo videli jedan primer implementacije CEP-a.

```
when TTCDanger over window:time(intervalTTC) >= minDangerTTC  
then trigger(HelpBreaking)
```

U primeru 2 za template smo naveli još jedno potencijalno mesto za implementaciju CEP-a.

Zaseban primer upotrebe CEP-a bismo implementirali kroz sračunavanje prosečne potrošnje goriva gde bismo agregirali potrošnju i poredili sa pređenom kilometražom. Merene potrošnje bi bile u mililitrima na kratak vremenski interval npr. sekund, nakon čega bi se merenja akumulirala i poredila sa pređenom kilometražom.

Backward chaining



Ovde smo naveli dva primera sistema koji se nalaze u automobilima. Pomoću informacija o ispravnosti određenog senzora ili komponente mi možemo doći do zaključka u kom delu je došlo do kvara.

Query

- Možemo videti sve komponente Motornog sistema ili sistema za klimu i ventilaciju.
- Možemo videti sve komponente pojedinačnog sistema npr. Napajanje gorivom.
- Možemo da vidimo sve komponente koje su u kvaru.
- Možemo da vidimo koji sve sistemi imaju kvarove u sebi.
- Možemo da vidimo da li se neki pojedinačni senzor nalazi unutar nekog određenog sistema.