**Programski flow – za referenco pisanja članka**

Tisto, kar je bilo na hitro narejeno za razumevanje. Z veseljem naredim kaj bolj berljivega, ko bo članek v pisanju ☺

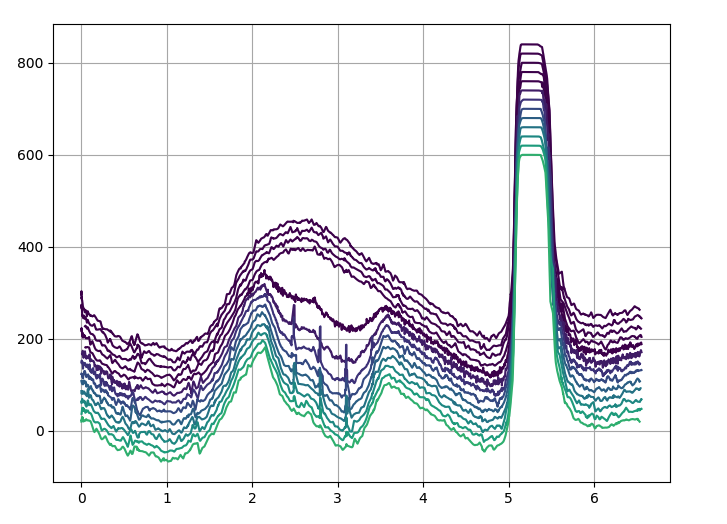


**Control loop:**



**Dodatne ideje worth noting:**

* Neaktivni senzorji – ko se joint premika stran, lahko kakšen senzor ignoriramo
* Force measurement iz tokov in referenca tudi za te
* Velikost prilagajanja odvisno od oddaljenosti objekta
* Multithreaded procesiranje
* Referenca je overfittana na 1000Hz, zato da se adaptira v tem slogu in na začetku dobro zgladi špice. Primer adaptacije na škatlo.

****

**Experiments – kot opisano v osnutku.**

**Motion reference**

Gib robota, kjer gremo v eno točko in ven iz nje po isti poti. Zato uporabimo pri izračunu točke v gibu tudi hitrost. Poleg tega se pa opiramo na to, da čas vedno teče omejen okvir naprej.

**A graph of different colored lines

Description automatically generated**A graph of a graph

Description automatically generated with medium confidence

**II.**

Sistem smo preverili tako, da smo program pustili delovati v praznem

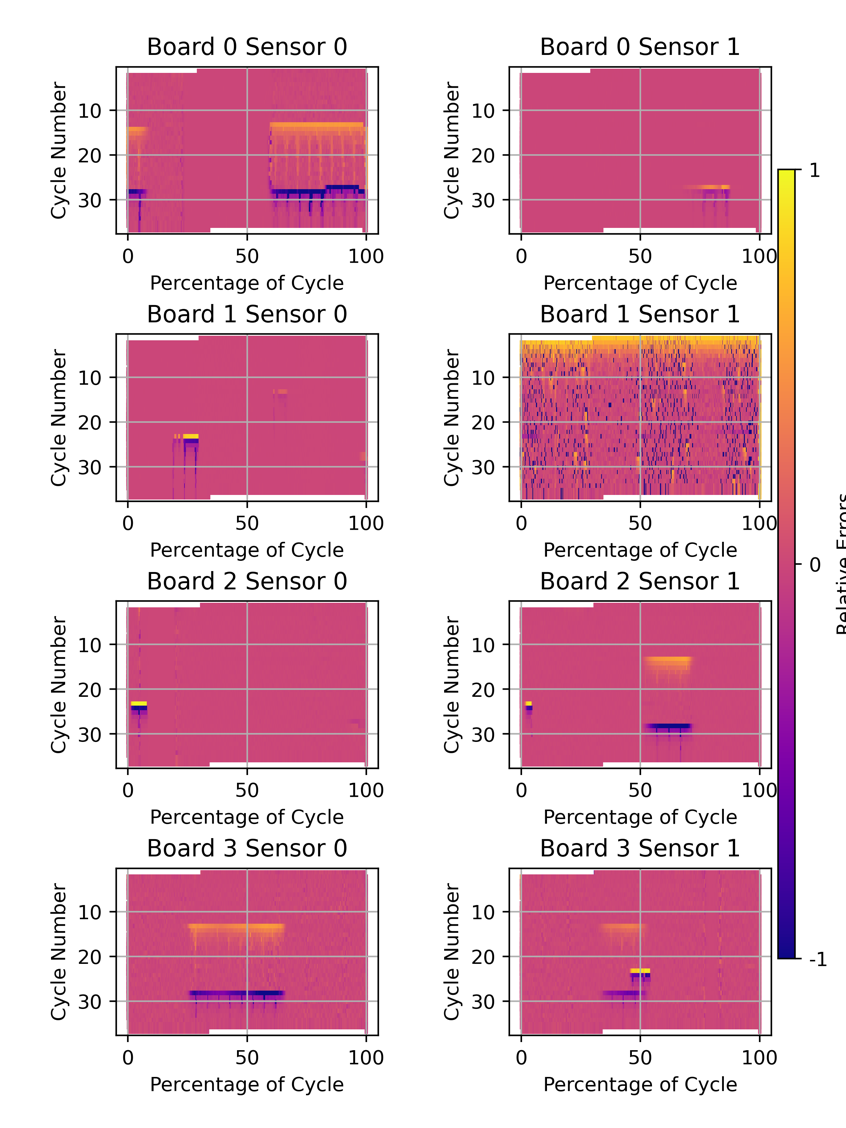
okolju, potem pa smo mu dodali oviro (škatlo) in opazovali prilagajanje.

Prikazani so grafi podatkov referenčne slike senzorjev za vsak cikel. Barva označuje relativno odstopanje od trenutne referenčne slike. Na 14. ciklu v robotovo okolje postavimo škatlo, ki pride v vidno polje senzorjev (0,0), (3,0) in (2,1) ob različnih delih cikla – opazna je rumena sled podatkov z veliko relativno napako, zaradi katere se robot upočasi. V naslednjih ciklih se referenčna slika postopoma adaptira na škatlo in rumena sled napake izginja.

Testiramo če sistem še vedno deluje, tako da delavec poseže v novo prilagojeno okolje na več mestih – vidno na senzorjih (1,0), (2,0), (2,1), (0,1) in (3,1). Robot se upočasni.

Ko škatlo umaknemo iz delovnega območja, je opazna negativna relativna napaka v meritvah senzorjev, saj so referenčne razdalje zdaj večje kot pričakovane. Robot se ne upočasni in spet postopoma prilagodi.

Komentar na senzor (1,1): zašumljena slika podatkov zaradi senzorja ki gleda v kable. Demonstriramo, da tak šum ne vpliva na sistem, saj se robot zaradi njega ne ustavlja.



**III.**

Postavili smo ga tudi v dinamično okolje, kjer je robot iz tekočega traku

pobiral zobnik in ga nasajal na ležišče in nazaj na tekoči trak.

Nič dodatnega, robot deluje normalno. Znova poskusimo z napako, na katero reagira. Komentar na negativno napako po pozitivni – prilagajanje. Bele črte vidne na istem delu cikla, kjer je robot pri miru (odlaganje in nasajanje zobnika) – tam ne more odreagirati na napako, zato čaka.

A screenshot of a computer screen

Description automatically generated

**IV.**

Preverili smo tudi sodelovanje z delavcem, kejr je zobnik v odložišče

postavljal delavec, robot pa ga je pobral in nasadil na ležišče.

Znova vidimo prilagoditev referenčne slike skozi zmanjševanje relativne napake in odmik delavca. Slika je bolj zašumljena zaradi nekonstantnih gibov človeka, s pravim thresholdom se robot vseeno prilagodi in ne ustavlja več.

A screenshot of a computer screen

Description automatically generated

**VI.**

Eksperiment iz točke IV lahko nadgradimo s tem, da človeškega operaterja

nadomestimo z drugim robotom. S tem pokažemo, da naš sistem ni

občutljiv na to, kaj posega v njegov delovni proctor (človek/stroj), in da je

zelo prialgodljiv.

Robot se prilagodi, bolje kot na človeka ker je manj zašumljeno, upočasniti je potrebno oba robota. Proti koncu še z mahanjem preverimo, če sistem še vedno zazna poseg človeka – pravi poseg zazna tudi na zašumljenem senzorju (1,1)

A screenshot of a diagram

Description automatically generated