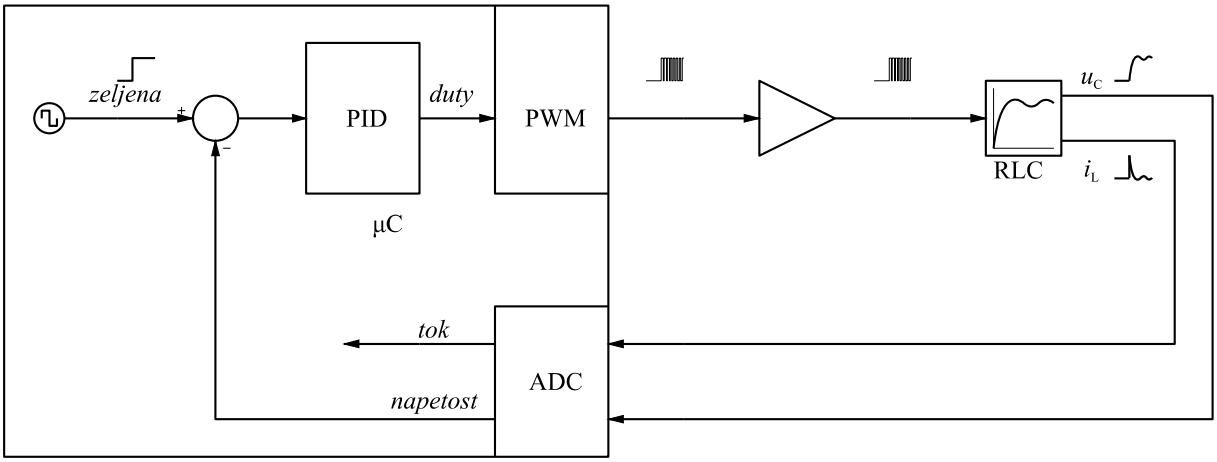
Vaja 1: Napetostni PID regulator – floating point

Cilj vaje:

Napetost na kondenzatorju RLC člena regulirajte s PID regulatorjem. Parametre PID regulatorja določite eksperimentalno, tako da bo odziv stabilen in v stacionarnem stanju ne bo preostale napake.

Opis sistema:

Sistem je sestavljen iz dveh tiskanih vezji. Na prvem se nahaja mikrokrmilnik TMS320F28069 s pripadajočim vmesnikom za povezavo z osebnim računalnikom. Na drugem pa se nahaja ojačevalnik, ki tokovno ojača PWM izhod mikrokrmilnika, RLC vezje in meritev napetosti na kondenzatorju RLC vezja ter meritev toka skozi dušilko RLC vezja. Meritev toka in napetosti je pripeljana na analogna vhoda mikrokrmilnika. Shema sistema je prikazana na sliki 1.



Slika 1: Shema sistema za regulacijo napetosti na kondenzatorju RLC vezja

Predloga programa se nahaja na Git repozitoriju na:
https://github.com/DPM2-2018-2019/DP2_reg_float.git

Predloga programa je v grobem sestavljena iz treh modulov (slika 2).

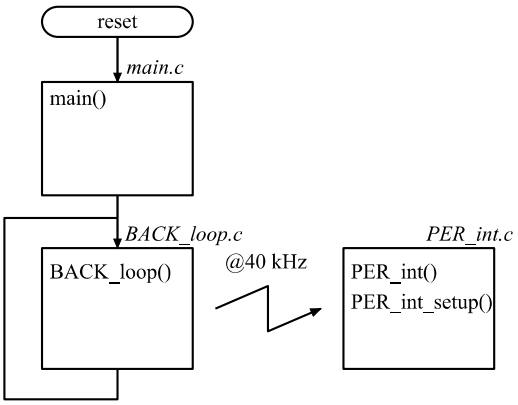
V datoteki *main.c* se nahaja funkcija *main()*, v kateri se izvede:
- inicializacija mikrokrmilnika
- inicializacija AD pretvornika
- inicializacija PWM modula
- inicializacija prekinitvene funkcije
- požene časovnik PWM modula, ki tudi proži prekinitev in AD pretvorbo

V datoteki *BACK_loop.c* se nahaja funkcija *BACK_loop()* v kateri je neskončna zanka, ki se izvaja potem ko se zaključi inicializacija.

Vaja 1: Napetostni PID regulator – floating point

V datoteki *PER_int.c* pa se nahajata dve funkciji. Funkcija *PER_int_setup()* izvede:
- inicializacijo registrov potrebnih za izvajanje prekinitve
- inicializacijo podatkovnih strukture, ki so v uporabi v prekinitvi

Prekinitvena funkcija *PER_int()* pa se izvaja periodično s taktom 40 kHz. V njej:
- se prebereta rezultata AD pretvorbe
- prvo sekundo (40000 vzorcev) se izvede kalibracija preostale napetosti za rezultat AD pretvorbe iz tokovne sonde
- se preračuna rezultat AD pretvorbe
- generira želena vrednost. Želena vrednost ima obliko stopničastega signala s periodo 1 sekunde (40000 vzorcev), ki spreminja vrednost iz 0,25 V na 2,5 V. Na nivoju 2,5 V ostane 12,5 ms (500 vzorcev).



Slika 2: Shema izvajanja programa

Predlagan potek:

1. Iz GitHub-a sklonirajte repozitorij v izbrano mapo. Ustvarite in preklopite na novo vejo (Branch).
2. Razvojno okolje poženite v izbrani mapi.
3. Uvozite projekt.
4. Na podlagi električne sheme določite faktorja za preračun vrednosti AD pretvorbe za napetost in tok (*napetost_gain*, *tok_gain*).
5. Preverite osnovno delovanje sistema tako da ročno spreminjate vklopno razmerje (*duty*). Napetost na kondenzatorju (*napetost*) bi morala ustrezno slediti vklopnemu razmerju.
6. Najprej realizirajte P regulator in ga ustrezno nastavite. Pri nastavljanju si pomagajte z grafičnim prikazom odziva na spremembo želene vrednosti (uporaba modula *DLOG_gen*).
7. Nato dodajte integralni (I) člen ter ga ustrezno nastavite. Po potrebi tudi ponastavite P člen.
8. Na koncu dodajte še diferencialni člen (D). Končen PID regulator ustrezno nastavite.
9. Regulator parametrirajte za optimale odziv na spremembo želene vrednosti in na odziv na motnjo (vkop bremena)
10. Končno verzijo programa pošljite na GitHub.

Vaja 1: Napetostni PID regulator – floating point

Zahteve:

Vse spremenljivke, ki jih potrebujete za realizacijo PID regulatorja naj bodo tipa *float*.
Regulator naj bo napisan tako, da v primeru spremembe vzorčne frekvence (*SAMPLE_FREQ*) ni potrebe po spreminjanju parametrov PID regulatorja.

Rezultati:

Kakšna je obremenjenost mikrokrmilnika (*cpu_load*). Katera je najvišja vzorčna frekvenca pri kateri bi sistem še lahko deloval?

Katera je najnižja vzorčna frekvenca pri kateri je sistem še stabilen?

Kakšni so parametri Kp, Ti, Td pri vzorčni frekvenci 40 kHz za optimalni odzviv na spremembo želene vrednosti?

Kakšni so parametri Kp, Ti, Td pri vzorčni frekvenci 40 kHz za optimalni odzviv na motnjo?

Prikažite odziv na spremembo želene vrednosti in odziv na motnjo.