

Мехатронички Системи: Управување со Повратна Врска

Галевски Марко 1172, Никола Мучев 1017, Елена Наумовска 1019

16/12/17

1 Вовед

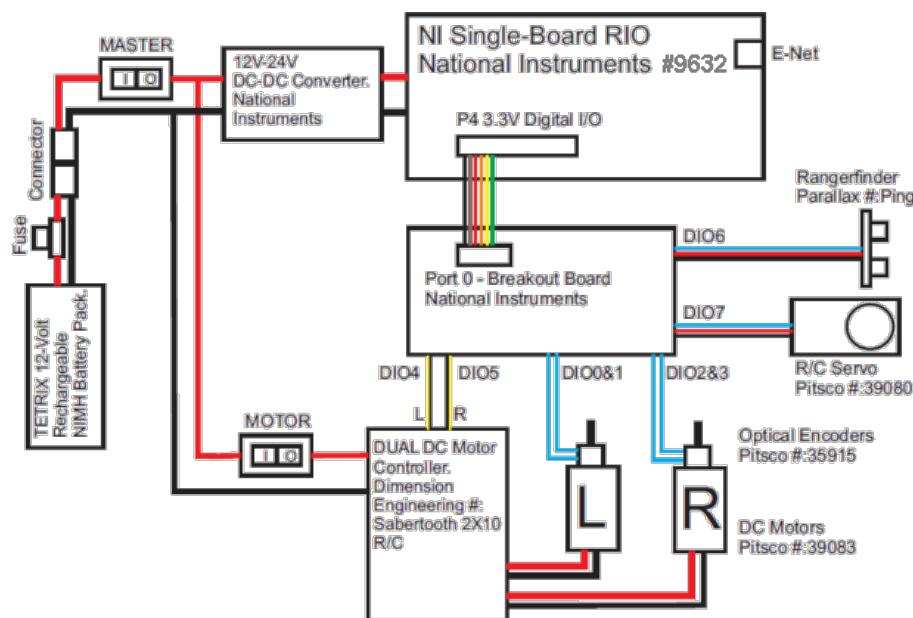
Роботика и автоматика стануваат потребни и основни делови од инженерството и следствено се многу важни теми за проучување од страна на студенти на инженерство и наука. Понатаму, роботика е изградена врз темели како карактеризација на трансдусери, контрола на мотори, собирање на податоци, механика на движење, мрежна комуникација, компјутерска перцепција, препознавање на шеми, кинематика, планирање на траекторија, и други кои исто така се темели за други полинија како на пример производство. National Instruments (NI) LabVIEW комплетот за роботика заедно со LabVIEW програмскиот пакет нудат додаток на традиционалните учебници за роботика, со можност за активно/практично учење во компактен и проширен комплет.

Проектот беше изработен со цел да го прикаже концептот и принципот позади управувањето на некој систем со повратна врска. За остварувањето на задачата употребивме еден почетен кит за роботика од National Instruments, роботот DaNI 2.0. Овој роботски кит ги содржи сите компоненти потребни за успешна реализација на задачи и проекти поврзани со управувањето и задвижувањето на еден општ робот во форма на „Rover“.

Во текот на овој документ ќе се разговара за структурата на роботот - односно неговите сензори и актуатори, како и за управувачката единица sbRIO, исто од NI. Потоа, ќе се претставаат и образложаат 3 алгоритми во LabVIEW. Првите два ќе бидат базирани на ПИД управување со повратна врска, додека третиот алгоритам ќе биде едно детално објаснување на почетната програмата дадена од NI како пример за можните способности на DaNI. За крај, ќе се продискутира за некои предизвици со кои се соочивме во текот на изработката на овој проект, и ќе се разговара за понатамошна работа со DaNI за постигнување на некоја корисна функција.

2 DaNI

National Instruments (NI) LabVIEW комплетот за роботика го содржи DaNI: скlopен робот со рамка, тркала, актуатори, сензори, sbRIO единица и кабли за поврзување. Хардверот може да биде проучуван, обратно инженириран, и модифициран од студенти. Меѓутоа, главната цел е роботска перцепција и контрола кои се имплементирани во LabVIEW софтвер развиен на одделен сервер (компјутер) и спуштен на роботскиот компјутер.



2.1 Актуатори

2.1.1 DC Мотори

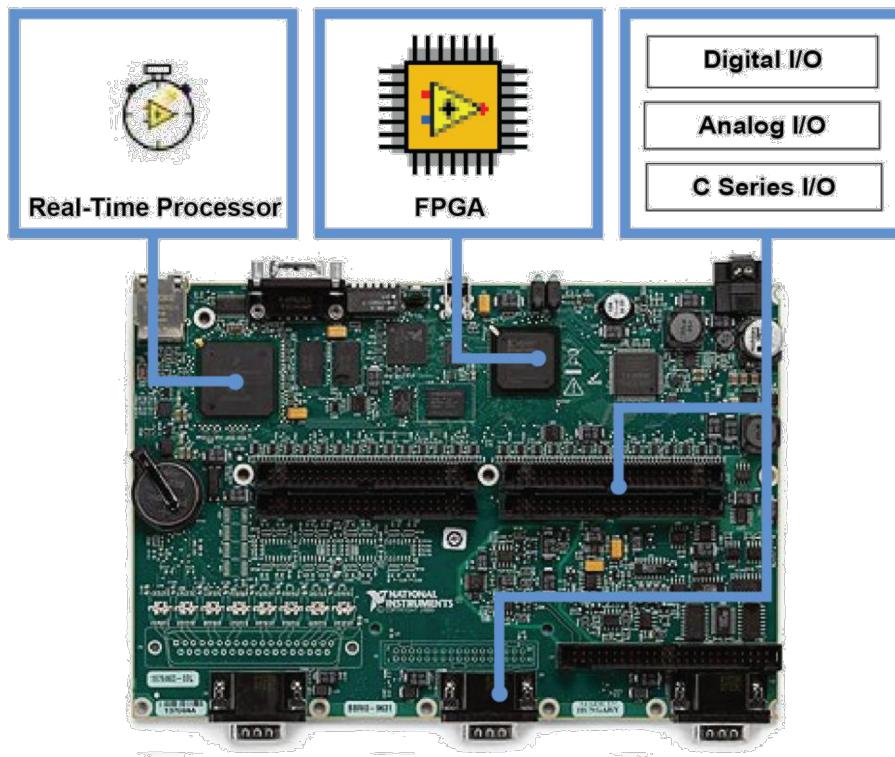
2.1.2 Серво Мотор

2.2 Сензори

2.2.1 Енкодер

2.2.2 Ултразвучен Сензор

3 NI sbRIO 9632



sbRIO (single-board Reconfigurable Input/Output) управувачките единици од National Instruments претставуваат компјутер монитран на една плочка (single board) наменета за случаи каде што е потребно решение со управување во реално време (Real Time). Оперативен систем кој работи во реално време (RTOS: Real Time Operating System) е изработен со особената цел да извршува функции кои барат големо ниво на временска прецизност и висок степен на надежност. За некој систем да се смета за RTOS, тој мора да има познато максимално време на извршување за секоја од неговите клучни операции. Системите кои можат сигурно да обезбедат максимален временски одсив се вели дека работат во потполно реално време, додека системите кои можат само понекогаш да обезбедат максимален временски одсив се вели дека работат во делумно реално време.

Потребната брзина се постигнува со користењето на FPGA чип како процесорска единица (дали имаше и обичен процесор покрај него?)

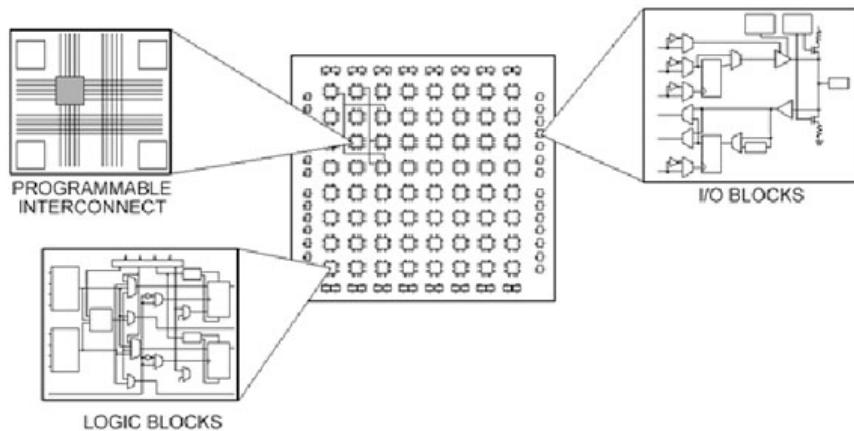
sbRIO поседува 110 дигитални влезови/излези, 100 од кои се оспособени за PWM (Pulse Width Modulation), а 10 од кои се наменети за ниско-фреквентна намена. sbRIO поседува и 32 аналогни влезови, и 8 аналогни излези (ПРОВЕРИ ГИ БРЗИНТЕ И ПРЕЦИЗНОСТИТЕ)

sbRIO содржи и 3 сериски „С“ портови за поврзувањето на надворешни модули од NI за проширување на можностите на sbRIO да опфатат и актуација и аквизиција на податоци.

3.1 PWM

3.2 FPGA

FPGA, или „Field Programmable Gate Array“, претставува репрограмибилно интегрално коло кој поседува голем број на програмабилни логични порти. Кај обичните микроконтролери, логиката за управување се пишува и компајлира од некои програмски јазик како C, BASIC или некој графички јазик како G (LabVIEW). Во текот на овој процес се сублимираат во процесорски наредби како ADD и MOV. Низата на можните наредби е единствена за секој микропроцесор. Кај FPGA колата, програмата не се сублимира на наредби, туку самата внатрешна архитектура на колото се подредува со електромагнетни полинња. Како резултат, се добива конфигурација на логични порти која ќе ја извршува задачата описана во првобитната програма. Бидејќи FPGA нудат хардверско решение, нивното работење е често многу пати побрзо од работењето на обичните микроконтролери, но се подраги.



FPGA колото што се наоѓа во sbRIO-то е моделот Xilinx Spartan 6, кој поседува 6 милиони репрограмибилни логични порти.

3.3 Споредба со Ардуино Уно

Ардуино претставува често применето open-source решение за управувачки задачи и во индустрија и на аматерскиот терен. Ардуино плочи често содржат микроконтролер, аналогни-дигитални и дигитални-аналогни претворувачи, и осцилаторски кристал. Додека sbRIO плочите се програмираат во LabVIEW, Ардуино плочите се програмираат со сопствениот „Ардуино“ јазик, која е проширена и поедноставена варијанта на C++ јазикот. Ардуина се популарни помеѓу хобисти, студенти, и за прототипување. Најпопуларната варијанта на Ардуино плочите е Ардуино Уно, која е споредена со sbRIO-то подолу:

	Arduino Uno	NI sbRIO 9632
Напојување	7 - 12 V DC	19 - 30 V DC
Дигитален I/O	14 влезови/излези	110 влезови/излези
Брзина	16MHz	400MHz
Аналогни влезови	6 влезови	32 single-ended влезови
Аналогни излези	6 дигитални пинови способни за PWM	4 16-bit излези, 100 PWM пинови
Меморија	2kB RAM, 32kB неволатилна	128MB RAM, 256MB неволатилна
Цена	€15	€1560

4 LabVIEW

LabVIEW (Laboratory Virtual Instrument Engineering Workbench) е софтверски пакет наменет за програмирањето на виртуелни уреди (инструменти) за мониторинг и управување на физички уреди. Инструментите што се програмираат во LabVIEW можат да се компајлираат и да се издават

како комплетно независни програми кои можат да се монтираат како главен управувач на мехатроничките уреди и машини. LabVIEW инструментите се програмираат користејќи го нивниот сопствен графички програмски јазик, „G“. Да се спомени дека јазикот G и G-Code немат никаква поврзаност меѓу себе. Основниот пакет на LabVIEW поседува многу од стандардните можности што се очекуваат од било кој програмерски јазик, како што се логички/булови операции, математички операции, и пристап до алатки за визуелизација на податоци. LabVIEW може да се прошири (и во некои случаи мора да се прошири) со додатни пакети. На пример, овие пакети можат да содржат готови под-инструменти (sub-VIs) за обработка на сигнали (Signal Processing), или да овозможуваат соработка помеѓу LabVIEW и некои други технологии (FPGA module). Од овие пакети ние ги употребуваме Real Time, FPGA, и Robotics пакетите. Првите два пакети ни го оспособуваат LabVIEW да програмира FPGA чипови и да управува во реално време, додека Robotics пакетот содржи во себе голем број на готови инструменти за обична и инверзна кинематика, отчитување од сензорите, и практиче наредби на моторите. Robotics пакетот исто поседува едно подмножество на инструменти наменети само за DaNI 2.0, и со тие го управуваме DaNI роботот.

4.1 Robotics Модул: Starter Kit 2.0

Во Robotics модулот на LabVIEW, од интерес ни се готовите VIs за Starter Kit 2.0, кои содржат во нив потпрограми за иницијализација и деиницијализација на роботот, отчитување од сензорите, и задавање на брзина на ротација на моторите. Тука ќе се набројат и објаснат овие блокови.



Блок за иницијализација:

Првиот блок е за иницијализацијата на роботот. Како влез се дава IP адресата на DaNI роботот, и според зададената адреса започнува комуникацијата со DaNI. При иницијализација на DaNI се создава „објект“ кој го претставува роботот. Програмерскиот термин „објект“ дефинира конгломерат на податоци и функции кој го опишува некој апстрактен предмет. Овие предмети се често неопипливи, но во случаи како нашиот, овој „објект“ е еден кибернетски претставнички меѓуслој кој ни овозможува комуникација со физичкиот систем во прашање. Како излез на оваа функција го добиваме објектот на DaNI, што ни претставува предуслов за употребата на било која од другите функции. Error-in и error-out портите на VI-то се за заштита при некоја грешка во системот. Ако грешката е 1 (има грешка), целата функцијата на иницијализација се заменува со куса врска, и нема излезен објект.

5 Управување со повратна врска

За ЕЛЕНА: Гледај во кодот да ги видиш примерите за како се пишат равенки.

Пример со наброени равенки:

$$a = b^2 + \sqrt{\frac{7 \cdot x}{\ddot{x} + 3 \cdot \dot{x} + \int_a^b x dt}} \quad (1)$$

Пример со нова линија:

$$a = b^2 + \sqrt{\frac{7 \cdot x}{\ddot{x} + 3 \cdot \dot{x} + \int_a^b x dt}}$$

Пример со left alignment:

$$a = b^2 + \sqrt{\frac{7 \cdot x}{\ddot{x} + 3 \cdot \dot{x} + \int_a^b x dt}}$$

Исто, равенки можат inline да се пишат вака $\int_{birth}^{death} struggle dt = life$.
Good Luck, have fun.

6 Изработени Алгоритми

Во текот на нашата работа со DaNI роботот, развишме 2 алгоритми за неговото управување. Првиот алгоритам ја имаше задачата да го држи роботот на фиксно растојание од пречката (што се детектираше со ултразвучниот сензор), додека вториот алгоритам беше изработен со цел да се изедначат брзините на двете тркала, бидејќи не се совпаѓае. Двата алгоритми се базираат на управување со повратна врска, употребувајќи ПД контролери. Третиот алгоритам е примерниот алгоритам за избегавање на пречки даден од NI за демонстрација.

6.1 Оддржување на фиксно растојание со ПД контролер

6.2 Компензација на брзинска грешка со ПД контролер

6.3 Избегавање пречки со хистограм на векторски-полиња

7 Предизвици

8 Заклучок

9 Понатамошна работа