

**Robot Półpająk**

**bo ma 4 nogi**

LP1 C1, EADI-3

Bełch Jakub

Rzeszów, XI 2016

1. Plan do zrobienia

* robot na 4 nogach, sterowanie przez arduino lub (jka nie da rady to PC albo raspbery), sterowanie rowniez reczne z pilota (BT HC-05) (albo z kompa przez BT) - chyba ze ogarneli cos oprocz LUA na ESP8266
* noga o 3 stopniach swobody xyz, notacja DH, kinematyka odwrotna i prosta
* 4 nogi - robot chodzacy na 4 nogach (sterowanie poprzez zadanie XY robota)
* noga,serwo, robot: klasa w C++(Visual Micro)
* sterowanie za pomoca sieci petriego (to juz mam w C++11, i xamarin) tylko trza przerobic na arduino IDE/atmegaC (VisualMicro) albo musze ogarnac include klasy vectro z STL, albo przesylanie tablicy 2D do funkcji
* zeby chodzil prosto - zaprogramowane/obliczone wczesniej sekwencje posrednie
* rejestrowal swoje polozenie XY (+Z ale trza by cos do pomiary wysokosci, albo na podstawie przechylow ale to chyba szkoda czasu),

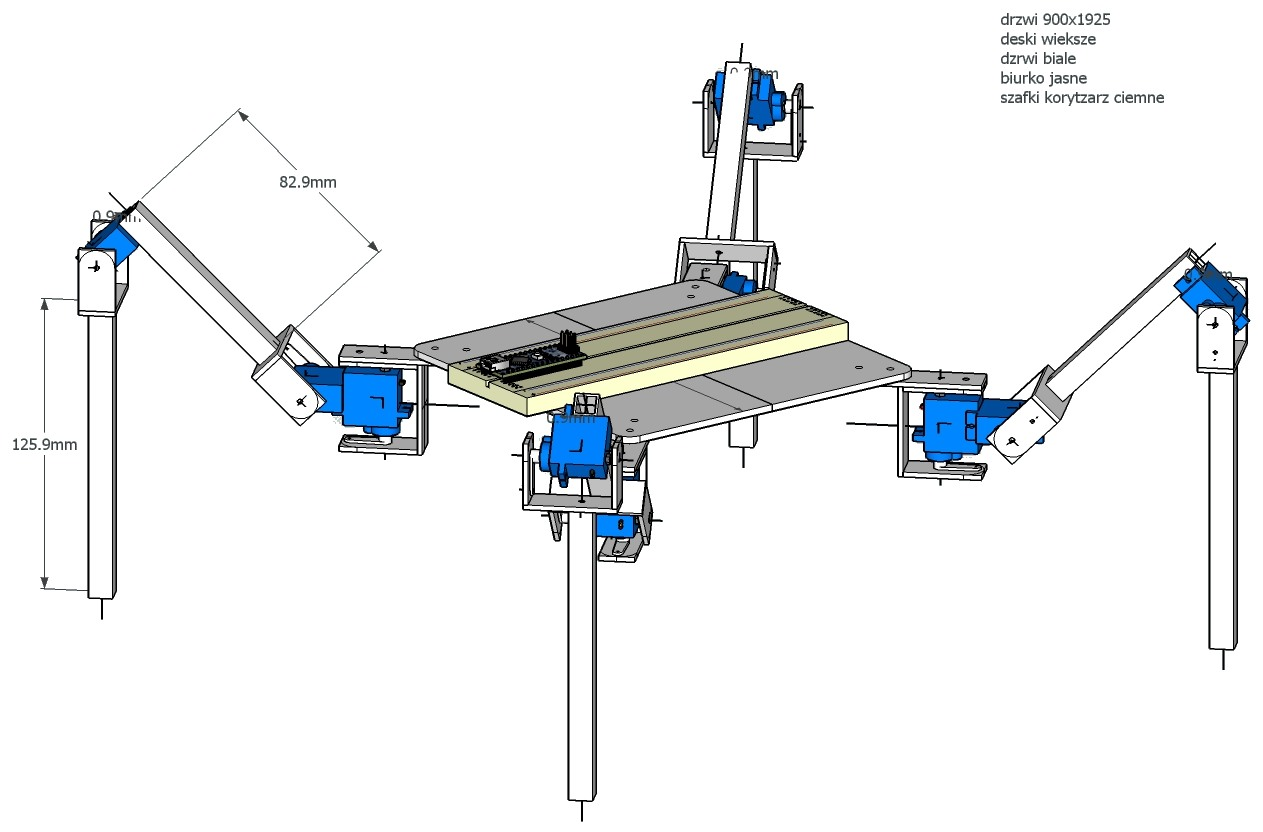
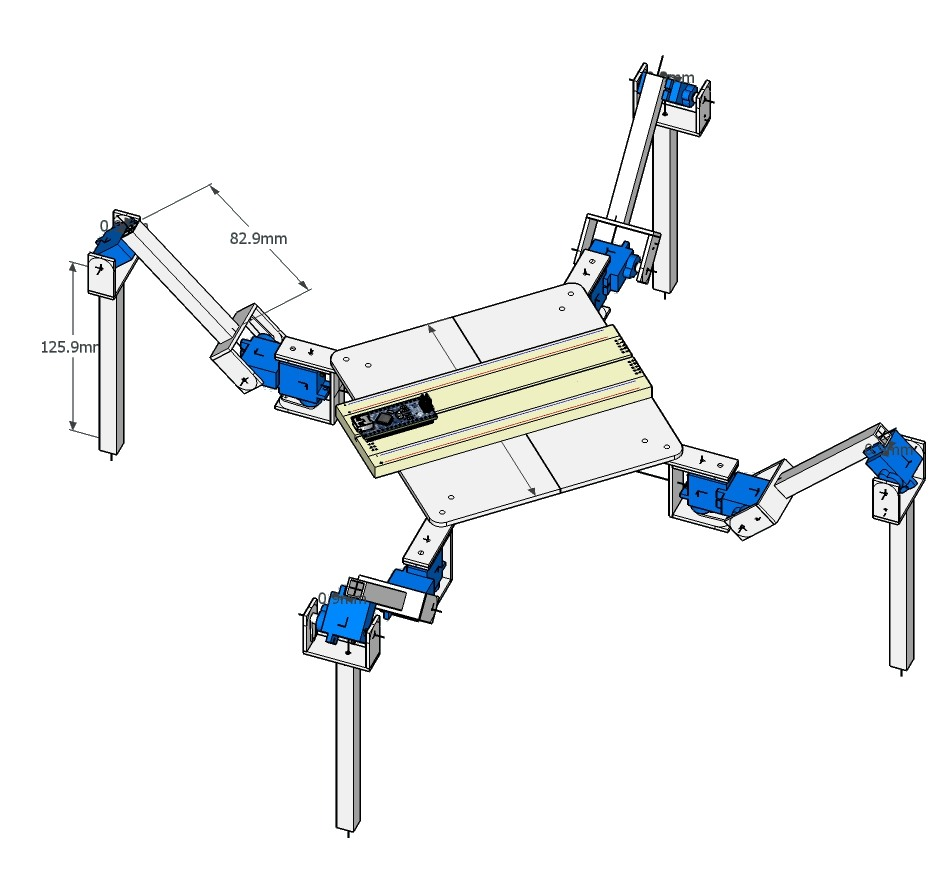
1. kiedys jak starczy czasu

* zeby skrecal i podnosil sie i obnizal, robil jakies proste figury (sekwencje ktore da rade zrobic)
* wizualizajca/interfejs na PC/Android w c#/Xamarin, (moze w Unity tylko czy po cos?) – komunikacja Bluetooth
* przesylanie z zewnatrz zadanej trasy XY i wykonywanie jej
* trzeci segment do nog na ok 4cm ktory bybly caly czas pionowo (moze jakas "stopa" amortyzowana na koncu)
* glowa - 8 diod losowo zapalanych/gaszonych

1. kiedys tam... jak juz mega starczy czasu

* sterowanie komunikacja ethernet wifi przez ESP (dostep do interfejsu przez WWW)
* platforma na robocie na 3 serwach/silnikach krokowych samopoziomujaca sie (moze regulator PID do szybszej reakcji)
* ramie na robocie, cos jak ta noga, identyczna klasa tylko z chwytakiem/manipulatorem, moze dluzszym, tylko praca na postoju bo problem z ogarnieciem srodka ciezkosci przy chodzeniu, chociaz mozna zrobic wiekszy "rdzen przekroju" i da sie wtedy cos przenosic
* kamerka na ramieniu 2D (rotZ i rotX) stopniach swobody (rzesyl obrazu/ nalozenie HUD i analiza)
* przy kamerce jakis wskaznik np laserowy, namierzanie celu, sledzenie, przewidywanie ruchu (walka robotow), albo rurka ze sprezynka/gumka zaczepem i kulka w srodku...

omijanie przeszkod, podazanie za swiatlem, podazanie do celu z wykorzystaniem ANN (tylko czy arduino to uciagnie), ale to najpierw musze to ogarnac



1. Cel i zakres pracy

Celem pracy jest opracowanie algorytmów sterowania, komunikacji, interfejsu uzytkownika oraz wykonanie fizycznego modelu symetrycznego robota 4-nożnego. Dzieki przyjeciu takiej konstrukcji robota mozliwe jest uzywanie go w trudnym, nierownym terenie, gdzie moglby sprawdzic sie lepiej niz roboty kolowe bądź gasienicowe. Ponadto zastosowanie tylko 4 nog obniza koszty wykonania.

Robot taki moze byc wykorzystany jako platforma do wykonywania innych zadan, po uprzednim montazu odpowiednich akcesoriow: ramienia z kamera, chwytakiem lub innym manipulatorem, nadajnika/odbiornika wiekszej mocy, jako swego rodzaju "naziemny dron" (trudniejszy do zauwazenia i rozpoznania) mogacy wykonywac wiekszosc zadan dronow latajacych - namierzanie, kontrola, rozpoznanie.

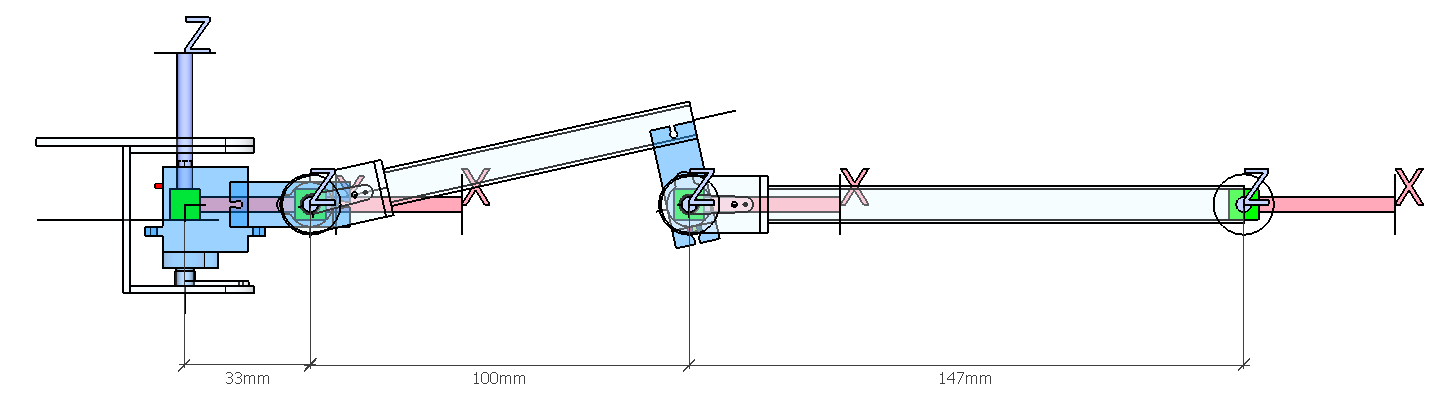
Po przeskalowaniu robota, przy minimalnych modyfikacjach moglby sluzyc jako platforma dla maszyn i urzadzen w przemysle np. w przy karczowaniu lasow, poszukiwaniach geofizycznych, w rolnictwie, budownictwie itp..

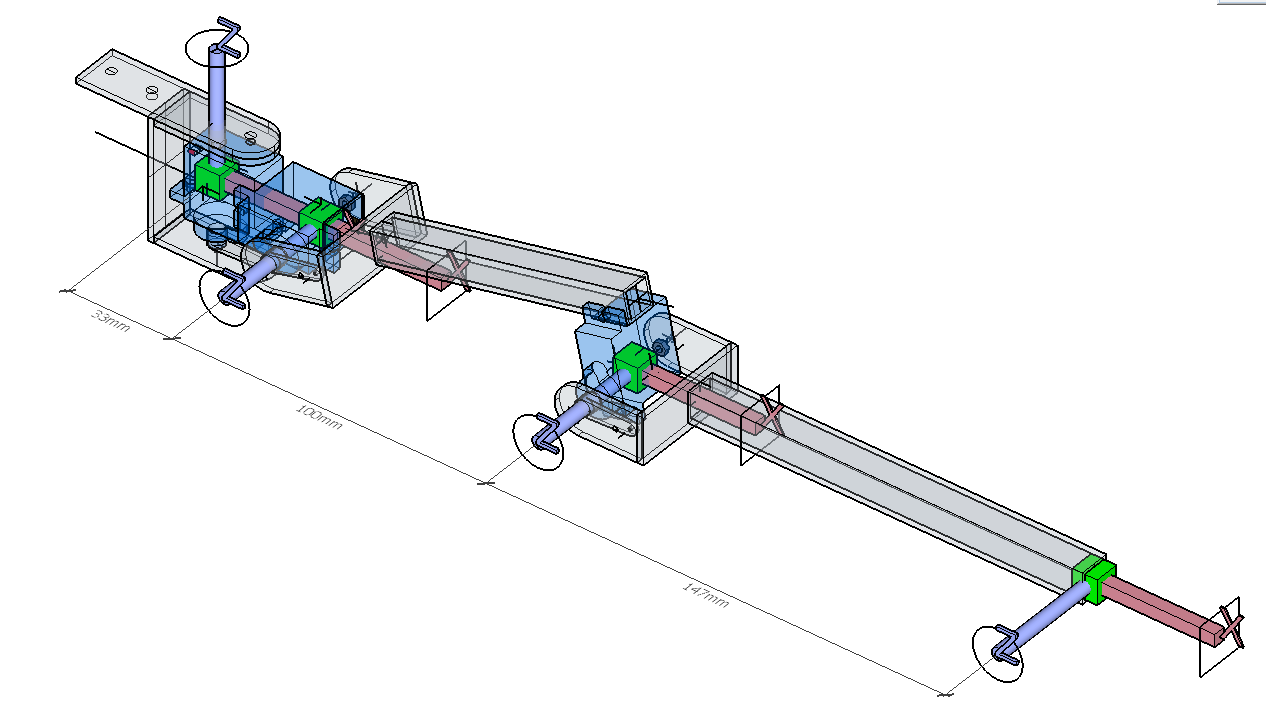
Niewielka powierzchnia zajmowana przez tak przyjeta konstrukcje robota umozliwia prace w gesto zastawionym terenie: place budowy, rozladunkowe bez koniecznosci tyczenia sciezek/droga jak dla pojazdow kolowych/gasienicowych, oczywiscie po montazu dodatkowych czujnikow na nogach zapobiegajacych uszkodzeniu innych przedmiotow/ludzi.

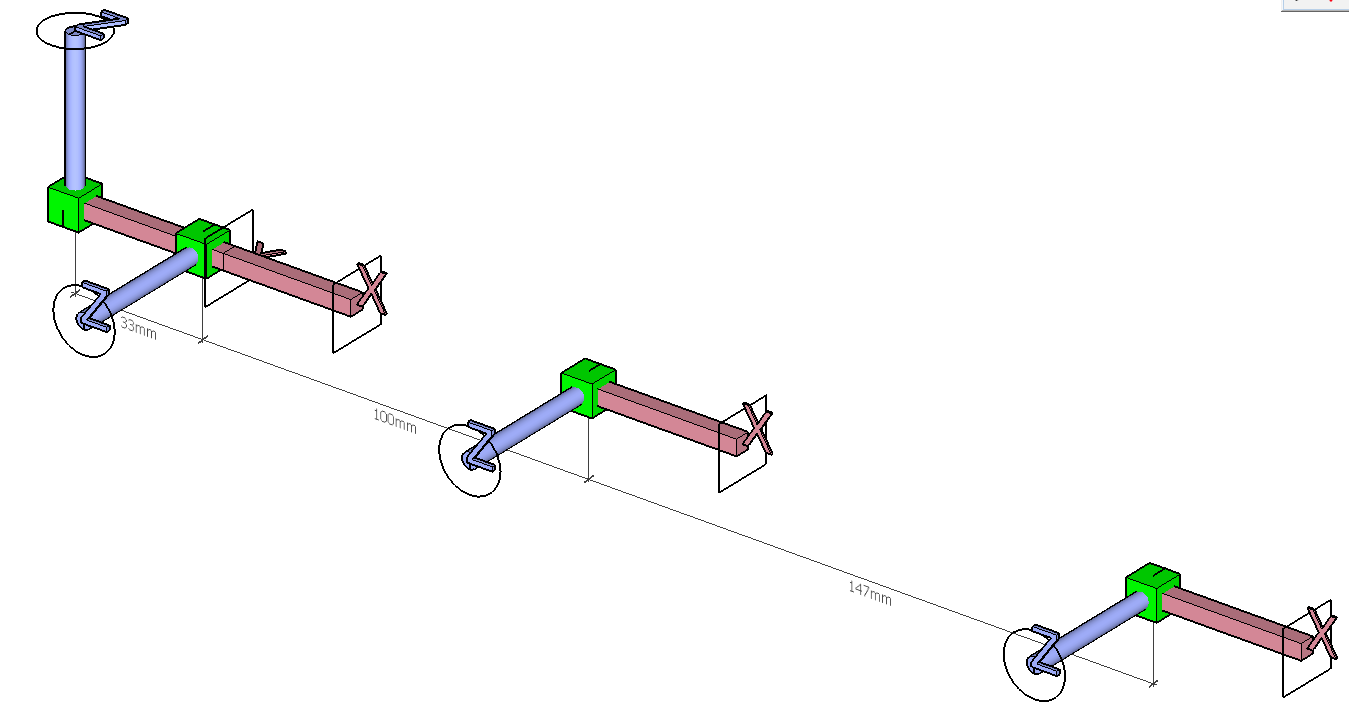
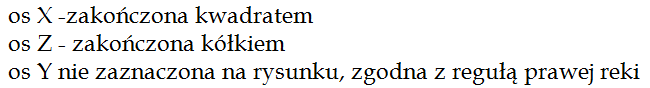
Polaczona ze soba wieksza liczba takich robotow moglaby wspolddzialac ze soba jako jakis system rozproszony, jednak opracowanie go wykracza poza zakres niniejszej pracy.

* 1. Szczegolowy zakres i zalozenia pracy obejmują:
* wykonanie modelu robota na 4-nogach, kazda z nog o 3 stopniach swobody (tranXYZ)
* glowny uklad sterujacy - procesor z rodziny STM32XX
* oprogramowanie w jezyku VisualC++ przy jak najbardziej obiektowym podejsciu do napotkanych problemow - opracowanie odpowiednich klas i metod
* zdarzeniowy algorytm sterowania - sieci Petriego (w razie koniecznosci/mozliwosci niezalezne od siebie)
* sterowanie reczne dedykowanym pilotem, lub przez interfejs na komputerze PC (komunikacja Bluetooth - modul HC-05)
* podazanie po zadanej trasie na plaszczyznie i rejestracja przebytej drogi i polozenia
* wykonywanie zaprogramowanych wczesniej figur: obnizanie/podnoszenie korpusu, przechylanie sie chodzil prosto - zaprogramowane/obliczone wczesniej sekwencje posrednie

­Schemat robota[[1]](#footnote-2) :







os X -zakończona kwadratem

os Z - zakończona kółkiem

os Y nie zaznaczona na rysunku, zgodna z regułą prawej reki

1. Dane wejsciowe

tabelka DHsym z wartościami symbolicznymi:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Θ | d | a | α |
| 1 | Θ1,var | 0 | a1 | 90 |
| 2 | Θ2,var | 0 | a2 | 90 |
| 3 | Θ3,var | d1 | 0 | 270 |
| 4 | Θ4,var | 0 | a3 | 0 |

tabelka DHnum z wartościami konstrukcyjnymi robota:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Θ | d | a | α |
| 1 | Θ1,var | 0 | 50 | 90 |
| 2 | Θ2,var | 0 | 100 | 90 |
| 3 | Θ3,var | 50 | 0 | 270 |
| 4 | Θ4,var | 0 | 150 | 0 |

Kat teta1 (kierunek)wyznaczyc mozna szybko z tan(PX/PY), a dalej w O1 mamy zwykle ramie planarne tylko z pionie, dlatego po uproszczeniu tabelka do obliczen wyglada:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Θ | d | a | α |
| 1 | Θ1,var | 0 | a1 | 90 |
| 2 | Θ2,var | 0 | a2 | 90 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Θ | d | a | α |
| 1 | Θ1,var | 0 | 100 | 90 |
| 2 | Θ2,var | 0 | 150 | 90 |

Podobnie wygiete pod katem prostym ramie (aby mozna bylo zlozyc segmenty rownolegle do siebie), do obliczeni mozna zredukowac do jego przeciwprostokatnej (100mm).

Pierwszy i jedyny mozliwy do zmiany kat eluera φ to suma wyznaczonych wczesniej Θ1,var i Θ2,var .

1. Rownania z matlaba
   1. kinematyka odwrotna - XYZ -> qi

poczatek w ukladzie 1 – pamietac zeby dodac/odjac ramie 1[[2]](#footnote-3)

th1sym =

atan2(py, px) - atan2(-a2\*(1 - (a1^2 + a2^2 - px^2 - py^2)^2/(4\*a1^2\*a2^2))^(1/2), a1 - (a1^2 + a2^2 - px^2 - py^2)/(2\*a1))

atan2(py, px) - atan2(a2\*(1 - (a1^2 + a2^2 - px^2 - py^2)^2/(4\*a1^2\*a2^2))^(1/2), a1 - (a1^2 + a2^2 - px^2 - py^2)/(2\*a1))

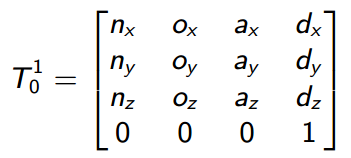
th2sym =

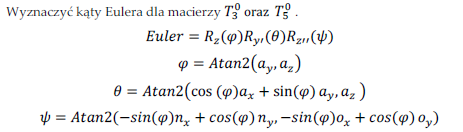
pi + acos((a1^2 + a2^2 - px^2 - py^2)/(2\*a1\*a2))

pi - acos((a1^2 + a2^2 - px^2 - py^2)/(2\*a1\*a2))

1. kąty Eulera

wg tabelki i wzorów (konwencja ZYZ):





1. Implementacja Sieci Petriego

wady i zalety sieci pteriego ze fajne porstsze latwe do modyfikacji mozna dawac kilka sieci nara niezaleznych

latwa intuicyjna reakcja na zdarzenia latwey do przedtsaiwenia

ze sieci niekolrowe o roznych wagach

tranzycje: bez kodu po rostu warunek logiczny, stany chwilowe tylko na czas przejsci ado kolejnego miejsca

miejsca: porcedury mozna wpisac do wykoania, miejsca stany stale

wykorzystany program petrnetsim ta praca dyoplmowa tego gostka

Mimo iż temat sieci petriego neijest nowy, i jest dostepnych wiele materialow teoretycznych, sama idea jest dosc prosta jednak konkretnych latow dostepnych rozwiazan i algorytmow brak - autor szukal w intenecie i nie znalazl w ziwazku z cyzm musial zaglenic sie w temat i obadal go sobei sam

zalozeniem bylo nie zaglebianie sie w teoretyczne rozwania na temat zytonosci, osiaglanosci itp a odpalenie sieci a by po obliczeniu koniecznych jej stalych specyficznych dla niej wlasciwosci oraz zadaniu wektora znakowan poczatkowych dzialala sobie sama

dokladny projekt sieci, testowanie

teoretycznie nie powinno to byc trudne i take nie bylo, rzecz sprowadzac powinna sie w sumie do mnozenia odpowiednich macierzy.

na podstawie , opracowano m-plik w matlabie , nastepnie na jego podstawie dla sprawdzenia poprawnosci dzialania algorytmu napisalem odpowiednie klasy i metody na poczatek w konsolowej aplikacji windows majac na uwadze szybkie przelozenie kodu na mikrokontoroler (poczatkowo platforma arduino na atmega8).

W przyjeto konwencje macierzy incyndencji [wiersze x kolumny <=> miejsca x tranzycje], stad macierz incydencji do obliczen jest transponowana, za to w algorytmie w w toku obliczen nie wystepuje transponowanie innych macierzy.

rzecz na ktora tzreba zwrocic uwage to ze mimo ze mozna przmenoszy cala macierz xxxpopraw to ladniexxx nalezy odpalac tranzycje (wiersze macierz Teye) po kolei poneiwaz odpalenie kazdej pojedynczej tranzycji wplywa nam na biezace znakowanie co z kolei wplywac moze na kolejne tranzycje.

ma to rowniez te zalete ze poniewaz macierz ta jest jednostkowa i diagonalna a odplamay naraz jedna tranzycje nie jest konieczne mozenie tcyh macierzy poniewaz wystaczy do wektroa znakowan iezacyhc dodac i-ty wiersz (wektor)transponowanej macierzu incdencji dla i tej tranzycji pod warunkiem ze jest gotowa do odpalenia co sprawdzamy odpowiednia metoda.

druga taka rzecza zwiazana ze specyfika zastosowania jest opracowanie metody aby unikncan dokladania zetonu przez caly czas trwania stanu a dodajacej zeton tylko w momencie wystapaienia stanu (zbocza nrasatacjacego) i kotntroli tego, aby przy trwaniu stanu dokladac zetonu przy kazdym prrzejsciu przez siec (przez caly czas aktywacji czujnika (

rysunke algorytm

ZAŁ I – nieliniowosc serw SG90 ­- kalibracja

# Bibliografia

1. **Krzysztof Sacha, prof. dr hab. inż.** http://www.ia.pw.edu.pl/~sacha/petri.html. [Online]

2. www.stackoverflow.com. [Online]

1. dodatnie wartosci theta -> CCW, układy współrzędnych jak w tabelkach w pkt.6. [↑](#footnote-ref-2)
2. na dzis rowne 33mm [↑](#footnote-ref-3)