

**Robot Półpająk**

**bo ma 4 nogi**

**opis i takie tam**

LP1 C1, EADI-2

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
* 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 (, przesyl 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. Uwagi

nie jest to praca inzynierska wiec prosze o wyrozumialosc poneiwaz nie dbalem zbytnio o gramatyczno naukowa strone, wiele rzeczy pisalem wlasnymi slowami nie tracac czasu na ubieranie tego w skomplikowane slowa, tym bardziej ze docelowo chce przetlumaczyc to na angielski i wrzucic na *instructables.com* aby bylo to zrozumialem dla zwyklego czlowieka, problem tu opisany podejrzewam jest dobrze rozpoznany wsrod fachowcow wiec nie jest do nich adresowany

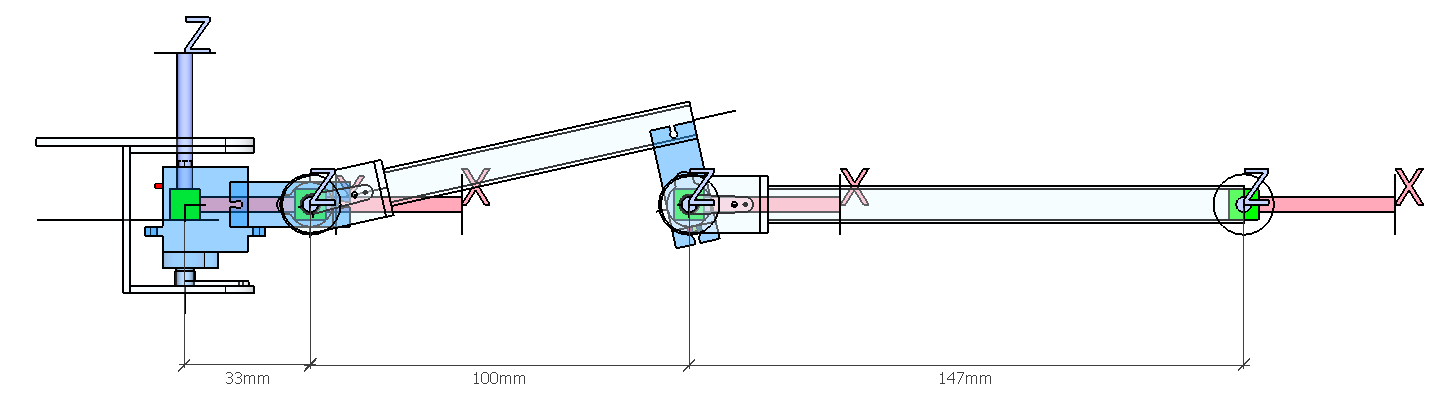
do chodzenia zawsze rownania z 1 wiersza bo zawsze maja th1 (kat 1go ramienia) dodatnie, a do pokazywania tez zawsze ten dodatni

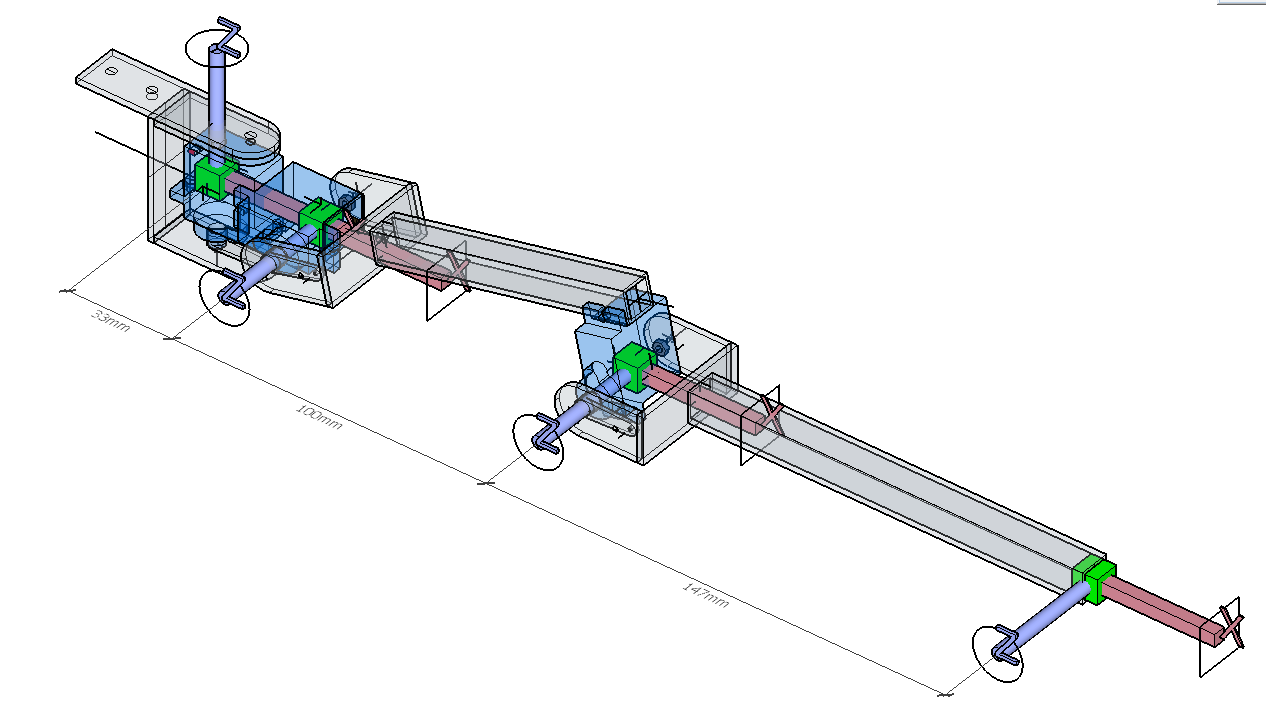
dolicz katy eulera to by mozna zrobic podawacz (pilnuje poziomu chwytaka), jeno trza dokupic serwo pilnujace poziomu i sciskajace chwytak+krancowka zeby wiedzial ze juz scisnal+3 pokretla, albo komunikacja do zadanego polzoenia itp

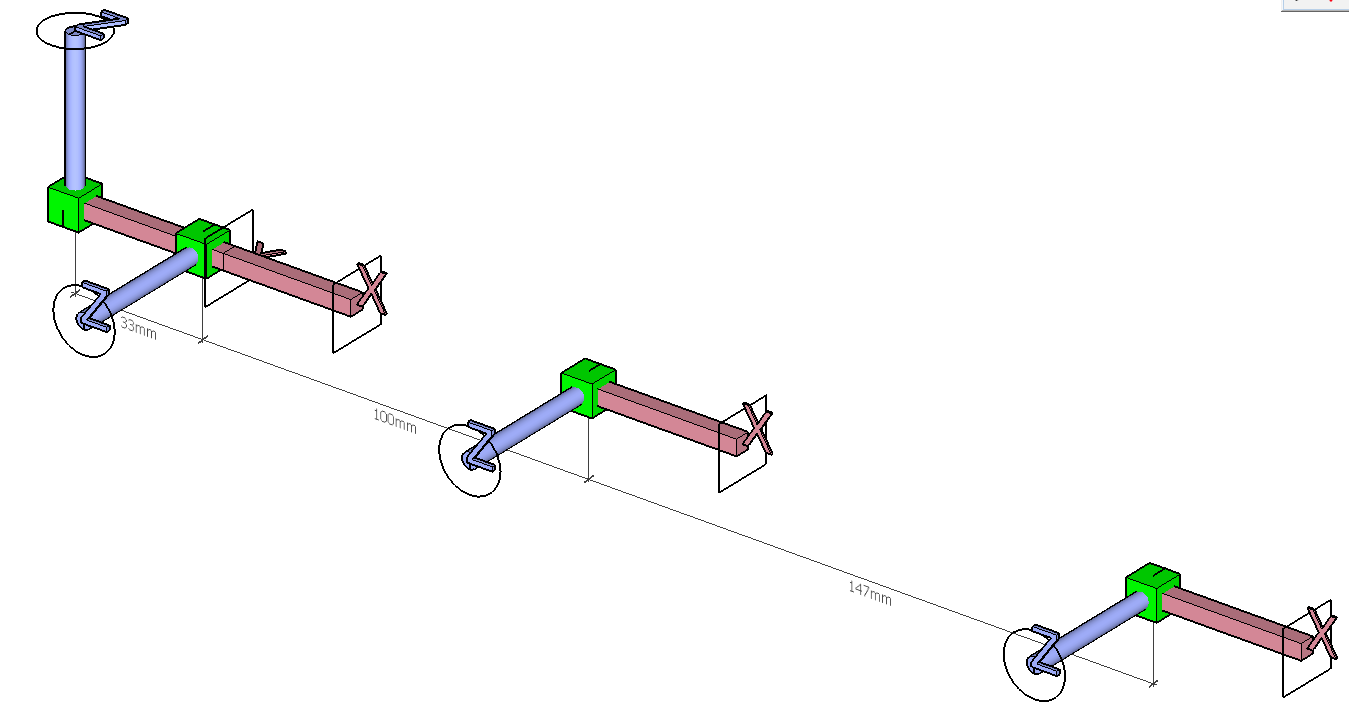
ZAŁ I – nieliniowosc serw SG90 i rozwiazanie

1. Schemat robota (układy współrzędnych jak w tabelce):

dodatnie wartosci theta -> CCW







os X -zakonczona kwadratem

os Z - zakonczona kolkiem

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

1. Dane wejsciowe (numeracja od 1 a nie od 0 jak w instrukcji do ćwiczenia)

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)

1. Rownania z matlaba

poczatek w ukladzie 1 – pamietac zeby dodac/odjac ramie 1

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. Kinematyka prosta qi -> XYZ
2. Macierze T14

[ cos(theta2 + theta4)\*cos(theta1), -sin(theta2 + theta4)\*cos(theta1), sin(theta1), cos(theta1)\*(a1 + a4\*cos(theta2 + theta4) + a2\*cos(theta2) + d3\*sin(theta2))]

[ cos(theta2 + theta4)\*sin(theta1), -sin(theta2 + theta4)\*sin(theta1), -cos(theta1), sin(theta1)\*(a1 + a4\*cos(theta2 + theta4) + a2\*cos(theta2) + d3\*sin(theta2))]

[ sin(theta2 + theta4), cos(theta2 + theta4), 0, a4\*sin(theta2 + theta4) - d3\*cos(theta2) + a2\*sin(theta2)]

[ 0, 0, 0, 1]

1. wektor P04

>> Psym(:,4)

cos(theta1)\*(a1 + a4\*cos(theta2 + theta4) + a2\*cos(theta2) + d3\*sin(theta2))

sin(theta1)\*(a1 + a4\*cos(theta2 + theta4) + a2\*cos(theta2) + d3\*sin(theta2))

a4\*sin(theta2 + theta4) - d3\*cos(theta2) + a2\*sin(theta2)

>> Pnum(:,6)

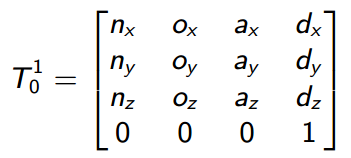
cos(theta1)\*(150\*cos(theta2 + theta4) + 100\*cos(theta2) + 50\*sin(theta2) + 50)

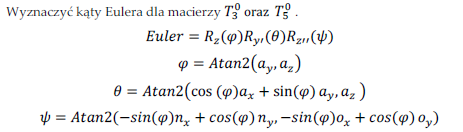
sin(theta1)\*(150\*cos(theta2 + theta4) + 100\*cos(theta2) + 50\*sin(theta2) + 50)

150\*sin(theta2 + theta4) - 50\*cos(theta2) + 100\*sin(theta2)

1. kinematyka odwrotna - XYZ -> qi
2. kąty Eulera dla T03 I T05

wg tabelki i wzorów:





Wyznaczone kąty po sprawdzeniu numerycznym są zapisane w konwencji ZYZ

>> Eulsym(:,4)

ans = -(pi\*sign(cos(th2)))/2

(pi\*sign(cos((pi\*sign(cos(th2)))/2)\*sin(th2) + sin((pi\*sign(cos(th2)))/2)\*cos(th2)))/2

atan2(cos((pi\*sign(cos(th2)))/2)\*cos(th3 + th4)\*sin(th2) + sin((pi\*sign(cos(th2)))/2)\*cos(th3 + th4)\*cos(th2), - cos((pi\*sign(cos(th2)))/2)\*sin(th3 + th4)\*sin(th2) - sin((pi\*sign(cos(th2)))/2)\*sin(th3 + th4)\*cos(th2))

>> Eulsym(:,6)

ans = atan2(sin(th3 + th4 + th5)\*sin(th2), -cos(th3 + th4 + th5))

angle(- cos(th3 + th4 + th5) - (sin(th3 + th4 + th5)\*sin(th2)\*(imag(cos(th3 + th4 + th5)) - real(sin(th3 + th4 + th5)\*sin(th2)))\*i)/abs(cos(th3 + th4 + th5) - sin(th3 + th4 + th5)\*sin(th2)\*i) - (sin(th3 + th4 + th5)\*cos(th2)\*(real(cos(th3 + th4 + th5)) + imag(sin(th3 + th4 + th5)\*sin(th2)))\*i)/abs(cos(th3 + th4 + th5) - sin(th3 + th4 + th5)\*sin(th2)\*i))

angle(((real(cos(th3 + th4 + th5)) + imag(sin(th3 + th4 + th5)\*sin(th2)))\*(cos(th2)\*cos(th6) + cos(th3 + th4 + th5)\*sin(th2)\*sin(th6)))/abs(cos(th3 + th4 + th5) - sin(th3 + th4 + th5)\*sin(th2)\*i) + ((real(cos(th3 + th4 + th5)) + imag(sin(th3 + th4 + th5)\*sin(th2)))\*(cos(th2)\*sin(th6) - cos(th3 + th4 + th5)\*cos(th6)\*sin(th2))\*i)/abs(cos(th3 + th4 + th5) - sin(th3 + th4 + th5)\*sin(th2)\*i) + ((sin(th2)\*sin(th6) + cos(th3 + th4 + th5)\*cos(th2)\*cos(th6))\*(imag(cos(th3 + th4 + th5)) - real(sin(th3 + th4 + th5)\*sin(th2)))\*i)/abs(cos(th3 + th4 + th5) - sin(th3 + th4 + th5)\*sin(th2)\*i) + ((cos(th6)\*sin(th2) - cos(th3 + th4 + th5)\*cos(th2)\*sin(th6))\*(imag(cos(th3 + th4 + th5)) - real(sin(th3 + th4 + th5)\*sin(th2))))/abs(cos(th3 + th4 + th5) - sin(th3 + th4 + th5)\*sin(th2)\*i))

>>