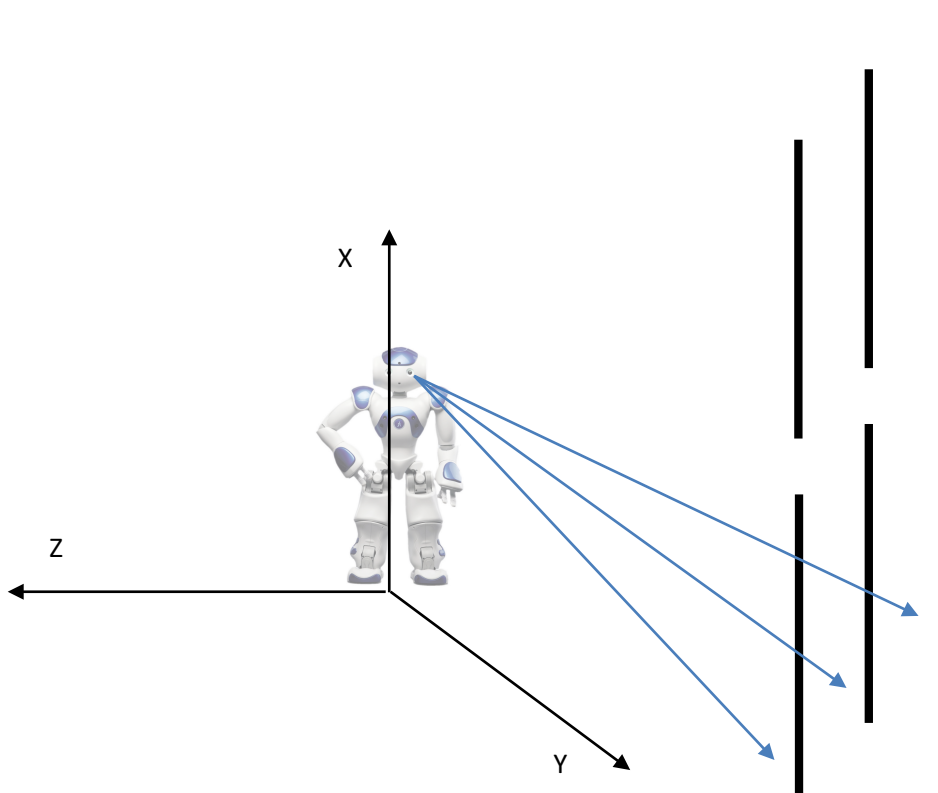


Analýza

Rotácia agenta je určená tromi atribútmi X,Y a Z. Jeho X a Y hodnoty dokážeme vypočítať na základe pohybu pevných bodov.



Obr. 1: Ukážka vnímania agenta pevných bodov (zástaviek)

Z obrázku môžeme vidieť ako vníma agent pevné body. Keďže vieme ich absolútnu pozíciu a vieme ich pozíciu, ktorú vnímame. Vieme povedať ako sa poloha odlišuje od pôvodnej. To však len v prípade, že poznáme pozíciu troch alebo viacerých pevných bodov.

Toto sa v projekte počíta v AgentRotationCalculator.java. Tá sa má dáta o pevných bodoch, ktoré robot videl a má dáta o tom, kde majú absolútnu pozíciu.

To môžeme podľa vzorca:

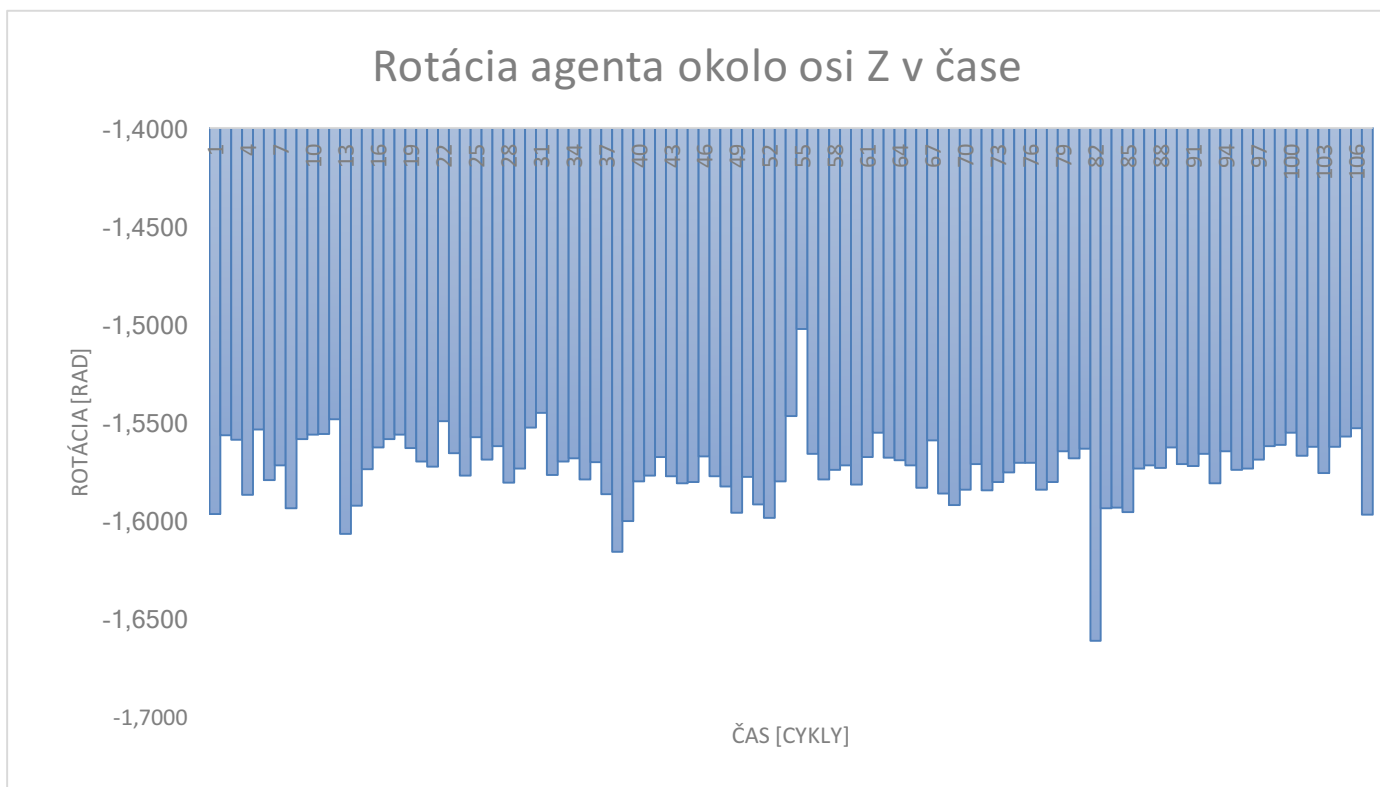
$$Z_{needed} = Z_{known} - \frac{(Y_2 - Y_1) \times (Z_{known} - Y_1) \times (Y_2 - Y_1)}{(Y_2 - Y_1) \cdot (Y_2 - Y_1)}$$

Potom dokážeme rotáciu vypočítať pomocou goniometrických funkcií:

```
double rotationX = asin(-xAxis.getZ());  
double rotationZ = atan2( -xAxis.rotateOverX(rotationX).getY() /  
cos(rotationX), xAxis.rotateOverX(rotationX).getX() / cos(rotationX));  
double rotationY = atan2( -yAxis.rotateOverX(rotationX).getZ() /  
cos(rotationX), zAxis.rotateOverX(rotationX).getZ() / cos(rotationX));
```

Pokus

Urobili sme pokus ako agent vníma svoju rotáciu v čase. Pri testovacom scenári, kde sa zo vzpriamenej polohy dostal na uhol 90°. V grafe 1 sú vyobrazené namerané hodnoty



Graf 1: Zobrazenie hodnôt rotácie agenta okolo osi Z

Z grafu 1 vidíme, že rotácia okolo osi Z nie je meraná v tejto implementácii správne.

Záver

Algoritmus sme testovali na testovacom scenári, kde sa hráč predkloní do úplného predklonu (teda 90 stupňov) v priebehu 10 sekúnd. Z pokusu sme zistili, že hodnoty vypočítané týmto algoritmom, neboli správne.

Toto je možné nesprávnou implementáciou prepočtu z videných bodov. Svoju úlohu zohráva aj zašumenie dát.