TestCode

In deze map kunnen alle geteste codes gevonden worden. Deze codes zijn geschreven om de bewegingsfuncties te testen.
Maar ook voor het testen van de verschillende aanstuur mogelijkheden via de controller.
Alle codes welke geschreven worden om een functionaliteit te testen worden in deze map gezet.

Codes

- MoveJ.py deze code is gebruikt in het testplan/experimenteel onderzoek voor het kiezen van een bewegingsfunctie.
- MoveCart.py deze code is gebruikt in het testplan/experimenteel onderzoek voor het kiezen van een bewegingsfunctie.
- JOGcontroller.py de gebruikte code om de vloeiende versnellende beweging te maken via de controller.
- Controller2.py de gebruikte code om de schokkende rotatie code te krijgen.
- calc.py de gebruikte code om de nieuwe coordinate uit te rekenen voor de Controller2.py code.

Controller2.py

Dit is de code van de schokkende versie welke wordt aangestuurd via de controller. De controller wordt gebruikt om de coordinate van de arm te vergroten of verkleinen. Als er een nieuw coordinaat is worden via de `calc.py` code de juiste coordinate berekent. Dit door de kop als het ware als een cirkel rond zijn middenpunt te draaien.
Door op de controller de linker joystick horizontaal te bewegen worden de X coordinaat verhoogt of verlaagd. Hierna wordt het juiste Y coordinaat gevonden om de afstand tussen de kop van de arm en het middenpunt te behouden. Dit wordt gedaan door middel van een omgeschreven cirkel formule. $y^2 = x^2 + r^2$. Door x en r mee te geven in de functie wordt de Y waarden berekent. Als X groter wordt dan de straal wordt de negative Y genomen en zal de x afnemen i.p.v. toenemen.
Bij deze code kan er alleen versneld worden door de stappen welke gemaakt worden te vergroten. En door de punt tot punt bewegingen geeft dit een schokkend eindresultaat.

JOGcontroller.py

Dit is de code van de vloeiende versie welke wordt aangestuurd via de controller. De controller wordt gebruikt om de robot op een enkele as te verplaatsen. Door de `JOG` functie te gebruiken wordt de verplaatsing alleen uitgevoerd op een enkele as. In tegenstelling tot de `Controller2.py` blijft deze beweging vloeien tot de joystick wordt losgelaten. De snelheid is hierdoor wel variabel te maken door de intensiteit van de input. Deze snelheid wordt berekend door de inputwaarde van de joystick om te zetten naar procenten door deze keer 100 te doen. Voor de verplaatsing op de Z-as wordt de input eerst verhoogd met 1 omdat de standaard waarde van deze joysticks -1 is. Om te voorkomen dat de snelheid boven de 100% komt, wordt na de vermenigvuldiging de waarden gehalveerd.
Het nadeel is dat de snelheid alleen kan worden veranderd als de functie opnieuw wordt aangeroepen. Daarom moet de beweging tussen elke verandering worden stopgezet. Hierdoor kan het zijn dat de functie niet goed opnieuw gestart wordt. Dit kan er voor zorgen dat de arm zijn beweging compleet zal stopzetten na een verandering in snelheid. Ook kan er alleen bewogen worden in lijnen welke evenwijdig zijn aan het assen-stelsel. Dus alleen de x, y of z waardes kunnen per keer veranderd worden.

calc.py

Hierin staan alle berekeningen welke nodig zijn voor `Controller2.py`. Er wordt gebruik gemaakt van de cirkel formule in combinatie met de stelling van Pythagoras.

Omdat de oorsprong (0,0) altijd het middelpunt van de robot vormt, kunnen we de stelling van Pythagoras toepassen: de lengtes van de zijden van de driehoek komen dan overeen met de coordinate in het assenstelsel.

`getA(x,y)`: In deze functie wordt door het gebruik van de tanges de hoek van het kopstuk tenopzichten van de x-as berekent.

`getR(x,y)`: In deze functie wordt door middel van de cirkel formule $x^2 + y^2 = r^2$ de afstand tussen de kop en het middelpunt van de robot gevonden. Dit is de straal van de cirkel welke wordt gebruikt voor de verplaatsingen.

`rIncrement(r, a)`: In deze functie wordt een nieuwe straal doorgegeven. Samen met de hoek van de eerder berekende a worden er nieuwe x en y coordinate berekent voor de nieuwe straal.
Dit wordt gedaan door gebruik te maken van de cosinus (nieuw x coordinaat) en de sinus (nieuw y coordinaat). Door de formule om de hoek te berekenen om te schrijven kan er een nieuwe lengte van de x of y as berekend worden.

`calcPoint(x, r, takeNeg)`: In deze functie wordt het bijpassende Y coordinaat berekent voor een nieuwe x waarde. Om dit te doen wordt er gebruik gemaakt van de standaard cirkel formule: $(x-a)^2 + (y-b)^2 = r^2$. Omdat het a en b het middenpunt van de cirkel is en deze altijd gelijk is aan (0,0) kunnen deze worden weggelaten. Hierna eindigen we met $x^2 + y^2 = r^2$ dit kan worden omgeschreven worden naar $y^2 = r^2 - x^2$ hiermee krijgen we dus een nieuw y coordinaat. Als de robot een negative y waarden nodig heeft, wordt de berekende y waarde negatief gemaakt. Dit kan omdat bij een machtsvergelijking er altijd een positive en negative waarden wordt uitgerekend.

MoveCart

Deze code is gebruikt in het testplan movement en wordt hier veder in uitgelegd.

MoveJ

Deze code is gebruikt in het testplan movement en wordt hier veder in uitgelegd.

Changelog

Wie	Datum	Wijziging
Ivo Bruinsma	25-09-2025	Aanmaak document
Ivo Bruinsma	10-11-2025	Introductie & opsomming codes
Ivo Bruinsma	14-11-2025	Uitleg per code