

Inbetriebnahme und Bedienung des Open Manipulator X mittels Handlungsplanung (als Vorbereitung für den Lehrbetrieb)

Fabian Claus

Bachelorarbeit

Technische Hochschule Brandenburg
Prof. Dr. Jochen Heinsohn, Ingo Boersch

Inhaltsverzeichnis

1	Einführung	5
1.1	Aufgabe	5
1.2	Abgrenzung/Motivation	5
2	Grundlagen	7
2.1	OpenMANIPULATOR-X	7
2.1.1	ROS2	8
2.2	Planung	8
2.2.1	Allgemein	8
2.2.2	STRIPS	8
2.2.3	PDDL	8
2.2.4	PDDL-Plugin für VS Code	8
2.2.5	Partial Order Planning Forward	8
2.2.6	Behavior Tree	8
3	Konzept	9
3.1	Struktur Nodes	9
4	Implementierung	11
4.1	Inbetriebnahme Greifarm	11
4.1.1	Zusammenbau	11
4.1.2	Virtuelle Maschine	12
4.2	Steuerung Greifarm	13
4.2.1	OMX-Controller	13
4.2.2	Topics	13
4.2.3	Kinematik	13
4.3	PlanSys2	13
4.3.1	PDDL-Domain	13
4.3.2	Action Nodes	13
5	Ergebnis	15

Kapitel 1

Einführung

1.1 Aufgabe

Im Rahmen der Bachelorarbeit soll der Greifarm OpenMANIPULATOR-X der Firma Robotis in Betrieb genommen sowie die Möglichkeiten der Steuerung erprobt werden. Weiterhin soll die Steuerung Mittels Handlungsplanung ermöglicht werden. Hierfür sind bestehende Bibliotheken und Frameworks zu evaluieren und ein ausgewähltes zu Implementieren.

1.2 Abgrenzung/Motivation

- Bewegungsplanung

Kapitel 2

Grundlagen

2.1 OpenMANIPULATOR-X

Der OpenMANIPULATOR-X ist ein von der Firma ROBOTIS¹ hergestellter Greifarm nach den Prinzipien “OpenSoftware“ und “OpenHardware“. OpenSoftware steht hierbei dafür, dass es ein OpenSource Projekt ist und auf dem OpenSource Projekt ROS (Robot Operating System) basiert. OpenHardware steht dafür, dass die meisten Komponenten als STL-Dateien zur Verfügung stehen und als Ersatzteile oder zum Anpassen des Greifarms mittels eines 3D-Druckers selbst hergestellt werden können.

Der OMX(Greifarm?,Abk?) ist eine 5DOF (5 Degrees of Freedom) Plattform, welche mittels 5 Servomotoren² gesteuert wird. Dies ist aufgeteilt in 4DOF für den Arm sowie 1DOF für den Greifer. Es kann eine Last bis 500g getragen werden

¹<http://en.robotis.com>

²DYNAMIXEL XM430-W350-T

2.1.1 ROS2

2.2 Planung

2.2.1 Allgemein

2.2.2 STRIPS

2.2.3 PDDL

2.2.4 PDDL-Plugin für VS Code

2.2.5 Partial Order Planning Forward

2.2.6 Behavior Tree

Kapitel 3

Konzept

3.1 Struktur Nodes

Um eine einfache sowie übersichtliche Steuerung des Greifarms zu ermöglichen wird die Funktionalität in mehrere ROS2 Nodes aufgeteilt. Generell werden folgende Funktionalitäten benötigt: der Planer, die Speicherung des aktuellen Zustands der Welt, die Ausführung des Plans sowie die Möglichkeit Eingaben zu verarbeiten und an die entsprechende Stelle weiterzuleiten.

Diese Aufteilung entspricht auch einer guten Aufteilung und Trennung der Verantwortungen um daraus ROS2 Nodes zu machen. Hier haben die Nodes folgende Verantwortlichkeiten:

Die Planungs-Node muss mit einer gegebenen Domäne und einem Problem einen Plan bestehend aus einer Reihe von Aktionen zurückgeben.

Die Welt-Node hält den aktuellen Zustand der Welt bzw. des Problems und muss diesen konsistent halten.

Die Ausführungs-Node muss entsprechend eines gegebenen Plans die beinhalteten Aktionen ausführen.

Die Eingabe-Node ermöglicht die Erstellung eines Problems mit einem Welt-Zustand und einem Ziel.

Kapitel 4

Implementierung

4.1 Inbetriebnahme Greifarm

Der OMX wird als Bausatz geliefert. Für die Inbetriebnahme ist daher der Zusammenbau und die Installation der entsprechenden Software nötig.

4.1.1 Zusammenbau

- Bausatz ca 40 Teile (ohne Schrauben)
- Rausbrechen Plastik bei Vorbereitung Servos
- Servos einzeln anschließen und per Dynamixel Wizard ID setzen

Der Bausatz des OMX besteht aus ca. 60 Teilen (ohne Schrauben, s. Abbildung 4.1). Einige der mit den Servomotoren mitgelieferten Teile werden dabei nicht benötigt, da der Bausatz des OMX diese auch enthält oder ersetzt (z.B. längere Kabel). Von allen Schrauben wurde außerdem Ersatz mitgeliefert.

Der Zusammenbau erfolgte nach der auf der Webseite verfügbaren Bauanleitung VERWEIS ANLEITUNG. Zu beachten ist, dass hier vorausgesetzt wird, dass den Servos bereits die IDs 11 (Basis des Greifarms) bis 15 (Greifer) zugewiesen wurde. Dies kann über die Software DYNAMIXEL Wizard¹ gemacht werden: die Servos einzeln über das U2D2² an den PC anschließen, die ID setzen und den Servo entsprechend markieren oder die ID merken. Weiterhin müssen bei den Abdeckungen der Servos 12 und 14 die vorgestanzten Abdeckungen herausgebrochen werden. Dies ist in der Anleitung leicht zu übersehen.

¹VERWEIS ODER LINK

²ERKLÄRUNG ODER VERWEIS HINZUFÜGEN



Abbildung 4.1: Bausatz für den OpenMANIPULATOR-X

4.1.2 Virtuelle Maschine

- Ubuntu 20.04
- Auflösung Einstellung "keine"
- Installation ROS2 Foxy
- Installation OMX

Zur Nutzung des Greifarms wurde eine virtuelle Maschine (VM) mit Virtual-Box ³ von Oracle aufgesetzt. Als Betriebssystem der VM wurde das für ROS2 Foxy empfohlene[1] Ubuntu 20.04 ⁴ gewählt. Danach wurde entsprechend der Anleitung für den OpenMANIPULATOR-X [2] zuerst ROS 2 Foxy über das Installations-Script von ROBOTIS und im Anschluss die für den Greifarm benötigten Packages installiert.

³<https://www.virtualbox.org>

⁴<https://releases.ubuntu.com/20.04/>

4.2 Steuerung Greifarm

4.2.1 OMX-Controller

4.2.2 Topics

4.2.3 Kinematik

4.3 PlanSys2

4.3.1 PDDL-Domain

- Durative Actions
- Gripper + Blockworld
- keine existential/negative Preconditions

4.3.2 Action Nodes

Kapitel 5

Ergebnis

Tabellenverzeichnis

Abbildungsverzeichnis

4.1	Bausatz für den OpenMANIPULATOR-X	12
-----	---	----

Listings

Literaturverzeichnis

- [1] Installing ROS 2 on Ubuntu Linux. URL: <https://docs.ros.org/en/foxy/Installation/Ubuntu-Install-Binary.html>.
- [2] Openmanipulator-x. URL: https://emanual.robotis.com/docs/en/platform/openmanipulator_x/quick_start_guide/#install-ros-on-pc.