

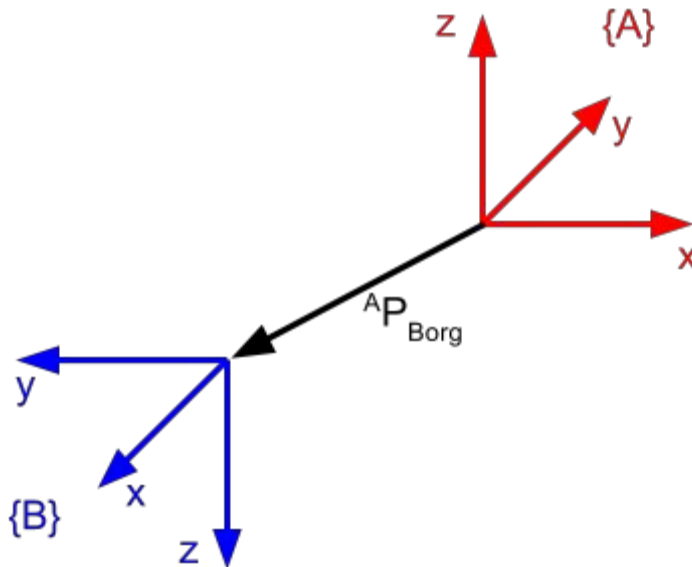
3. Übungszettel Robotik WS15/16

Prof. Daniel Göhring, Zahra Boroujeni
Institut für Informatik, Freie Universität Berlin
Abgabe online bis Dienstag, 10.11.2015, 12 Uhr s.t.

Fassen Sie Ihre Ergebnisse (Bilder und Beschreibung der Ergebnisse) in einer PDF-Datei zusammen und benennen Sie diese "RO-03-<Nachnamen der Studenten>.pdf". Quellcode soll *nicht* im PDF erscheinen.

1. Aufgabe (4 Punkte): Koordinaten-Transformation (Schriftlich im Pdf)

a. (3 Punkte) Gegeben seien die folgenden Koordinatensysteme:



Geben Sie die homogene Transformationsmatrix ${}^A_B T$ an, die einen im blauen Koordinatensystem B repräsentierten Vektor aus Sicht des roten Koordinatensystems A beschreibt. Die Translation zwischen den beiden Koordinatensystemen ist dabei

$${}^A P_{Borg} = (-3, -2, -2).$$

b. (1 Punkt) Transformieren Sie den Vektor $v_B = (1, -2, 3)$ mit der Transformationsmatrix ${}^A_B T$ aus Aufgabe a) in das Koordinatensystem A . Geben Sie den Rechenweg mit an.

2. Aufgabe (6 Punkte): Kommunikation zwischen ROS-Nodes (Programmieraufgabe)

a). (3 Punkte) Lesen Sie hierzu im Tutorial

<http://wiki.ros.org/ROS/Tutorials/>

die Punkte 11, 12 und 13, bei denen zwei miteinander kommunizierende Nodes (Talker, Chatter) beschrieben werden.

Implementieren Sie die Talker-Chatter-Nodes einmal in der Programmiersprache C++ **oder** Python. Wandeln Sie das Beispiel so ab, dass der Talker zum Listener eine initial ausgewürfelte Zahl (aus dem Bereich der ganzen Zahlen von 1-100) kommuniziert, der Listener addiert diese Zahl zu seiner eigens ausgewürfelten Zahl (aus dem Bereich der ganzen Zahlen von 1-100) und kommuniziert die Summe an den Talker. Der Talker addiert diese Zahl zu seiner letzten Summe und kommuniziert das Ergebnis wieder an den Listener usw.

Geben Sie die 20 Ausgaben (für 10 Schritte) aus, wobei sowohl der Talker als auch der Listener jeweils 10 Summen ermittelt (Abgabe in Pdf-Datei).

Beispiel: Talker Initial: 4; Listener Initial: 8; Listener Summe: $8+4 = 12$;

Talker Summe: $4+12 = 16$; Listener Summe: $12+16 = 28$

Geben Sie den Source-Code für die beiden Nodes als 2 Dateien ab.

b). (3 Punkte) Verwenden Sie den Roboterarm aus dem letzten Aufgabenblatt, holen Sie sich in dem dafür vorgesehenen Verzeichnis ein Update für das Repository

```
git pull https://github.com/ZahraBoroujeni/arm2r
```

In der display.launch wurde folgende Zeile hinzugefügt, um Namenskonflikte der rostopic zu vermeiden:

```
<rosparam param="source_list">[/calibrated/joint_states]</rosparam>
```

mit folgendem Kommandozeilenbefehl können dann (nach starten der display.launch) die Gelenke des Roboters verändert werden:

```
rostopic pub -r 10 calibrated/joint_states sensor_msgs/JointState '{header: auto, name: [\'joint_base_first\', \'joint_first_second\'], position: [1, 0.5418], velocity: [], effort: []}'>
```

Modifizieren Sie nun eine rosnode, die die Gelenkwinkel des Roboters dynamisch verändert, z.B. in Form von Sinuskurven. Die Datei src/arm2_tf_listener.cpp wurde dazu bereits im Repository angelegt.

Beschreiben Sie die von Ihnen gewählten Gelenkwinkelfunktionen und geben Sie die Distanz des Endeffektors zum Basispunkt über der Zeit in einem Diagramm aus. Um den Punkt des Endeffektors zu bestimmen, ist es hilfreich, ein virtuelles Gelenk (virtual joint) im URDF-File hinzuzufügen.