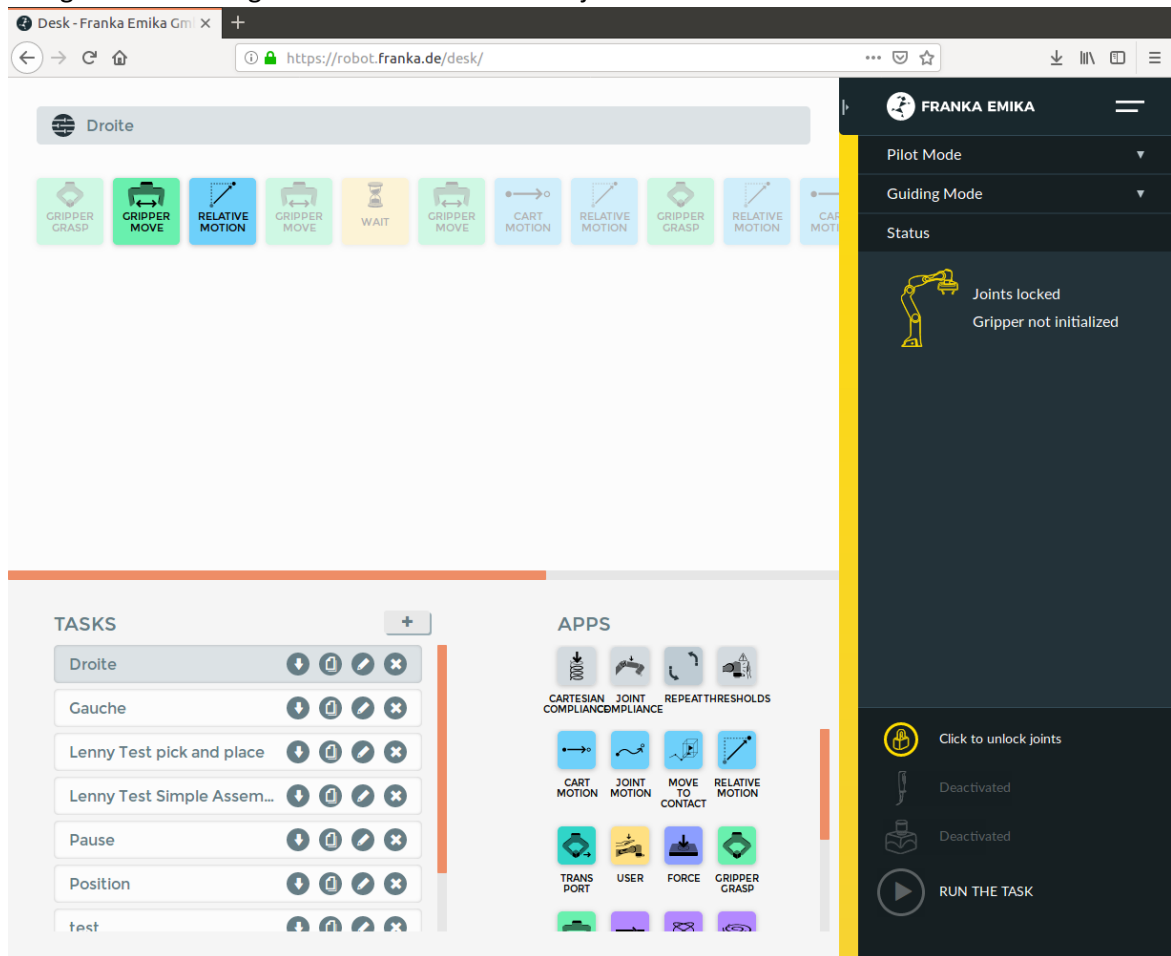


## Start Franka desktop with Ubuntu 16 installed on it.

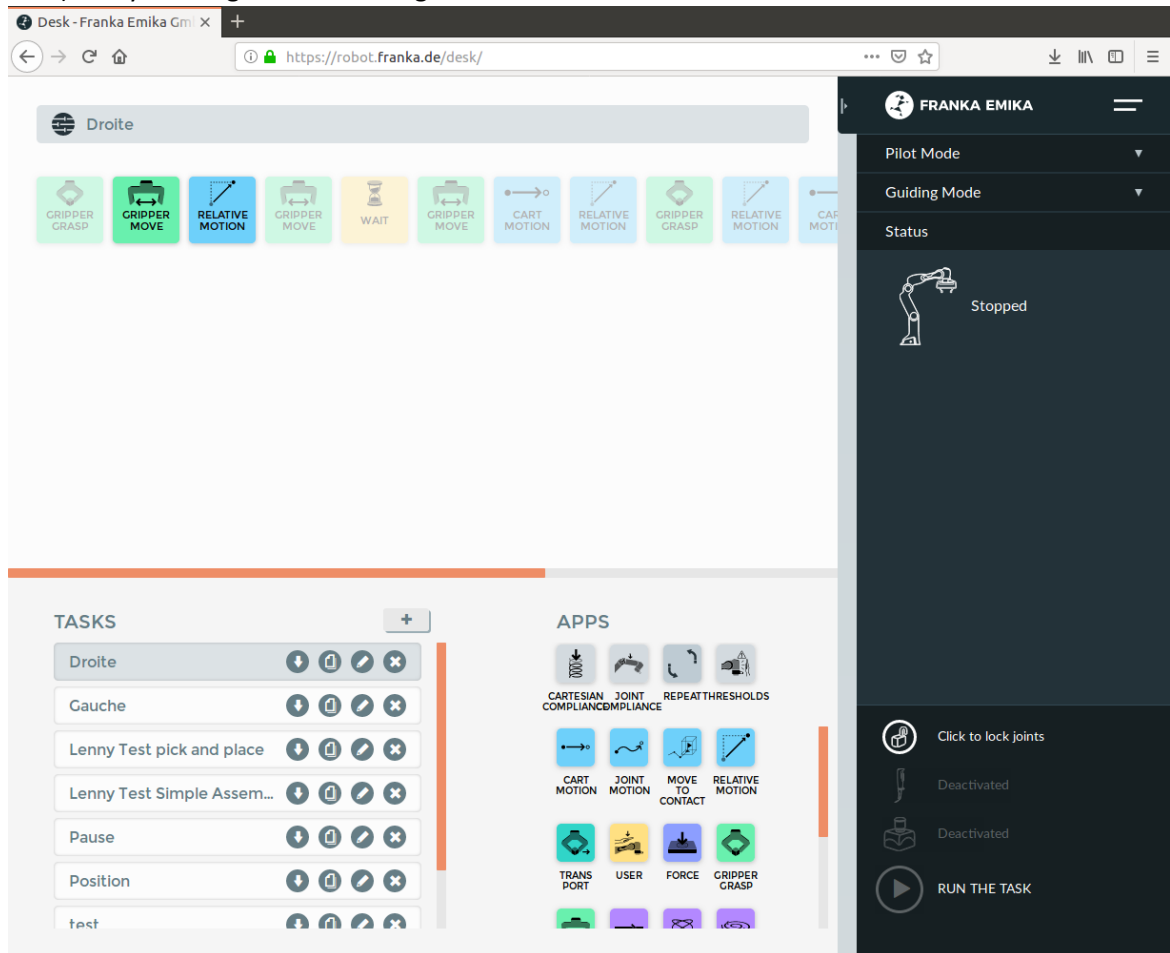
- 1) When starting the computer, press “shift” button till you get a menu where you can choose which Ubuntu version you want to run.
- 2) Choose “Advanced options for Ubuntu”.
- 3) Choose the first “rt” (real time kernel) version: “Ubuntu, with Linux 4.16.18-rt12”.
- 4) The password of this computer is “franka123”, its IP address “192.168.2.105”.

## Start the Franka Panda robot via Desk.

- 1) Turn on the Franka Control (black box).
- 2) Use an ethernet cable to connect the Franka Base and the Franka desktop.
- 3) Choose “Wired connection 1”. See below for specifications of this network.
- 4) Open Firefox Web Browser.
- 5) Go to <https://robot.franka.de/desk/>
- 6) The username is “admin”, the password is “franka123”.
- 7) You get the following window when the robot’s joints are locked.



- 8) To unlock the joints, click on “click to unlock the joints”. You will hear 7 clicks (the robot will move a bit) and you will get the following window.



## Specifications of “Wired connection 1”

The image displays two screenshots of the 'Editing Wired connection 1' window, showing different tabs.

**Top Screenshot (General Tab):**

- Connection name: Wired connection 1
- General | Ethernet | 802.1x Security | DCB | IPv4 Settings | IPv6 Settings
- ☒ Automatically connect to this network when it is available
- ☒ All users may connect to this network
- ☐ Automatically connect to VPN when using this connection
- Firewall zone: Default
- Buttons: Cancel, Save

**Bottom Screenshot (Ethernet Tab):**

- Connection name: Wired connection 1
- General | Ethernet | 802.1x Security | DCB | IPv4 Settings | IPv6 Settings
- Device: F0:4D:A2:DA:8A:50
- Cloned MAC address: (empty)
- MTU: automatic - + bytes
- Wake on LAN:
  - ☒ Default
  - ☐ Phy
  - ☐ Unicast
  - ☐ Multicast
  - ☐ Ignore
  - ☐ Broadcast
  - ☐ Arp
  - ☐ Magic
- Wake on LAN password: (empty)
- Buttons: Cancel, Save

**Editing Wired connection 1**


Connection name:

General | Ethernet | **802.1x Security** | DCB | IPv4 Settings | IPv6 Settings

☐ Use 802.1X security for this connection

Authentication:

Username:

Password:  

☐ Show password

**Editing Wired connection 1**

Connection name:

General | Ethernet | 802.1x Security | **DCB** | IPv4 Settings | IPv6 Settings

☐ Use Data Center Bridging (DCB) for this connection

Feature	Enable	Advertise	Willing	Priority	
FCoE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="text" value="default"/>	<input type="text" value="Fabric"/>
iSCSI	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="text" value="default"/>	
FIP	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="text" value="default"/>	
Flow Control	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="button" value="Options..."/>
Priority Groups	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="button" value="Options..."/>

Editing Wired connection 1

Connection name: Wired connection 1

General Ethernet 802.1x Security DCB IPv4 Settings IPv6 Settings

Method: Automatic (DHCP)

Addresses

Address	Netmask	Gateway

Add Delete

Additional DNS servers:

Additional search domains:

DHCP client ID:

☐ Require IPv4 addressing for this connection to complete

Routes...

Cancel Save

Editing Wired connection 1

Connection name: Wired connection 1

General Ethernet 802.1x Security DCB IPv4 Settings IPv6 Settings

Method: Automatic

Addresses

Address	Prefix	Gateway

Add Delete

Additional DNS servers:

Additional search domains:

IPv6 privacy extensions: Disabled

☐ Require IPv6 addressing for this connection to complete

Routes...

Cancel Save

## Control the Franka Panda robot via Franka Control Interface (FCI).

- 1) Disconnect the Franka Base from the Franka desktop. Use the same ethernet cable to connect the Franka Control with the Franka desktop.
- 2) Choose "Control". See below for specifications of this network.
- 3) Check that you can receive data by for example using the echo code.

## Run a code

My programs and Harry's programs can be found in the examples folder in Libfranka.

- 1) Open a terminal and build
  - a. `cd libfranka/build`
  - b. `cmake -DCMAKE_BUILD_TYPE=Release ..`
  - c. `cmake --build .` (you have to do this every time you change something in your code)
- 2) Open a new terminal and run your code
  - a. `cd libfranka/build/examples/Kelly` (the folder where you can find your program)
  - b. `./echo_robot_state_Kelly`
- 3) When you get the error "command not possible in the current mode!", you probably have to open the emergency-stop device (such that lights are blue).
  - a. Emergency-stop device pushed = lights of the robot are white
  - b. Emergency-stop device open = lights of the robot are blue

white	Interactive	safe interaction with Panda is possible
blue	Attention! Activated	Attention: Panda is enabled for movement and could start any moment
green	Automatic execution	Panda is carrying out an automatic program and is moving independently
yellow	Locked	Panda is locked mechanically or cannot be used
pink	Conflict	Panda is receiving conflicting enable signals
red	Error	an error has occurred

## Specifications of “Control”

**Editing Control**

Connection name: **Control**

General | Ethernet | 802.1x Security | DCB | IPv4 Settings | IPv6 Settings

- ☒ Automatically connect to this network when it is available
- ☒ All users may connect to this network
- ☐ Automatically connect to VPN when using this connection

Firewall zone: Default

Cancel Save

---

**Editing Control**

Connection name: Control

General | Ethernet | 802.1x Security | DCB | IPv4 Settings | IPv6 Settings

Device: [dropdown]

Cloned MAC address: [text box]

MTU: automatic [minus] [plus] bytes

Wake on LAN:

- ☒ Default
- ☐ Phy
- ☐ Unicast
- ☐ Multicast
- ☐ Ignore
- ☐ Broadcast
- ☐ Arp
- ☐ Magic

Wake on LAN password: [text box]

Cancel Save

**Editing Control**


Connection name:

General | Ethernet | **802.1x Security** | DCB | IPv4 Settings | IPv6 Settings

☐ Use 802.1X security for this connection

Authentication:

Username:

Password:  

☐ Show password

**Editing Control**

Connection name:

General | Ethernet | **802.1x Security** | **DCB** | IPv4 Settings | IPv6 Settings

☐ Use Data Center Bridging (DCB) for this connection

Feature	Enable	Advertise	Willing	Priority	
FCoE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="text" value="default"/>	<input type="text" value="Fabric"/>
iSCSI	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="text" value="default"/>	
FIP	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="text" value="default"/>	
Flow Control	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="button" value="Options..."/>
Priority Groups	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="button" value="Options..."/>



**Editing Control**

Connection name:

General Ethernet 802.1x Security DCB IPv4 Settings IPv6 Settings

Method:

**Addresses**

Address	Netmask	Gateway
192.168.2.1	24	

DNS servers:

Search domains:

DHCP client ID:

☐ Require IPv4 addressing for this connection to complete

**Editing Control**

Connection name:

General Ethernet 802.1x Security DCB IPv4 Settings IPv6 Settings

Method:

**Addresses**

Address	Prefix	Gateway

Additional DNS servers:

Additional search domains:

IPv6 privacy extensions:

☐ Require IPv6 addressing for this connection to complete

The address in IPv4 Settings is chosen as “192.168.2.1” because the robot’s address is “192.168.2.105”, see screenshot below.

Admin - Franka Emika

https://robot.franka.de/admin/network

FRANKA EMIKA

DASHBOARD

NETWORK

USERS

END-EFFECTOR

MODBUS TCP

SYSTEM

admin

Robot network

Defines the IP address and the address range of the DHCP server running at the robot's base network interface. The network is fixed to a 24bit network mask and the DHCP server assigns IP addresses with last octets in the range of 100 - 150.

For example, 192.168.1.1 would have the robot serve IP addresses 192.168.1.100 - 150 while all 192.168.1.x IP addresses can reach it.

192 . 168 . 0 . 1

Shop Floor network

DHCP Client ☐

Address

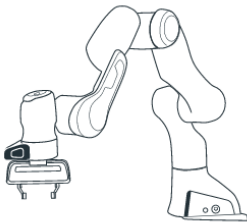
192.168.2.105


Netmask

255.255.255.0

Gateway

DNS





RESET DEFAULT

DISCARD

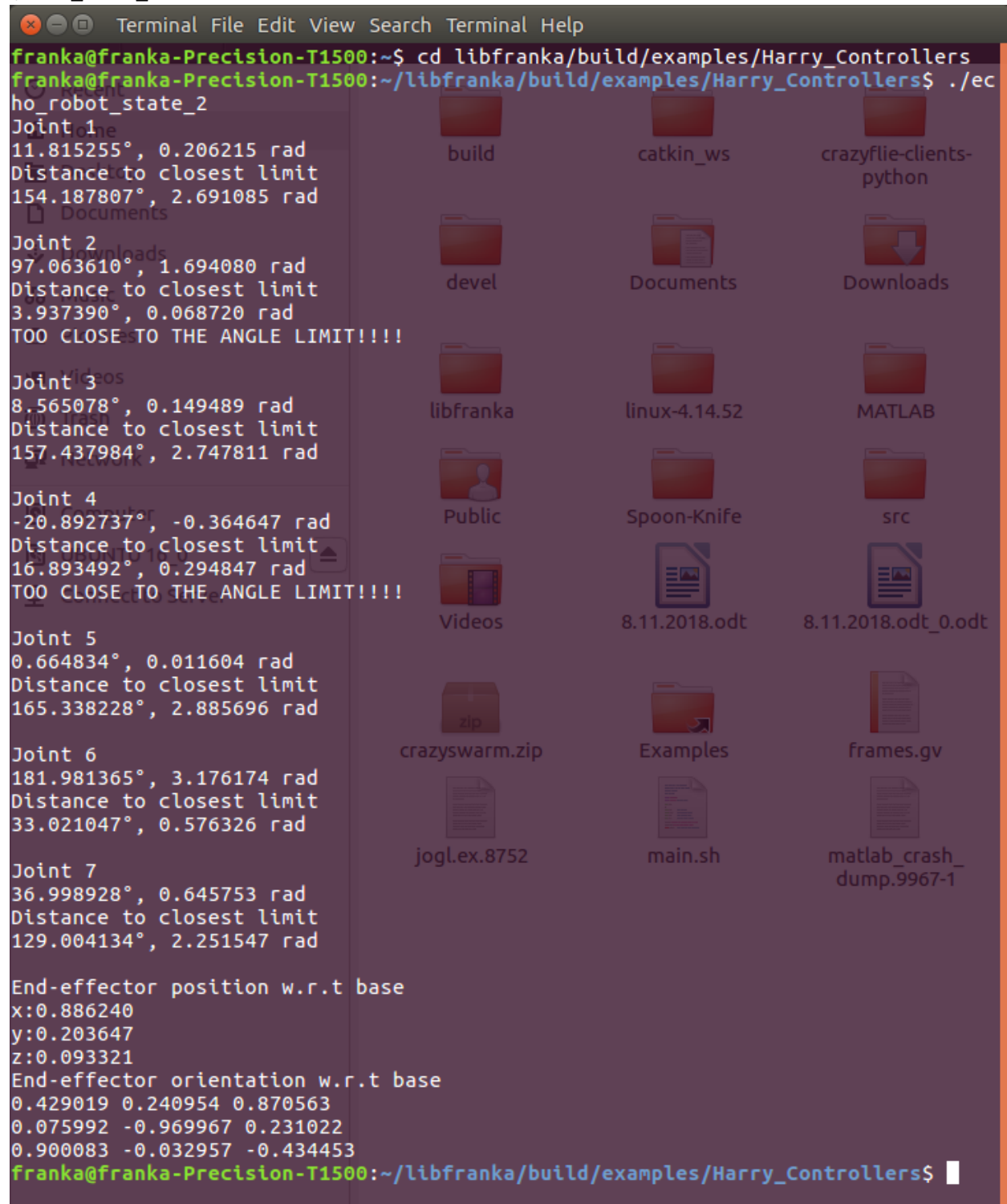
APPLY

Check that the connection works.

New echo robot state code can be found in folder “Kelly”

cd libfranka/build/examples/Kelly

./echo\_robot\_state



```
franka@franka-Precision-T1500:~$ cd libfranka/build/examples/Harry_Controllers
franka@franka-Precision-T1500:~/libfranka/build/examples/Harry_Controllers$ ./echo_robot_state_2
Joint 1
11.815255°, 0.206215 rad
Distance to closest limit
154.187807°, 2.691085 rad
Joint 2
97.063610°, 1.694080 rad
Distance to closest limit
3.937390°, 0.068720 rad
TOO CLOSE TO THE ANGLE LIMIT!!!!
Joint 3
8.565078°, 0.149489 rad
Distance to closest limit
157.437984°, 2.747811 rad
Joint 4
-20.892737°, -0.364647 rad
Distance to closest limit
16.893492°, 0.294847 rad
TOO CLOSE TO THE ANGLE LIMIT!!!!
Joint 5
0.664834°, 0.011604 rad
Distance to closest limit
165.338228°, 2.885696 rad
Joint 6
181.981365°, 3.176174 rad
Distance to closest limit
33.021047°, 0.576326 rad
Joint 7
36.998928°, 0.645753 rad
Distance to closest limit
129.004134°, 2.251547 rad


End-effector position w.r.t base
x:0.886240
y:0.203647
z:0.093321
End-effector orientation w.r.t base
0.429019 0.240954 0.870563
0.075992 -0.969967 0.231022
0.900083 -0.032957 -0.434453
franka@franka-Precision-T1500:~/libfranka/build/examples/Harry_Controllers$
```

## Get Vicon data on the Franka desktop

- 1) Keep the ethernet connection between the Franka desktop and the Franka control. Since the Franka desktop has only one ethernet port, the connection between the Franka desktop and the Vicon (Windows) desktop has to be done wireless.
- 2) Make and start an Ad-Hoc network on the Vicon desktop.

- a. Press “Windows + x”, select “Windows PowerShell (administrator)”, and set hostednetwork called “adhocname” and key “password”.


You don’t always have to change/set the hostednetwork name and password. In case you want to use the already created hostednetwork, go to step c/d.

 Administrator: Windows PowerShell

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\WINDOWS\system32> netsh wlan set hostednetwork mode=allow ssid=adhocname key=password
```

- b. You get a message that the hostednetwork you just created is set.

 Administrator: Windows PowerShell

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\WINDOWS\system32> netsh wlan set hostednetwork mode=allow ssid=adhocname key=password
The hosted network mode has been set to allow.
The SSID of the hosted network has been successfully changed.
The user key passphrase of the hosted network has been successfully changed.

PS C:\WINDOWS\system32>
```

- c. Show more information about this hostednetwork. Now you can see it is not yet started.

```
PS C:\WINDOWS\system32> netsh wlan show hostednetwork
```

```
Hosted network settings
-----
Mode                : Allowed
SSID name           : "adhocname"
Max number of clients : 100
Authentication       : WPA2-Personal
Cipher              : CCMP

Hosted network status
-----
Status              : Not started
```

- d. Start the hostednetwork.

```
PS C:\WINDOWS\system32> netsh wlan start hostednetwork
The hosted network started.

PS C:\WINDOWS\system32>
```

- e. Show more information about this hostednetwork. Now you can see it is started and that it has 0 clients.

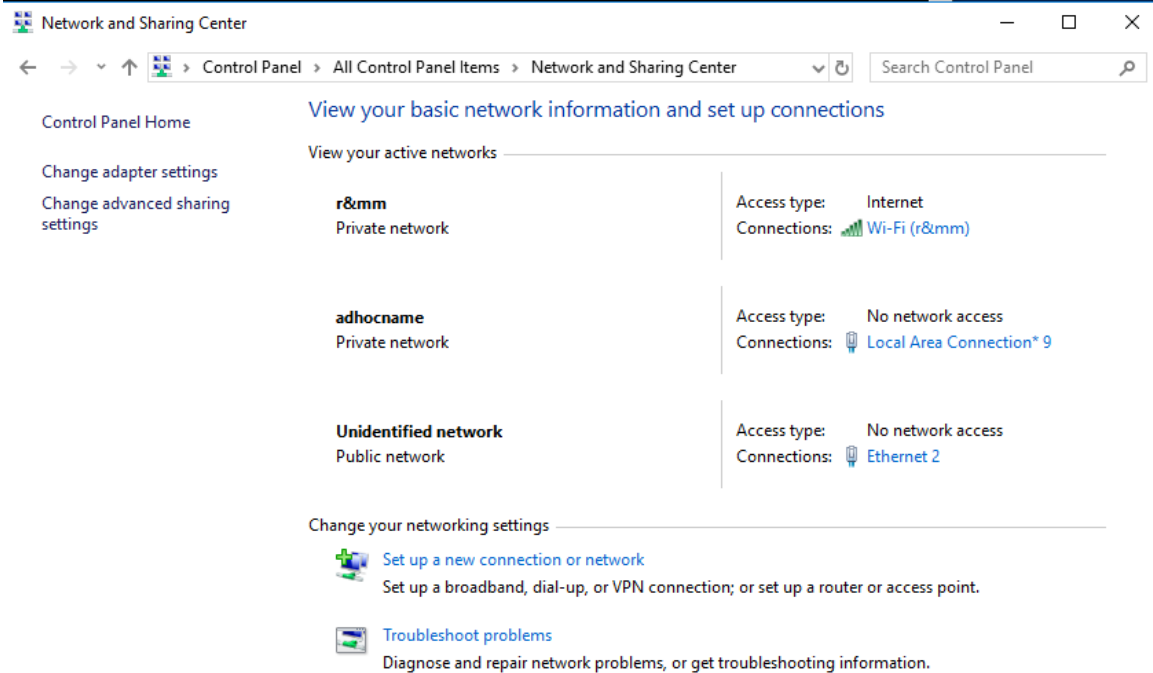
```
PS C:\WINDOWS\system32> netsh wlan show hostednetwork

Hosted network settings
-----
Mode                : Allowed
SSID name           : "adhocname"
Max number of clients : 100
Authentication       : WPA2-Personal
Cipher              : CCMP

Hosted network status
-----
Status              : Started
BSSID               : 00:1b:11:be:0b:a6
Radio type          : 802.11g
Channel             : 2
Number of clients    : 0

PS C:\WINDOWS\system32>
```

- 3) Check in “Network & Sharing Center” the created network. You can see it under the name “adhocname” and with connection type “Local Area Connection\*9”.



- 4) Check the IP address of the created network.
- Open a command prompt.
  - Type “ipconfig”.
  - Under “Wireless LAN adapter Local Area Connection\* 9” you can see that the IP address of the created network is “192.168.137.1”.

```

C:\Users\VICON>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Ethernet adapter Ethernet 3:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Ethernet adapter Ethernet 4:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Ethernet adapter Ethernet 5:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Ethernet adapter Ethernet 2:

    Connection-specific DNS Suffix  . :
    IPv4 Address. . . . . : 192.168.10.1
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . :

Wireless LAN adapter Wi-Fi:

    Connection-specific DNS Suffix  . :
    Link-local IPv6 Address . . . . . : fe80::4d71:3348:dc0f:4d7b%7
    IPv4 Address. . . . . : 192.168.1.176
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.1.1

Wireless LAN adapter Local Area Connection* 9:

    Connection-specific DNS Suffix  . :
    Link-local IPv6 Address . . . . . : fe80::90da:3919:7b88:91e%19
    IPv4 Address. . . . . : 192.168.137.1
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . :

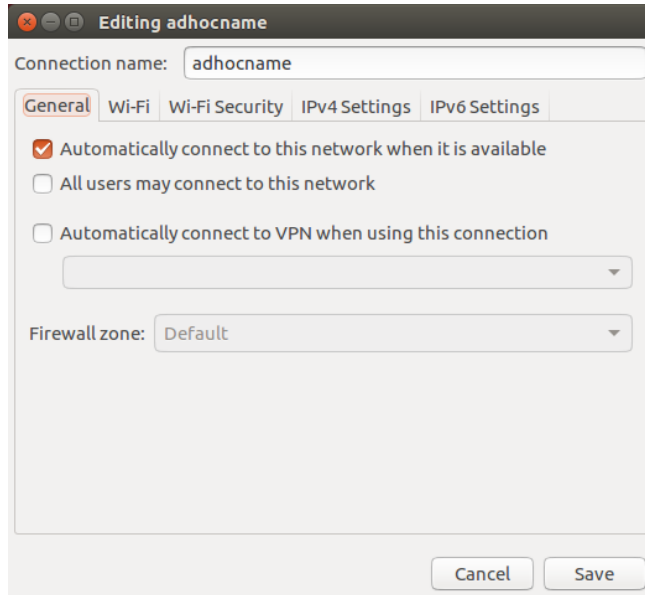
```

- 5) Use a Wi-Fi dongle to have Wi-Fi on the Franka desktop (Ubuntu) while having an ethernet connection.



6) Make a new wireless network on the Franka desktop with following properties. Go to step 7) if you already created this wireless network.

a. General



The screenshot shows the 'Editing adhocname' window with the 'General' tab selected. The 'Connection name' field is set to 'adhocname'. The 'General' tab is active, showing options for automatic connection and firewall zone. The 'Automatically connect to this network when it is available' checkbox is checked. The 'All users may connect to this network' checkbox is unchecked. The 'Automatically connect to VPN when using this connection' checkbox is unchecked. The 'Firewall zone' is set to 'Default'. The 'Cancel' and 'Save' buttons are at the bottom right.

Editing adhocname

Connection name: adhocname

General Wi-Fi Wi-Fi Security IPv4 Settings IPv6 Settings

☒ Automatically connect to this network when it is available

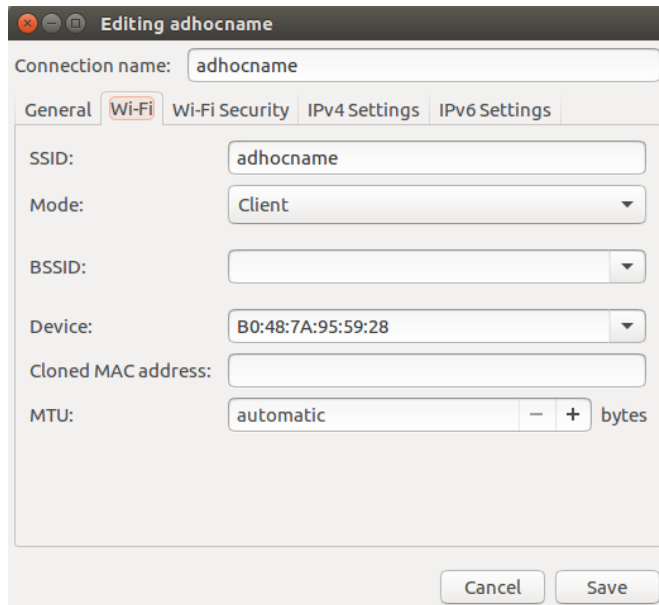
☐ All users may connect to this network

☐ Automatically connect to VPN when using this connection

Firewall zone: Default

Cancel Save

b. Wi-Fi



The screenshot shows the 'Editing adhocname' window with the 'Wi-Fi' tab selected. The 'Connection name' field is set to 'adhocname'. The 'Wi-Fi' tab is active, showing fields for SSID, Mode, BSSID, Device, Cloned MAC address, and MTU. The 'SSID' field is set to 'adhocname'. The 'Mode' is set to 'Client'. The 'BSSID' field is empty. The 'Device' is set to 'B0:48:7A:95:59:28'. The 'Cloned MAC address' field is empty. The 'MTU' is set to 'automatic'. The 'Cancel' and 'Save' buttons are at the bottom right.

Editing adhocname

Connection name: adhocname

General Wi-Fi Wi-Fi Security IPv4 Settings IPv6 Settings

SSID: adhocname

Mode: Client

BSSID:

Device: B0:48:7A:95:59:28

Cloned MAC address:

MTU: automatic - + bytes

Cancel Save

c. Wi-Fi Security

The screenshot shows the 'Editing adhocname' window with the 'Wi-Fi Security' tab selected. The 'Connection name' field contains 'adhocname'. The 'Security' dropdown is set to 'WPA & WPA2 Personal'. The 'Password' field contains 'password' and has a 'Show password' checkbox checked. The 'Cancel' and 'Save' buttons are at the bottom right.

Editing adhocname

Connection name: adhocname

General Wi-Fi Wi-Fi Security IPv4 Settings IPv6 Settings

Security: WPA & WPA2 Personal

Password: password

☒ Show password

Cancel Save

d. IPv4 Settings

The screenshot shows the 'Editing adhocname' window with the 'IPv4 Settings' tab selected. The 'Connection name' field contains 'adhocname'. The 'Method' dropdown is set to 'Automatic (DHCP)'. There is a table for 'Addresses' with columns 'Address', 'Netmask', and 'Gateway', and buttons 'Add' and 'Delete'. Below the table are fields for 'Additional DNS servers:', 'Additional search domains:', and 'DHCP client ID:'. There is a checkbox 'Require IPv4 addressing for this connection to complete' and a 'Routes...' button. The 'Cancel' and 'Save' buttons are at the bottom right.

Editing adhocname

Connection name: adhocname

General Wi-Fi Wi-Fi Security IPv4 Settings IPv6 Settings

Method: Automatic (DHCP)

Addresses

Address	Netmask	Gateway

Add

Delete

Additional DNS servers:

Additional search domains:

DHCP client ID:

☐ Require IPv4 addressing for this connection to complete

Routes...

Cancel Save



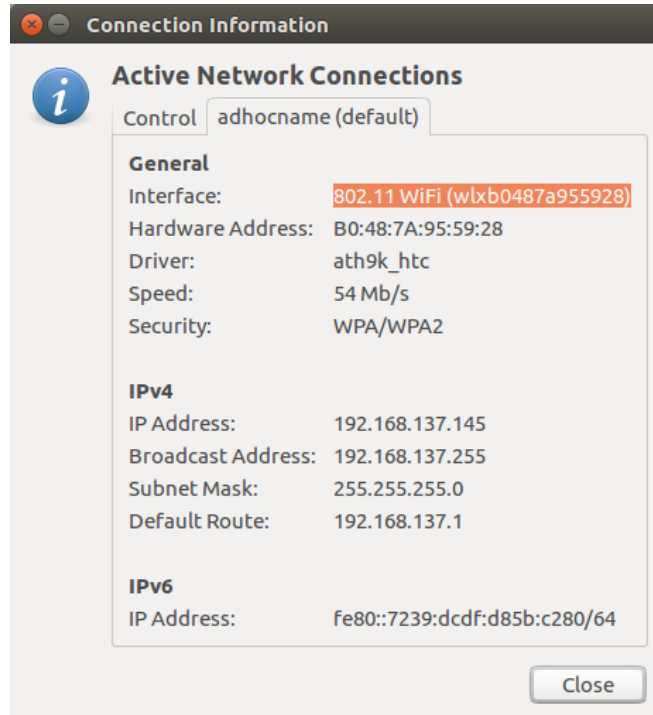
e. IPv6 Settings

The screenshot shows a window titled "Editing adhocname" with several tabs: General, Wi-Fi, Wi-Fi Security, IPv4 Settings, and IPv6 Settings. The "IPv6 Settings" tab is active. It contains a "Method" dropdown menu set to "Automatic". Below this is an "Addresses" section with a table for adding and deleting IP addresses, and buttons for "Add" and "Delete". There are also input fields for "Additional DNS servers" and "Additional search domains", a dropdown for "IPv6 privacy extensions" set to "Disabled", and a checkbox for "Require IPv6 addressing for this connection to complete". A "Routes..." button is at the bottom right of the settings area. At the very bottom of the window are "Cancel" and "Save" buttons.

- 7) The wireless network with the name "adhocname" will show up in the list of wireless networks. If not, exert and insert the Wi-Fi dongle again. Connect to this network.
- 8) Check the connection information on the Franka desktop.
  - a. Ethernet connection with the name "Control" to Franka Panda Control.

The screenshot shows a window titled "Connection Information" with a sub-header "Active Network Connections". There are two tabs: "Control" and "adhocname (default)". The "Control" tab is active. It displays network details for an Ethernet interface. The "General" section shows the interface as "Ethernet (enp2s0)", hardware address "F0:4D:A2:DA:8A:50", driver "tg3", speed "1000 Mb/s", and security "None". The "IPv4" section shows IP address "192.168.2.1", broadcast address "192.168.2.255", and subnet mask "255.255.255.0". The "IPv6" section shows IP address "fe80::ca61:6617:638c:4042/64". A "Close" button is at the bottom right.

- b. Wireless connection with the name “adhocname” to Vicon desktop.  
Here you can see the IP address of the Vicon desktop, i.e. Default Route: 192.168.137.1, and of the Franka desktop, i.e. IP Address: 192.168.137.145.



- 9) Check that the Franka desktop can send and receive packages to and from the Vicon desktop.  
Open a terminal (Ctrl+Alt+t), type “ping 192.168.137.1”.

```
franka@franka-Precision-T1500:~$ ping 192.168.137.1
PING 192.168.137.1 (192.168.137.1) 56(84) bytes of data.
64 bytes from 192.168.137.1: icmp_seq=1 ttl=128 time=1.30 ms
64 bytes from 192.168.137.1: icmp_seq=2 ttl=128 time=1.39 ms
64 bytes from 192.168.137.1: icmp_seq=3 ttl=128 time=0.969 ms
64 bytes from 192.168.137.1: icmp_seq=4 ttl=128 time=1.20 ms
64 bytes from 192.168.137.1: icmp_seq=5 ttl=128 time=1.87 ms
64 bytes from 192.168.137.1: icmp_seq=6 ttl=128 time=0.882 ms
64 bytes from 192.168.137.1: icmp_seq=7 ttl=128 time=1.38 ms
64 bytes from 192.168.137.1: icmp_seq=8 ttl=128 time=1.36 ms
64 bytes from 192.168.137.1: icmp_seq=9 ttl=128 time=3.10 ms
64 bytes from 192.168.137.1: icmp_seq=10 ttl=128 time=1.38 ms
64 bytes from 192.168.137.1: icmp_seq=11 ttl=128 time=1.55 ms
64 bytes from 192.168.137.1: icmp_seq=12 ttl=128 time=1.55 ms
64 bytes from 192.168.137.1: icmp_seq=13 ttl=128 time=4.93 ms
64 bytes from 192.168.137.1: icmp_seq=14 ttl=128 time=1.00 ms
64 bytes from 192.168.137.1: icmp_seq=15 ttl=128 time=1.76 ms
64 bytes from 192.168.137.1: icmp_seq=16 ttl=128 time=5.27 ms
64 bytes from 192.168.137.1: icmp_seq=17 ttl=128 time=1.52 ms
^C
--- 192.168.137.1 ping statistics ---
17 packets transmitted, 17 received, 0% packet loss, time 16085ms
rtt min/avg/max/mdev = 0.882/1.910/5.275/1.262 ms
```

- 10) Check that the Vicon desktop can send and receive packages to and from the Franka desktop. Open a command prompt, type “ping 192.168.137.145”.

```
C:\Users\VICON>ping 192.168.137.145

Pinging 192.168.137.145 with 32 bytes of data:
Reply from 192.168.137.145: bytes=32 time=2ms TTL=64
Reply from 192.168.137.145: bytes=32 time=2ms TTL=64
Reply from 192.168.137.145: bytes=32 time=1ms TTL=64
Reply from 192.168.137.145: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.137.145:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 2ms, Average = 1ms
```

- 11) Show more information about the hostednetwork in the Windows PowerShell (Administrator). Now you can see the created Ad-Hoc network has 1 client.

```
PS C:\WINDOWS\system32> netsh wlan show hostednetwork

Hosted network settings
-----
Mode                : Allowed
SSID name           : "adhocname"
Max number of clients : 100
Authentication      : WPA2-Personal
Cipher              : CCMP

Hosted network status
-----
Status              : Started
BSSID               : 00:1b:11:be:0b:a6
Radio type          : 802.11b
Channel             : 2
Number of clients   : 1
                    b0:48:7a:95:59:28    Authenticated
```

- 12) Launch the vicon\_bridge (bridge between Vicon and ROS) on the Franka desktop.
- Go to the “vicon.launch” file in the folder: catkin\_ws/src/vicon\_bridge/launch
  - Change the IP address at the 5<sup>th</sup> line to the IP address of the Windows pc. In this example 192.168.137.1. Save this file.
  - Open a new terminal (Ctrl+Alt+t), type “cd catkin\_ws/devel”, “source setup.bash”, “cd ..”, and “roslaunch vicon\_bridge vicon.launch”.
  - If it works, you will get the following window. Objects that are set visible in the Vicon software will be created and will be advertised. In case the object is occluded, a warning will show up.

```

Franka@franka-Precision-T1500:~/catkin_ws$ cd devel/
Franka@franka-Precision-T1500:~/catkin_ws/devel$ source setup.bash
Franka@franka-Precision-T1500:~/catkin_ws/devel$ cd ..
Franka@franka-Precision-T1500:~/catkin_ws$ roslaunch vicon_bridge vicon.launch
... logging to /home/franka/.ros/log/0c066c04-136b-11e9-8cbe-f04da2da8a50/roslau
ch-franka-Precision-T1500-4400.log
Checking log directory for disk usage. This may take awhile.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://franka-Precision-T1500:45709/

SUMMARY
=====

PARAMETERS
* /rostdistro: kinetic
* /rosversion: 1.12.14
* /vicon/datastream_hostport: 192.168.137.1:801
* /vicon/stream_mode: ClientPull
* /vicon/tf_ref_frame_id: /world

NODES
/
  vicon (vicon_bridge/vicon_bridge)

auto-starting new master
process[master]: started with pid [4410]
ROS_MASTER_URI=http://localhost:11311

setting /run_id to 0c066c04-136b-11e9-8cbe-f04da2da8a50
process[rosout-1]: started with pid [4423]
started core service [/rosout]
process[vicon-2]: started with pid [4426]
[ INFO] [1546968611.259579825]: Connecting to Vicon DataStream SDK at 192.168.13
7.1:801 ...
[ INFO] [1546968611.261424521]: .
[ INFO] [1546968612.261728647]: ... connected!
[ INFO] [1546968612.261844835]: Setting Stream Mode to ClientPull: Success
[ INFO] [1546968612.261882539]: Axis Mapping: X-Forward Y-Left Z-Up
[ INFO] [1546968612.261920818]: Version: 1.3.0
[ INFO] [1546968612.261950744]: setting up grab_vicon_pose service server ...
[ INFO] [1546968612.263205411]: setting up segment calibration service server ..

[ WARN] [1546968612.265299264]: grab frame returned false
[ WARN] [1546968612.268935340]: Wand occluded, not publishing...
[ INFO] [1546968612.280461397]: creating new object Wand/seg ...
[ WARN] [1546968612.282750878]: unable to load zero pose for Wand/seg
[ INFO] [1546968612.282799675]: ... done, advertised as " vicon/Wand/seg"
[ WARN] [1546968624.440503362]: Wand occluded, not publishing...
[ WARN] [1546968627.689481120]: Wand occluded, not publishing...
[ WARN] [1546968629.837934180]: Wand occluded, not publishing...

```

13) Check if you can get data from an object created and visible in Vicon on the Franka desktop.

Open a new terminal (Ctrl+Alt+t), type "rostopic echo /vicon/markers". The name "markers" is in this case Wand/seg. In general it's the object name / the segment name.

For my objects I had to write "rostopic echo /vicon/pickobject/pickobject"

```
franka@franka-Precision-T1500:~/catkin_ws$ rostopic echo /vicon/Wand/seg
header:
  seq: 17032
  stamp:
    secs: 1546969082
    nsecs: 633474370
  frame_id: "/world"
child_frame_id: "vicon/Wand/seg"
transform:
  translation:
    x: -2.17020675687
    y: 0.546741545302
    z: -0.829786395472
  rotation:
    x: -0.290364028967
    y: -0.676780054507
    z: -0.253996787546
    w: 0.627011260202
---
```

```
auto-starting new master
process[master]: started with pid
ROS_MASTER_URI=http://localhost:11
setting /run_id to a88b215a-136b-1
process[rosout-1]: started with pi
started core service [/rosout]
process[vicon-2]: started with pid
[ INFO] [1546968873.849286195]: Co
7.1:801 ...
[ INFO] [1546968873.852109336]: .
[ INFO] [1546968874.852521519]: ..
[ INFO] [1546968874.852586223]: Se
[ INFO] [1546968874.852649724]: Ax
[ INFO] [1546968874.852700585]: Ve
[ INFO] [1546968874.852736700]: se
[ INFO] [1546968874.854193349]: se
```

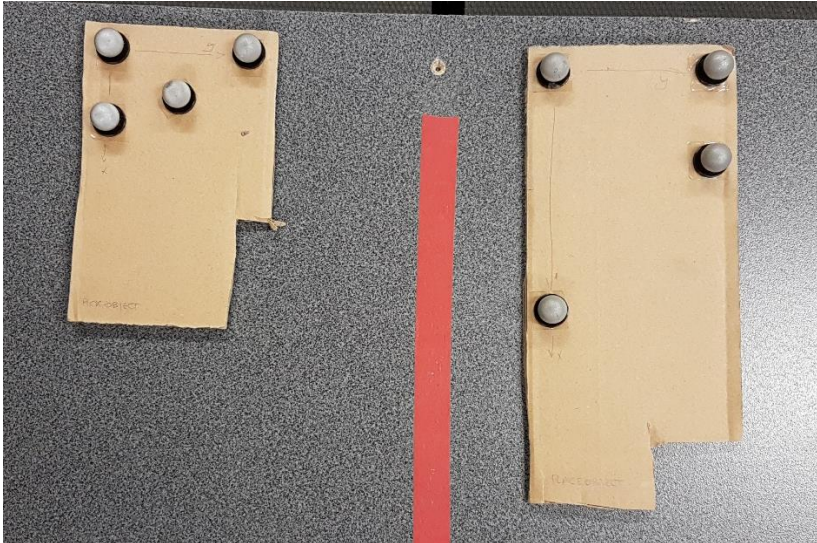
## Make objects in Vicon Nexus

Look to Louis' report and to Harry's second report.

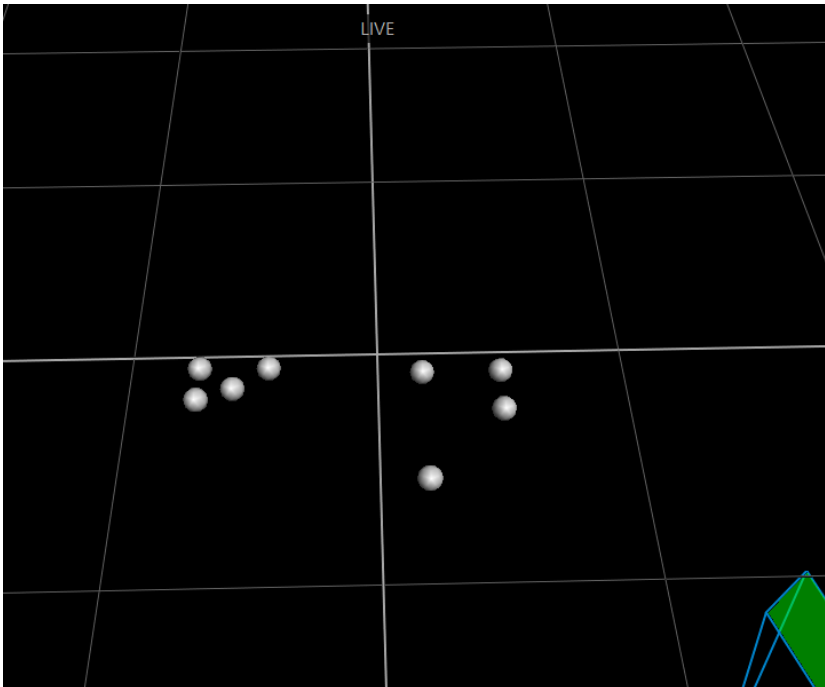
## Make objects in Vicon Tracker

1. Put markers on your object.

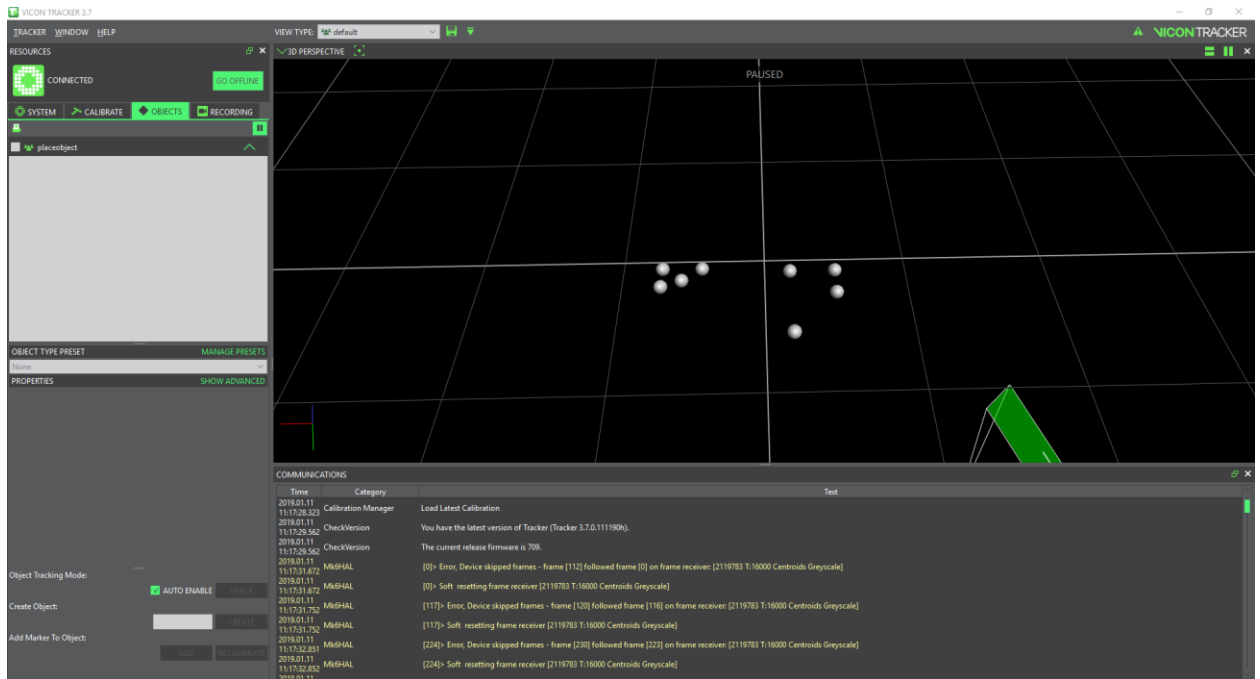
Here you can see the "pickobject" and the "placeobject", 4 reflective markers are placed on both.



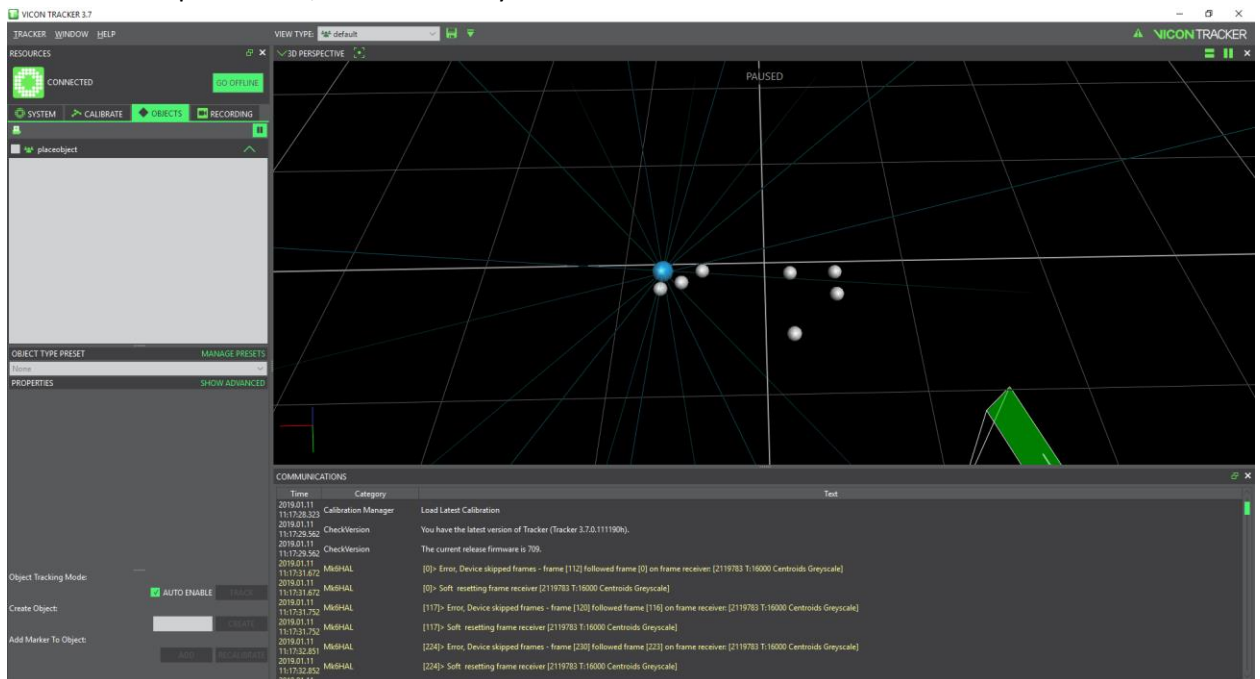
2. You can see them in Vicon Tracker as separated markers when no object is selected.



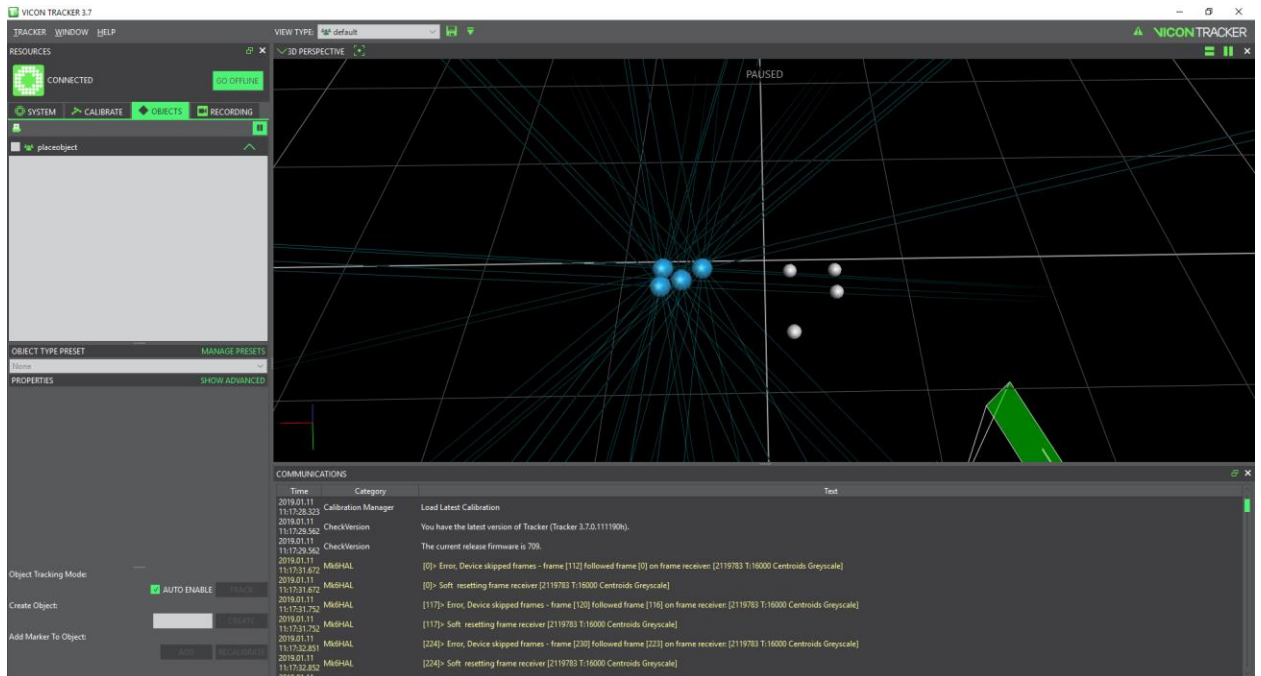
### 3. Pause the Vicon Tracker screen.



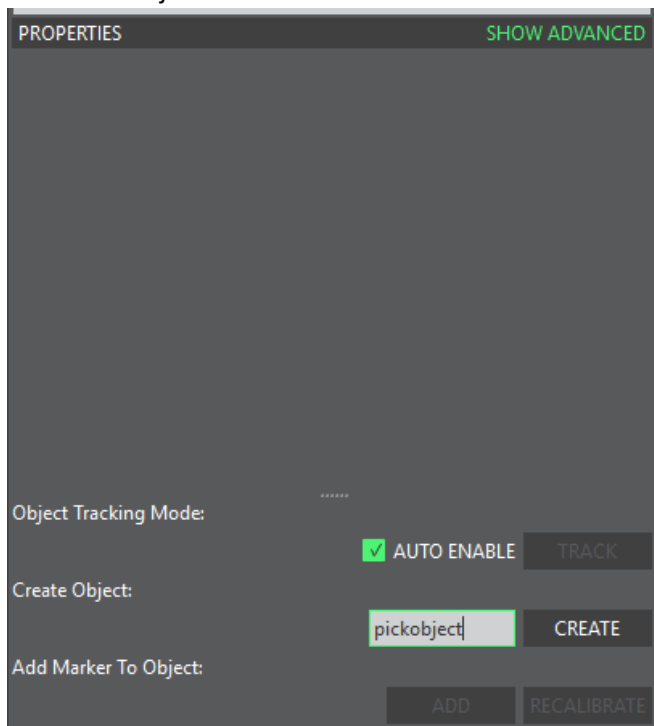
### 4. Select marker per marker, use the Ctrl key.





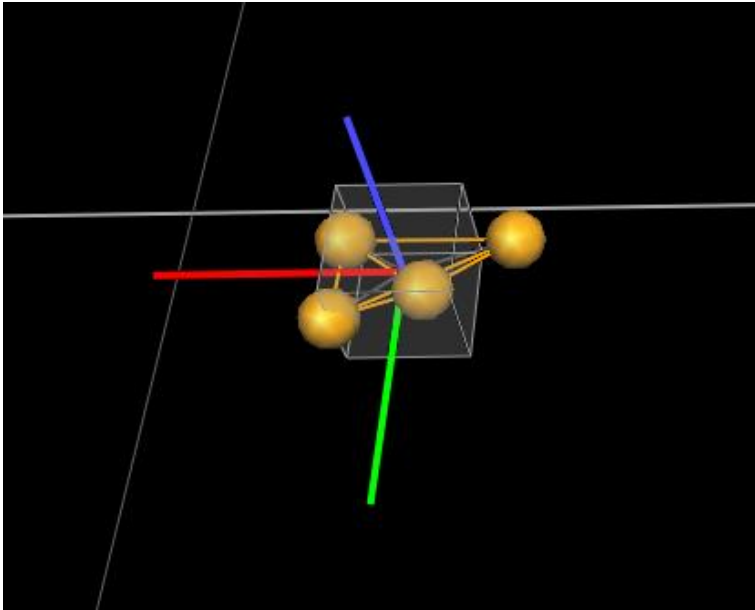


5. Create an object.

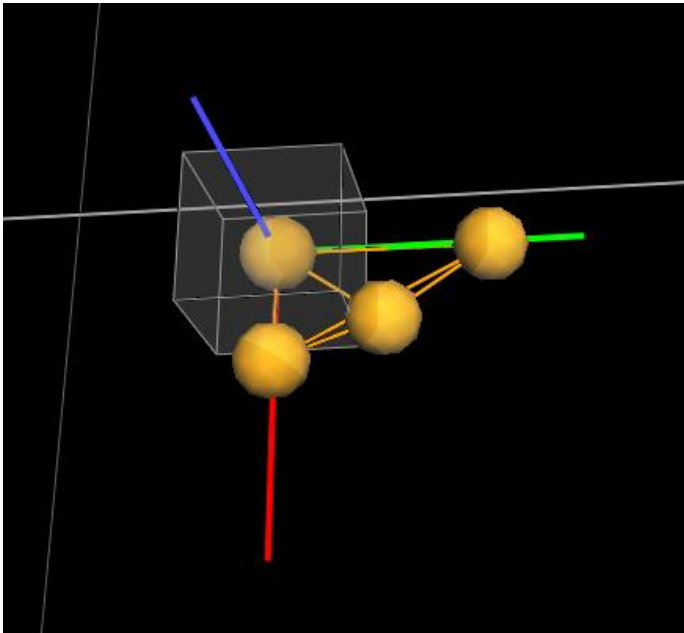




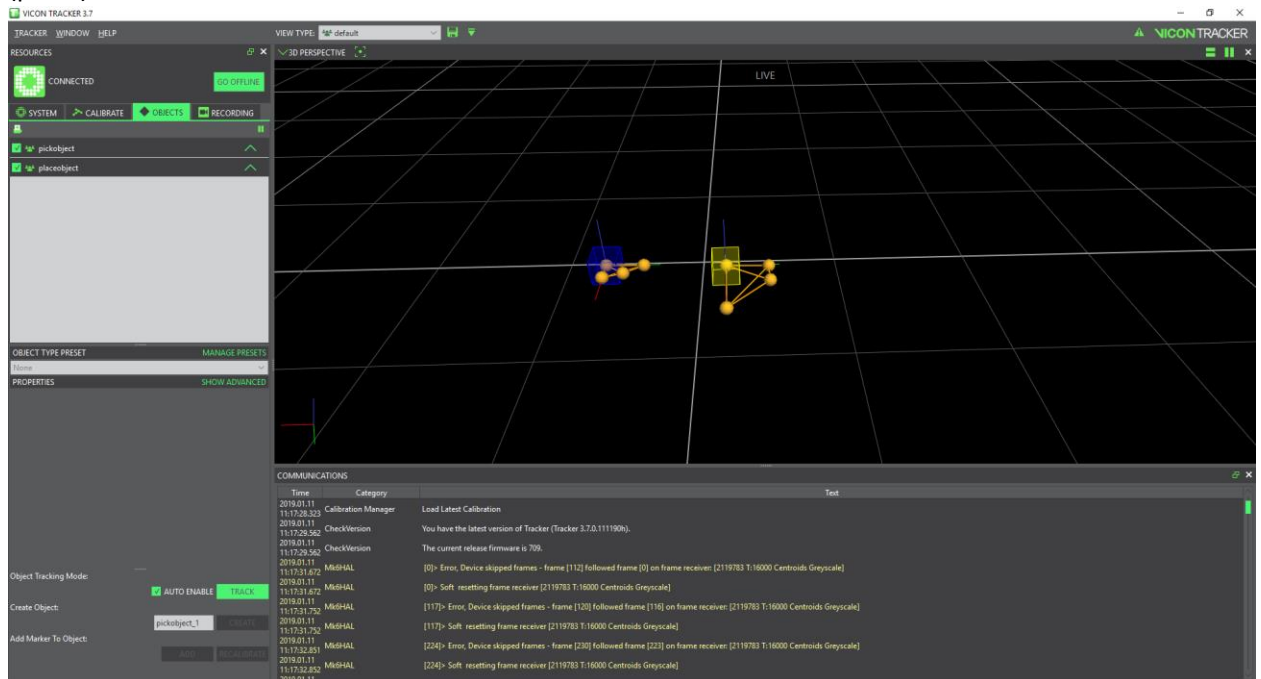
6. You will see its x-axis (red), y-axis (green), and z-axis (blue). You can change the position and orientation of this frame. This is the “pickobject”.



7. Do the same for a second object. This is the “placeobject”.

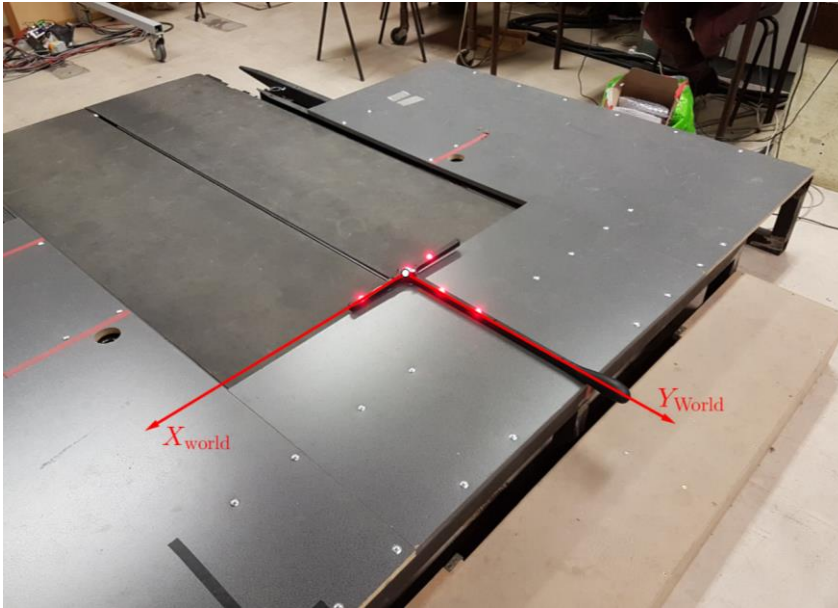


8. The two objects with their center (positioned and oriented myself) in blue (pick) and yellow (place).



## Start of “Vicon\_” codes

1. In Vicon we have set the world frame (origin + xyz axes) with the Vicon wand.



2. We put three markers on the robot to know the translation of the robot with respect to the world frame in x-, y-, and z-direction . In the code, we call this the robot\_wrt\_world frame.



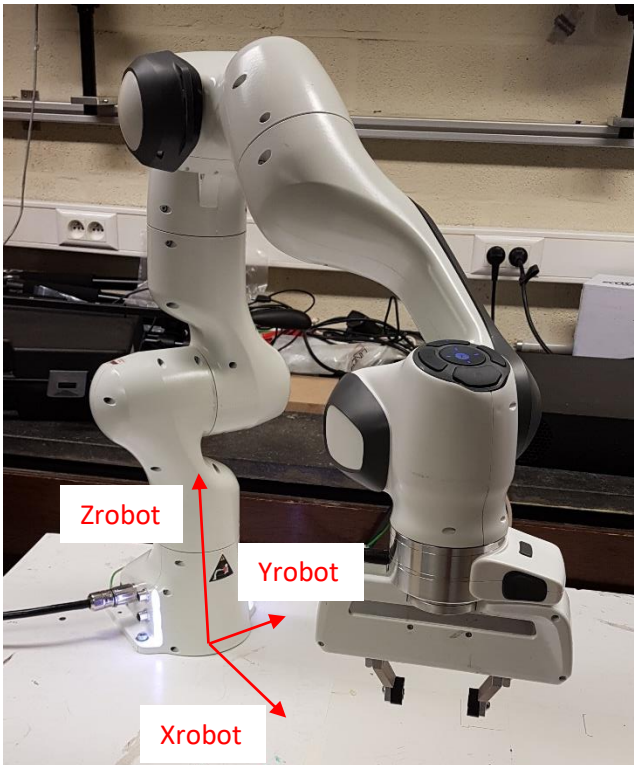
```
robot_wrt_world(0) = msg->markers[8].translation.x; // [mm]  
robot_wrt_world(1) = msg->markers[8].translation.y; // [mm]  
robot_wrt_world(2) = msg->markers[8].translation.z; // [mm]
```

3. We also put three markers in the corner of the table where the robot is mounted on to know the translation of the table with respect to the world frame in x-, y-, and z-direction . In the code, we call this the table\_wrt\_world frame.

```
table_wrt_world(0) = msg->markers[11].translation.x; // [mm]
table_wrt_world(1) = msg->markers[11].translation.y; // [mm]
table_wrt_world(2) = msg->markers[11].translation.z; // [mm]
```

4. We want to define the “real” robot origin with respect to the world frame set by the Vicon wand. The real robot frame, as set by Franka, is a frame of which the origin is positioned in the base of the robot and oriented as shown on the figure below.

For the real x- and real y-position, we use the x- and y-coordinates of the robot\_wrt\_world frame. For the real z-position, we use the z-coordinate of the table\_wrt\_world frame.



These positions have to be filled in manually.

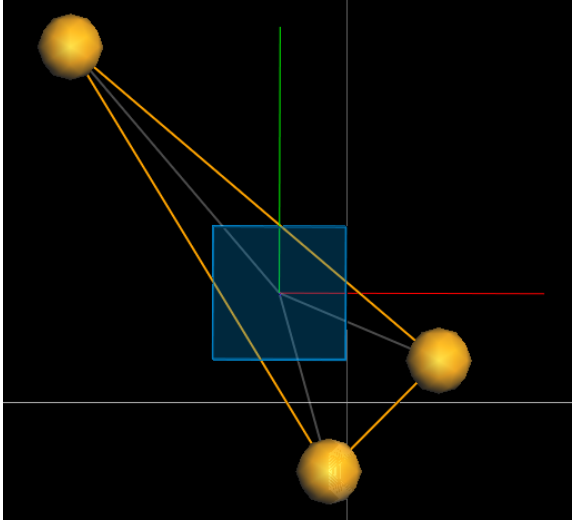
```
robot_origin(0) = -81.8/1000; // = robot_wrt_world(0) [m]
robot_origin(1) = 2011.3/1000; // = robot_wrt_world(1) [m]
robot_origin(2) = 462.7/1000; // = table_wrt_world(2) [m]
```

Every time you place the robot and table back in the Vicon room, you have to repeat this process, because the position can vary. If you don't move the table, then you don't have to do this.

```
robot_origin(0) = -465.0/1000; // = robot_wrt_world(0) [m]
robot_origin(1) = 2015.7/1000; // = robot_wrt_world(1) [m]
robot_origin(2) = 447.6/1000; // = table_wrt_world(2) [m]
```

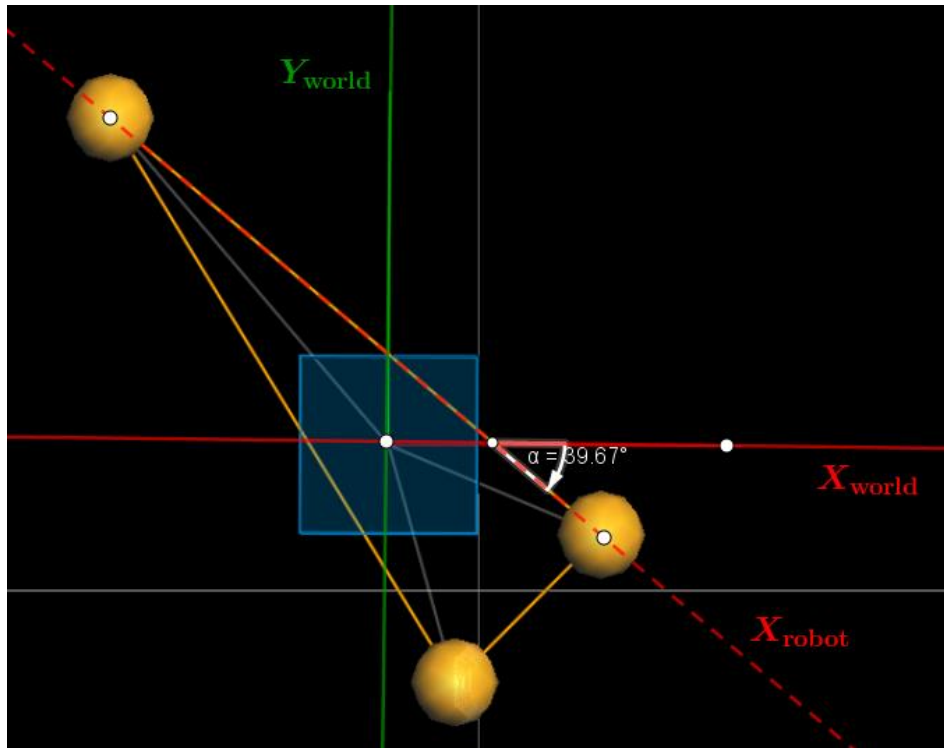
5. For the orientation of the robot with respect to the orientation of the world frame set by Vicon, we look again to the three markers we placed on the robot.

The long segment is in the x-direction of the robot and the short segment in the y-direction of the robot. The red line denotes the x-axis and the green line denotes the y-axis of the world frame set by the Vicon wand.



From the positive  $X_{world}$  axis to the positive  $X_{robot}$  axis, we need to do an rotation with an angle  $\alpha=40^\circ$  about the negative  $Z$  axis, so with an angle of  $-40^\circ$  about the positive  $z$ -axis.

$$\begin{bmatrix} x_R \\ y_R \\ z_R \end{bmatrix} = \begin{bmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{world} \\ y_{world} \\ z_{world} \end{bmatrix} = \begin{bmatrix} \cos(-40^\circ) & \sin(-40^\circ) & 0 \\ -\sin(-40^\circ) & \cos(-40^\circ) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{world} \\ y_{world} \\ z_{world} \end{bmatrix}$$



6. The “pickobject” and “placeobject” made in Vicon Tracker software with respect to the world frame (=set by the Vicon wand) can be received by the following commands.

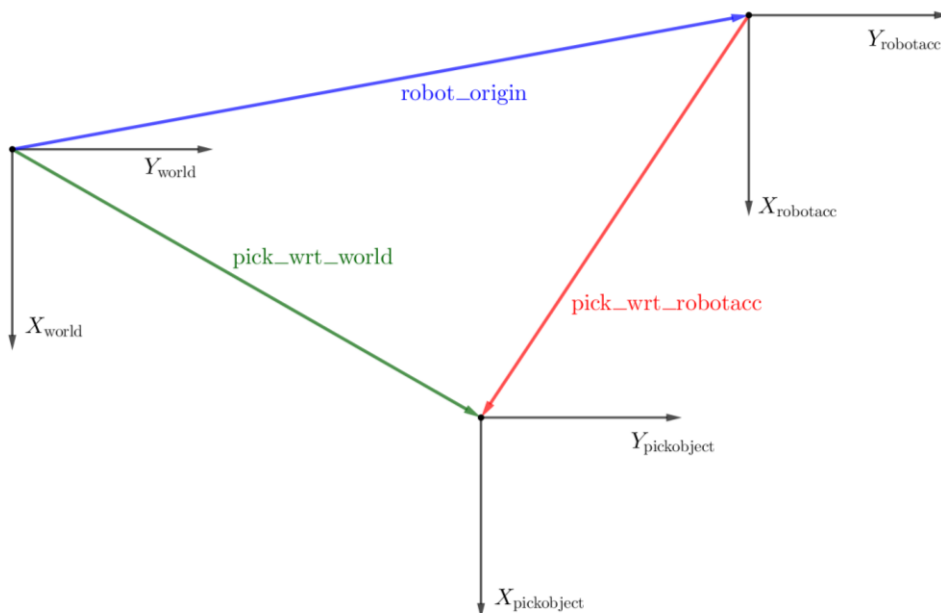
The marker number you use depends on the sequence you added the markers and the objects in Vicon Tracker.

- When creating a new object in Vicon Tracker, you have to select all the markers placed on the object. The sequence you use for selecting the markers “fixes” the number that marker gets.
- Selecting objects in Vicon Tracker also “fixes” the number that the markers get.
- In the pick and place example, the “origin” of each object was selected first and each object had four markers. When starting the experiment, I first selected my “pickobject” and afterwards my “placeobject”. That’s why I have to call markers[0] for the origin of my “pickobject” and markers[4] for the origin of my “placeobject”.

```
// coordinates of pick and place frame w.r.t. Vicon frame (Vicon frame = world frame)
// (check marker number)
pick_wrt_world(0) = msg->markers[0].translation.x; // [mm]
pick_wrt_world(1) = msg->markers[0].translation.y; // [mm]
pick_wrt_world(2) = msg->markers[0].translation.z; // [mm]
place_wrt_world(0) = msg->markers[4].translation.x; // [mm]
place_wrt_world(1) = msg->markers[4].translation.y; // [mm]
place_wrt_world(2) = msg->markers[4].translation.z; // [mm]
```

7. We don’t want the positions of our objects with respect to the world frame, but with respect to the robot frame. Therefore, we do the following transformations.

```
// coordinates of pick and place frame w.r.t. robot frame
pick_wrt_robotacc(0)=pick_wrt_world(0)/1000-robot_origin(0); // [m]
pick_wrt_robotacc(1)=pick_wrt_world(1)/1000-robot_origin(1); // [m]
pick_wrt_robotacc(2)=pick_wrt_world(2)/1000-robot_origin(2); // [m]
place_wrt_robotacc(0)=place_wrt_world(0)/1000-robot_origin(0); // [m]
place_wrt_robotacc(1)=place_wrt_world(1)/1000-robot_origin(1); // [m]
place_wrt_robotacc(2)=place_wrt_world(2)/1000-robot_origin(2); // [m]
```





And take into account the rotation about the z-axis with an angle  $\alpha = -40^\circ$ , see explanation in point 5, with “robotacc = world” (when only looking to orientation) and “robot = robot” to have the orientation of our object with respect to the real robot frame fixed by Franka.

[https://frankaemika.github.io/docs/control\\_parameters.html#denavithartenberg-parameters](https://frankaemika.github.io/docs/control_parameters.html#denavithartenberg-parameters)

```
// take into account rotation: world frame -> rotation about z with -42° -> robot frame
pick_wrt_robot(0)=cos(alpha)*pick_wrt_robotacc(0)+sin(alpha)*pick_wrt_robotacc(1); // [m]
pick_wrt_robot(1)=-sin(alpha)*pick_wrt_robotacc(0)+cos(alpha)*pick_wrt_robotacc(1); // [m]
pick_wrt_robot(2)=pick_wrt_robotacc(2)+z_offset; // [m]
place_wrt_robot(0)=cos(alpha)*place_wrt_robotacc(0)+sin(alpha)*place_wrt_robotacc(1); // [m]
place_wrt_robot(1)=-sin(alpha)*place_wrt_robotacc(0)+cos(alpha)*place_wrt_robotacc(1); // [m]
place_wrt_robot(2)=place_wrt_robotacc(2)+z_offset; // [m]
```

