Universidad del Valle de Guatemala
Facultad de Ingeniería
Departamento de Ingeniería Electrónica,
Mecatrónica y Biomédica

Instalación y configuración Crazyswarm2 + MOCAP4ROS

Brandon Garrido, Miguel Zea

Este documento tiene como principal objetivo describir el proceso para la instalación de las librerías que permitan el levantamiento de una infraestructura para el entorno de pruebas multidrones con crazyflies 2.1 y el sistema de captura de movimiento Optitrack.

1 Consideraciones

Previo a la instalación de las librerías correspondientes, asegúrese de contar con una partición de sistema operativo basado en Linux de Ubuntu LTS 22.04. Además, de tener instalada la versión de ROS2 Humble, siguiendo los pasos del enlace proporcionado e instalando la versión de escritorio.

Asimismo, es importante tener instalado dentro del mismo entorno de trabajo la versión más reciente y estable de Python 3.10.

Ubuntu	Python	ROS 2
22.04	3.10	Humble, Iron

Figure 1: Requerimientos pre-instalación librerías.

2 Mocap4ROS2

2.1 Instalación

Mocap4ROS2 una librería especializada en Motion Capture (Captura de Movimiento) diseñada para trabajar en el entorno de ROS 2 (Robot Operating System 2). Esta librería se utiliza para integrar datos de sistemas de captura de movimiento en aplicaciones y proyectos que funcionan con ROS 2, basandose en el protocolo de red NatNet.

Para instalar la biblioteca es necesario dentro del entorno de trabajo de ROS, crear una carpeta llamada 'mocap4ros2_ws' y a través de la linea de comandos de Ubuntu, con los respectivos permisos de superusuario ejecutar uno por uno los siguientes comandos:

```
$ mkdir src && cd src
```

```
$ git clone https://github.com/MOCAP4ROS2Project/
    mocap4ros2_optitrack.git

$ git clone https://github.com/MOCAP4ROS2-Project/mocap.git

$ git clone https://github.com/MOCAP4ROS2-Project/mocap_msgs.git

$ vcs import < mocap/dependency_repos.repos
cd .. && rosdep install --from-paths src --ignore-src -r -y

$ colcon build --symlink-install
```

Los comandos anteriores permitirán clonar el repositorio de la librería de Mocap4Ros2 que contiene todos los paquetes y drivers correspondientes para la comunicación con OptiTrack, así como compilar y construir los paquetes dentro del espacio de trabajo ROS 2 para su ejecución en el sistema linux.

2.2 Configuración

Para el funcionamiento correcto de la comunicación del servidor basado en Linux con el streaming de datos proporcionadas por Motive en una máquina de Windows, es necesario copiar y reemplazar las carpetas de archivos proporcionados en el github que corresponden al siguiente enlace, que se encuentran dentro de la carpeta mocap4ros_ws/src, lo que permitirá acceder a una versión funcional de la librería.

Además, tomar especial atención que para establecer las IP de servidor y cliente del streaming de datos del sistema de captura, así como los puertos correspondientes de comando y data para la comunicación multicast con el servidor, modificar los parámetros correspondientes mostrados en la figura 2, dentro del archivo *mocap_optitrack_driver/config/mocap_optritrack_driver_params.yaml*.

```
mocap_optitrack_driver_node:
  ros__parameters:
    connection_type: "Multicast" # Unicast / Multicast
    server_address: "192.168.50.200"
    local_address: "192.168.50.170"
    multicast_address: "239.255.42.99"
    server_command_port: 1510
    server_data_port: 1511
```

Figure 2: Configuracion parametros del controlador

Finalmente correr las siguiente dos lineas de comando para configurar correctamente el entorno de ROS 2 y permitir el uso de los paquetes y ejecutables instalados en el espacio de trabajo.

```
$ . install/local_setup.bash
$ source install/setup.bash
```

Lo siguiente será ejecutar en una consola el nodo que permitirá acceder a la comunicación mediante NatNet con el Mocap. Asímismo se requerirá en otra consola ejecutar la herramienta RQT que permite visualizar y controlar datos en entornos de ROS para publicar los tópicos y servicios necesarios tales como las lecturas de los single marker y los cuerpos rígidos dentro del sistema de captura OptiTrack.

```
#ejecutar el nodo
$1 ros2 launch mocap_optitrack_driver optitrack2.launch.py

#activar los servicios y t picos
$2 ros2 run rqt_gui rqt_gui --force-discover


#lecturas de los markers
$3 ros2 topic echo /markers
$4 ros2 topic echo /rigid_bodies
```

3 Crazyswarm 2

3.1 Instalación

Para la instalación de Crazyswarm en su versión dos, es importante seguir los pasos detallados en la documentación de Crazyswarm desde el segundo inciso (instalación de dependencias) hasta el cuarto (construcción del espacio de trabajo en ROS2). Una vez completados estos pasos, es necesario ejecutar los comandos para actualizar el entorno, ubicados al final de la documentación.

Es importante que para poder ejecutar los nodos de Crazyswarm 2, correctamente es necesario ubicarse en la carpeta ../src dentro del espacio de trabajo de ros2_ws y ejecutar los comandos para la configuración del entorno de ROS2 en cada consola.

```
. install/local_setup.bash
source install/setup.bash
```

3.2 Configuración

Además de seguir los pasos de instalación, es crucial, al igual que con la librería Mocap4ROS2, copiar y reemplazar las carpetas de archivos proporcionados en el github que corresponden al siguiente enlace, que se encuentran dentro de la carpeta ros2_ws/src.

Además, para garantizar una comunicación efectiva tanto con los drones Crazyflie 2.1 como con el sistema de captura de movimiento (Mocap) se deberá modificar y configurar de acuerdo a las necesidades, los siguientes archivos:

- crazyflies.yaml: Este archivo configura los parámetros específicos de cada Crazyflie dentro del enjambre, como el URI, el controlador, la posición inicial, entre otros. Se destaca su importancia para el correcto funcionamiento del sistema, ya que permite la detección y comunicación con los Crazyflies, así como su integración con el sistema de captura de movimiento.

```yaml
robots:
  cf233:
    enabled: true
    uri: radio://0/80/2M/E7E7E7E7E3
    type: cf21  # see robot_types


robot_types:
  cf21:
    motion_capture:
      enabled: true
      # only if enabled; see motion_capture.yaml
      marker: default_single_marker
      dynamics: default
    big_quad: false
    battery:
      voltage_warning: 3.0  # V
      voltage_critical: 2.7 # V

# global settings for all robots
all:
  # firmware logging for all drones (use robot_types/type_name to set per type, or
  # robots/drone_name to set per drone)
  firmware_logging:
    enabled: true
    default_topics:
      # remove to disable default topic
      pose:
        frequency: 10 # Hz
  firmware_params:
    commander:
      enHighLevel: 1
    stabilizer:
      estimator: 2 # 1: complementary, 2: kalman
      controller: 2 # 1: PID, 2: mellinger
```

Figure 3: Configuracion de los crazyflies

- motion_capture.yaml: Este archivo configura los paquetes esenciales para la integración del control con el sistema de captura de movimiento.

```
/motion_capture_tracking:
  ros__parameters:
    type: "optitrack"
    hostname: "192.168.50.200"

    mode: "libobjecttracker" # one of motionCapture,libRigidBodyTracker
```

Figure 4: Configuracion del sistema de captura

Una vez completada la configuración de estos archivos, es necesario ejecutar los siguientes comandos en una consola aparte:

```
$1 - ros2 launch crazyflie launch.py
$2 - ros2 run crazyflie_examples hello_world
```

Esto permitirá establecer la comunicación con los Crazyflies 2.1 a través de la CrazyRadio PA que deberá estar conectada en el servidor de Ubuntu, así como ejecutar el código en Python que contendrá las rutinas de movimiento para los drones.

4 Referencias

- Documentacion ROS2 Humble :
  https://docs.ros.org/en/humble/Installation/Ubuntu-Install-Debians.html
- Documentacion Crazyswarm :
  https://imrclab.github.io/crazyswarm2/installation.html
- Documentacion MOCAP4ROS2 :
  https://mocap4ros2-project.github.io/