Mehdi Raza Khorasani
S6164555

Homework 1

COGAR
May 20, 2024

1 Sending goals to mars

The satellite component is a publisher which publishes satellite data to a topic. This is being subscribed by an image processing component which pre processes the data. Then these processed images are published on a topic which is subscribed by the Exploration component. This component is responsible to generate possible goal positions which are of scientific value. These goals are published on a topic which is subscribed by a Human in the loop (HITL) component. The purpose of this component is to filter and inspect the generated goals, which will then be sent to the mars base station.
Once the processing is complete, we expect to have a list of goal poses in as the output of HITL component. We call the network service to send this data (along with the name of the topic, say **earth/goalPoses** its coming from). This is done so that the message broker on mars can decide where to send this data.

2 Receiving goals on mars base station

Once the communication component receives the goals, it publishes it on a general topic with generic type. The general topic is subscribed by the **message broker** component.

2.1 Message broker

This is the central component of our system, responsible to redirect the incoming data to their respective streams/topic/services. Please note that the **general topic** has an abstract data type (similar to an http request [header, payload, metadata, etc])

Once data is received by the **message broker**, it is responsible to parse the data (by topic-information in the header) and request the **goal-storage service** to save it.

3 Pulling next set of waypoints to the robot

The message-broker on the robot is subscribed to the current-mode of the robot (the mode is maintained by a parameter-server pub/sub on the robot). Once we receive 'idle' mode, we would create a request-next-goal message in the message broker and call the communication components **sender**.

3.1 On mars base

Once the communication component on mars-base receives the goal-request message, the message-broker will be responsible to set the mode in the param-server to 'planning'. The param-server on the mars base implements a pub/sub (similar to that of the robot), which is subscribed by the global path planner component. Once the global-path-planner receives the mode='planning', it will make a service-call to the goals-database to send in the next goal for the robot (please note that, the goals-database component maintains internally the id of the last sent goal, so it can figure out which goal to send next) and will start planning. Once the plan is complete, the generated set of way-points will be sent through a service call to the message broker so that it sends it to the robot.

3.2 On the robot

On the robot, once we receive the waypoints-message in the message broker component, a sub-routine inside it will be responsible to call the **mode selector** component to set the mode to path planning and also publish the waypoints to the local path planner in a pub sub mechanism (we can use an action server here to get feedback on the progress of the planner). the local planner will generate set-points for the controller and send it through pub/sub (action server) architecture. The local planner will need sensor information and dynamic obstacle info (from robot sensors which can be cameras, sonar etc), so the relevant pub/subs are added to receive this.

3.2.1 Positive case: goal reached

if the local planner reaches the last waypoint successfully, it will send a goal-success message (with a snapshot of current sensor readings of the environment) which is subscribed by the message broker. The message broker after receiving this message forwards it to mars base station.

mars base:
on the base station, once this message is received by the message broker, it calls the mode selector service which is responsible to decide if we should go to a nearby place or just start collecting scientific data. The service responds with an augmented goal which is sent back to the robot and the local path planning pipeline is triggered again.

All in all, once the mode-selector component decides if the robot should start scientific data collection, it responds with a proceed-to-data-collection message which is sent to the robot through the message broker.

On the robot:
On the robot, when the proceed-to-data-collection is received, the message broker sets the mode to **data-collection**.

3.3 Negative case: goal unreachable or danger encountered

While planning, the danger detection component is responsible to continuously subscribe to the sensor data and check for danger (this can be done using the terrain analysis or xyz algorithm, let us not focus on the specifics for now). In case a danger is detected, the danger detection component publishes to the **safety** topic. This topic is subscribed by the mode selector component, which on receiving this danger message, sets the mode to idle. Once mode idle is set, the parameter server publishes the new status on the relevant topic, and message broker being subscribed to the param-server, receives this message and cancels the local-planners current goal. Hence the robot comes to a stop state. The message broker also sends this update to the base station, in return expecting a message with a new set of waypoints. All in all, if a danger situation is encountered, we cancel the current goal and expect the base station to send the next goal. The procedure to fetch the next goal and trigger the planner has been explained previously.

4 Data collection pipeline: Robot

Once the mode on the robot's param-server is set to **data-collection**, the scientific-equipment-controller receives this change of mode, and starts deployment of the scientific equipment. If we require that each location only deploys only certain set of sensors and also take a certain set of samples, we can send this information along with the waypoints and store it in the parameter-server. But to keep things simple, we ignore this detail, which can be left to the discretion of the developer. The scientific information is collected and stored in the storage-service through a req/res procedure. Once enough samples are collected (this can be determined based on some custom criterion relating to the goal position), we call the mode-selector service to set the mode to 'idle' so that the next goal pose can be sent (this is already explained above).

5 Data retrieval from robot to base station

Since all scientific data collected is stored in the database, once the robot's mode is set to 'idle', and the robot is requesting for a new goal, the base-station's message broker is responsible to first check if any new data is available in the robot database. if so, we fetch it using the retrieve-api of the storage component. Secondly, we clear the robot's storage after successfull retrieval and persistance in the base-station's storage component. Then the global path planning pipeline is triggered (already explained above) which generates the next set of waypoints and sends it to the robot.

6 Data retrieval from mars base station to earth

Data to earth is only sent when earth requests for it (pull mechanism). We expose an interface from the base-station comm component which lets earth pull out the latest data from the base-station. The latest data is retrieved from the database and sent to earth. Please note that we dont delete the data from mars base station unless:

1. It is specifically requested by earth
2. The storage space on the base-station is full, we then remove old data to make space for new data (this can be a complex behavious to implement, so for simplicity, we can just clear the database once it is full, or delete data partially. It all depends on the requirements of the system).