Notas Projecto Integrador

# Arquitetura De Software e Integraçao

Esta àrea de trabalho foi liderada por Martinho Figueiredo.

# Objectivos

Esta area tevo como objectivo: - Criar um ambiente de desemvolvimento grafico  
- Criar um sistemas para rercolhar e distribuicao de dados de sensores

# Metodologia

## Introducao

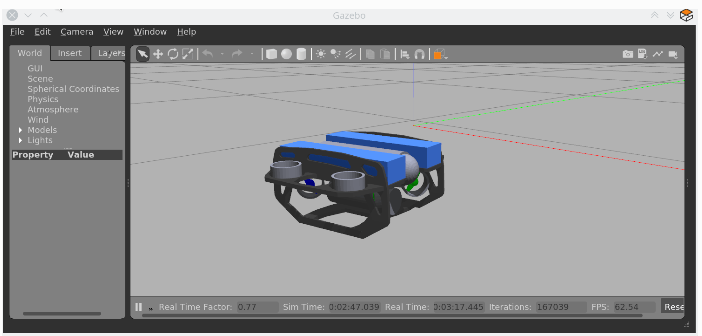
Para atigir os objectivos propostos foram avaliados os requisitos: - Funcionais: - Ambiente Grafico para Visualizar resultados - Capacidade de Gravar Ensaios - Flexibilidade de Distribuicao do ambiente final - Nao Funcionais: - Tempo de atraso - Acrescimo de processamento

### Recolha e Distribuicao de dados de Sensores

De modo interagir com o ROV, foi necessario estudar os protocolos de comunicacao disponibilizados por este, e conceber um sistema que distribui estes dados em simultaneo aos variados scripts criados, para os monitirizar e modificar. A simultaneidade destes processos necessita de uma camada agregadora que orquestra o fluxo da informacao e, com esse intuito, escolhemos uma arquitectura baseada no Robot Operating System (ROS).Esta ferramenta consiste num modelo publisher-subscriber, em que varios processos estao associados a um objecto, que e a repressentacao digital do estado do ROV. Os processos do tipo publisher servem para actualizar o modelo com informacao em tempo real do robot. Quando estes acabam de o fazer, os processos subscriber sao notificados, para que possam processar esta nova informacao e criar os seus resultados, sejam estes comandos de movimento ou valores numericos de estimativas de posicao. Este mecanismo garante alguma coesao temporal, pois sincroniza todos os processos com a chegada de nova informacao. Com este formato em mente, fizemos uso de uma blioteca do ambiente ROS, MAVROS, que funciona como publisher no nosso sistema.

### Ambiente Grafico

O ambiente grafico permite traduzir o resultado das medidas efectuadas por todos os sensores numa representacao digital do ambiente em que o ROV se encontra. Este objectivo foi solucionado com a ferramenta Gazebo que e um simulador 3d para robotica que nos permitiu incluir um modelo 3d do submarino e da piscima criando assim uma representaçao do nosso ambiente de teste. Isto é facilitado pelo facto do Gazebo funcionar como um subscriber do Objecto Disponibilizado pelo ROS



img1

## Arquitetura

Como todas estas ferramentas tem que ser executadas em simultaneo, com parametros bem definidos, para que funcionem de maneira interligada, recorremos ao Docker para criar *containers* que sirvam como blocos fundamentais para montar todo o nosso sistema de controlo. O docker permite nos tambem facilmente transportar o software de PC para PC, facilitando assim a instalação de todo o software necessario

graph LR  
 subgraph BlueROV2  
 CC["Companion Computer"]--> ARDP[ArduPilot]  
 ARDP --> M["Actuatores"]  
 end  
   
 subgraph TopSide Computer  
 MR[MAVROS] <-..-> CC  
 end  
   
 subgraph Engine  
 MR <-..-> RE[ROS Host]  
 RE --> DB[ROS Database]  
 RE --> GZB[Gazebo Simulator]   
 SCRPT[Scripts] <-..-> RE  
 end

Para isso criamos 2 containers: - O container *topside*, com MAVROS para puder comunicar com o Ros host e passar os dados que recebe atraves da porta ethernet fisica - Um container com o software necessario para: - executar *ros engine* - *front end* de gazebo

Separar assim os serviços necessarios permite nos ter acesso a mais configuraçoes de hardware no futuro.

graph BT   
 subgraph CONFIG1  
 subgraph PC  
 TS-->RE  
 end  
 ROV --> TS  
 end  
 subgraph CONFIG2  
 subgraph PC1  
 RE1  
 end  
 subgraph PC2  
 TS1  
 end  
 TS1-->RE1  
 ROV1 --> TS1  
 end

## Aprendizagem e Dificuldades

Uma das dificuldades desta area de trabalho foi aprender todos estas novas ferramentas e garantir que funcionam umas com as outras, como por exemplo garantir que todas a network de containers e serviços estao apontados na direção certa para que possam comunicar. Numa outra perspectiva, os resultados desta area de trabalho são dificeis de quantificar, pois trata-se de trabalho de preparação e estruturação do codigo, com alguns melhoramentos na facilidade de desenvolvimento para os outros processos, sendo apenas os resultados qualitativos.

|  | Refs |
| --- | --- |
| 1 | [Docker And ROS](https://roboticseabass.com/2021/04/21/docker-and-ros/) |
| 2 | [ROS](https://ros.org) |
| 3 | [ROS and Pixhawk](https://docs.px4.io/v1.12/en/robotics/) |
| 4 | [3d MAPs from sensor data](https://docs.px4.io/v1.12/en/simulation/gazebo_octomap.html) |