



*Monitoraggio in tempo reale di dispositivi robotici tramite i sensori di uno smartphone e l'utilizzo del framework ROS*

**Relatore:** Prof. Enrico Masala

**Candidato:** Davide Brau

### **Introduzione**

Nell'ambito della robotica, l'attività di monitoraggio è una componente fondamentale per il funzionamento dell'intero sistema. Con questo termine infatti, si intende identificare quel processo che ha lo scopo di acquisire le informazioni provenienti dalle varie parti di un sistema robotico, utili per poter esercitare il controllo su di esso. Oltre all'acquisizione occorre focalizzarsi anche sulle modalità di interfacciamento e trasmissione, poiché quasi sempre ci si ritrova a lavorare all'interno di un sistema distribuito, dove sia il monitoraggio che il controllo vengono spesso svolti da remoto. L'obiettivo di questa attività di tesi è stato quello di esplorare le possibilità e le soluzioni, utili per svolgere il monitoraggio in questo contesto, utilizzando e analizzando le opportunità fornite dal framework: ROS.

### **ROS: Robot Operating System**

Questo framework, spesso definito come un meta-sistema operativo di tipo open-source, è costituito da un insieme di processi, in esecuzione simultanea su una o più macchine, capaci di comunicare tra loro per raggiungere un obiettivo comune. ROS è in grado di fornire tutti quei servizi che ci si aspetta da un qualsiasi sistema operativo, definendo inoltre, gli strumenti e le librerie indispensabili per: creare, compilare ed eseguire il codice. L'architettura di un sistema ROS può essere rappresentata attraverso un grafo orientato, dove i processi comunicano tra loro grazie a un'infrastruttura di comunicazione pensata per garantire maggiore robustezza attraverso un elevato livello di modularità e di disaccoppiamento, favorendo in questo modo la scomposizione di attività complesse in singoli elementi, più semplici da trattare rispetto alle equivalenti soluzioni monolitiche; contribuendo nel complesso, alla scalabilità del sistema.

ROS implementa diversi tipi di trasmissione, includendo comunicazioni sincrone attraverso quelli che vengono identificati come "ROS Service" e comunicazioni asincrone tramite i "topic" (secondo il paradigma: *Publish-Subscribe*). I processi autonomi in esecuzione nel sistema ROS vengono identificati come nodi, questi sono in grado di comunicare tra loro grazie alla presenza di un master, scambiandosi dei dati serializzati che prendono il nome di "messaggi". Infine, i vari nodi vengono organizzati in package, che a loro volta, sulla base di dipendenze reciproche e funzionalità comuni, possono essere raggruppati in "stack" per facilitarne la distribuzione.

### **Multimedia in ambito ROS**

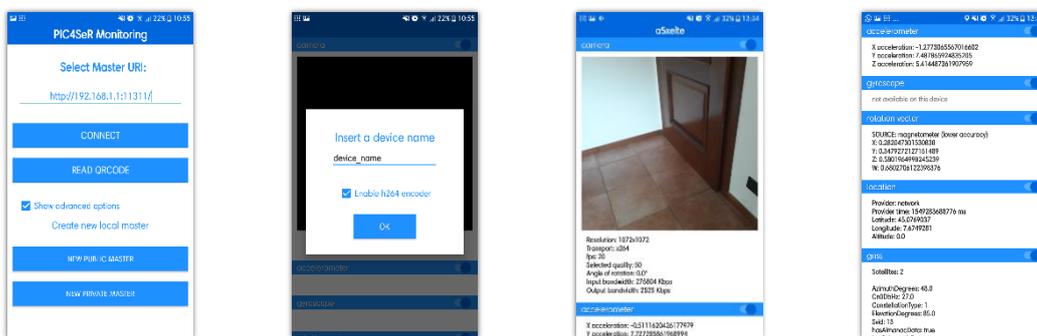
Le immagini all'interno di un'architettura ROS vengono gestite all'interno dei singoli nodi che le elaborano in maniera autonoma, queste subiranno una serie di trasformazioni lungo il percorso dalla sorgente di acquisizione fino alla loro visualizzazione o memorizzazione, secondo un'architettura pipeline. A differenza degli altri tipi di dati, le immagini possono essere rappresentate attraverso numerose codifiche e formati, che rendono necessaria l'adozione di meccanismi specifici, in grado di garantire una gestione adeguata. È possibile individuare quattro fasi quando si parla di gestione delle immagini:

- *acquisizione*
- *trasmissione*
- *elaborazione*
- *visualizzazione*

A differenza degli altri nodi, associati a dati e formati specifici, quelli relativi alle immagini necessitano di un approccio differente per quanto riguarda i meccanismi di trasmissione. In questo caso, spesso si tende a voler specializzare le strategie di trasporto, ad esempio attraverso l'utilizzo di algoritmi di compressione. Tutto ciò porta con sé una serie di complicazioni, in quanto l'introduzione di un nuovo formato compresso comporterebbe la necessità di dover ridefinire tutti i nodi preesistenti, associati alle immagini, che intendono utilizzarlo. Per far fronte a questo problema è stato introdotto un package chiamato: "image\_transport", che definendo un nuovo livello intermedio, consente di disaccoppiare la parte di elaborazione dell'immagine da quella di trasmissione. Per definire un nuovo tipo di trasporto, si ricorre all'utilizzo di particolari package che hanno la funzione di plugin. La distribuzione ufficiale di ROS definisce i tipi di trasporto "compressed" (JPEG) e "theora", ma sono disponibili anche altre soluzioni in grado di utilizzare librerie specifiche per la codifica H.264, migliorando ulteriormente le prestazioni.

### Sviluppo di un'applicazione Android basata su ROS

In seguito alla precedente fase di approfondimento, si è cercato di mettere in pratica le nozioni acquisite, sviluppando un'applicazione Android basata su ROS. La maggior parte delle applicazioni Android di questo tipo, consentono di utilizzare lo smartphone per il controllo a distanza, tuttavia, poche prevedono l'utilizzo opposto, ovvero sfruttare quest'ultimo come sorgente di informazioni. Lo scopo di questa applicazione consiste nel permettere l'accesso a un insieme di dati utili per il monitoraggio, ricavati grazie all'utilizzo della grande varietà di sensori presenti normalmente all'interno degli smartphone odierni, rendendoli poi disponibili in modo relativamente semplice al resto del sistema, per mezzo delle funzionalità offerte da ROS.



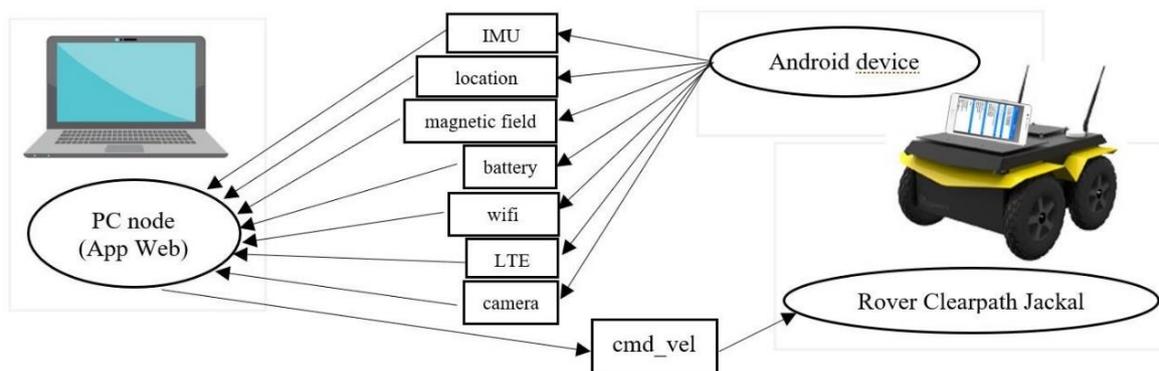
È possibile ottenere informazioni sulla cinematica utilizzando: l'accelerometro, il giroscopio, il magnetometro e il servizio di localizzazione GNSS. Tuttavia, il monitoraggio può anche coinvolgere l'ambiente circostante in cui il robot si muove, per tale motivo vengono acquisiti i dati riguardanti la temperatura, la pressione e l'illuminamento; qualora questi fossero disponibili sullo smartphone. Inoltre, vengono anche fornite un insieme di informazioni relative alla rete cellulare, allo stato del Wi-Fi e gli aggiornamenti riguardanti la batteria. L'applicazione consente la visualizzazione da remoto delle immagini acquisite dalle fotocamere, attraverso le due modalità di trasporto: "compressed" e "x264", basate rispettivamente sugli standard di compressione JPEG e H.264. In seguito, sono stati creati alcuni ROS-Service che permettono di regolare da remoto i parametri riguardanti queste due modalità. Durante lo sviluppo dell'applicazione sono stati definiti nuovi tipi di messaggi ROS per il trasporto di informazioni associate alle diverse classi Android. Inoltre, con l'introduzione di nuovi messaggi specifici per le classi appartenenti al sistema GNSS, è stato possibile sfruttare questi ultimi per la condisione e l'elaborazione di un insieme di informazioni dettagliate ottenute dai satelliti, note con il nome di "raw measurements" (disponibili negli smartphone più recenti). Questi ultimi, se utilizzati in modo

appropriato, potrebbero costituire una nuova frontiera nello sviluppo di applicazioni per il posizionamento ad alta precisione, dalle quali molti sistemi robotici potrebbero trarre grande vantaggio.

### Integrazione con interfaccia web basata su ROS e prove con rover

In seguito alla fase di sviluppo della applicazione Android sono state eseguite delle prove per verificare in funzionamento di quest'ultima nell'utilizzo congiunto con un'interfaccia web basata su javascript, anch'essa in grado di comunicare con l'ambiente ROS. Successivamente, una volta configurate entrambe le applicazioni, sono stati eseguiti dei test con l'ausilio del rover: *Clearpath Jackal*.

Lo scopo di queste prove, oltre alla verifica del normale funzionamento, è stato quello di osservare le diverse criticità che si possono riscontrare durante l'utilizzo in un contesto reale. È stato importante poter esaminare l'impatto che si ha sulla trasmissione dei dati pubblicati dallo smartphone nel momento in cui aumenta la distanza dall'access-point Wi-Fi o la reattività dei sensori quando il rover si muove a velocità sostenuta. I casi di maggiore interesse riguardano la trasmissione delle immagini. In questo caso gli elementi da tenere in considerazione sono: la qualità, la latenza e il framerate; questi sono legati alla velocità del link, ai protocolli di trasporto e alle tempistiche di compressione degli algoritmi utilizzati nelle due modalità "compressed" e "x264". Infatti, entrambe presentano vantaggi e svantaggi che vanno tenuti in considerazione nel momento in cui viene svolta un'attività di controllo, specialmente se di tipo real-time.



La modalità compressed ha il vantaggio di garantire una visualizzazione più stabile in quanto la codifica a livello di singole immagini, consente a queste di essere trasmesse senza che subiscano un deterioramento. Tuttavia, i limiti di questa modalità sono dovuti innanzitutto al framerate, che anche in condizioni ottimali risulta essere troppo basso. Infatti, sebbene i valori ottenuti possano essere tollerabili nel momento in cui il rover è fermo, diventano inaccettabili quando quest'ultimo si muove a una certa velocità. Ciò è dovuto principalmente alle tempistiche di compressione, che a loro volta dipendono: dai parametri utilizzati nell'algoritmo JPEG, dalla risoluzione delle immagini e dalla potenza di calcolo a disposizione. La situazione peggiora ulteriormente con l'aumento della distanza dal ricevitore, in questo caso la diminuzione della banda a disposizione determina, oltre a un ulteriore abbassamento del framerate, anche un aumento della latenza.

Per quanto riguarda la modalità x264, il bitrate medio prodotto in uscita dall'encoder è nettamente inferiore rispetto al caso precedente. Inoltre, il framerate e la latenza, grazie all'utilizzo dell'interfaccia MediaCodec (che lavorando a basso livello consente di evitare i rallentamenti dovuti alla compressione), risultano essere adeguati durante l'utilizzo in movimento del rover. Il problema di questa modalità è sempre legato all'aumento della distanza dall'access-point. In questo caso si assiste alla perdita di alcuni dati che compongono i frame, causando degli errori nella visualizzazione, i quali purtroppo vengono accentuati dall'utilizzo della codifica differenziale. Per entrambe le modalità è possibile migliorare il funzionamento: diminuendo la qualità o riducendo la risoluzione; attraverso l'utilizzo dei ROS-Service per la regolazione di questi parametri associati alla fotocamera.