



POLITECNICO DI TORINO

Corso di Laurea Magistrale in Ingegneria Informatica

Monitoraggio remoto di robot mediante tecnologie Web Based e framework ROS

Relatore: Prof. Ing. Enrico Masala

Candidato: Federico Barone

Introduzione

La robotica è un settore disciplinare che si prefigge come obiettivo la realizzazione e l'applicazione pratica di robot sia nelle attività di produzione industriale che di ricerca scientifica e tecnologica. Alla **robotica industriale** si affianca la **robotica avanzata**, che si occupa, principalmente, di realizzare applicazioni utili in ambienti ostici. Un settore in crescita della robotica avanzata è la **robotica di servizio**, i cui sviluppi appaiono piuttosto proficui per il progresso tecnologico. Fine ultimo della robotica di servizio è assistere ed interagire con l'essere umano. Un robot, pertanto, deve essere in grado di muoversi abilmente nell'ambiente circostante, di modo che non rechi danni ad oggetti o persone, e sia, altresì, in grado di sfruttare le proprie capacità svolgendo mansioni predefinite: applicazioni domestiche, assistenza medica, intrattenimento, agricoltura, education, monitoring, etc.

Obiettivi

L'obiettivo del presente elaborato è progettare e realizzare un'interfaccia web, definita **ROS Web Interface**, che permetta all'utente di interagire da remoto con il robot. La suddetta interfaccia consente il monitoraggio e il controllo dei vari test effettuati. Di fatto, l'interazione tra le applicazioni, android e web, è stata fondamentale per osservare i dati acquisiti dai relativi dispositivi di controllo. Dallo smartphone, collocato sul rover, non è stato possibile analizzare le informazioni ottenute dai sensori. L'interfaccia web, pertanto, ha lo scopo di osservare le informazioni rilevate. La matrice profonda di questo studio è rappresentata dal monitoraggio del robot a distanza da una stazione remota, nonché dal controllo dei suoi movimenti e dall'esercizio di ogni sua potenzialità. Si è reso, altresì, opportuno l'utilizzo di particolari sensori presenti nei dispositivi android e la gestione dei dati multimediali provenienti da essi. Per usufruire di tale tecnologia, è opportuno configurare correttamente l'ambiente di lavoro sul personal computer. In definitiva, il lavoro di tesi si avvale dell'apporto di studi incentrati sulla gestione remota di droni e rovers da parte dell'utente. Questi, insieme ad ulteriori analisi basate sulla gestione dei dati multimediali, sono i temi principali dell'elaborato.

Robot Operating System

ROS, acronimo di Robot Operating System, è il framework utilizzato per la realizzazione dell'interfaccia web. Individua uno standard de facto per la «prototipazione» e lo sviluppo di software per applicazioni di robotica. Il sistema ROS si basa sul concetto di grafico computazionale, che rappresenta la rete di processi ROS. Tale framework prevede una serie di concetti utili per poter comprendere l'architettura. Il «cuore» dell'architettura ROS è rappresentato dal

master, il quale ha il compito di orchestrare i **nodi** connessi, responsabili dell'esecuzione di calcoli e dell'elaborazione dei dati raccolti dai sensori. I nodi comunicano tra di loro mediante lo scambio di **messaggi**, attraverso un sistema di comunicazione di tipo *publish/subscribe*. I messaggi, di fatto, vengono pubblicati da un nodo su un dato **topic**, il quale può essere sottoscritto da un altro nodo se interessato ad un certo tipo di dato. Infine, sono presenti i **servizi**, i quali vengono offerti da un nodo server, mentre un nodo client li utilizza sfruttando il paradigma delle chiamate di procedura remote.

Configurazione ambiente ROS

L'interfaccia web progettata e realizzata per il monitoraggio da remoto di un rover prevede, anzitutto, una corretta configurazione dell'ambiente di lavoro, in modo tale che sia possibile la comunicazione tra i vari dispositivi collegati. Lo schema logico (presentato nella Figura 1) mostra i principali nodi che permettono una corretta interazione con la nostra interfaccia web. I nodi ROS presentati sono in grado di comunicare tra di loro mediante l'utilizzo di opportuni messaggi. Per concretizzare tale comunicazione, è necessario eseguire il nodo **ROS Master**. Questo, difatti, permette l'inizializzazione dell'ambiente ROS. I principali nodi utilizzati affinché l'interfaccia web sia in grado di monitorare da remoto i dispositivi mobili sono: **ROSBridge Web Socket**, il quale fornisce uno strato di livello intermedio in grado di interfacciare l'applicazione web al framework ROS; il nodo **Android Sensors** eseguito all'interno del dispositivo android mediante l'applicazione **PIC4SeR Monitoring**, dal quale è possibile ricavare i dati relativi ad alcuni dispositivi di controllo; **Network Sensors** permette di monitorare la scheda di rete NIC presente all'interno del personal computer. Una volta avviati i suddetti nodi, è possibile eseguire l'interfaccia web all'interno di un qualsiasi **Browser Web** sfruttando un server virtuale messo a disposizione da Visual Studio Code: **Live Server**.

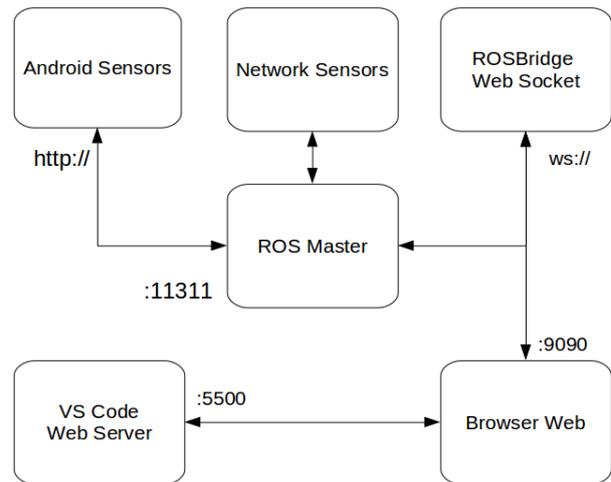


Figura 1: Configurazione dell'architettura ROS

Progettazione di ROS Web Interface

Il software applicativo **ROS Web Interface**, realizzato per il monitoraggio remoto del robot, sfrutta le caratteristiche messe a disposizione dalle tecnologie *Web Based*. Pertanto, l'interfaccia web è stata realizzata usufruendo di: **HTML5** per la creazione dello schema di formattazione; i principi del linguaggio **CSS3** per lo stile della pagina web; il linguaggio di scripting **JavaScript** per la realizzazione delle funzionalità in grado di interagire con il robot e con l'applicazione android. L'utilizzo delle funzionalità presenti all'interno della libreria **roslibjs** permette di interagire con il framework ROS. I grafici bidimensionali inseriti all'interno dell'interfaccia web, invece, sono stati realizzati mediante il supporto della libreria **RGraph.js**. All'interno dell'applicazione web è stata inserita una mappa che permette di localizzare il robot in movimento. Tale mappa sfrutta le API messe a disposizione dalla libreria **Mapbox**. L'interfaccia web è sta-

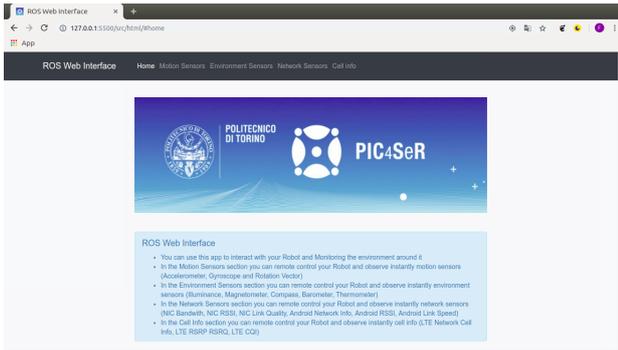


Figura 2: ROS Web Interface - Home Page

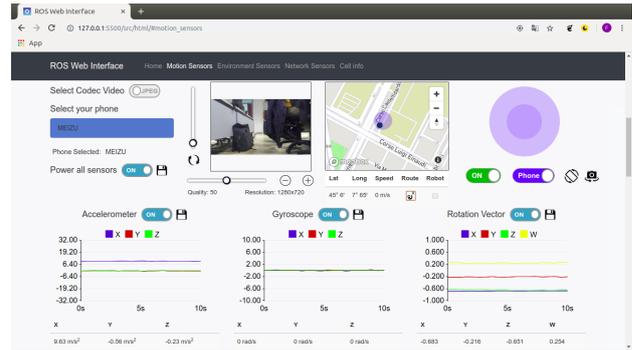


Figura 3: ROS Web Interface - Motion Sensors

ta suddivisa in sezioni in modo tale da visualizzare i sensori della medesima tipologia all'interno della stessa vista grafica. Nella *Home Page* (come mostrato in Figura 2) vengono descritte, in maniera concisa, le possibili azioni che l'utente può svolgere all'interno dell'applicazione web. In particolare, nel menu in testa, è possibile scegliere quale tipologia di sensori l'utente voglia analizzare. Tale menu è composto da quattro sezioni: la prima riguarda i *sensori di movimento*, ovvero accelerometro, giroscopio e vettore di rotazione (come mostrato nella Figura 3); a seguire vi sono i *sensori relativi all'ambiente circostante*, dunque illuminamento, campo magnetico, bussola, barometro e termometro; la terza mostra i *sensori di rete*, ovvero le informazioni sulla scheda di rete NIC e sul dispositivo android; infine, sono presenti le informazioni riguardo la *connessione LTE*. I grafici bidimensionali realizzati per la visualizzazione dei dati prevedono: un tasto di selezione per l'acquisizione dei dati e un pulsante per il salvataggio in opportuni file in formato **JSON**. A queste informazioni si aggiungono: il supporto per la *teleoperazione*, sfruttando un opportuno joystick realizzato usufruendo della libreria **Nipples**; l'acquisizione delle *immagini multimediali* provenienti dal dispositivo android o dalla telecamera ubicata a bordo del rover; le informazioni relative alla *geolocalizzazione* del robot (mostrate all'interno della mappa), le quali possono essere acquisite sfruttando il sensore GPS presente sul robot, oppure, quello collocato all'interno al dispositivo android stesso. L'interfaccia web prevede, inoltre, un elenco (aggiornato automaticamente in base ai dispositivi collegati) dedicato alla selezione del dispositivo android dal quale ricavare le informazioni sui sensori.

Conclusioni e sviluppi futuri

La progettazione e realizzazione dell'applicazione web è stata utile al monitoraggio e alla gestione dei dati multimediali provenienti dal dispositivo android e dal rover **Clearpath Jackal**. Analizzando particolari sensori, è stato possibile trarre considerazioni interessanti. Di fatto, le analisi condotte sulla rete wireless e sulla rete mobile LTE sono state oggetto di discussione. L'utilizzo di un unico *access point*, in entrambi i casi, non è stato sufficiente per il monitoraggio di ampi spazi. Inoltre, l'attenuazione dei valori dei parametri prestazionali utilizzati per l'analisi della rete conferma quanto detto. Nei contesti applicativi real time, l'utilizzo di più *access point* sarebbe, certamente, una delle possibili soluzioni. Questo, infatti, consentirebbe il monitoraggio di spazi più ampi. Ulteriori analisi potrebbero essere effettuate mediante l'utilizzo di opportuni aeromobili a pilotaggio remoto. Difatti, lo smartphone sostituirebbe i vari sensori collocati sul drone, diminuendo il peso di quest'ultimo e migliorando le sue prestazioni in volo. La robotica di servizio, pertanto, risulta fondamentale per la realizzazione ed integrazione delle applicazioni pratiche, tra le quali assumono sempre maggior rilievo le applicazioni web.