

Università degli Studi di Catania  
Corso di Laurea Specialistica in Ingegneria Informatica  
Corso di Sistemi Cognitivi e Interazione Uomo-Macchina

## **PROTOTIPO dell'APPLICAZIONE INTERATTIVA:**

R.R.M. (Remote Robot Monitoring)

**Versione:**

1.0

**Data:**

15 Gennaio 2009

**Autori:**

Nicolò Grippaldi - A63/000231

Loris Fichera - A63/000232

## 1. INTRODUZIONE

Il lavoro di progettazione dell'interfaccia descritta in questo paper si inserisce all'interno del lavoro di preparazione al campionato di robotica *Eurobot Open 2009*. Il campionato si svolge annualmente in una sede Europea (l'edizione 2009 si terrà a La Ferté-Bernard, in Francia) e prevede la partecipazione di team da ogni parte del mondo. Scopo dell'edizione 2009 è quello di costruire un robot in grado di edificare *templi* a partire dagli elementi presenti sul campo da gioco: pezzi di *colonne* e *architravi*. Il campionato si svolge secondo la formula del *match-race*: vengono giocate delle partite *uno contro uno* di 90 secondi ciascuna, al termine dei quali viene proclamato vincitore il team il cui robot ha edificato il maggior numero di templi.

Durante il lungo lavoro di preparazione al campionato è di fondamentale importanza poter eseguire numerose sessioni di test per la calibrazione dei sensori del robot, il testing delle strategie, il miglioramento dell'affidabilità del software che guida il robot. Tali test richiedono una interazione tra un operatore umano e il robot stesso: l'operatore invia dei comandi al robot, quindi ne osserva la risposta, quindi valuta la bontà delle soluzioni adottate. L'interfaccia da noi progettata si propone di facilitare tale interazione, andando a sostituire una potente - ma sicuramente scomoda - interazione a mezzo linea di comando.

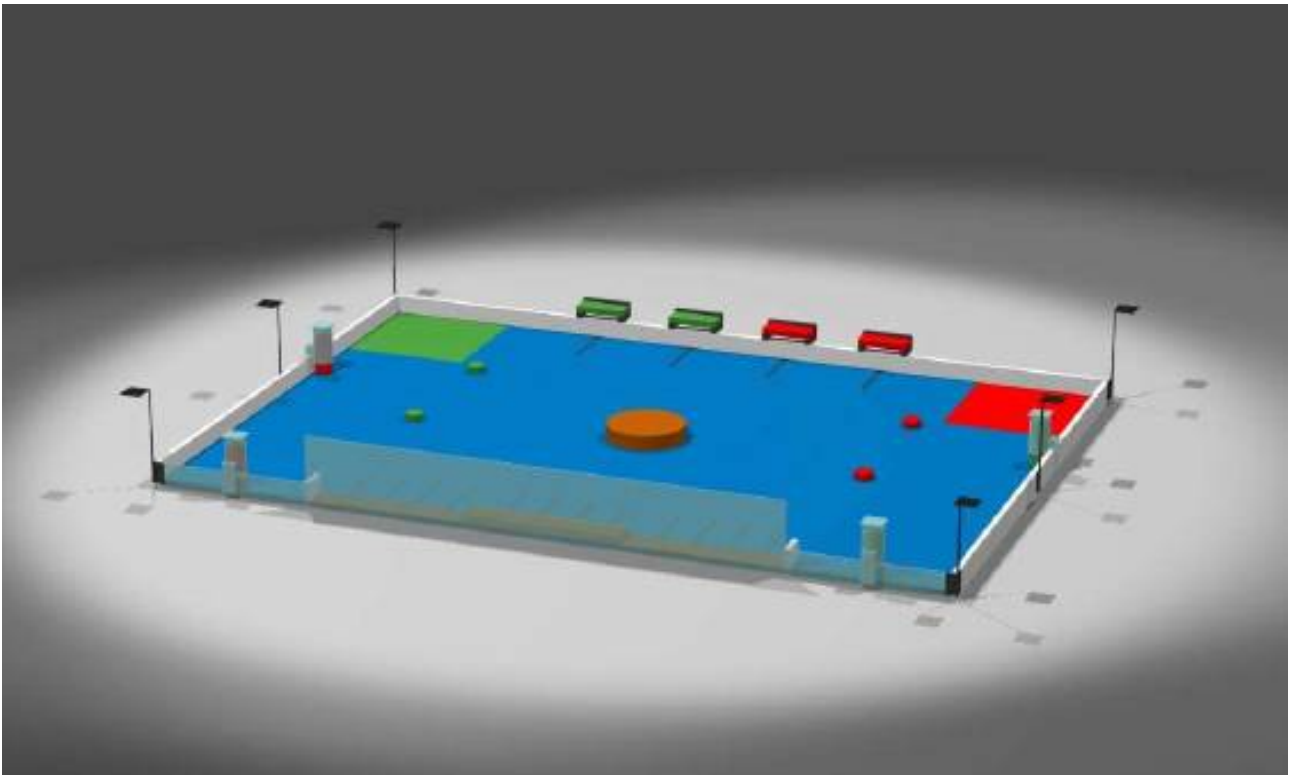


Illustrazione 1: Campo da gioco di Eurobot Open 2009

## 2. DESIGN dell'INTERFACCIA

### 2.1 FUNZIONALITÀ

Per permettere lo scambio di informazioni con il robot, è stato necessario implementare una funzionalità che permette all'utente di connettersi con il robot stesso. Poiché il robot deve muoversi sul campo da gioco per prendere pezzi di *colonne* e *architravi* utili alla costruzione dei *templi*, deve seguire determinate traiettorie che vengono fornite dall'utente tramite l'interfaccia. Una volta connesso al robot, l'utente, cliccando con il mouse sul campo da gioco, può selezionare i punti che costituiscono le traiettorie che il robot deve percorrere per raggiungere l'obiettivo nel minor tempo possibile. Definite le traiettorie, premendo con il mouse un pulsante di *start*, l'utente fa muovere il robot lungo le traiettorie stabilite. A Robot in movimento, se ha intenzione di interrompere lo spostamento premerà un pulsante di *stop*.

E' necessario consentire all'operatore anche la possibilità di monitorare sia la velocità istantanea sia l'accelerazione del Robot, che sono due parametri di fondamentale importanza per il controllo delle traiettorie.

Dopo aver premuto un pulsante di *start*, la traiettoria reale effettuata dal Robot dovrà essere tracciata in *real-time*. Tale rappresentazione visuale permette di testare i sensori di posizione, il sistema di localizzazione, il sistema di *obstacle avoidance*.

L'interfaccia dovrà permettere all'utente di monitorare lo stato delle batterie del robot grazie a un indicatore del livello di carica delle stesse.

### 2.2 VINCOLI

Il progetto è stato condizionato dalla presenza di alcuni vincoli:

- il robot è pilotato da un framework robotico scritto in Erlang ed eseguito sul pc di bordo
- il software verrà utilizzato all'interno di un laboratorio
- il software dovrà essere eseguito su notebook e/o mini-pc dotati di sistema operativo GNU/Linux

Tali vincoli hanno influito sulla scelta del linguaggio di programmazione e delle librerie grafiche utili alla costruzione dell'interfaccia.

### 2.3 UTENTI

Gli utenti che utilizzeranno l'interfaccia sono ingegneri, studenti di Ingegneria dell'Automazione e di Ingegneria Informatica, esperti di robotica, utenti esperti di linux. Proprio perché il prodotto è destinato ad utenti esperti, alla base del progetto c'è stata l'idea di velocizzare quanto più possibile l'interazione con il robot. A tal fine si è pensato di permettere l'accesso alle funzionalità sopra descritte non solo tramite voci di menu ma anche attraverso *combinazione rapida di tasti*. Inoltre, la modalità di rappresentazione dei dati e delle grandezze sarà coerente con le convenzioni e gli standard comunemente adottati nei laboratori di robotica: ad esempio, ciascun punto del campo da gioco, dovrà essere rappresentato sotto forma di coordinate cartesiane (x,y) invece di essere identificato da una lettera o da un nome specifico.

## 2.4 SCELTE DI DESIGN

Durante il progetto dell'interfaccia si è sempre cercato di raggiungere un buon compromesso tra *estetica* ed *utilità*, cioè abbiamo fatto in modo di implementare tutte le funzionalità di cui ha bisogno l'utente cercando di aiutarlo attraverso una buona organizzazione degli oggetti (finestre, pulsanti, menu). Inoltre il progetto è stato portato avanti anche con l'obiettivo di permettere all'utente di distinguere facilmente gli elementi *passivi*, che danno solo informazioni, (es: barra di stato collocata in basso che indica la posizione all'interno del campo da gioco man mano che l'utente muove il cursore su di esso), dagli elementi *attivi* che, invece, devono essere cliccati affinché l'utente possa eseguire determinate operazioni (pulsanti, voci di menu).

Per la realizzazione concreta dell'interfaccia è stato scelto di utilizzare un linguaggio interpretato per garantirne la portabilità: la scelta è ricaduta sul linguaggio di programmazione *Python* e sulle librerie grafiche *wxPython*.

Per aiutare l'utente a visualizzare meglio gli spostamenti del robot, la posizione degli ostacoli e la disposizione degli oggetti, il campo da gioco è stato rappresentato con visuale "dall'alto".

Come detto precedentemente, la funzionalità principale da sviluppare sarà l'immissione di traiettorie da fare effettuare al robot. L'interfaccia sarà costituita, quindi, per buona parte da una rappresentazione "dall'alto" del campo da gioco. Saranno necessari dei pulsanti di START e STOP e altre widget per la rappresentazione dello stato dei sensori.

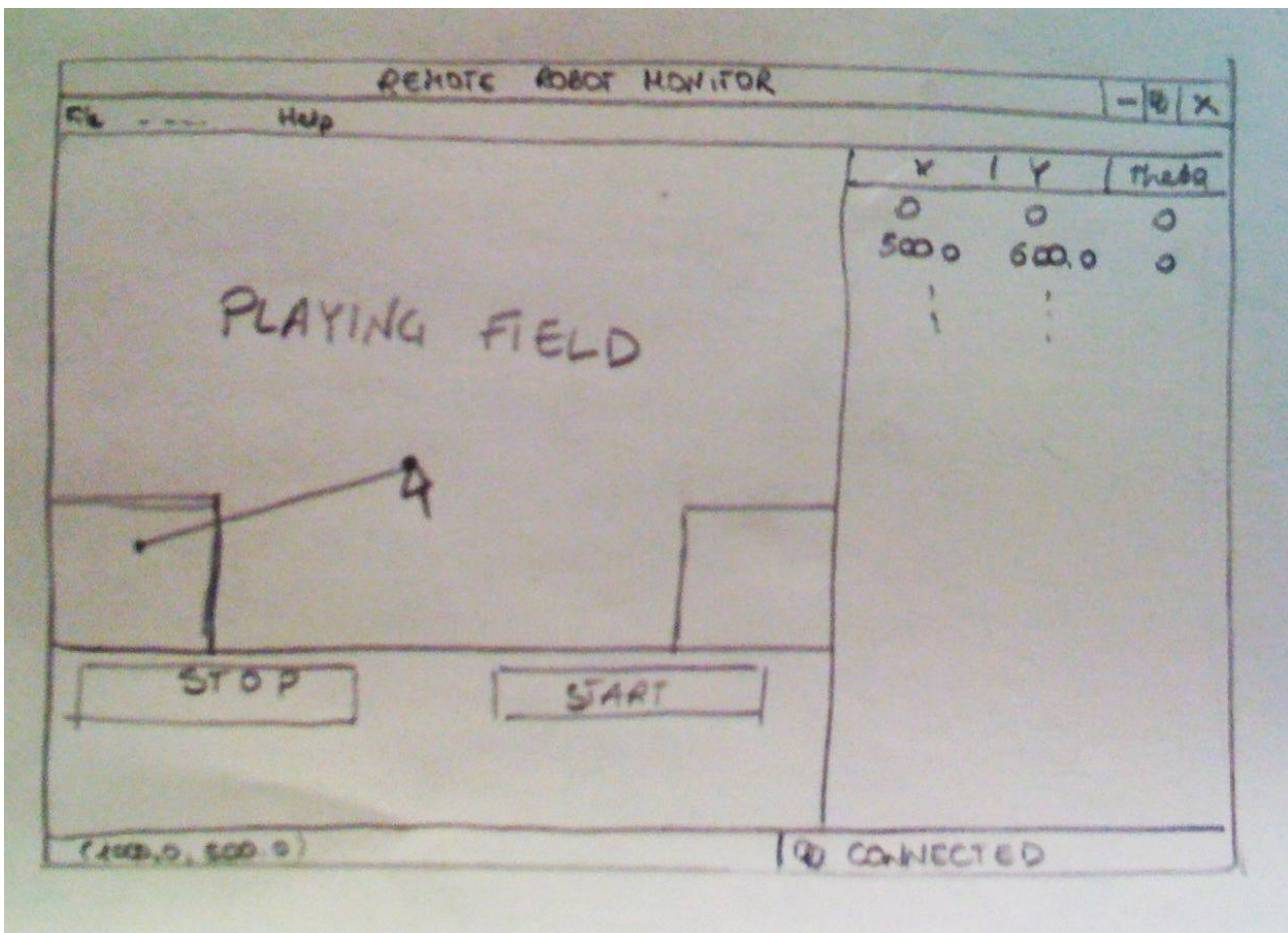


Illustrazione 2: Primo prototipo dell'interfaccia

L'utente potrà fissare i punti utili per generare le traiettorie cliccando con il tasto sinistro del mouse sulla posizione voluta all'interno del campo da gioco che rappresenta sicuramente l'azione più semplice che possa fare l'utente: rispetto al vecchio sistema di interazione a mezzo linea di comando, è stata spostata la complessità dall'utente al sistema in quanto, una volta fissati i punti, è il sistema stesso che si occupa di generare la traiettoria e di passarla al robot. Per una maggiore chiarezza e per dare la possibilità all'utente di individuare eventuali errori, sarà presente una widget che conterrà una rappresentazione tabellare della traiettoria impostata.

Un aspetto rilevante dell'interfaccia è l'uso coerente dei colori. Infatti, sapendo che per la maggior parte delle civiltà occidentali il colore rosso indica pericolo, mentre il verde è associato a qualcosa di positivo, abbiamo usato il rosso per il pulsante di STOP e il verde per il pulsante START. Inoltre nel campo da gioco, l'uso di colori diversi permette di identificare le diverse regioni che lo costituiscono e permette di metterlo in risalto rispetto al resto dell'interfaccia, caratterizzato da uno sfondo bianco che consente la visualizzazione sia della barra di stato sia della lista dei punti generati.

*Last, but not least*, per offrire un feedback continuo e immediato delle operazioni effettuate sarà presente una barra di stato in basso contenente informazioni sullo stato della connessione con il robot e che permetterà di visualizzare il punto del tavolo su cui il cursore del mouse si trova: questo faciliterà le operazioni di immissione della traiettoria.

### 3. STORYBOARD

In questa sezione, sarà descritto, a mezzo di uno *storyboard*, un possibile scenario di interazione. Per ogni step verranno indicati dei *tips*, ovvero dei suggerimenti per rendere l'interazione più rapida ed efficiente. Gli screenshots di seguito riportati sono stati catturati su un sistema *Debian*, dotato di *Python 2.6* e librerie *wxPython 2.8*.

**Scenario:** “L'operatore si collega al robot, imposta una traiettoria, la esegue e visualizza la traiettoria realmente effettuata dal robot”.

#### STEP 1: Connessione al Robot

Affinché il software possa interfacciarsi con il robot è necessario creare una connessione wireless tra il robot stesso (dotato di antenna wi-fi) e il pc su cui l'operatore lavora. L'operatore troverà nel menù “Actions” la voce “Connect to Robot”.

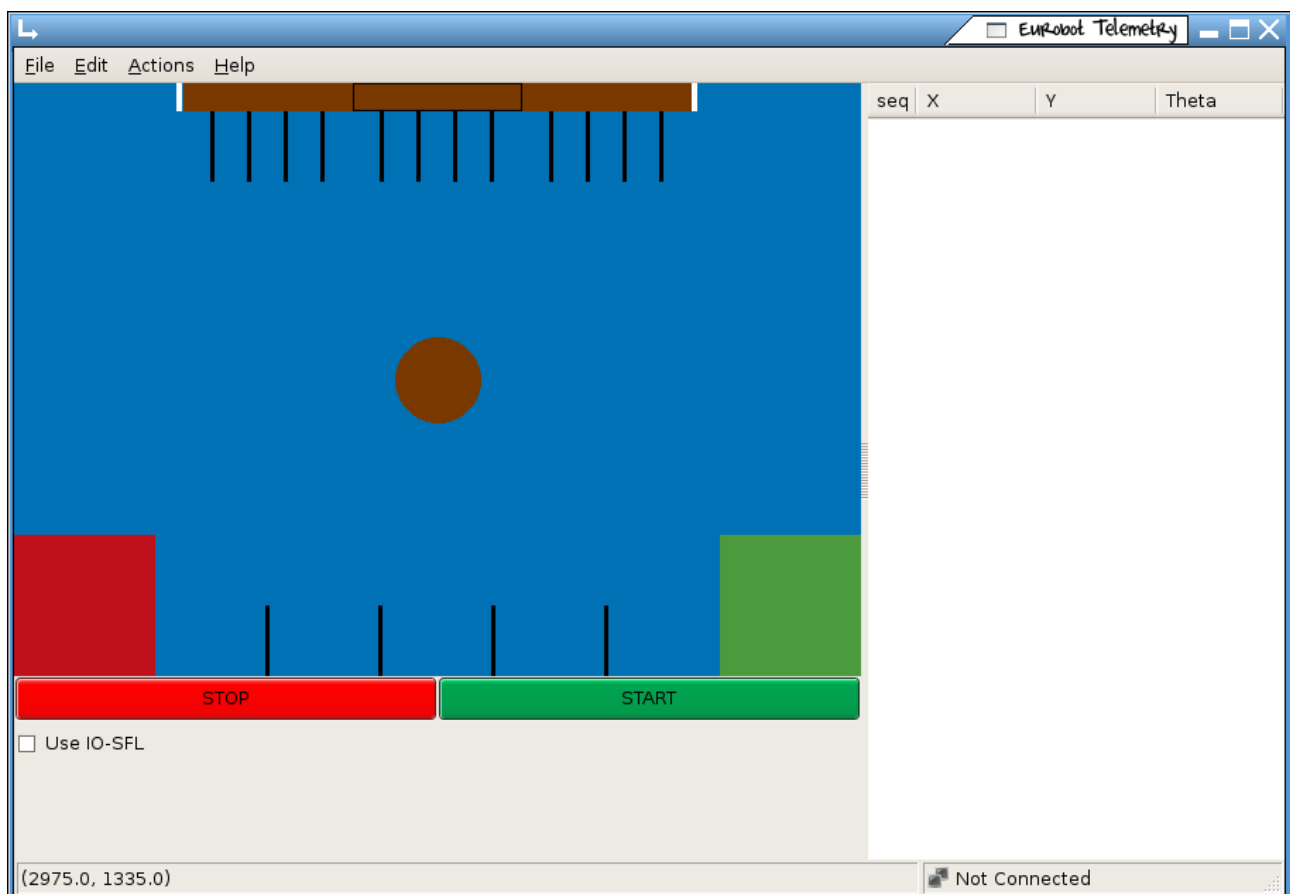


Illustrazione 3: Aspetto dell'Interfaccia

#### Tips:

- Per aprire rapidamente la finestra di dialogo della connessione è disponibile la scorciatoia da tastiera *ctrl+c*.

## STEP 2: IMPOSTAZIONE DELLA CONNESSIONE

Una finestra di dialogo richiede l'inserimento dell'indirizzo IP del Robot. Al termine dell'immissione, l'operatore premerà il tasto "OK".

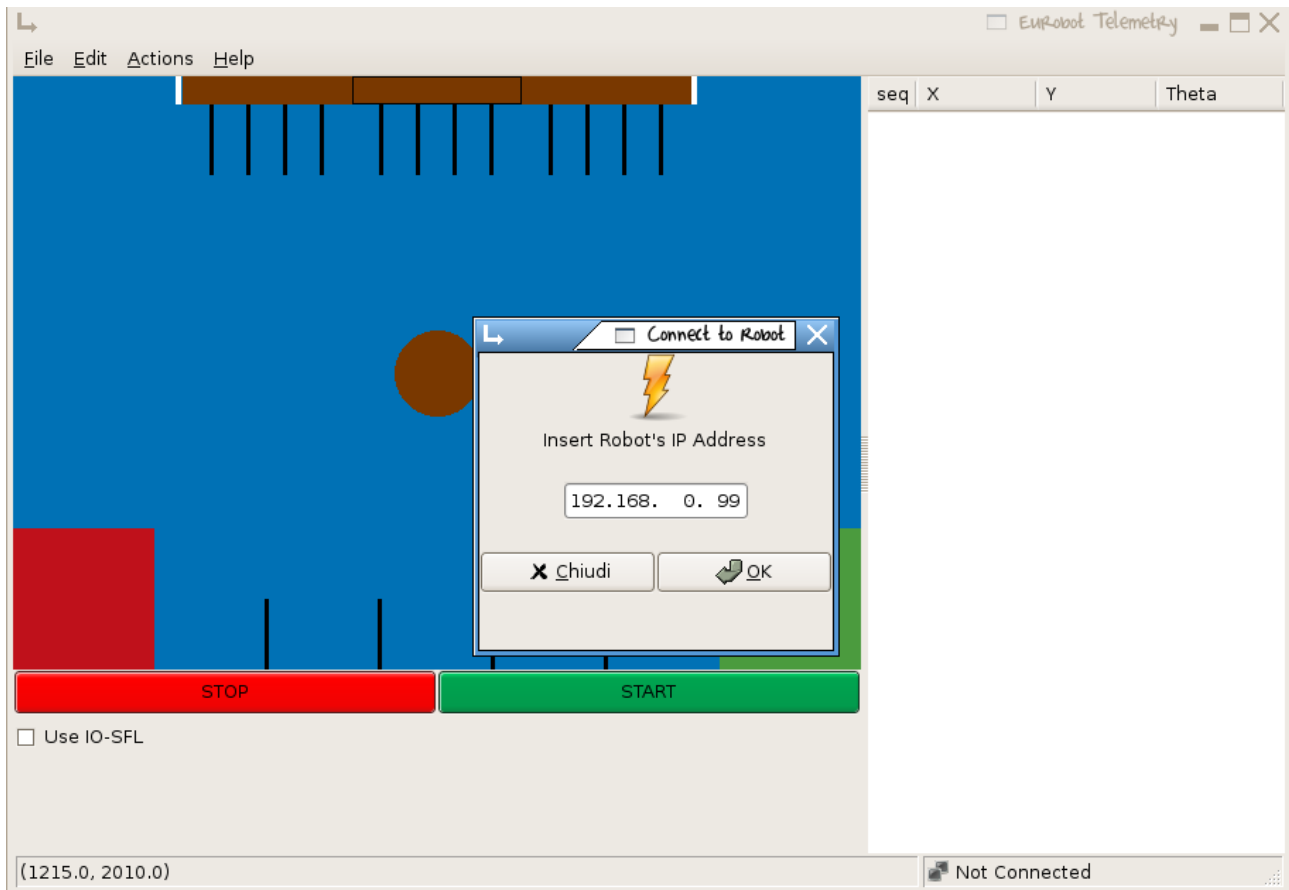


Illustrazione 4: Immissione indirizzo IP del Robot

### Tips:

- all'apertura, la finestra di dialogo propone un valore IP di default (che corrisponde all'ultimo indirizzo usato per connettersi al Robot).
- il campo IP viene colorato di giallo se l'indirizzo immesso non è valido (vedi figura 3).

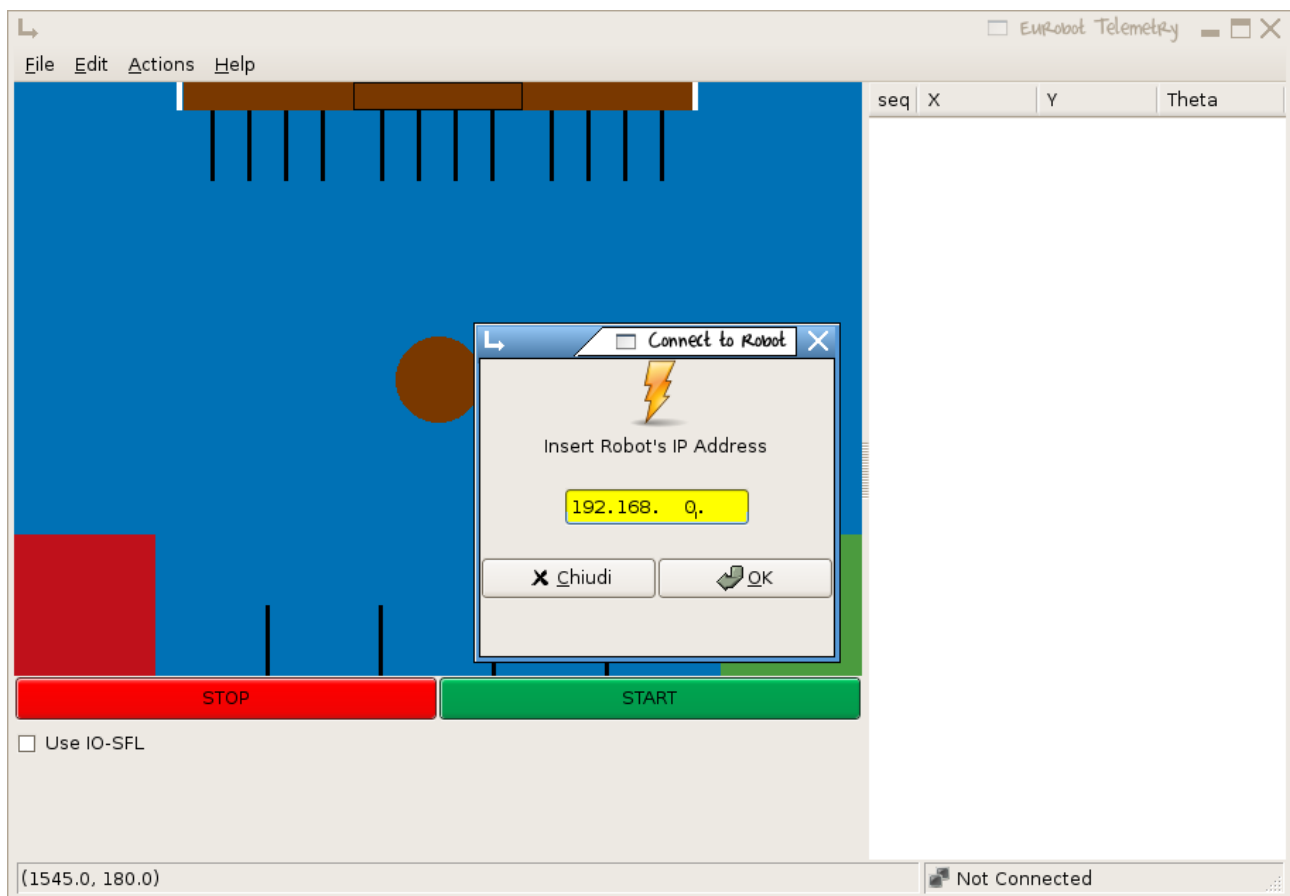


Illustrazione 5: Inserimento Indirizzo IP non valido



### STEP 3: ATTESA INSTAURAZIONE DELLA CONNESSIONE

Viene aperta una finestra di dialogo dotata di barra di scorrimento che informa del tentativo di connessione in corso.

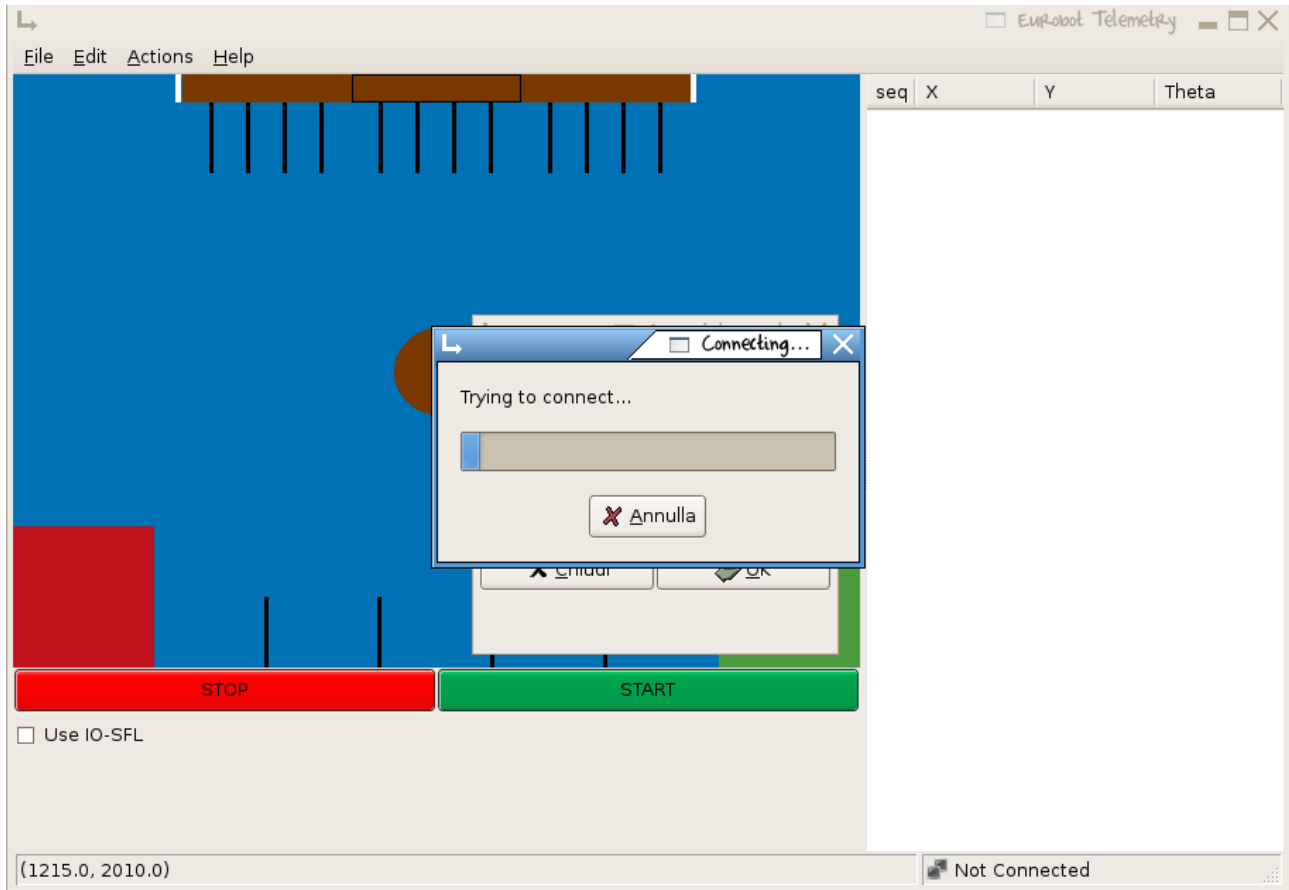


Illustrazione 6: Finestra di dialogo con barra di scorrimento

Al termine, viene confermata l'avvenuta connessione.

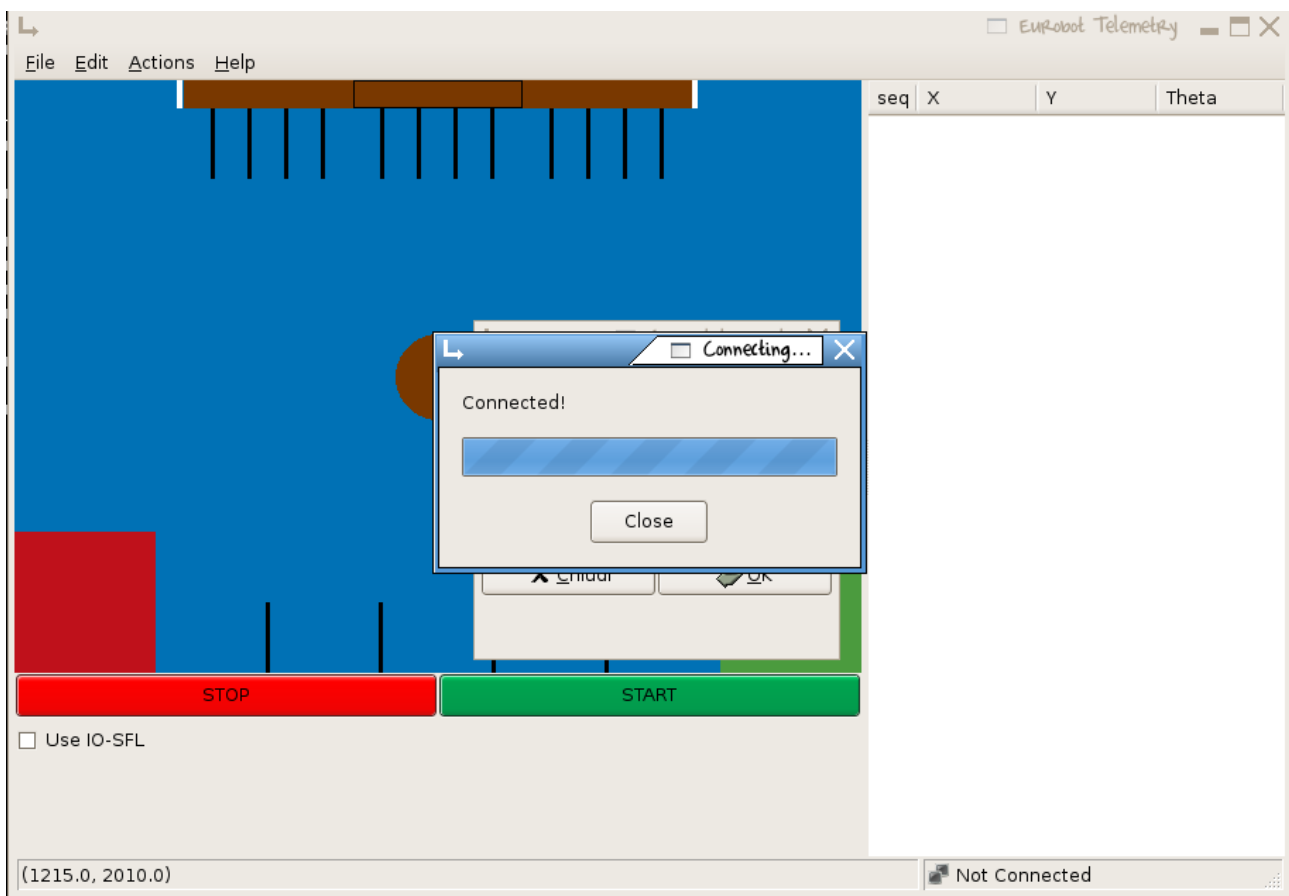


Illustrazione 7: A connessione avvenuta

**Tips:**

- il dialogo è provvisto di pulsante di chiusura.
- la parte della barra di stato che informa sullo stato della connessione al robot viene aggiornata con una nuova icona (con computer “illuminati”) e la dicitura “Connected to <indirizzo\_robot>”

#### STEP 4: IMPOSTAZIONE TRAIETTORIA

Cliccando sul tavolo da gioco viene creata una traiettoria per punti.

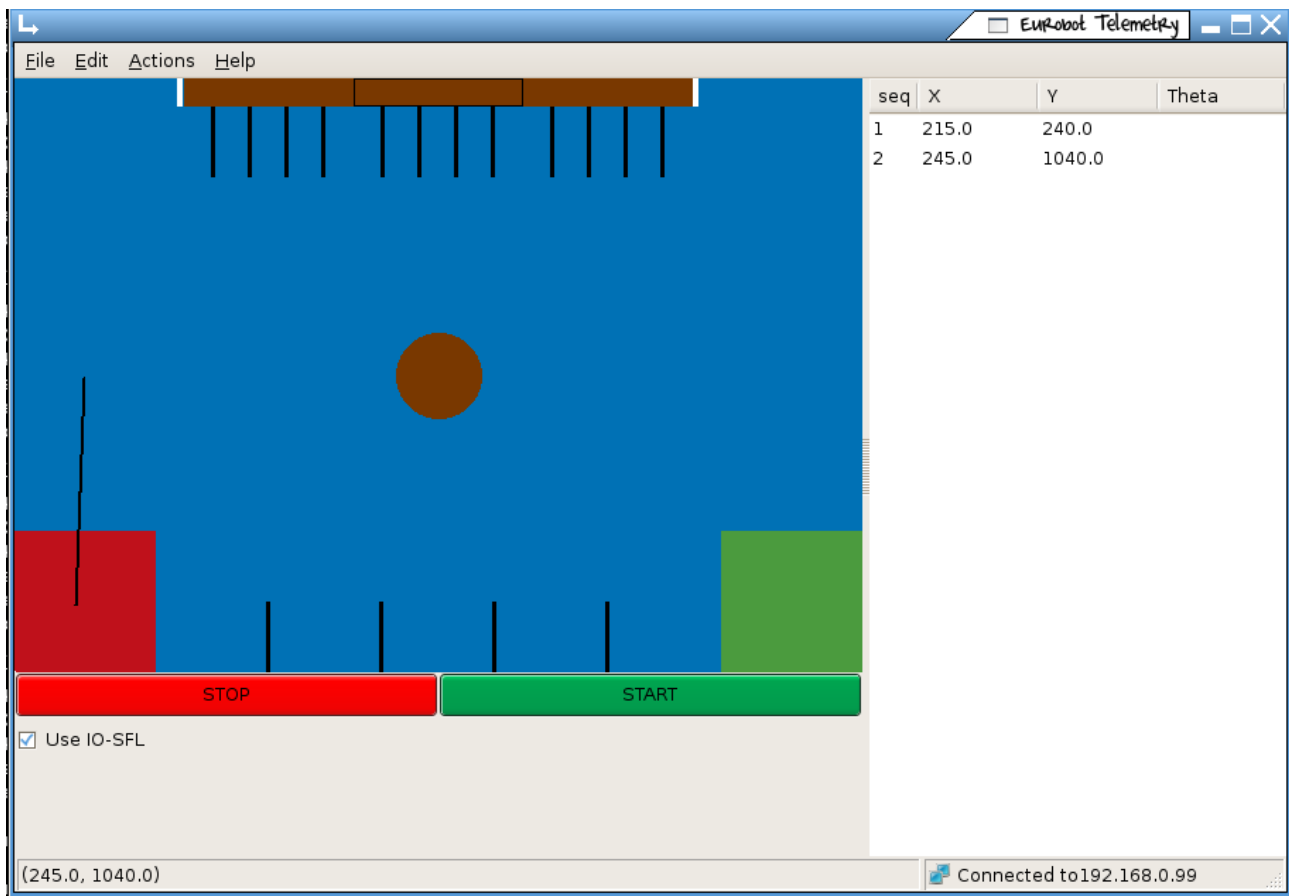


Illustrazione 8: Impostazione traiettoria 1/2

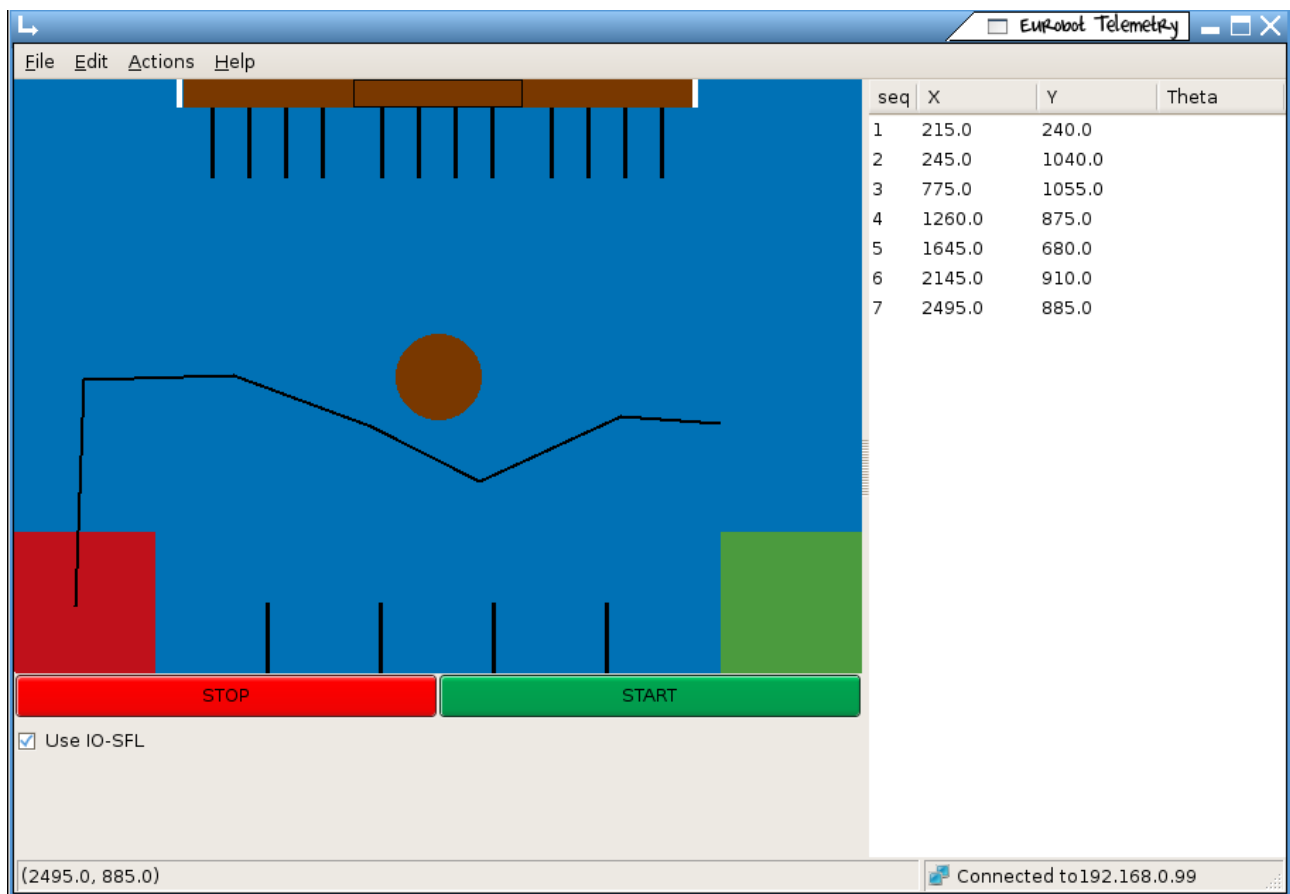


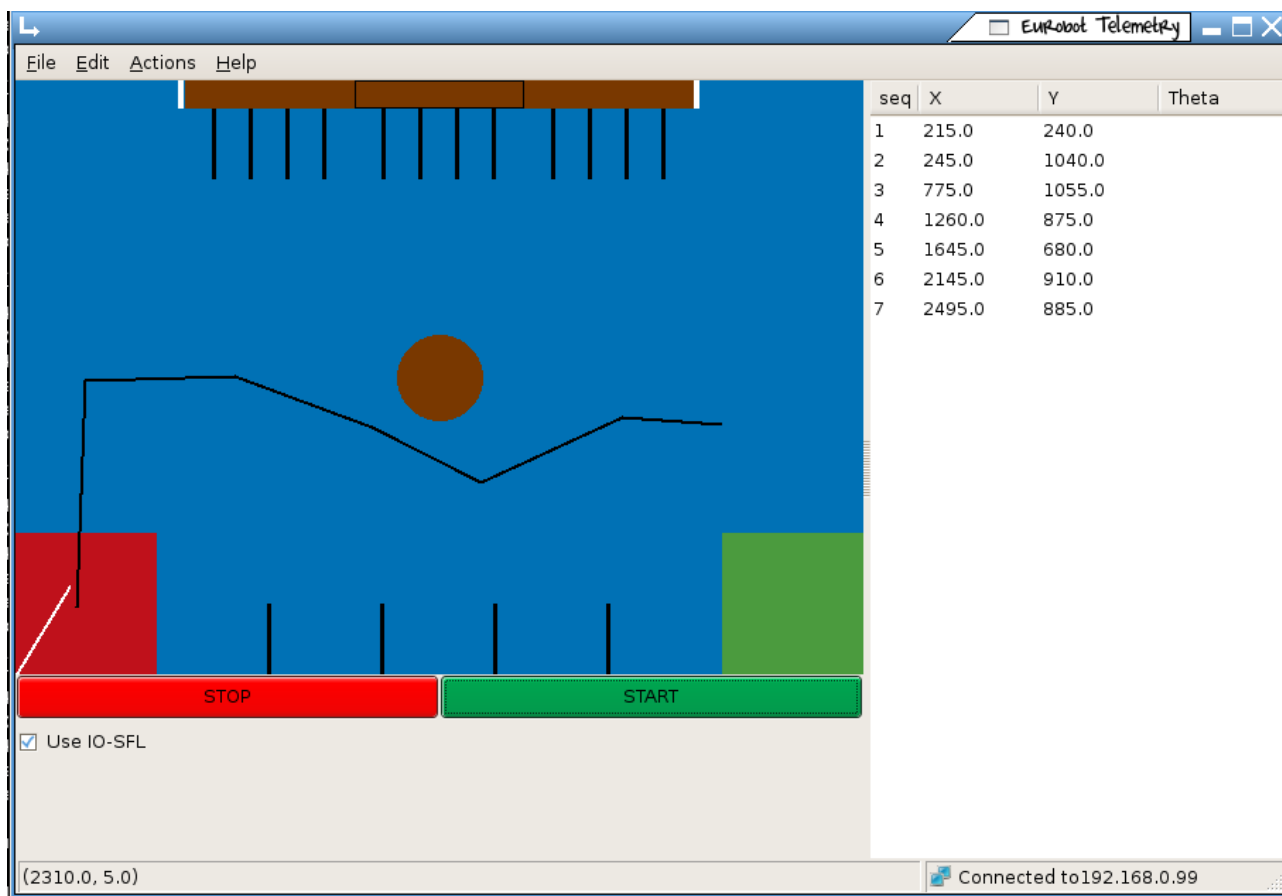
Illustrazione 10: Impostazione traiettoria 2/2

#### Tips:

- un punto immesso per sbaglio può essere rimosso cliccando su Edit>Undo (o, in alternativa, con la combinazione di tasti *ctrl+z*).
- possono essere impostate altre opzioni come IO-SFL.

## STEP 5: EFFETTUAZIONE TRAIETTORIA

Cliccando sul pulsante START, l'operatore osserva la traiettoria effettuata dal Robot.



### Illustrazione 11: Effettuazione della traiettoria 1/2

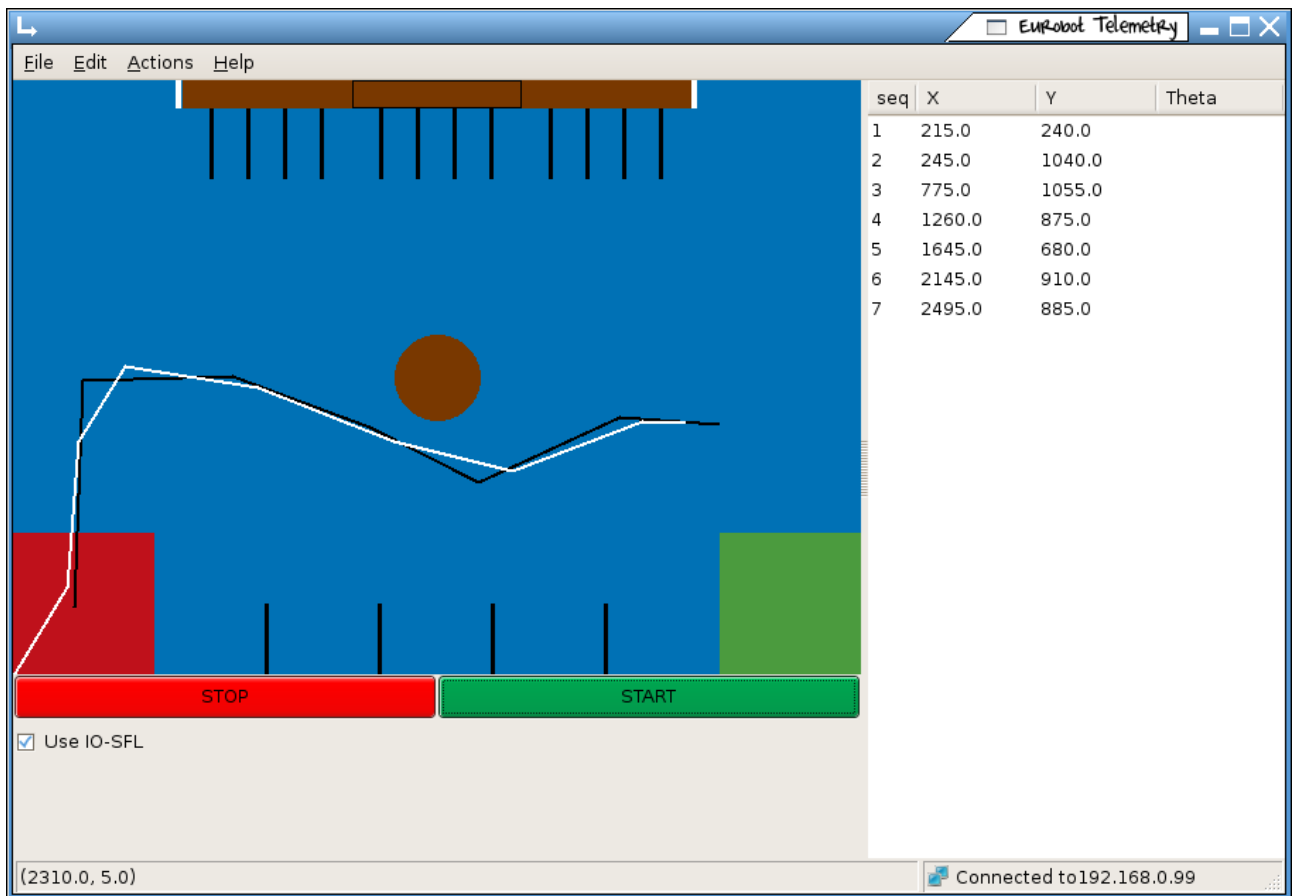


Illustrazione 12: Effettuazione della traiettoria 2/2

#### 4. FUTURE WORK

Quello proposto in questo paper è solo un prototipo dell'applicazione vera e propria: non tutte le funzionalità sono state sviluppate. I possibili sviluppi futuri di questo lavoro includono, quindi:

- sviluppo delle funzionalità proposte
- adeguata rappresentazione del robot sul campo da gioco
- funzionalità di salvataggio/ripristino della sessione di lavoro