TraCI4Matlab: Manual de usuario.

Facultad de Minas
GAUNAL
Grupo de Automática de la Universidad Nacional

POLITÉCNICO COLOMBIANO
JAIME ISAZA CADAVID

Versión 1

Febrero de 2014

Principales modificaciones por versión del documento

Historial de versiones.

Versión	Autores	Fecha	Descripción de la modificación
1	Andrés Felipe Acosta Gil	05/02/2014	Creación del documento

Contenido

1. ¿Qué es TraCI4Matlab?

TraCI4Matlab es una API (Application Programming Interface) desarrollada en Matlab que permite la comunicación entre cualquier aplicación desarrollada en este lenguaje y el simulador de tráfico urbano SUMO (Simulation of Urban Mobility). Las funciones que componen TraCI4Matlab implementan el protocolo de nivel de aplicación TraCI (Traffic Control Interface), que está construido sobre el stack TCP/IP, para que la aplicación desarrollada en Matlab, que es el cliente, pueda acceder y modificar el entorno de simulación que provee el servidor (SUMO). TraCI4Matlab permite el control de los objetos de SUMO, como lo son los vehículos, semáforos, calles, etc, lo que permite simular diferentes aplicaciones como el control predictivo de semáforos y la asignación dinámica de rutas, entre otras.

2. Instalación de TraCI4Matlab

2.1 Prerrequisitos

- Sistema operativo: Windows 7 x64 o superior.
- Matlab R2011b x64 o superior.
- SUMO 0.19.0 o superior. Las instrucciones de instalación se pueden encontrar en http://sumo-sim.org/userdoc/Installing.html

2.2 Instalación

- **Paso 1: Configuración de las variables de entorno**
  Establecer la variable de entorno SUMO_HOME con alcance de sistema, con un valor correspondiente al directorio raíz de la instalación de SUMO. Por ejemplo, si se instaló la versión de SUMO 0.19.0 en C:\, entonces la variable SUMO_HOME tendría un valor de C:\sumo-0.19.0. Las variables de entorno se pueden configurar de la siguiente manera: Primero, hacer clic en el botón de inicio de Windows, luego clic derecho en Equipo y clic en la opción propiedades, como se muestra en la figura 1. En la parte izquierda de la ventana que se abre, hacer clic en el enlace "Configuración avanzada del sistema", como lo muestra la figura 2. Luego, en la ventana que se abre, hacer clic en el botón "Variables de entorno". En el campo "Variables del sistema", hacer clic en el botón "Nueva" y configurar la variable de acuerdo con la instalación de SUMO, como se muestra en la figura 3. Finalmente, en el mismo campo, buscar la variable *"path"* y anexarle la ruta al directorio *bin* del directorio de instalación de SUMO. Esto se hace agregando un punto y coma al valor actual y luego, la ruta correspondiente, como se muestra en la figura 4.

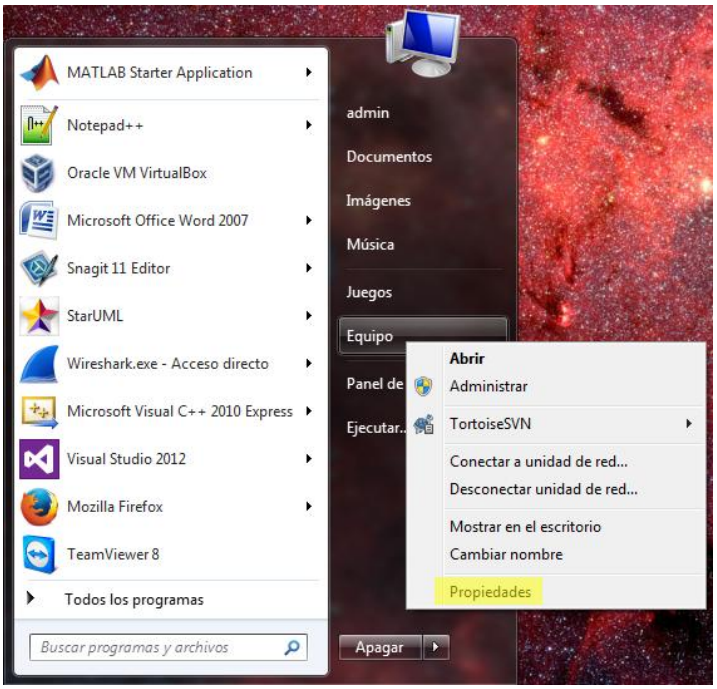

Figura 1 Configuración de variables de entorno, acceso a propiedades del sistema

- **Paso 2:** Descargar TraCI4Matlab y descomprimirlo en algún directorio que esté en el *path* de Matlab. La mayoría de las veces, éste se encuentra en Documentos\MATLAB. La verificación se puede hacer en Matlab, en el menú *File*, opción set *path*.

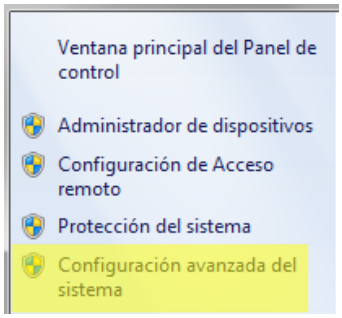

Figura 2 Configuración de variables de entorno, configuración avanzada del sistema

- **Paso 3: Agregar TraCI4Matlab al path de Matlab**
  En Matlab, ir al menú *File*, opción set *path*. Seleccionar el botón *"Add with subfolders"* y seleccionar el directorio TraCI4Matlab descomprimido en el paso anterior, como lo muestra la figura 5. Finalmente, clic en el botón *save* y en *close*.

- **Paso 4: Probar el correcto funcionamiento de TraCI4Matlab**
  Abrir el *script* traci_test2.m ubicado en el directorio examples del directorio TraCI4Matlab y ejecutarlo. Se debe abrir la interfaz gráfica de SUMO, donde se debe hacer clic en el botón *play.* Se muestra una simulación con seguimiento visual a un vehículo de color azul.

Luego de terminada la simulación, en el segundo 801, se muestra una gráfica de Matlab, como se observa en la figura 6.

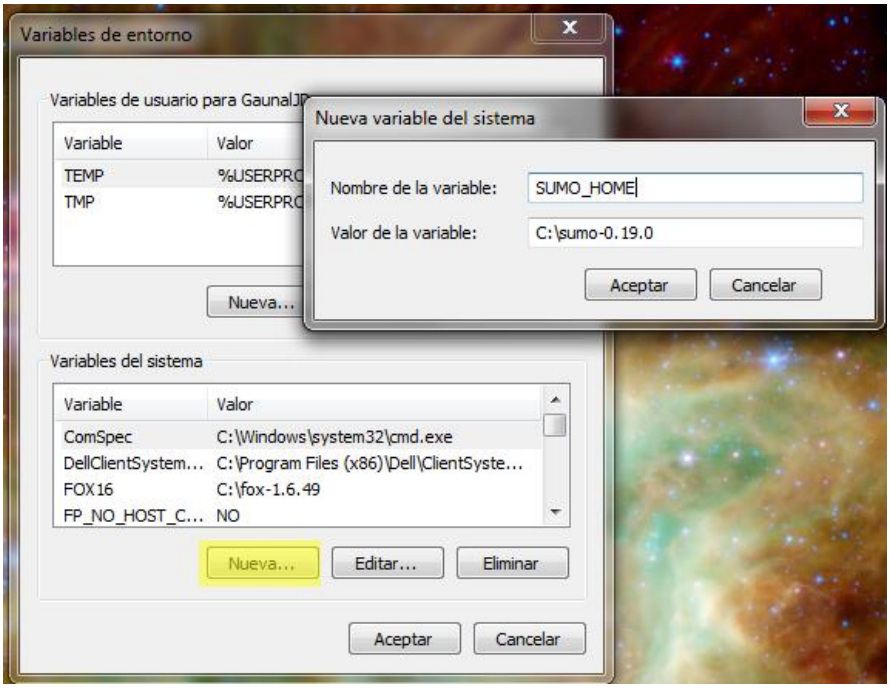

Figura 3: Configuración de variables de entorno, creación

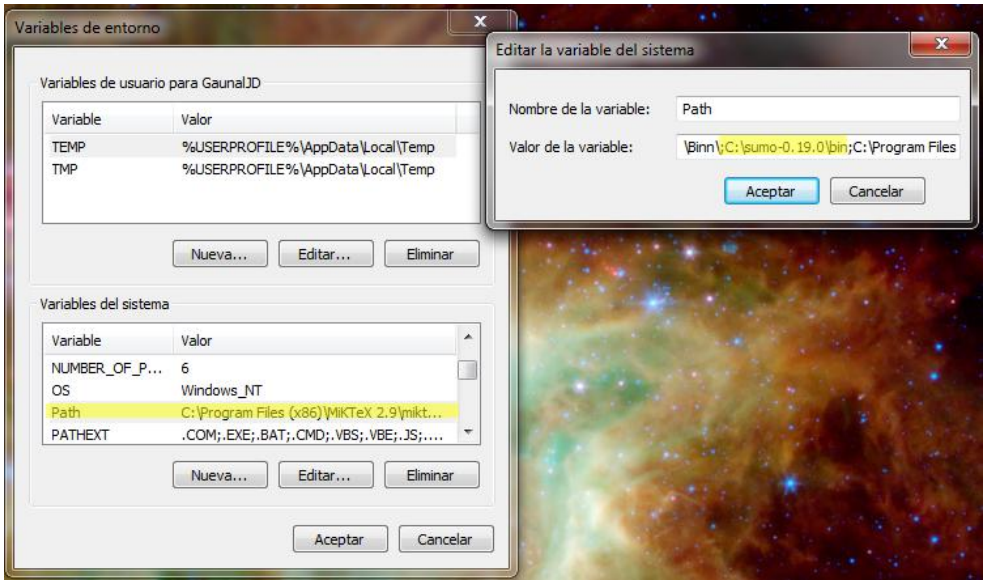

Figura 4: Configuración de variables de entorno, edición del path

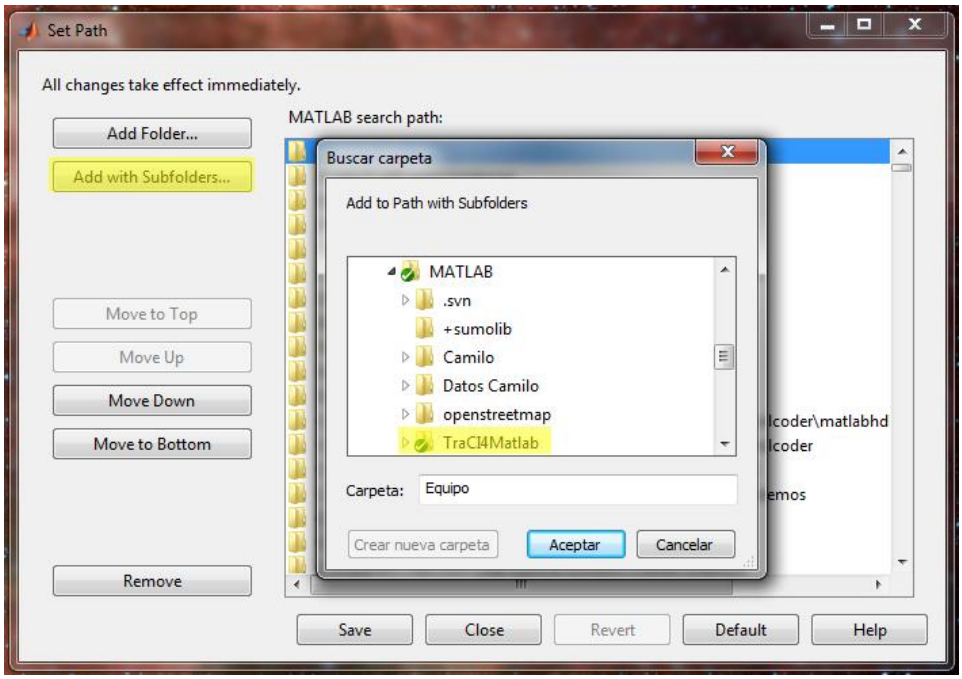

Figura 5: Agregar TraCI4Matlab al path de Matlab

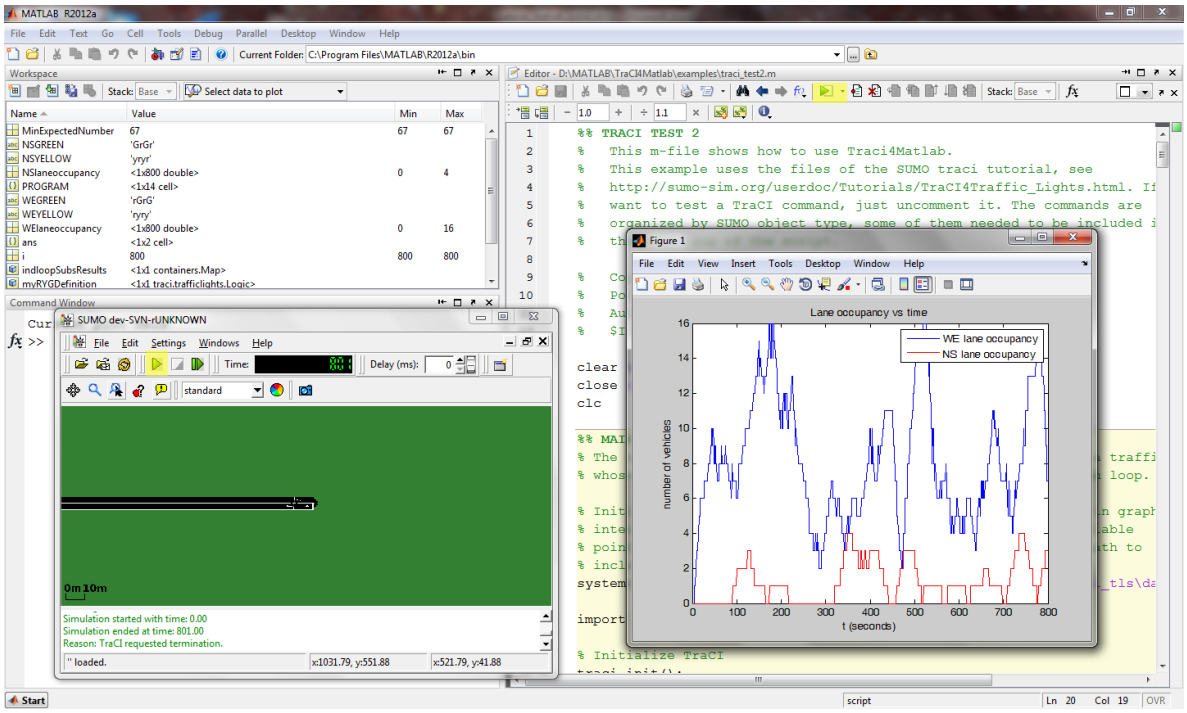

Figura 6: Comprobación del correcto funcionamiento de TraCI4Matlab

3. Utilización de TraCI4Matlab

3.1 Creación del escenario de simulación en SUMO.

Para utilizar TraCI4Matlab, el primer paso es crear un escenario de simulación de SUMO. La creación de dicho escenario se sale del alcance del presente manual, pero se puede acudir al tutorial oficial que se encuentra en la página http://sumo-sim.org/userdoc/Tutorials/Quick_Start.html.

3.2 Configurar SUMO en modo servidor.

Es muy importante anotar que para utilizar TraCI4Matlab, el archivo de configuración del escenario SUMO (aquel con extensión *.sumocfg*) debe incluir el elemento *traci_server* configurado en el puerto 8813, que es el puerto que se utiliza por defecto, tal como se muestra en la figura 7. Este elemento hace que SUMO no ejecute la simulación de inmediato, sino que se ponga en un estado de escucha por el puerto 8813.

```
1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <configuration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4
5      <input>
6          <net-file value="cross.net.xml"/>
7          <route-files value="cross.rou.xml"/>
8          <additional-files value="cross.det.xml"/>
9      </input>
10
11     <time>
12         <begin value="0"/>
13     </time>
14
15     <report>
16         <verbose value="true"/>
17         <no-step-log value="true"/>
18     </report>
19
20     <traci_server>
21         <remote-port value="8813"/>
22     </traci_server>
23
24 </configuration>
```

Figura 7: Inclusión del elemento *traci_server* en el archivo de configuración de SUMO

3.3 Creación de la aplicación en Matlab.

- **Paso 1: Ejecutar SUMO desde Matlab**

  Toda aplicación en Matlab que utilice TraCI4Matlab debe comenzar mediante la ejecución de los comandos: sumo si se quiere ejecutar el simulador sin visualización de vehículos o sumo-gui si se quiere ejecutar el simulador en modo de interfaz gráfica; especificando como parámetro la ruta donde se encuentra el archivo de configuración de la simulación de interés, mencionado en el paso anterior. Este requisito se logra por medio del comando *system* de Matlab, como lo muestra la figura 8.

```
%   Copyright 2013 Universidad Nacional de Colombia,
%   Politecnico Jaime Isaza Cadavid.
%   Authors: Andres Acosta, Jairo Espinosa, Jorge Espinosa.
%   $Id: traci_test2_clean.m 2 2013-12-21 21:39:57Z aacosta $

clear all
close all
clc

import traci.constants

system(['sumo-gui -c ' getenv('SUMO_HOME')...
    '\docs\tutorial\traci_tls\data\cross.sumocfg&']);
```

Figura 8: Ejecutar SUMO en desde Matlab

Opcionalmente se puede añadir el comando *import traci.constants* para facilitar la referencia a las constantes de TraCI, en caso de que se vayan a hacer suscripciones TraCI, que se explicarán más adelante. Así, por ejemplo, para hacer referencia al comando LAST_STEP_VEHICLE_NUMBER de las constantes TraCI, basta con escribir constants.LAST_STEP_VEHICLE_NUMBER envés de traci.constants.LAST_STEP_VEHICLE_NUMBER.

- **Paso 2: Inicializar la conexión**

  Luego de inicializar el servidor SUMO en el paso anterior, se procede a establecer la conexión, con la función traci.init, como se muestra en la figura 9. Si se configuró el servidor SUMO para utilizar el puerto 8813, esta función no necesita parámetros adicionales. Para utilizar parámetros adicionales, se recomienda leer la ayuda, digitando help traci.init en la ventana de comandos de Matlab.

```
% Initialize TraCI
traci.init();

traci.inductionloop.subscribe('0');
for i=1:length(steps)

    % Perform a simulation step (one second)
    traci.simulationStep();

    programPointer = min(programPointer+1, length(PROGRAM));

```

Figura 9: Inicialización de la conexión con el servidor SUMO

- **Paso 3: Desarrollar la aplicación**

  Normalmente, las aplicaciones de TraCI4Matlab incluyen un ciclo principal, en el que se ejecutan los pasos de tiempo de la simulación por medio del comando traci.simulationStep. Dentro de este ciclo, se accede a los valores y se modifican los atributos de los objetos encontrados en la simulación, que se agrupan en trece dominios, a saber: gui, lane, poi, simulation, trafficlights, vehicletype, edge, inductionloop, junction, multientryexit, polygon, route, y vehicle. Esto se logra por medio de comandos con la siguiente estructura general: traci.dominio.get/set_wrapper(), en donde dominio toma

uno de los valores listados anteriormente y get/set_wrapper() son las funciones para acceder (*get*) o modificar (*set*) los atributos del objeto de interés.
Si por ejemplo, se quiere conocer el valor de la velocidad del vehículo con ID *veh_1* en el paso de tiempo actual, se procede con el comando current_speed_veh1 = traci.vehicle.getSpeed('veh_1'). Para obtener una lista de todos los comandos de un dominio específico, digitar help traci.dominio en la ventana de comandos de Matlab, donde dominio puede tomar cualquier valor de los dominios listados anteriormente. La figura 10 muestra el ejemplo para el caso del dominio lane. Se puede obtener la ayuda de una función particular haciendo clic sobre ella.

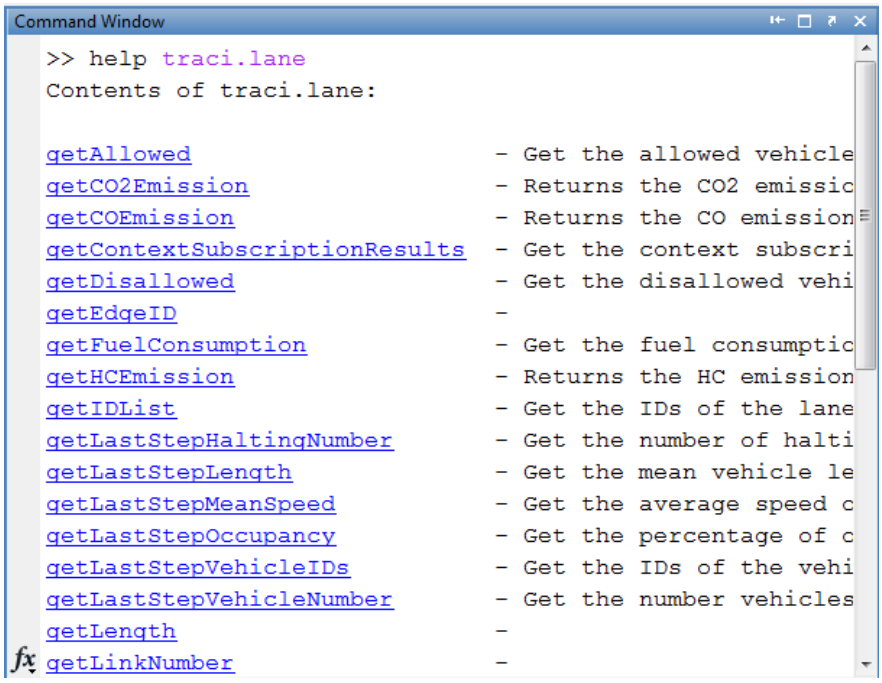

Figura 10: Obtener una lista de las funciones asociadas a un dominio de objetos SUMO.

El ciclo principal de la simulación se puede ejecutar hasta un tiempo fijo, o hasta que los todos los vehículos de la simulación hayan llegado a sus destinos, para lo cual se utiliza la función traci.simulation.getMinExpectedNumber, como lo muestra la figura 11.

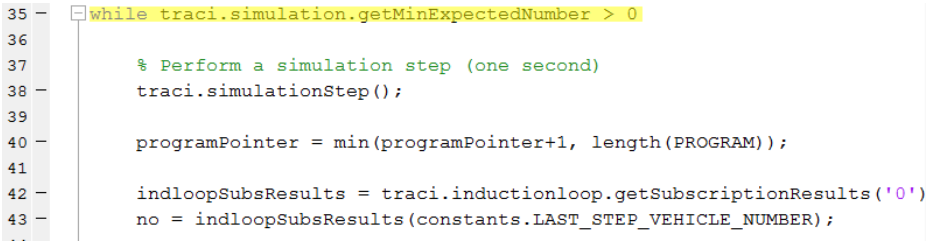

Figura 11: Condición para ejecutar la simulación hasta que todos los vehículos hayan llegado a sus destinos

TraCI4Matlab incorpora funciones para realizar suscripciones TraCI. Una suscripción TraCI permite retribuir varios atributos de un objeto SUMO por medio de un solo comando. Para

utilizar las suscripciones TraCI, es necesario conocer las constantes TraCI que contienen los códigos de diferentes atributos a los cuales se puede realizar la suscripción, que se pueden encontrar digitando edit traci.constants en la ventana de comandos de Matlab, en el campo "*VARIABLE TYPES*". Supóngase, por ejemplo, que se quiere realizar una suscripción TraCI para acceder a los valores de los atributos LAST_STEP_VEHICLE_NUMBER y LAST_STEP_MEAN_SPEED del detector tipo *induction loop* con ID '0'. En este caso, se procede con el comando mostrado en la figura 12. Nótese que se ha utilizado el comando import traci.constants al principio del *script*, como se explicó en el paso 1.

```
33
34 -    traci.inductionloop.subscribe('0',{constants.LAST_STEP_VEHICLE_NUMBER,..
35         constants.LAST_STEP_MEAN_SPEED});
```

Figura 12: Suscripciones TraCI

Ahora, para acceder a los valores a los cuales se realizó la suscripción, se utilizan los comandos mostrados en la figura 13, dentro del ciclo principal. Allí se puede ver que primero se almacenan todos los resultados en una variable que luego se indexa con las constantes de traci a las cuales se realizó la suscripción, para acceder a los valores de interés.

```
44 -    indloopSubsResults = traci.inductionloop.getSubscriptionResults('0');
45 -    no = indloopSubsResults(constants.LAST_STEP_VEHICLE_NUMBER);
46 -    lsms = indloopSubsResults(constants.LAST_STEP_MEAN_SPEED);
```

Figura 13: Acceso a los resultados de las suscripciones TraCI

- **Paso 4: Cerrar la conexión**

  Finalmente, se cierra la conexión con el servidor SUMO, como se muestra en la figura 14. Luego, se puede realizar post-procesamiento de los datos obtenidos con todas las ventajas de Matlab.

```
65 -    end
66
67 -    traci.close()
68
69 -    plot(steps, WElaneoccupancy)
70 -    hold;
71 -    plot(steps, NSlaneoccupancy, 'r')
72 -    legend('WE lane occupancy', 'NS lane occupancy')
73 -    title('Lane occupancy vs time')
74 -    xlabel('t (seconds)')
75 -    ylabel('number of vehicles')
```

Figura 14: Cerrar la conexión con el servidor