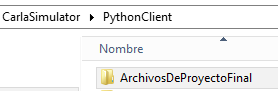
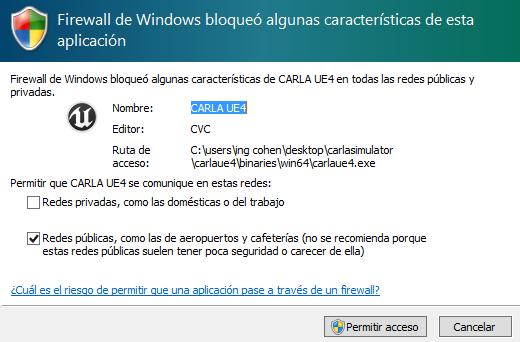
***Proyecto Final – Control Digital – Controlador Lateral y Longitudinal de un Vehículo Automanejado***

* Leer la sección de controladores
* Ver la sección de prerrequisitos de hardware y software.
* Active su ambiente de trabajo si utiliza Anaconda
* Descomprimir los archivos de proyecto final en PythonClient

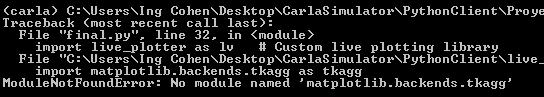


* Seguido lance Carla con el siguiente comando
  + CarlaUE4.exe /Game/Maps/RaceTrack -windowed -carla-server -benchmark -fps=30
* Si tiene una alerta de seguridad acepte incluir la excepción en el firewall

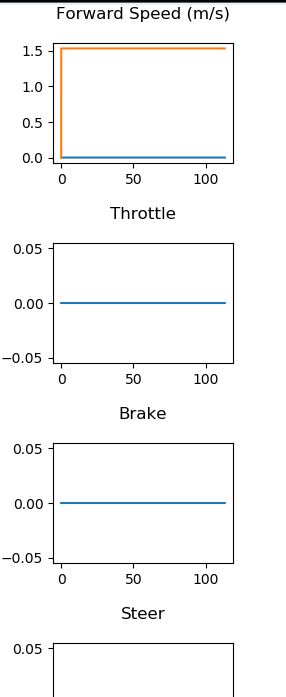
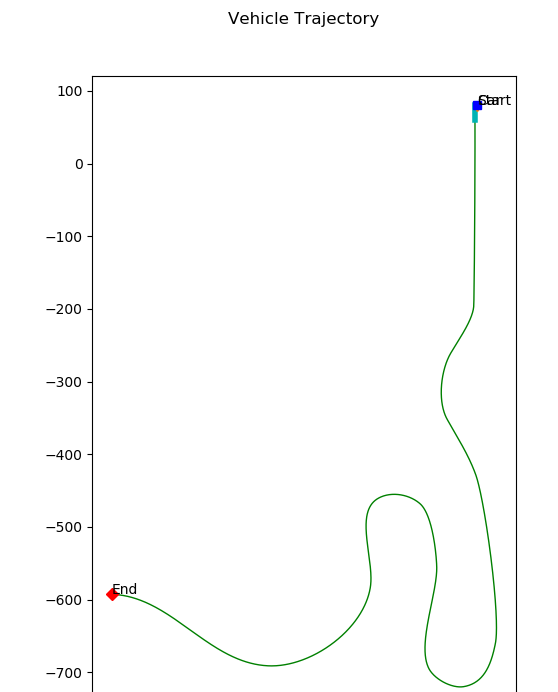


* Probar en control manual el siguiente ambiente de trabajo usando su ambiente de trabajo
  + En una pantalla correr el siguiente comando dentro de la carpeta CarlaSimulator
    - CarlaUE4.exe /Game/Maps/RaceTrack -windowed -carla-server
  + En otra ventana correr el siguiente comando dentro del a carpeta PythonClient
    - python manual\_control.py
  + Al principio Carla estará congelado hasta que se corra el comando último anterior luego podrá controlar manualmente.

Ejecutar el módulo de la aplicación a desarrollar.

* En la primera pantalla abrir el ambiente de desarrollo y ejecutar sobre la carpeta CarlaSimulator
  + CarlaUE4.exe /Game/Maps/RaceTrack -windowed -carla-server
* En la segunda pantalla abrir el ambiente de desarrollo y ejectuar sobre la carpeta de proyecto final
  + python final.py
* Si recibe algún error como el inferior ejecutar dentro del entorno de desarrollo
  + 
  + pip install matplotlib==3.0.0
  + ejecute nuevamente el archivo
* Debe presentarse la siguiente ventana



**Asignación de Control Vehicular**

En este proyecto ud. implementar controlarpara el simuldor CARLA. Su objetivo es controlar el vehículo a seguir la via por medio de unos puntos preseleccionados. El vehículo necesitará acogerse a estos puntos y ciertas velocidades así que se requerirá taoto control lateral como longitudinal.

Para realizar esto debe seguir los pasos de prerrequisito y también de las páginas anteriores.

Ud debe editar el archivo controller2d.py que viene en CARLA encontrado en la carpeta de proyecto final. El archivo contiene el controlar del vehpiculo, ud implementará la función update\_controls.

Aclaramos que todas las unidades están en SI (metros, segundos, radianes), y el simulador utiliza el sistemas de coordenada de mano izquierda, es decir que opera entre x, y y la rotación en el eje z llamada en inglés ‘yaw’.

Los waypoints son un python list en la forma [x, y, v] donde x y y son la posición y v es la velocidad de esta posicion respectivamente. Un ejemplo para acceder al punto 5 de y sería:

Waypoints[4][1] # Rcuerde que en python los índices comienzan de 0.

Los waypoints se actualizan cada paso en simulación así que no asuma que estos no cambian. Los waypoints fueron linelmente interpolados (de localización a velocidad) subset de los waypoints principales.

|  |  |  |
| --- | --- | --- |
| **Descripción** | **Nombre de Variable** | **Unidades** |
| Ubicación del Vehículo | [x, y] | [metros, metros] |
| Orientación del Vehículo | yaw | Radianes |
| Velocidad del Vehículo | v | Metros por segundo |
| Tiempo de Simulación | t | Segundo |
| Velocidad deseada | v\_desired | Metros por segundo |
| Waypoints a seguir (x, y, v) | wayponts | [metros, metros, m/s] |

Utilizando esta formulación en su controlador ud tendrá la salida del vehículo de aceleración, dirección y freno como sigue:

|  |  |  |
| --- | --- | --- |
| **Descripción** | **Nombre de Variable** | **Límites** |
| Aceleración | throttle\_output | 0 a 1 (en porcentaje) |
| Dirección | steer\_output | -1.22 a 1.22 (en radianes) |
| Freno | brake\_output | 0 a 1 (en porcentaje) |

Luego de desarrollado el controlador:

* En la primera pantalla abrir el ambiente de desarrollo y ejecutar sobre la carpeta CarlaSimulator
  + CarlaUE4.exe /Game/Maps/RaceTrack -windowed -carla-server
* En la segunda pantalla abrir el ambiente de desarrollo y ejectuar sobre la carpeta de proyecto final
  + python final.py

Se abrirar las ventanas mencionadas anteriormente. Estas ventanas, en especial la gráfica dará respuesta de la simulación deseada de velocidad deseada y actual. Esto generará una retroalimentación en el simulador CARLA por la cual ud no tiene que preocuparse enviando los comandos deseados.

Las velocidades esperadas de aceleración, freno y dirección son de 0 a 1, 0 a 1 y -1 a 1.

Si su simulación corre muy lentamente ud puede incrmenetar el período de gráfico y refresco modificando el archivo options.cfg de la carpeta de proyecto final.

|  |  |  |
| --- | --- | --- |
| **Parámetro** | **Descripción** | **Valor** |
| Live\_plotting | Habilitar o deshablitar gráfico | True/False |
| Live\_plotting\_period | Período (en segundos) del cual el gráfico se refresca. Ajustado a 0 para refrescar en cada simulación | segundos |

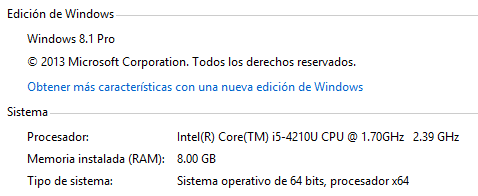
La aplicación de cliente se cerrará a 250 segundos de inactivodad de simulación. Al completarse el circuito de automóvil de darse un control correcto del auto generará un archivo de trayectorias en controller\_output/trajectory.txt en la carpeta de proyecto final

**PRE-REQUISITOS DE HARDWARE**

Lo siguiente es lo recomendado, pero no es obligatorio, simpelemnte es para mejorar el rendimiento.

* AMD QuadCore 2.5GHz o superior
* NVIDIA GEForce 4xx GTX o AMD Radeon 6xxxx HD
* 8GB de RAM
* 10GB de espacio en disco

Este es el ejemplo del setup de la máquina utilizada, es de bajos recursos y ejecuta el emulador.

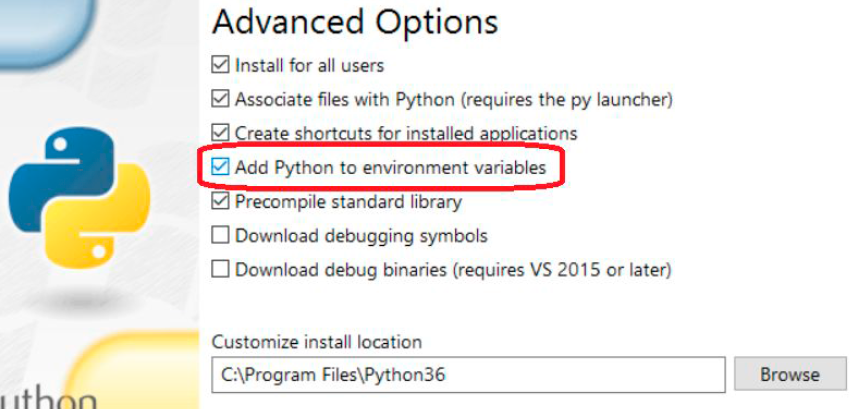


**PREREQUISITOS DE SOFTWARE**

* Windows 7 o superior (64 bits)
* Permitir los accesos del firewall a los puertos 2001 2002 tanto TCP como UDP

Para instalar CARLA necesita python 3.6, puede instalar Anaconda o instalar python directamente.

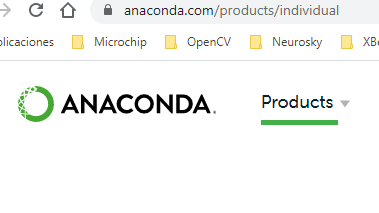
Asegurese de haber añadido python a las variables de entorno para realizar una correcta instalación, elija alguna de las dos siguientes instalaciones



*INSTALACIÓN DE PYTHON*

* Abrir luego una ventana de comandos de windows y verificar que python esté instalado
  + Python –version
* Esta ventana deberá haber retornado Python 3.X como versión actual instalada
* Intente instalar una librería
  + Pip install numpy

*INSTALACIÓN DE ANACONDA*



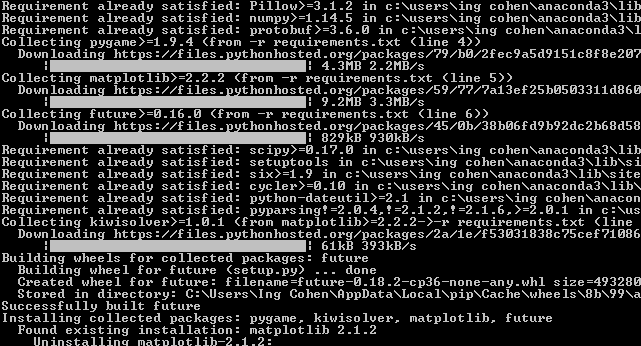
* Descargar e instalar Anaconda del sitio web e instalar
* Abra una ventana de comandos y crear el ambiente de trabajo
  + conda créate –n carla python=3.6
* Active el ambiente de trabajo
  + activte carla
* Intente instalar una librería por ejemplo
  + pip install jupyter notebook
* Desactive el ambiente de trabajo
  + Deactivate carla

INSTALACIÓN DE DEPENDENCIAS

Si utiliza Anaconda active su ambiente de desarrollo

Si utilizó la instalación común de python siga como mostramos en los pasos siguientes

Primeramente descomprima el zip… esta guía asume que tenemos descomprimido **CarlaUEWindows.zip** en el escritorio en la carpeta **CarlaSimulator**.

* Navegue a la carpeta CarlaSimulator
* Instale las dependencias
  + pip install –r requirements.txt
* Arranque el simulador con el siguiente comando
  + CarlaUE4.exe -windowed -carla-no-networking
* Puede ajustar el tamaño con los argumentos -ResX= -ResY=, por ejemplo 640x480
  + CarlaUE4.exe -windowed -carla-no-networking -ResX=640 -ResY=480



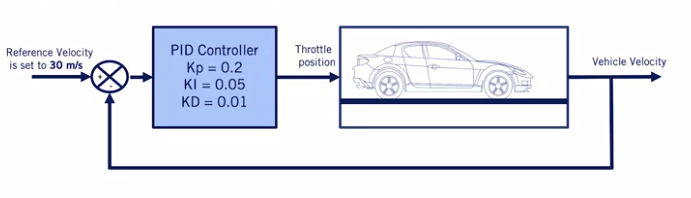
Para moverse sobre el simulador utiliza las teclas:

* A, S, D, W para mover el vehículo sobre el mapa.
* Use la tecla Q para conmutar entre reversa o adelante..
* Use la tecla P para activar el modo de piloto automatico (carro automanejado), nuevamente P deshabilitará el modo y volverá a manual, solo funciona en Town01 y Town02
* Referires a la documentación de commandos de Carla en la web para ver todos los commandos.

**EJEMPLO DE CONTROLADORES**

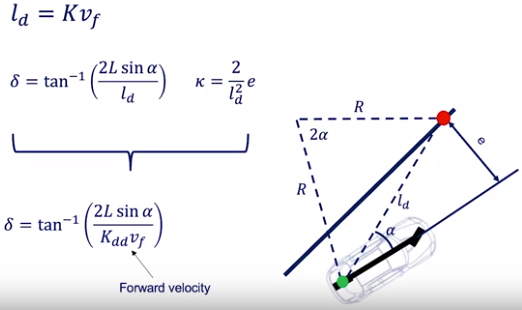
*Controlador Longitudinal*

* Puede implementar un control longitudinal de velocidad utilizando un PID
* Para mas información puede referirse a:
  + [**https://link.springer.com/chapter/10.1007%2F0-387-28823-6\_5**](https://www.springer.com/cda/content/document/cda_downloaddocument/9781461414322-c1.pdf?SGWID=0-0-45-1265143-p174267791)



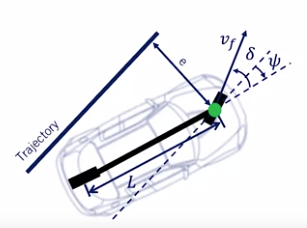
**Control Lateral**

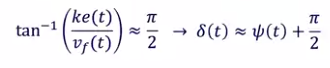
*Control Lateral Opción 1*



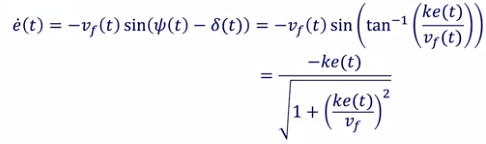
* Como observará se tiene un punto o varios puntos delanteros y se busca corregir el ángulo alpha que es el que lleva en medio de la pista.
* Igualmente puede utilizar otro PID para resolver este punto
* Ld = look-ahead distance
* K es una constante a ajustar, positiva y mayor a cero.
* e es el error de la posicion actual del vehículo y el punto de pista a seguir
  + e =(x,y)\_vehiculo – (x,y)\_deseada
* A este controlador se le conoce como ‘Pure Pursuit’
* Puede leer más aquí:
  + <https://www.ri.cmu.edu/pub_files/2009/2/Automatic_Steering_Methods_for_Autonomous_Automobile_Path_Tracking.pdf>.

*Control Lateral Opción 2*

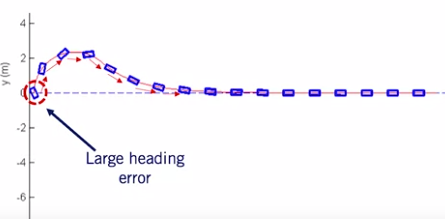


Para un error positivo largo truncar a: 

El error dinámico es: (nota: recuerde que e punto = e / dt)



Para errores muy pequeños: 



* Puede leer más de este controlador en
  + <https://www.ri.cmu.edu/pub_files/2009/2/Automatic_Steering_Methods_for_Autonomous_Automobile_Path_Tracking.pdf>.