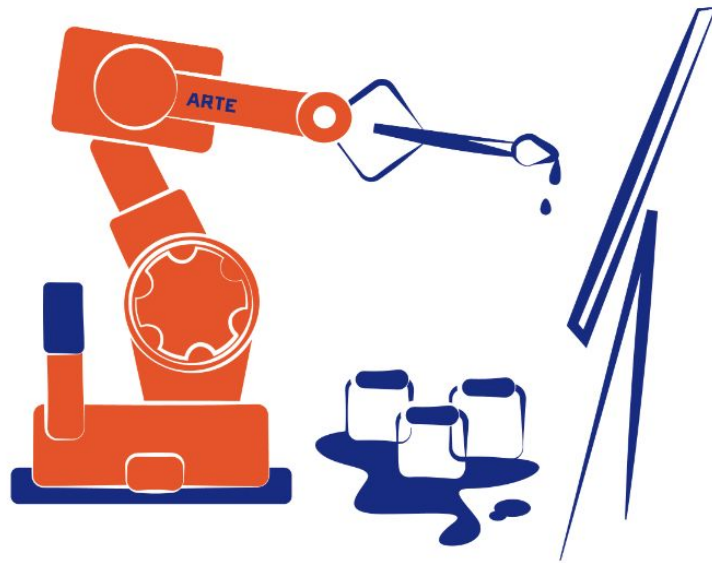


ARTE



A ROBOTICS TOOLBOX FOR EDUCATION

SESIÓN 0: INTRODUCCIÓN A ARTE

Arturo Gil Aparicio

arturo.gil@umh.es



ÍNDICE

1. **Instalación.**
2. **Cargar robots.**
3. **Uso de la GUI (Graphical User Interface)**

1 Instalación de la librería

Instalar la librería en Matlab es sencillo. Simplemente, sigue los siguientes pasos:

- a) Descarga y descomprime la librería desde <http://arvc.umh.es/arte>
- b) Copia el directorio llamado `arte` a tu disco duro, por ejemplo en el escritorio de tu PC.
- c) A continuación, navega hasta tu directorio dentro de Matlab. Por ejemplo, si copiaste la librería al Escritorio, deberías tener algo parecido a esto (Linux):

```
>> pwd

ans =

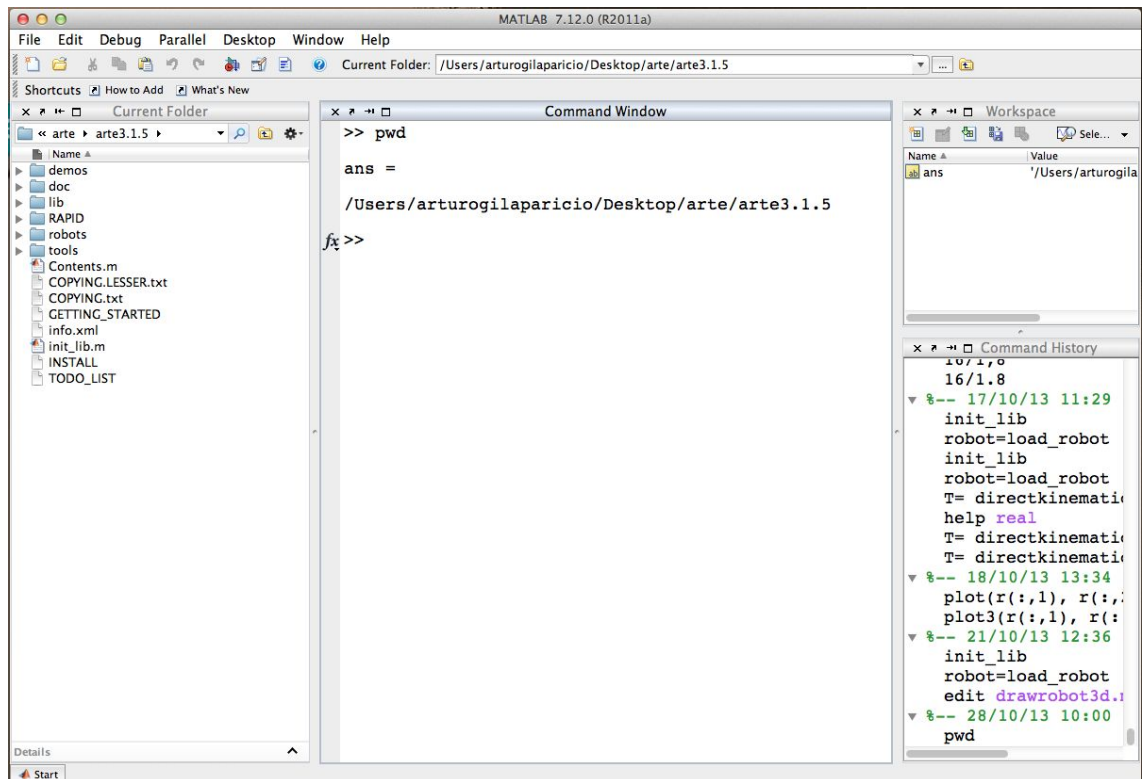
/Users/arturo/Desktop/arte
```

Si usas Windows, puedes ver algo como esto:

```
>> pwd

ans =

C:\Documents and Settings\arturo\Desktop\arte
```



d) Seguidamente, en la línea de comandos de Matlab escribe lo siguiente:

```
>> init_lib

% ARTE (A Robotics Toolbox for Education)
% Copyright (C) 2012 Arturo Gil Aparicio, arturo.gil@umh.es
% http://arvc.umh.es/arte
%
% This program is free software: you can redistribute it and/or modify
% it under the terms of the GNU Lesser General Public License as published by
% the Free Software Foundation, either version 3 of the License, or
% any later version.
%
% This program is distributed in the hope that it will be useful,
% but WITHOUT ANY WARRANTY; without even the implied warranty of
% MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
% GNU Lesser General Public License for more details.
%
% You should have received a copy of the GNU Lesser General Public License
% along with this program. If not, see <http://www.gnu.org/licenses/>.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% To begin with, try loading a robot:
% >> robot = load_robot('ABB','IRB140');
%
% Next, draw it on its zero position:
% >> drawrobot3d(robot,[0 0 0 0 0 0])
%
% Next, try a different pose:
% >> drawrobot3d(robot,[0 pi/2 -pi/2 0 0 0])
%
% Finally, use the teach pendant to move the robot:
% >> teach
```

Puedes encontrar una introducción a la librería aquí: An introduction to the library can be viewed here.

<http://www.youtube.com/watch?v=s8QQydJ9PwI&list=PLClKgnzRFYe72qDYmj5CRpR9ICNnQehup&index=18>

2 Cargar un robot

Toda la librería está basada en la estructura de datos que llamaremos robot. Esta estructura de datos de Matlab guarda todos los parámetros necesarios para definir un robot. Esta estructura de datos se guarda en el fichero parameters.m de cada robot que se encuentra en estos momentos definido en la librería. Es sencillo añadir más robots a la librería. Para cargar cualquier robot a la librería podéis hacer:

```
>> robot=load_robot('ABB','IRB140')

ans =

/Users/arturogilaparicio/Desktop/arte/arte3.1.5/robots/ABB/IRB140

Reading link 0
/Users/arturogilaparicio/Desktop/arte/arte3.1.5/robots/ABB/IRB140/link0.stl
EndOfFile found...
Reading link 1
/Users/arturogilaparicio/Desktop/arte/arte3.1.5/robots/ABB/IRB140/link1.stl
EndOfFile found...
Reading link 2
/Users/arturogilaparicio/Desktop/arte/arte3.1.5/robots/ABB/IRB140/link2.stl
EndOfFile found...
Reading link 3
/Users/arturogilaparicio/Desktop/arte/arte3.1.5/robots/ABB/IRB140/link3.stl
EndOfFile found...
Reading link 4
/Users/arturogilaparicio/Desktop/arte/arte3.1.5/robots/ABB/IRB140/link4.stl
EndOfFile found...
Reading link 5
/Users/arturogilaparicio/Desktop/arte/arte3.1.5/robots/ABB/IRB140/link5.stl
EndOfFile found...
Reading link 6
/Users/arturogilaparicio/Desktop/arte/arte3.1.5/robots/ABB/IRB140/link6.stl
EndOfFile found...
robot =

          name: 'ABB_IRB140_M2000'
           DH: [1x1 struct]
            J: []
inversekinematic_fn: [1x33 char]
 directkinematic_fn: [1x25 char]
           DOF: 6
          kind: 'RRRRRR'
    maxangle: [6x2 double]
      velmax: [6x1 double]
    accelmax: [6x1 double]
 linear_velmax: 2.5000
           T0: [4x4 double]
```

```

        debug: 0
        q: [6x1 double]
        qd: [6x1 double]
        qdd: [6x1 double]
        time: []
        q_vector: []
        qd_vector: []
        qdd_vector: []
        last_target: [4x4 double]
        last_zone_data: 'fine'
        tool0: [1x19 double]
        wobj0: []
        tool_activated: 0
        path: [1x65 char]
        graphical: [1x1 struct]
        axis: [1x6 double]
        has_dynamics: 1
        dynamics: [1x1 struct]
        motors: [1x1 struct]

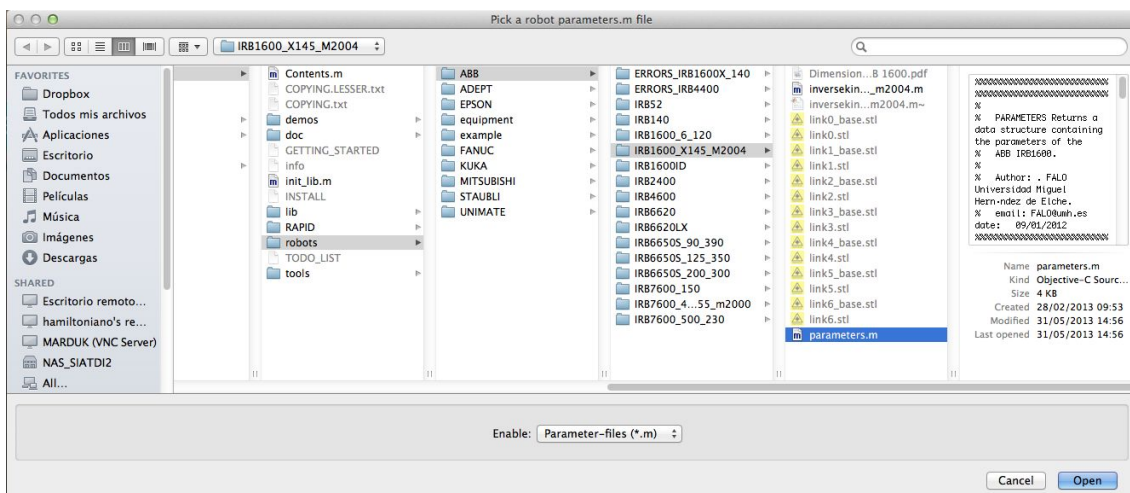
```

Esto carga el robot IRB140 del fabricante ABB, guardado en el directorio robots/ABB.

La función `load_robot` también puede ser llamada sin parámetros:

```
>> robot=load_robot
```

De esta manera, es posible navegar a cualquiera de los directorios de robots de los fabricantes y hacer click en el fichero **parameters.m**. Si todo se ha realizado correctamente, debería aparecer una figura con el robot que se acaba de cargar en la librería.



```
>> robot=load_robot
```

```
ans =
```

```

/Users/arturogilaparicio/Desktop/arte/arte3.1.5/robots/ABB/IRB1600_X145_M2004

Reading link 0

/Users/arturogilaparicio/Desktop/arte/arte3.1.5/robots/ABB/IRB1600_X145_M2004/
link0.stl
EndOfFile found...
Reading link 1

/Users/arturogilaparicio/Desktop/arte/arte3.1.5/robots/ABB/IRB1600_X145_M2004/
link1.stl
EndOfFile found...
Reading link 2

/Users/arturogilaparicio/Desktop/arte/arte3.1.5/robots/ABB/IRB1600_X145_M2004/
link2.stl
EndOfFile found...
Reading link 3

/Users/arturogilaparicio/Desktop/arte/arte3.1.5/robots/ABB/IRB1600_X145_M2004/
link3.stl
EndOfFile found...
Reading link 4

/Users/arturogilaparicio/Desktop/arte/arte3.1.5/robots/ABB/IRB1600_X145_M2004/
link4.stl
EndOfFile found...
Reading link 5

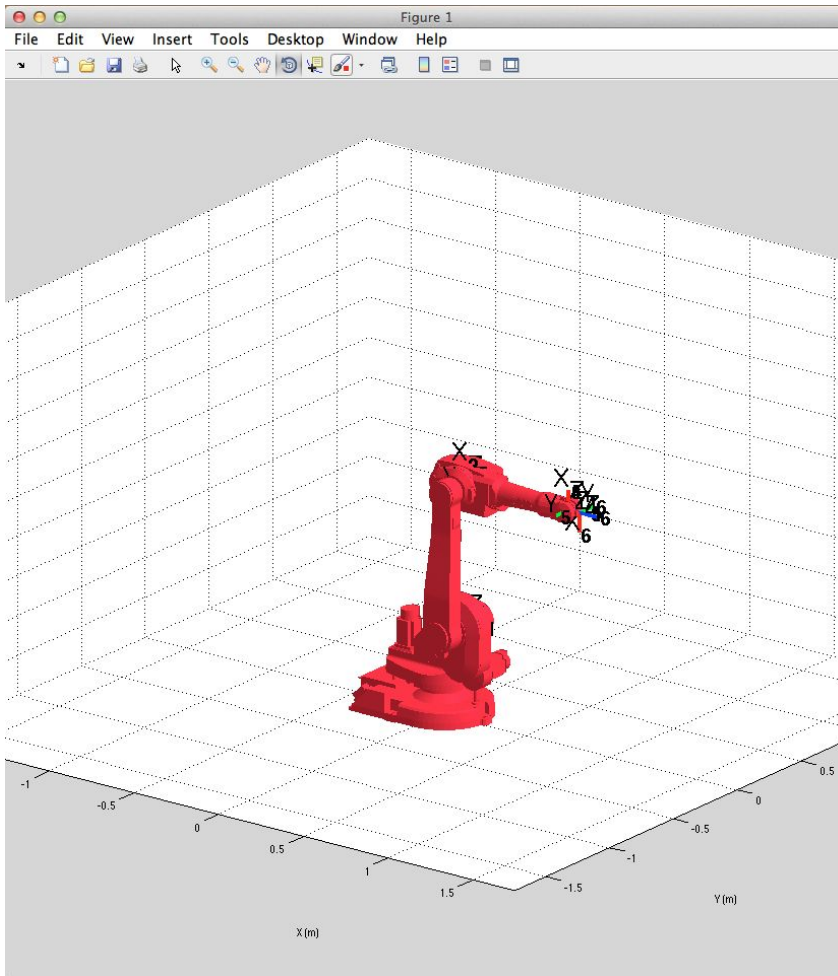
/Users/arturogilaparicio/Desktop/arte/arte3.1.5/robots/ABB/IRB1600_X145_M2004/
link5.stl
EndOfFile found...
Reading link 6

/Users/arturogilaparicio/Desktop/arte/arte3.1.5/robots/ABB/IRB1600_X145_M2004/
link6.stl
EndOfFile found...
robot =

        name: 'abb_irb1600_X145_m2004'
        DH: [1x1 struct]
        J: []
inversekinematic_fn: [1x49 char]
        DOF: 6
        kind: 'RRRRRR'
        maxangle: [6x2 double]
        velmax: [6x1 double]
        accelmax: [6x1 double]
        linear_velmax: 2.5000
        T0: [4x4 double]
        debug: 0
        q: [6x1 double]
        qd: [6x1 double]
        qdd: [6x1 double]
        time: []
        q_vector: []
        qd_vector: []
        qdd_vector: []
        last_target: [4x4 double]
        last_zone_data: 'fine'
        tool0: [1x19 double]
        wobj0: []
        tool_activated: 0
        path: [1x77 char]
        graphical: [1x1 struct]
        axis: [1x6 double]

```

```
has_dynamics: 1
dynamics: [1x1 struct]
motors: [1x1 struct]
```



Notad que, después de hacer:

```
>> robot=load_robot
```

y seleccionar un robot en la librería, los parámetros del robot se encuentran almacenados en la variable `robot`. Podemos guardar este tipo de dato en cualquier otra variable con otro nombre:

```
>> abb=load_robot('ABB','IRB140')
```

Después de esto, es posible usar la variable `abb` para cualquier llamada a las funciones de la librería. Muchas de estas funciones toman como parámetro una variable de tipo `robot`.

```
>> directkinematic(abb, [0 0 0 0 0 0])
>> drawrobot3d(abb)
```

Sin embargo, la variable robot está reservada y es muy útil en la librería. En concreto, es una variable declarada global, con lo que puede ser visible desde cualquier script de la librería. Por ejemplo, la aplicación gráfica teach accede a la variable robot:

```
>> robot=load_robot
>> teach
```

3 Uso de la aplicación teach

En la línea de comandos de Matlab, escribe:

```
>> robot=load_robot
>> teach
```

La siguiente aplicación gráfica debería aparecer (puede que tenga una apariencia ligeramente diferente: Figure 1):

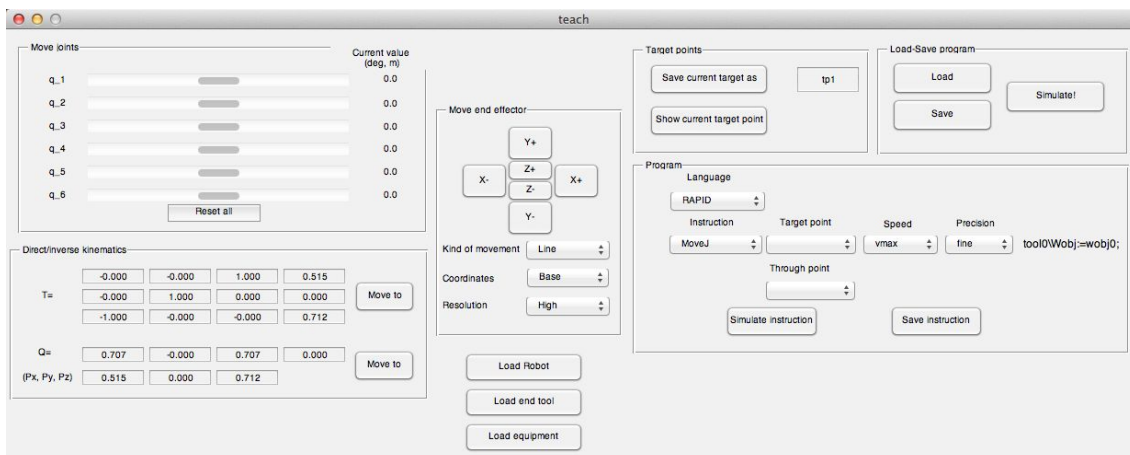


Figure 1

Esta aplicación gráfica cubre diferentes cuestiones, en concreto:

- La cinemática directa del robot.
- La cinemática inversa.
- Cuaternios.
- Sistemas de referencia.

