PGT - A path generation toolbox for Matlab
(v0.1)

Maria Nyström, Mikael Norrlöf
Division of Automatic Control
Department of Electrical Engineering
Linköpings universitet, SE-581 83 Linköping, Sweden
WWW: http://www.control.isy.liu.se
E-mail: nystrom@isy.liu.se, mino@isy.liu.se

5th September 2003

Report no.: LiTH-ISY-R-2542
Submitted to

Technical reports from the Control & Communication group in Linköping are available at http://www.control.isy.liu.se/publications.
Abstract

This user manual introduces the basic ideas of the PGT - Path Generation Toolbox for Matlab. The main features of the toolbox is to give the user the possibility to 1) build paths in Cartesian (3DOF) space, 2) transform the Cartesian paths into jointspace. All the most important functions are explained and examples of how to use them are also included.

Keywords: zone interpolation, cartesian path, joint path, cubic splines, MATLAB functions
1 Introduction

This toolbox is intended for geometrical path generation with application industrial robots. A path consists of a number of path primitives, called segments. For further information segments and how the segments relate to sections see e.g. [1] and [2]. Each section and each path is here represented by a data structure and a number of functions for path generation that operates on these have been developed.

2 Data structures

Each section is represented as a data structure with the following fields

$$\text{sec} = pseg: \quad cseg: \quad zinfo: \quad \text{range:} \quad jsp:$$

The Cartesian representation of the two segments that builds up the section, is stored in the fields $pseg$ (previous segment) and $cseg$ (current segment). Information about the zone path generation is stored in $zinfo$. Each section has an index. Four different values of this index are stored in the $\text{range}$ field: the index value at the beginning of the section, the index value in the middle of the zone, the index value at the end of the zone and finally, the index value at the end of the section. An index value at a certain point is equal to the length of the section from the beginning of the section to that point. If the section has been transformed to a path in joint space, the joint space path is stored in the $jsp$ (joint space path) field.

2.1 The segment field

Information about the segments in Cartesian coordinates is stored in a data structure with the following fields

$$\text{sec}.pseg \quad \text{or} \quad \text{sec}.cseg$$

ans =

$$\begin{align*}
\text{type:} \\
\text{descr:} \\
\text{range:} \\
\text{tdi:}
\end{align*}$$
The segment type is stored as a two character string in the `type` field. Predefined types are ‘sp’, which represents a spline, ‘ci’, which is a circular arc, and ‘em’, which represents an “empty” segment containing just one point. A mathematical description of the segment can be found in `descr`. This field will have different appearances for different segment types. The index value for each segment at the end of the first zone and at the beginning of the second zone, is stored in the `range` field. Here, the index of a point represents the length of the segment from the beginning of the segment to that point. Finally time dependent information for the segment is stored in the `tdi` field.

### 2.2 The zone information field

Necessary zone information is stored in a data structure named `zinfo`, which has the following fields

```matlab
>> sec.zinfo
ans =
    segindex:
    ztype:
```

To be able to construct the zone, normalized indices, for the two points where the zone intersects the segments, are needed. These values are stored in the `segindex` field. The indices are calculated as \( \frac{\text{length to zone intersection point}}{\text{entire segment length}} \). The `ztype` field contains a number, which determines what interpolation method to use when creating the zone path. If for example `ztype = 1` a parabolic path is received.

### 2.3 Path data structure

Several sections can be gathered to form a complete path. A path data structure has the following fields

```matlab
>> rpath
rpath =
    descr:
    range:
```

`descr` is an array containing the section data structures. At the point where each section ends, the length of the path is stored in the `range` field, beginning with the value zero. `range` is consequently a vector of size “number of sections”+1.
2.4 Information for transformation to joint space

For the transformation from Cartesian coordinates to joint space, a number of variables must be given specific values, and a number of functions, for different types of calculations, must be specified. This information is gathered in a data structure stored in a separate file. The data structure must have the following fields:

- **robot**: a complete robot description.
- **brp**: number of breakpoints in each spline function.
- **maxerror**: maximum deviation of the desired Cartesian path from the actual robot path.
- **delta**: initial length between the breakpoints.
- **ikine**: function handle for inverse kinematic calculations. The result should be a [3x1] vector.
- **dq.method**: function handle to a function that differentiates the joint angles with respect to the index in a given point. The result should be a [3x1] vector.
- **dq.var**: Optional variable. Given as argument to the function specified by dq.method.
- **aidx.method**: function handle to a function that handles the adjustment of the length between the breakpoints. The result is the new length, i.e. a float.
- **aidx.var**: Optional variable. Given as argument to the function specified by aidx.method.

The robot description should be generated with Robotic Toolbox, see [3]. An example of the data structure can be found in the file infofcn.m.
3 Functions

Syntax
sec = emptysec(p);

Description
Creates an empty section with just one point p. Is used as psec in moveline and movecirc when creating the first section on a path.

Example
Creation of a start point with coordinates [1 0.02 0.35]

>> esec = emptysec([1 0.02 0.35])
esec =
  cseg: [1x1 struct]
  range: 0

Syntax
sec = moveline(psec,p,zone,f,tdi);

Description
Creates, in Cartesian coordinates, a line section connected to the section psec. The section ends at the point p, which is enclosed by a zone of radius zone. The zone radius must be smaller than half the length of each segment in the zone. What interpolation method to use when creating the zone path in the first zone, around the last point of psec, is defined by the number f. If psec is an empty section, f can be omitted and will then be given the value zero. Time dependent information for the line segment can be given with the parameter tdi.

Example
Creation of a line section where psec is the empty section created in the emptysec example above, and the end point is p = [0.7 0.2 0.4]. The radius of the zone around the point p is 0.1 meters. f is omitted while the section does not start with a zone.

>> lsec = moveline(esec,[0.7 0.2 0.4],0.1)
lsec =
  pseg: [1x1 struct]
  cseg: [1x1 struct]
  zinfo: [1x1 struct]
  range: [0 0 0 0.2534]
Syntax  \[ \text{sec} = \text{movecirc}(\text{psec}, p1, p2, \text{zone}, f); \]

Description Creates, in Cartesian coordinates, an arc section connected to the section \( \text{psec} \). \( p2 \) specifies the last point on the arc and \( p1 \) an intermediate point. \( p2 \) is enclosed by a zone of radius \( \text{zone} \). The zone radius must be smaller than half the length of each segment in the zone. The path in the first zone, around the last point of \( \text{psec} \), is defined by the number \( f \). If \( \text{psec} \) is an empty section, \( f \) can be omitted and will then be given the value zero. Time dependent information for the circle segment can be given with the parameter \( \text{tdi} \).

Example Creation of a circle section which is connected to the line section created in the \text{moveline} example. The radius of the zone around the point \( p2 \) is \( 0.1 \) meters. The interpolation method to use in the zone around the first point, defined by \( \text{psec} \), is given by \( f = 1 \). This gives a parabolic path. The radius of this zone was specified in the \text{moveline} example.

\[ \gg \ csec = \text{movecirc}(\text{lsec}, [0.85 \ 0.35 \ 0.45], ...
\quad [0.74 \ 0.49 \ 0.5], 0.1, 1) \]

csec =
  pseg: [1x1 struct]
  cseg: [1x1 struct]
  zinfo: [1x1 struct]
  range: [0 0.0605 0.1271 0.3621]

Syntax  \[ \text{rpath} = \text{makepath}(\text{sec}_1, \ldots, \text{sec}_n); \]

Description Creates a path with the sections \( \text{sec}_1, \ldots, \text{sec}_n \) where \( \text{sec}_1 \) is the first section of the path and \( \text{sec}_n \) the last. No check will be made to see whether the sections are connected to each other or not. Empty sections are not allowed.

Example Creation of a path with the sections created in the \text{moveline} and \text{movecirc} examples

\[ \gg \ rpath = \text{makepath}(\text{lsec}, \text{csec}) \]
rpath =
  descr: [1x2 struct]
  range: [0 0.2534 0.6155]
Syntax

`rpath = appendpath(rpath, sec);`

Description

A section given by `sec` is appended to the path `rpath`. No check will be made to see whether the path and the section are connected to each other or not.

Example

A section created as `sec = moveline(csec, [0.35 0.6 0.5], 0.1, 1)` is being connected to the path created in the `makepath` example

```matlab
>> arpath = appendpath(rpath, sec)
arpath =
    descr: [1x3 struct]
    range: [0 0.2534 0.6155 1.0210]
```

Syntax

`value = evalsec(sec, x, ac);`

Description

Evaluates the section `sec` at the point given by index `x`, which must satisfy `x ∈ sec.range`. The accuracy in the zone is given by `ac` which can be omitted. Default value is then $10^{-7}$.

Example

The section created in the `movecirc` example is evaluated at the point given by index 0.09 with accuracy $10^{-6}$ if within a zone.

```matlab
>> value = evalsec(csec, 0.09, 1e-6)
value =
    0.7645
    0.2122
    0.4036
```

Syntax

`value = evalpath(rpath, x, ac);`

Description

Evaluates the path `rpath` at the point given by index `x`, which must satisfy `x ∈ rpath.range`. The accuracy in the zone is given by `ac` which can be omitted. Default value is then $10^{-7}$.

Example

The path created in the `makepath` example is evaluated at the point given by index 0.42 with accuracy $10^{-6}$ if within a zone.

```matlab
>> value = evalpath(rpath, 0.42, 1e-6)
value =
    0.8202
    0.2618
    0.4198
```
Syntax: \texttt{plotsec(sec,f,ac);}  
Description: Plots the section \texttt{sec} in Cartesian coordinates. \texttt{ac}, which is an integer, relates to the number of points plotted in the figure. If \texttt{f} = 1, only the actual path of the robot will be drawn. If \texttt{f} = 2 the parts of the segments inside the zone will also be drawn, and if \texttt{f} = 3, only the segments will be drawn and not the actual path. \texttt{f} and \texttt{ac} can be omitted. Default value for \texttt{f} is 1 and for \texttt{ac} 50.
Example: The section created in the \texttt{movecirc} example is plotted.

\begin{verbatim}
>> plotsec(csec)
\end{verbatim}

![Figure 1: Function plotsec](image)

Syntax: \texttt{plotpath(rpath,f,ac);}  
Description: Plots the path \texttt{rpath} in Cartesian coordinates. \texttt{ac}, which is an integer, relates to the number of points plotted in the figure. If \texttt{f} = 1, only the actual path of the robot will be drawn. If \texttt{f} = 2 the parts of the segments inside the zone will also be drawn, and if \texttt{f} = 3, only the segments will be drawn and not the actual path. \texttt{f} and \texttt{ac} can be omitted. Default value for \texttt{f} is 1 and for \texttt{ac} 50.
Example: The path created in the \texttt{makepath} example is plotted with \texttt{f} = 2.

\begin{verbatim}
>> plotpath(rpath,2)
\end{verbatim}

![Figure 2: Function plotpath](image)

Syntax: \texttt{tdi = gettdi(rpath,x);}  
Description: The information stored in the \texttt{tdi} field of a segment, defined by index \texttt{x}, is returned. The path or section \texttt{rpath} contains the segment.
Syntax  \( \text{sec} = \text{getfsec}(\text{rpath}); \)
Description Returns the first section of the path \( \text{rpath}. \)

Syntax  \( \text{sec} = \text{getlsec}(\text{rpath}); \)
Description Returns the last section of the path \( \text{rpath}. \)

Syntax  \( \text{sec} = \text{getsec}(\text{rpath},i); \)
Description Returns the \( i \):th section of the path \( \text{rpath}. \)

Syntax  \( \text{value} = \text{dsec}(\text{sec},x,h); \)
Description Differentiates the section \( \text{sec} \), in Cartesian coordinates, with respect to the path index, i.e. \( \frac{d\text{P}(l)}{dl} \), and returns the first derivative evaluated at the point given by index \( x \). If inside a zone the derivative is calculated with finite different approximation otherwise an analytical solution is given. \( H \) is the distance between the points in the finite different approximation. \( H \) can be omitted and will be given the value \( 10^{-6} \).
Example The section created in the \text{movecirc} example is differentiated and evaluated in the point given by index 0.12.

\[
\begin{align*}
\gg \text{value} &= \text{dsec}(\text{csec},0.12) \\
\text{value} &= \\
&0.7764 \\
&0.5992 \\
&0.1957
\end{align*}
\]

Syntax  \( \text{value} = \text{d2sec}(\text{sec},x,h); \)
Description Calculates the second derivative of the section \( \text{sec} \), in Cartesian coordinates, with respect to the path index, i.e. \( \frac{d^2\text{P}(l)}{dl^2} \), and returns the second derivative evaluated at the point given by index \( x \). If inside a zone the derivative is calculated with finite different approximation otherwise an analytical solution is given. \( H \) is the distance between the points in the finite different approximation. \( H \) can be omitted and will be given the value \( 10^{-6} \).
Example The section created in the \text{movecirc} example is differentiated and evaluated in the point given by index 0.12.

\[
\begin{align*}
\gg \text{value} &= \text{d2sec}(\text{csec},0.12) \\
\text{value} &= \\
&-0.8178 \\
&0.8860 \\
&0.5775
\end{align*}
\]
Syntax

nrpath = cart2jsp(rpath, fcnhandle, 'fieldname_1', value_1, ..., 'fieldname_n', value_n);

Description

Transforms the Cartesian path or section rpath to cubic spline functions in joint space. Returns the path or section object nrpath, which is equal to rpath but with a description of the path in joint space attached. fcnhandle is a function handle to a function generating information necessary for the transformation, see also 2.4. If the information is to be changed temporarily, this can be done by specifying the field name that should be changed and then specifying the new value.

Example

The path created in the makepath example is to be transformed into joint space. Information about the transformation is received from the function handle @infofcn. The maximum deviation of the generated robot path from the desired Cartesian path, is temporarily set to $10^{-5}$ meters.

```
>> nrpath = cart2jsp(rpath, @infofcn, 'maxerror', 1e-5)
```

nrpath =

```
    descr: [1x2 struct]
    range: [0 0.2534 0.6155]
```

Syntax

plotbrp(rpath, f);

Description

Plots the breakpoint locations for the path rpath in joint space. If f is present the zone borders are marked with dotted lines.

Example

The breakpoints for the path created in the cart2jsp example is being plotted with the zone borders marked.

```
>> plotbrp(nrpath, 1)
```

![Figure 3: Function plotbrp](image-url)
Syntax: plotjder(rpath,d1,d2,f);

Description: Plots the first and second derivative of a path or section in joint space, which is stored in rpath. The plots are created at a number of calculation points, which are all at distance d1 from each other for the first derivative, and distance d2 for the second derivative. Default value for d1 and d2 is 0.005 meters. If f is present the zone borders are marked with dotted lines.

Example: The joint space path created in the cart2jsp example is differentiated and plotted with d2 = 0.001.

```matlab
>> plotjder(nrpath,[],0.001)
```

![Figure 4: Function plotjder](image)

Syntax: plotcder(r,rpath,d1,d2,f);

Description: Points on the path or section in joint space, given by rpath, is transformed back to Cartesian space using forward kinematics. The norm of the first and second derivative are then calculated with finite different approximation and plotted. r is the robot for which the path is constructed. d1 is the distance between the calculation points for the first derivative and d2 the same for the second derivative. Default value for d1 and d2 is 0.005 meters. If f is present the zone borders are marked with dotted lines.

Example: The joint space path created in the cart2jsp example is plotted with plotcder with the zone borders marked. The robot is generated with the function r3fcn.

```matlab
>> plotcder(robot,nrpath,[],[],1)
```
Syntax
plotce(r,rpath,d,f);

Description Points on the path or section in joint space, given by rpath are transformed to Cartesian coordinates and the compared to the desired Cartesian path or section, also given by rpath. The norm of the error are then plotted. r is the robot for which the path is constructed. d is the distance between the calculation points and if f is present the zone borders are marked with dotted lines.

Example The joint space path created in the cart2jsp example is compared to the desired Cartesian path with d = 0.001 and the zone borders marked. The robot is generated with the function r3fcn.

>> plotce(robot,nrpath,0.001,1)
3.1 Demo functions

Syntax
\[ \text{r3 = r3fcn; } \]

Description
Uses Robotic Toolbox, [3], to construct a three-link robot.

Syntax
\[ \text{infofcn; } \]

Description
Generates an example of the data structure described in Section 2.4.

3.2 Functions for use in cart2jsp

Syntax
\[ \text{theta = ikina(r,T); } \]

Description
Calculates the inverse kinematics using geometric inspection for a three-link robot \( r \) defined by the function \( r3fcn \). For each point defined by the homogeneous transformation matrix \( T \) four different solutions are given. At the end of the program one of these is returned. With that, \( \text{theta} \) is a \( [4 \times 3] \) matrix.

Syntax
\[ \text{dq = fda(r,sec,q,x,ST); } \]

Description
Approximates the derivative of the joint space path \( q(l_c) \) with respect to the path index using finite different approximation, i.e. \( \frac{q(l_c+h) - q(l_c)}{h} \). \( r \) is the robot for which the path is constructed. \( \text{sec} \) is the section in Cartesian coordinates, \( \text{q} \) is the joint angles where the derivative is to be evaluated, i.e. the value of \( q(l_c) \) in the equation above, and \( x \) is the value of \( l_c \) at that point. \( \text{ST} \) has two fields: \( \text{ST.h} \) and \( \text{ST.@fcnhandle} \). \( h \) is the distance between the two points in the finite different approximation and this value must be smaller than the maximum allowed deviation error. \( \text{@fcnhandle} \) is a function handle to a function for inverse kinematic calculations. Returns a \( [3 \times 1] \) vector.

Syntax
\[ \text{dq = diffinvjac(r,sec,q,x,ST); } \]

Description
Differentiates the joint space path \( q(l_c) \) with respect to the path index using \( \frac{dq(l_c)}{dl_c} = J^{-1}_v \frac{dp(l_c)}{dl_c} \) where \( J_v \) is the Jacobian. \( r \) is the robot for which the path is constructed. \( \text{sec} \) is the section in Cartesian coordinates, \( \text{q} \) is the joint angles where the derivative is to be evaluated and \( x \) is the value of \( l_c \) at that point. The variable \( \text{ST} \) is not used. Returns a \( [3 \times 1] \) vector.
Syntax

\[ value = \text{varic}(\text{delta,er,maxer,ST}); \]

Description
Adjusts the length between the breakpoints in a spline. \text{delta} is the current length between the breakpoints. \text{er} is the current deviation error and \text{maxer} is the maximum allowed deviation error. The variable \text{ST} contains two values. \text{ST}(1) is the smallest value that the current length will be multiplied with and is used if the step length is to be increased. \text{ST}(2) is the smallest value that the current length will be reduced with and is used if the step length is to be reduced. Returns a float.

Syntax

\[ value = \text{fixic}(\text{delta,er,maxer}); \]

Description
Adjust the length between the breakpoints in the spline with a fix set of adjustment factors. \text{delta} is the current length between the breakpoints. \text{er} is the current deviation error and \text{maxer} is the maximum allowed deviation error. Returns a float.

3.3 How to add new zone paths

The zone path is defined by a number when the section is created. This number then determines how the zone path is generated. New methods for generating zone paths can be added to the existing ones. This is done in the file \text{czindex.m}. The new method is given an integer value and the index calculations are inserted into the file. For more information about how the zone paths is created see [1] and [2].

References


PGT - A path generation toolbox for Matlab (v0.1)

Maria Nyström, Mikael Norrlöf

This users manual introduces the basic ideas of the PGT - Path Generation Toolbox for Matlab. The main features of the toolbox is to give the user the possibility to 1) build paths in Cartesian (3DOF) space, 2) transform the Cartesian paths into jointspace. All the most important functions are explained and examples of how to use them are also included.