Exercise 33: Using a Round-Robin realtime multi-tasking system

2017-03-05: Tim C. Lueth, MIMED - Technische Universität München, Germany (URL: http://www.mimed.de)

Contents

- Summary of the already explained aspects of the VLFL-Toolbox
- Motivation for this tutorial
- List of supported functions
- Intended use of RRrun or RRshell (both are the same)
- 1. Starting the shell
- 2. Adding realtime tasks and define the stop time
- 3. Adding realtime tasks and change the cycle time
- 4. Adding realtime tasks and change the cycle time and kill the task
- 5. Run a plotting task and save the task list by using "save"
- 6. Run the saved task a second time by using "load"
- 7. Execute a command file
- 8. Edit a command file
- Final Remarks

Summary of the already explained aspects of the VLFL-Toolbox

The previous examples of this series covered the following topics:

- Example 01: First Steps Using the VLFL-Toolbox for Solid Object Design
- Example 02: Using the VLFL-Toolbox for STL-File Export and Import
- Example 03: Closed 2D Contours and Boolean Operations in 2D
- Example 04: 2.5D Design Using Boolean Operators on Closed Polygon Lists (CPL)
- Example 05: Creation, Relative Positioning and Merging of Solid Geometries (SG)
- Example 06: Relative Positioning and Alignment of Solid Geometries (SG)
- Example 07: Rotation of Closed Polygon Lists for Solid Geometry Design
- Example 08: Slicing, Closing, Cutting and Separation of Solid Geometries
- Example 09: Boolean Operations with Solid Geometries
- Example 10: Packaging of Sets of Solid Geometries (SG)
- Example 11: Attaching Coordinates Frames to Create Kinematik Models
- Example 12: Define Robot Kinematics and Detect Collisions
- Example 13: Mounting Faces and Conversion of Blocks into Leightweight-structures
- Example 14: Manipulation Functions for Closed Polygons and Laser Cutting (SVG)
- Example 15: Create a Solid by 2 Closed Polygons
- Example 16: Create a Solid along a contour
- Example 20: Programmaticaly Interface to SimMechanics Multi-Body Toolbox
- Example 21: Programmaticaly Design of SimMechanisms Mechanisms and Simulink Signals

- Example 22: Creation of fixed nodes for the assembly of joints
- Example 23: Automatic Creation of a Missing Link and 3D Print of a Complete Model
- Example 24: Automatic Creation of a Joint Limitations
- Example 25: Automatic Creation of Video Titels, Endtitels and Textpages
- Example 26: Create Mechanisms using Universal Planar Links
- Example 27: Fourbar-Linkage: 2 Pose Syntheses and Linkage Export for 3D Printing
- Example 28: Fourbar-Linkage: 3 Pose Syntheses and Linkage Export for 3D Printing
- Exercise 30: Create Mechanisms using Universal Planar Links
- Exercise 31: Importing 3D Medical DICOM Image Data and converting into 3D Solids
- Exercise 32: Exchanging Data with a FileMaker Database
- Exercise 33: Using a Round-Robin realtime multi-tasking system

Motivation for this tutorial

Matlab can be converted relatively simply to a realtime-multi-tasking environment according to the round-robin method. The following conditions must apply: A) There is a fixed time base for all tasks B) All tasks are called up one after the other in the fixed time clock

The environment presented in this tutorial has the following properties:

- A) Shell character = While the real-time environment is running, Matlab commands can still be typed as before.
- B) All variables of the real-time environment can be modified directly.

List of supported functions

The input-interpreter routine of the real-time environment has additional commands that superimpose comparable commands on the Matlab command line.

- ***HELP*** - shows a help text
- ***EXIT*** or ***QUIT*** - ends the environment
- ***SHOWLIST*** or ***TASKS*** - shows all tasks
- ***KILLTASKS*** or ***KILLALL*** - removes all tasks
- ***ADD*** - appends a task at the task list
- ***KILLLAST*** - removes the last task
- ***BREAK*** or ***STOP*** - stops the realtime execution
- ***STEP*** - runs the realtime loop only ones
- ***GO*** or ***START*** or ***CONT*** - starts the realtime loop
- ***KILL*** Tasknumber - removes a task with that number
- ***SAVE*** - saves the current task list on disk
- ***LOAD*** - loads the current task list on disk
- ***EXE*** or ***EXECUTE*** filename - executes a command line file
- ***edit*** filename edits a command line file
- ***whos*** shows the variables

Intended use of RRrun or RRshell (both are the same)

- ***RRrun*** - starts the Shell
- ***RRrun*** commandstring (lines are spearated by \r)

1. Starting the shell

The following commands could be typed in absolute in the same way as part of the command string. So please try to type them also manually instead of using them als input parameter of RRrun There is no other chance to create a publishable document as to describe them as input parameter.

```
RRrun 'quit'    % Quit immediately
```

```
RRkeyboardLine =

quit

====LOOP STARTS 22-Mar-2017 23:23:30 for 600 SECONDS with CYCLETIME 0.100 SECONDS===========>
> END =======
RRrun>>
TERMINATED by User
================LOOP ENDS 22-Mar-2017 23:23:30 USING 0.10 SECONDS ========================
```

```
RRrun 'RRstop=RRcputime+1;'  % Quit after 1 second
```

```
RRkeyboardLine =

RRstop=RRcputime+1;

====LOOP STARTS 22-Mar-2017 23:23:31 for 600 SECONDS with CYCLETIME 0.100 SECONDS===========>
> END =======
RRrun>>
================LOOP ENDS 22-Mar-2017 23:23:32 USING 1.20 SECONDS ========================
```

```
RRrun 'LIST \r QUIT' % show the tasks and quit
```

```
RRkeyboardLine =

LIST  QUIT

====LOOP STARTS 22-Mar-2017 23:23:32 for 600 SECONDS with CYCLETIME 0.100 SECONDS===========>
> END =======
RRrun>>
RRtasklist =

  struct with fields:

       t0: 0.1000
    twarn: 0.1000
    tstop: 1
      cnt: 0
```

```
    tlist: []


TERMINATED by User
================LOOP ENDS 22-Mar-2017 23:23:32 USING 0.20 SECONDS ========================
```

```
RRrun 'whos \r QUIT' % show the variables and quit
```

```
RRkeyboardLine =

whos  QUIT

====LOOP STARTS 22-Mar-2017 23:23:33 for 600 SECONDS with CYCLETIME 0.100 SECONDS===========>
> END =======
RRrun>>  Name                Size            Bytes  Class               Attributes

  RRbreak             1x1                 1  logical
  RRcurs              0x0                 0  double              persistent
  RRdelay             1x1                 8  double              global
  RRkeyboardLine      1x6                12  char                global
  RRlastcommand       1x4                 8  char                global
  RRlastexectime      1x1                 8  double
  RRlasttime          1x1                 8  double              global
  RRmaxtime           1x1                 8  double              global
  RRpause             1x1                 8  double
  RRprompt            1x5                10  char                global
  RRstart             1x1                 8  double              global
  RRstop              1x1                 8  double              global
  RRtasklist          1x1               912  struct              global
  RRwindow            1x1                 8  matlab.ui.Figure    global
  RRwindowNr          1x1                 8  double              global
  cmd                 1x4                 8  char
  remain              0x0                 0  char
  token               1x4                 8  char
  varargin            1x1               136  cell


TERMINATED by User
================LOOP ENDS 22-Mar-2017 23:23:33 USING 0.20 SECONDS ========================
```

```
RRrun ('fprintf(''%.2f\n'',cputime) \rQUIT') % show the cputime and quit
```

```
RRkeyboardLine =

fprintf('%.2f\n',cputime) QUIT

====LOOP STARTS 22-Mar-2017 23:23:33 for 600 SECONDS with CYCLETIME 0.100 SECONDS===========>
> END =======
RRrun>>7233.72
```

```
TERMINATED by User
================LOOP ENDS 22-Mar-2017 23:23:33 USING 0.20 SECONDS ========================
```

2. Adding realtime tasks and define the stop time

```
RRrun 'ADD fprintf(''%.2f\n'',cputime) \r RRstop=RRcputime+1;' % show the cputime every cycle
 and quit after 1 second
```

```
RRkeyboardLine =

ADD fprintf('%.2f\n',cputime)  RRstop=RRcputime+1;

====LOOP STARTS 22-Mar-2017 23:23:34 for 600 SECONDS with CYCLETIME 0.100 SECONDS===========>
> END =======
RRrun>>7234.54

7234.64
7234.75
7234.86
7234.96
7235.06
7235.16
7235.26
7235.36
7235.46
7235.57
7235.67
================LOOP ENDS 22-Mar-2017 23:23:35 USING 1.30 SECONDS ========================
```

3. Adding realtime tasks and change the cycle time

```
RRrun 'ADD fprintf(''%.2f\n'',cputime) \r RRtasklist.t0=0.05 \r RRstop=RRcputime+1;'
```

```
RRkeyboardLine =

ADD fprintf('%.2f\n',cputime)  RRtasklist.t0=0.05  RRstop=RRcputime+1;

====LOOP STARTS 22-Mar-2017 23:23:35 for 600 SECONDS with CYCLETIME 0.100 SECONDS===========>
> END =======
RRrun>>7236.32

RRtasklist =

  struct with fields:

       t0: 0.0500
    twarn: 0.1000
    tstop: 1
      cnt: 2
    tlist: {' fprintf('%.2f\n',cputime)'  [Inf]  [1]}
```

```
7236.37

7236.42
7236.47
7236.52
7236.58
7236.64
7236.69
7236.74
7236.79
7236.84
7236.89
7236.94
7236.99
7237.04
7237.09
7237.19
7237.24
7237.30
7237.35
7237.40
7237.45
7237.50
================LOOP ENDS 22-Mar-2017 23:23:37 USING 1.30 SECONDS ========================
```

4. Adding realtime tasks and change the cycle time and kill the task

```
RRrun 'ADD fprintf(''%.2f\n'',cputime) \r RRtasklist.t0=0.05 \r KILLLAST \r RRstop=RRcputime+1;'
```

```
RRkeyboardLine =

ADD fprintf('%.2f\n',cputime)  RRtasklist.t0=0.05  KILLLAST  RRstop=RRcputime+1;

====LOOP STARTS 22-Mar-2017 23:23:37 for 600 SECONDS with CYCLETIME 0.100 SECONDS===========>
> END =======
RRrun>>7238.15

RRtasklist =

  struct with fields:

       t0: 0.0500
    twarn: 0.1000
    tstop: 1
      cnt: 2
    tlist: {' fprintf('%.2f\n',cputime)'  [Inf]  [1]}


7238.20

RRtasklist =

  struct with fields:
```

```
       t0: 0.0500
    twarn: 0.1000
    tstop: 1
      cnt: 2
    tlist: {0×3 cell}


================LOOP ENDS 22-Mar-2017 23:23:38 USING 1.35 SECONDS ========================
```

5. Run a plotting task and save the task list by using "save"

```
RRrun 'KILLALL \r global PL; PL=[0 0 0]; \r  ADD global PL; PL=[PL; PL(end,:)+rand(1,3)]; \r
ADD global PL; delete(gca); VLplot(PL,''b.-'',2); view(-30,30); grid on; \r save \r copyplot
\r RRstop=RRcputime+3;'
```

```
RRkeyboardLine =

KILLALL  global PL; PL=[0 0 0];   ADD global PL; PL=[PL; PL(end,:)+rand(1,3)];  ADD global PL
; delete(gca); VLplot(PL,'b.-',2); view(-30,30); grid on;  save  copyplot  RRstop=RRcputime+3
;

====LOOP STARTS 22-Mar-2017 23:23:39 for 600 SECONDS with CYCLETIME 0.100 SECONDS===========>
> END =======
RRrun>>
RRtasklist =

  struct with fields:

       t0: 0.1000
    twarn: 0.1000
    tstop: 1
      cnt: 0
    tlist: []



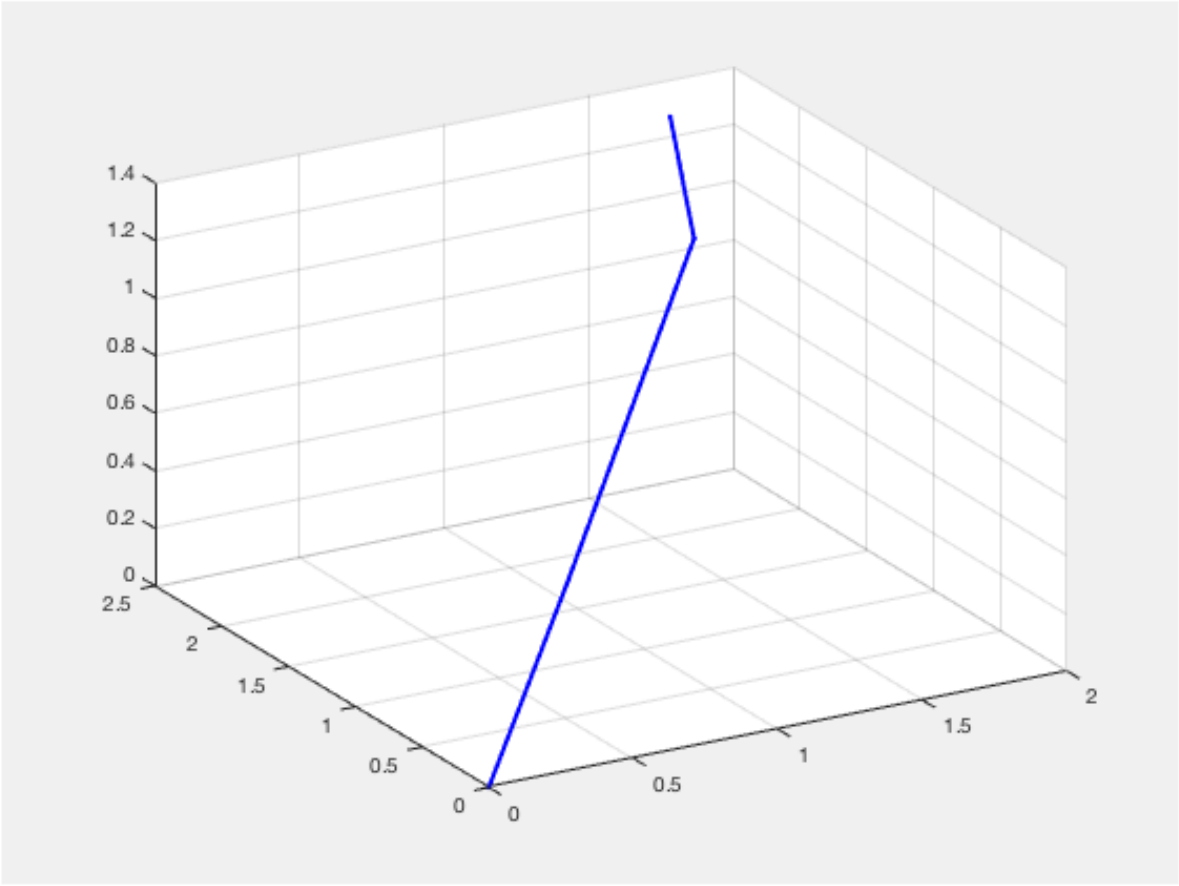
RRtasklist saved in RRtasklist.mat
  Name            Size            Bytes  Class     Attributes

  RRtasklist      1x1              1832  struct    global



RRrun: realtime condition broken by 247 milliseconds

================LOOP ENDS 22-Mar-2017 23:23:43 USING 4.08 SECONDS ========================
```



6. Run the saved task a second time by using "load"

```
RRrun 'load \r start \r copyplot \r RRstop=RRcputime+3;'
```

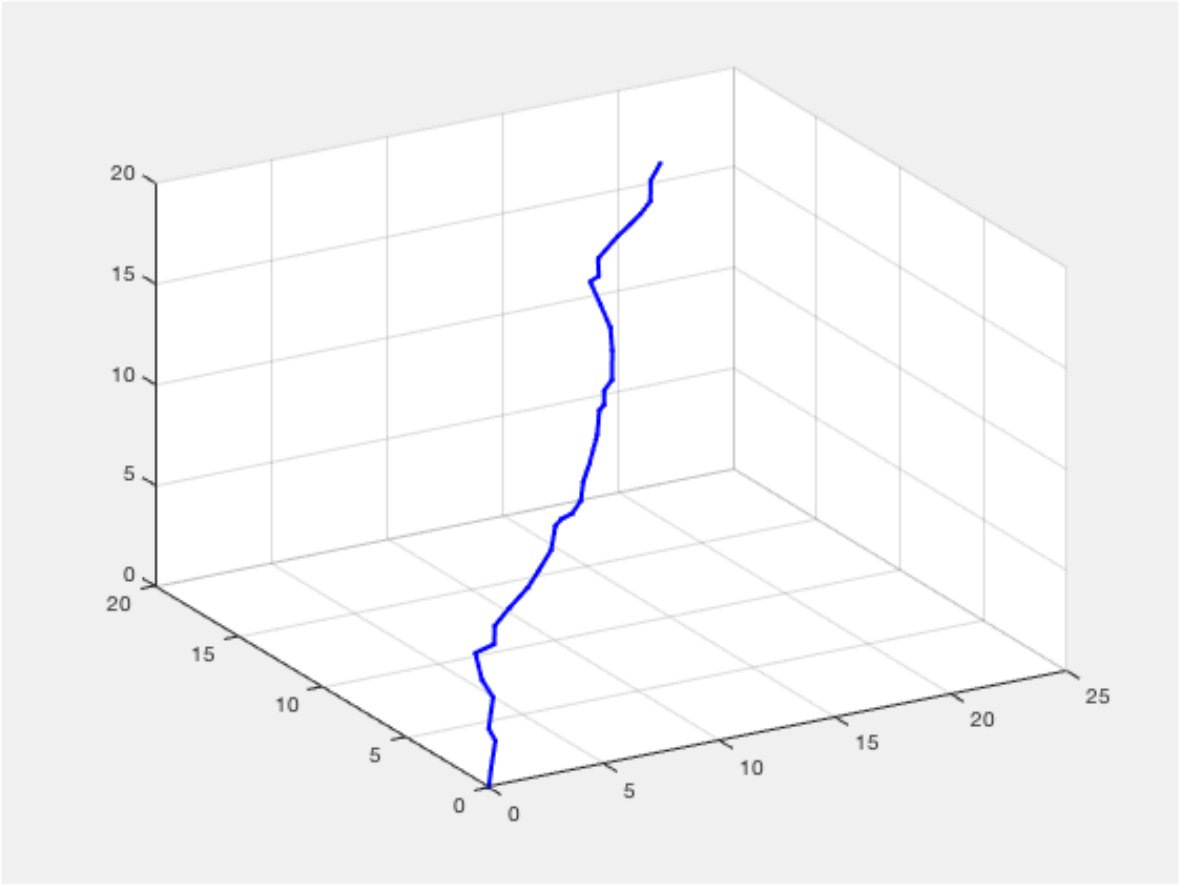
```
RRkeyboardLine =

load  start  copyplot  RRstop=RRcputime+3;

====LOOP STARTS 22-Mar-2017 23:23:43 for 600 SECONDS with CYCLETIME 0.100 SECONDS===========>
> END =======
RRrun>>
New RRtasklist loaded from RRtasklist.mat


RRrun: realtime condition broken by 218 milliseconds

================LOOP ENDS 22-Mar-2017 23:23:47 USING 3.75 SECONDS ========================
```



7. Execute a command file

```
RRrun 'execute RRrun_testcommands.txt'
```

```
RRkeyboardLine =

execute RRrun_testcommands.txt

====LOOP STARTS 22-Mar-2017 23:23:48 for 600 SECONDS with CYCLETIME 0.100 SECONDS===========>
> END =======
RRrun>>
fname =

RRrun_testcommands.txt

Warning: Inputs must be character vectors, cell arrays of character vectors, or
string arrays.

RRtasklist =

  struct with fields:

       t0: 0.1000
    twarn: 0.1000
```

```
    tstop: 1
      cnt: 0
    tlist: []



RRtasklist =

  struct with fields:

       t0: 0.1000
    twarn: 0.1000
    tstop: 1
      cnt: 4
    tlist: {2×3 cell}


ans =

  2×3 cell array

    ' global PL; PL=[P...'    [Inf]    [1]
    ' global PL; delet...'    [Inf]    [3]

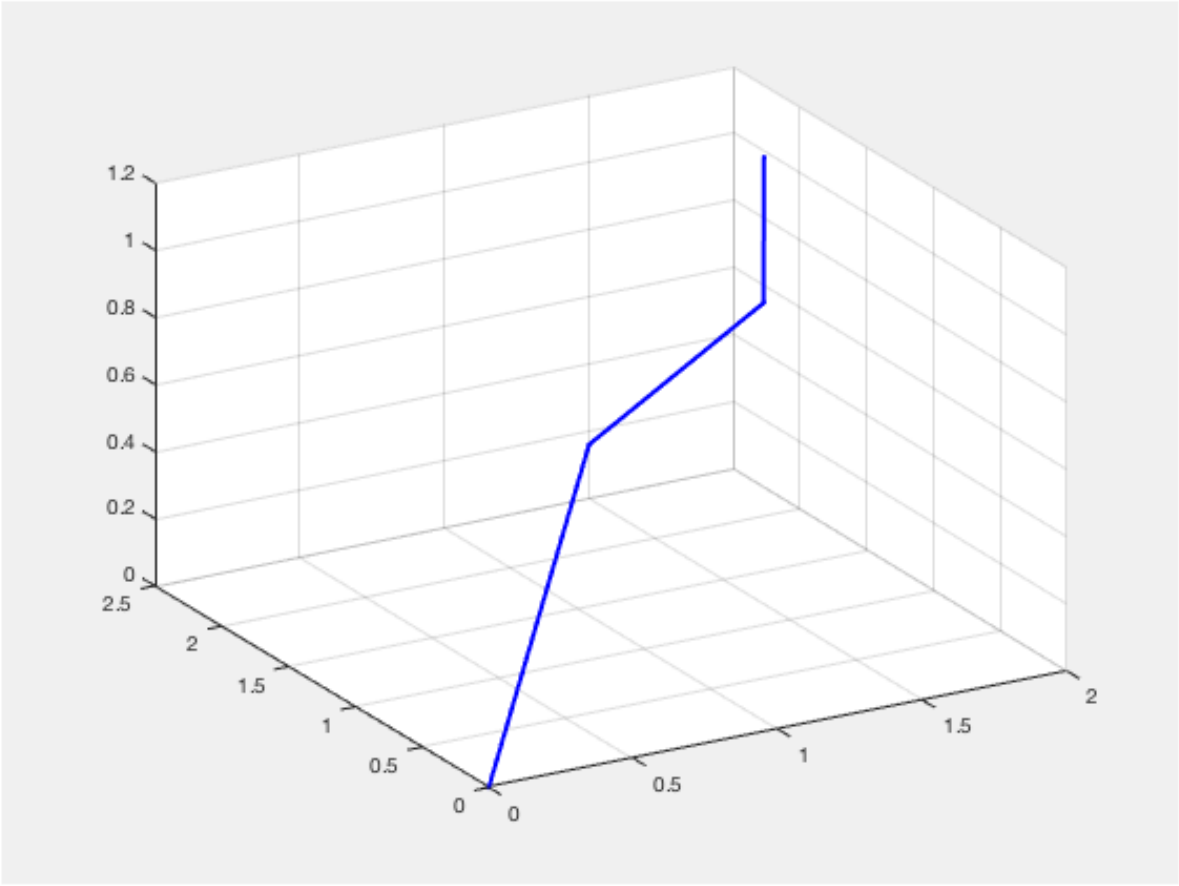
RRtasklist currently stopped. Use "START" or "STEP" to start task execution.

Elapsed time is 0.010238 seconds.



RRrun: realtime condition broken by 219 milliseconds

RRrun>>================LOOP ENDS 22-Mar-2017 23:23:52 USING 3.95 SECONDS ==================
=====
```



8. Edit a command file

```
edit 'RRrun_testcommands.txt'
```

Final Remarks

```
close all
VLFLlicense
```

```
This VLFL-Lib, Rel. 3.6 (2017-Mar-23), is for limited non commercial educational use only!
Licensee: 3.6 Tim Lueth for academic and education 05-Mar-2017 !
Please contact Tim Lueth, Professor at TU Munich, Germany!
WARNING: This VLFL-Lib (Rel. 3.6) license will exceed at 18-Mar-2018!
Executed 22-Mar-2017 23:23:54 by 'lueth' using Matlab 9.1.0.441655 (R2016b) on a MACI64
======================================== Used Matlab products: ==============================
========
database_toolbox
image_toolbox
map_toolbox
matlab
robotics_system_toolbox
simmechanics
```

```
simscape
simulink
================================================================================
========
```

Published with MATLAB® R2016b