Contents

Usage of CRG_TEST_GEN_ROAD

Introducing the usage of crg_test_gen_road. Examples are included. The file comments are optimized for the matlab publishing makro.

```
%   Copyright 2005-2010 OpenCRG - Daimler AG - Klaus Mueller
%
%   Licensed under the Apache License, Version 2.0 (the "License");
%   you may not use this file except in compliance with the License.
%   You may obtain a copy of the License at
%
%       http://www.apache.org/licenses/LICENSE-2.0
%
%   Unless required by applicable law or agreed to in writing, software
%   distributed under the License is distributed on an "AS IS" BASIS,
%   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
%   See the License for the specific language governing permissions and
%   limitations under the License.
%
%   More Information on OpenCRG open file formats and tools can be found at
%
%       http://www.opencrg.org
%
%   $Id: crg_test_gen_road.m 41 2009-10-30 11:00:00Z muellek $
```

Test proceedings

The tests generates synthetic crg-files as following:

- create additional information (curv, banking, slope)
- display result

```
% DEFAULT SETTINGS
% clear enviroment
clear all;
close all;

err_cnt  = 0;
warn_cnt = 0;
txtnum   = 0;

% global settings for the road
uinc = 0.5;
vinc = 0.2;

ubeg = 0;
uend = 1000;
u    = [ubeg uend];

lb = 5;
rb = -3;
v  = rb:vinc:lb;

inc = uinc;

% Create a minimal road and check for valid data

data.u = u;
data.v = v;
data.z = zeros(size(ubeg:uinc:uend,2), size(v,2), 'single');
txtnum = txtnum + 1; data.ct{txtnum} = 'CRG generated artificial road with a smooth surface
data   = crg_check(data);
```

curvature

select or deselect for test purposes as shown below

```
LC1  = 100;  R1s  = inf;  R1e  = inf;
LC2  = 50;   R2s  = inf;  R2e  = 50;
LC3  = 400;  R3s  = 50;   R3e  = 50;
% LC31 =  50;  R31s = 50;   R31e = 50;
LC4  = 130;  R4s  = 50;   R4e  = -25;
LC5  = 100;  R5s  = inf;  R5e  = inf;

c = { LC1   {1/R1s  ( 1/R1e  - 1/R1s )/LC1 }  ...
    ; LC2   {1/R2s  ( 1/R2e  - 1/R2s )/LC2 }  ...
    ; LC3   {1/R3s  ( 1/R3e  - 1/R3s )/LC3 }  ...
%     ; LC31  {1/R31s ( 1/R31e - 1/R31s)/LC31}  ...
    ; LC4   {1/R4s  ( 1/R4e  - 1/R4s )/LC4 }  ...
    ; LC5   {1/R5s  ( 1/R5e  - 1/R5s )/LC5 }  ...
    };

% simple check of curvature data
csum = 0;
for ii = 1:size(c,1)
    len = c{ii,1};
    if len < 0 || ~isequal(rem(len,uinc), 0)            % todo: bound should be greater 0, a
        warning('CRG:checkWarning', [num2str(ii) '. length = ' curvatue num2str(len) ' is ne
        err_cnt = err_cnt + 1;
    end
    csum = csum + len;
    % maybe a check greater than (uend-ubeg) is useful
end
c = [ c; { max(0,(uend-ubeg)-csum) {0} } ];    % if required add straight line up to the end
txtnum = txtnum + 1; data.ct{txtnum} = '... curvature added';
```

slope

select or deselect for test purposes as shown below

```
LS1  = 75;   S1s  =  0.0;    S1e  =  0.0;
LS2  = 300;  S2s  =  0.0;    S2e  =  0.01;
LS3  = 400;  S3s  =  0.01;  S3e  =  0.01;

s = { LS1   { S1s ( S1e  - S1s )/LS1 }  ...
    ; LS2   { S2s ( S2e  - S2s )/LS2 }  ...
    ; LS3   { S3s ( S3e  - S3s )/LS3 }  ...
    };

% simple check of slope data
ssum = 0;
for ii = 1:size(s,1)
    len = s{ii,1};
    if len < 0 || ~isequal(rem(len,uinc), 0)            % todo: bound should be greater 0, a
        warning('CRG:checkWarning', [ num2str(ii) '. slope length = ' num2str(len) ' is nega
        err_cnt = err_cnt + 1;
    end
    ssum = ssum + len;
    % maybe a check greater than (uend-ubeg) is useful
end
s = [ s; { max(0,(uend-ubeg)-ssum) {0} } ];    % if required keep last value up to the end
txtnum = txtnum + 1; data.ct{txtnum} = '... slope added';
```

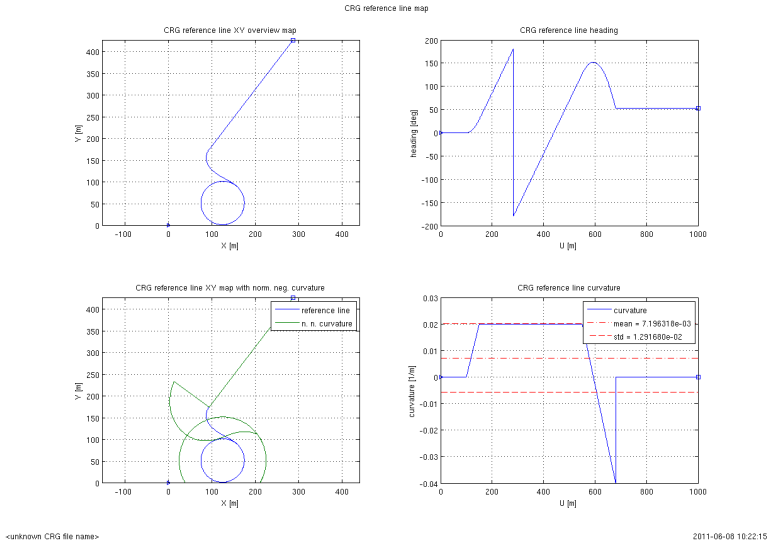
banking

select or deselect for test purposes as shown below

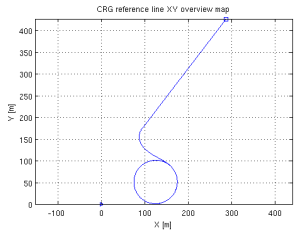
```
LB1  = 100;  B1s  =  0;     B1e  = 0;
LB2  = 50;   B2s  =  0;     B2e  = -0.02;
LB3  = 400;  B3s  = -0.02;  B3e  = -0.02;
% LB31 =  50;  B31s = 0.02;   B31e = 0.02;
LB4  = 125;  B4s  = -0.02;  B4e  = +0.02;
LB5  = 100;  B5s  = +0.02;  B5e  = 0.1;

b = { LB1   {  ( B1e  - B1s )/LB1 }  ...
    ; LB2   {  ( B2e  - B2s )/LB2 }  ...
    ; LB3   {  ( B3e  - B3s )/LB3 }  ...
%   ; LB31  {  ( B31e - B31s)/LB31}  ...
    ; LB4   {  ( B4e  - B4s )/LB4 }  ...
    ; LB5   {  ( B5e  - B5s )/LB5 }  ...
    };

% simple check of slope data
bsum = 0;
for ii = 1:size(b,1)
    len = b{ii,1};
    if len < 0 || ~isequal(rem(len,uinc), 0)            % todo: bound should be greater 0, a
        warning('CRG:checkWarning', [ num2str(ii) '. banking length = ' num2str(len) ' is ne
        err_cnt = err_cnt + 1;
    end
    bsum = bsum + len;
    % maybe a check greater than (uend-ubeg) is useful
end
b = [ b; { max(0,(uend-ubeg)-bsum) {0} } ];    % if required keep last value up to the end
txtnum = txtnum + 1; data.ct{txtnum} = '... banking added';

% check and warn if the data mismatch (simple and straight forward)

if ~isequal(uend-ubeg, csum, ssum, bsum)
    warning('CRG:checkWarning',  'Mismatch between the length of parameter descriptions.');
    warning('CRG:checkWarning', ['range of u-coordinate : ' num2str(uend-ubeg,'%.10g')]);
    warning('CRG:checkWarning', ['range of    curvature : ' num2str(csum,'%.10g')]);
    warning('CRG:checkWarning', ['range of        slope : ' num2str(ssum,'%.10g')]);
    warning('CRG:checkWarning', ['range of      banking : ' num2str(bsum,'%.10g')]);
    warn_cnt = warn_cnt + 1;
end

if any([warn_cnt err_cnt])
    disp('Summary of check:')
```

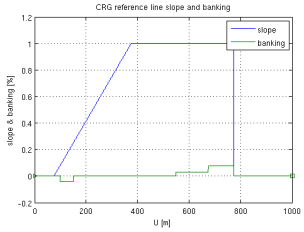
```
    if warn_cnt > 0
        warning('CRG:checkWarning', ['Total Warnings: ' num2str(warn_cnt)]);
    end
    if err_cnt > 0
        error('CRG:checkError', ['Total   Errors: ' num2str(err_cnt)]);
    end
end

% generate the road
data = crg_gen_csb2crg0(inc, u, v, c, s, b);

% show the data
txtnum = txtnum + 1; data.ct{txtnum} = '... finished';
crg_write(crg_single(data), 'crg_test_gen_road.crg');

% show the data
data = crg_show(data);
```

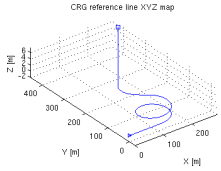
```
Warning: Mismatch between the length of parameter descriptions.
Warning: range of u-coordinate : 1000
Warning: range of    curvature : 780
Warning: range of        slope : 775
Warning: range of      banking : 775
Summary of check:
Warning: Total Warnings: 1
```

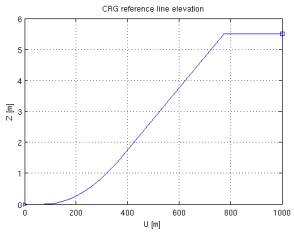


CRG reference line elevation

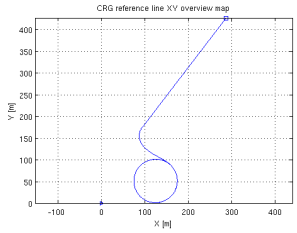


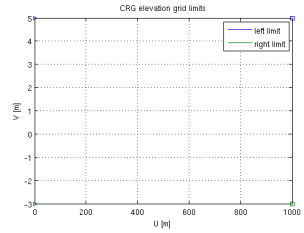


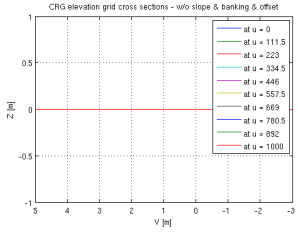


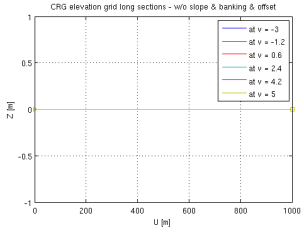


CRG elevation grid cuts and limits

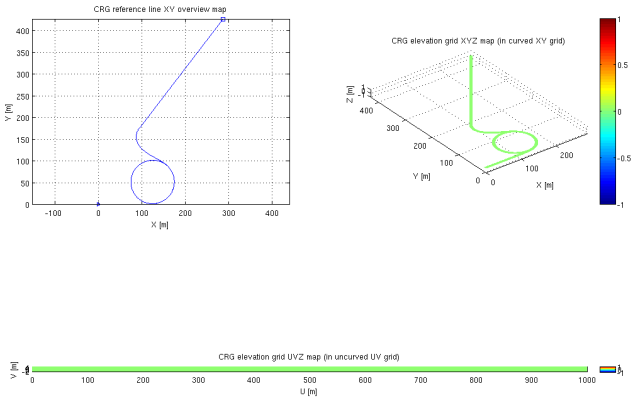






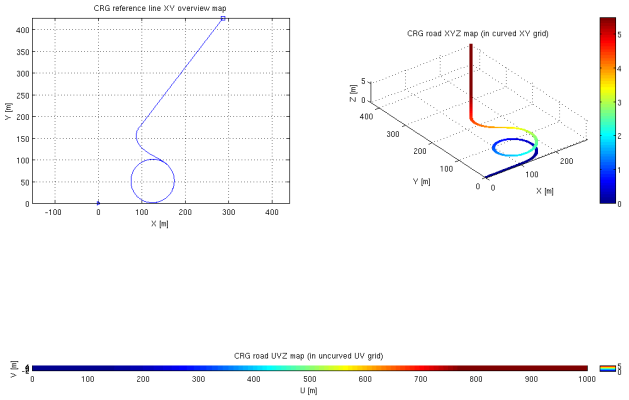


CRG elevation grid surface - w/o slope and banking



<unknown CRG file name> 2011-06-08 10:22:18

CRG road surface



<unknown CRG file name> 2011-06-08 10:22:18

```
CRG elevation grid size:
length                     =          4111
width                      =            41

CRG header data:
reference line start u     =             1
reference line end u       =          3111
reference line increment   =           1.5
long section v right       =             5
long section v left        =             5
long section v increment   =           1.2
reference line start x     =             1
reference line end x       =  461.49458543
reference line offset x    =             1
reference line start y     =             1
reference line end y       =  421.39263342
reference line offset y    =             1
reference line start z     =             1
reference line end z       =  5.4914999313
reference line offset z    =             1
reference line start phi   =             1
reference line end phi     =  1.9368346922
reference line offset phi  =             1
reference line start s     =             1
reference line end s       =             1
reference line start b     =             1
reference line end b       =             1

CRG modifier data:
grid nan mode              =             2
refpoint u fraction        =             1
refpoint u offset          =             1
refpoint v                 =             1
refpoint v offset          =             1
refpoint x                 =             1
refpoint y                 =             1
refpoint z                 =             1
refpoint phi               =             1

CRG option data:
border mode u              =             2
border mode v              =             2
border offset u            =             1
border offset v            =             1
border smooth ubeg         =             1
border smooth uend         =             1
refline continuation       =             1
refline search far         =           1.5
refline search close       =           1.5
check eps                  =          1e-16
check inc                  =          1.111
check tol                  =         1.1111

CRG derived data:
crossing ubeg              =             1
crossing uend              =          3111
crossing ulen              =             1
```

```
CRG comment data:

CRG minimal artificial road and flat surface
... curvature added
... slope added
... banking added
... finished
```