Contents

Usage of continuesTrack 1

Test proceedings 2

Test1 (continues track - intersection) 3

Test2 (continues track - parallel) 10

Usage of continuesTrack

Test scripts to display results of continues track (option) settings. Examples are included. The file comments are optimized for the matlab publishing makro.

```
%   Copyright 2005-2011 OpenCRG - VIRES Simulationstechnologie GmbH -
%   Holger Helmich
%
%   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_continuesTrack.m 1 2011-06-07 11:33:00Z hhelmich $
```

Test proceedings

- gen synth. crg-file
- set option: bdmu, bdmv, rflc
- display extrapolation > uend
- display extrapolation < ubeg

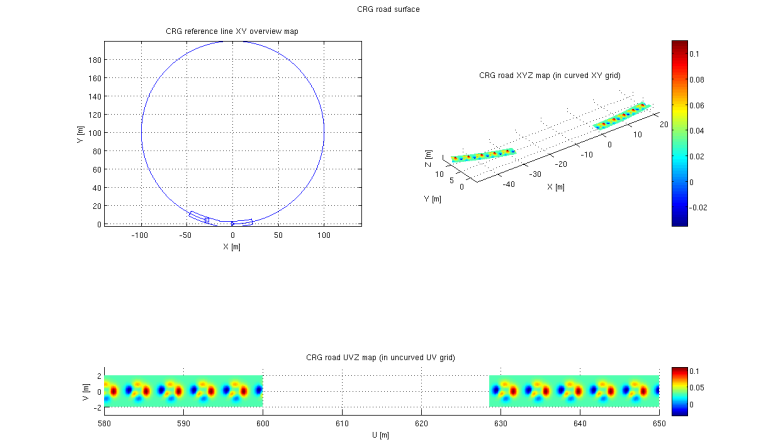
```
% DEFAULT SETTINGS
% display results
bmode = [0 4];      % border mode [0 4] <=> (0 to 4); [1 1] <=> (only 1)
use_b = 1;          % incl. banking (0/1)
use_s = 1;          % incl. slope (0/1)

d_uinc  = 0.1;      % debug: uinc (0 => data.dved.uinc)
d_vinc  = 0.1;      % debug: vinc (0 => data.dved.vinc)
d_suo   = 50;       % debug: ubeg offset
d_euo   = 50;       % debug: uend offset
d_svo   = 1;        % debug: vmin offset
d_evo   = 1;        % debug: vmax offset
```

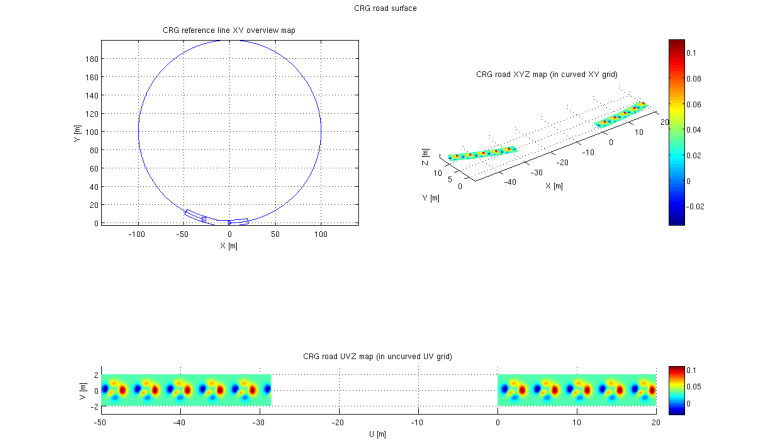
Test1 (continues track - intersection)

```
% gen synthetical track
ulength = 600;
LC1  = 600; C1s = 0.0;      C1e = 6.0;
LS1  = 600; S1s = 0.0003;   S1e = 0.0;
LB1  = 600; B1s = 0.0;      B1e = 0.03;

c = { LC1    { (C1e-C1s)/LC1 } ...
    };

s = { LS1   { (S1e-S1s)/LS1 } ...
    };

b = { LB1    { (B1e-B1s)/LB1 } ...
    };

if ~use_b, b = []; end
if ~use_s, s = []; end

data = crg_gen_csb2crg0([0.1,0.1], ulength, 2, c, s, b);

% z-values
[nu nv] = size(data.z);
nunv_max = ceil(nu/nv);

z = 0.01*peaks(nv);
z = repmat(z, nunv_max, 1);
data.z = single(z(1:nu, :));
data.z = data.z + 0.03;

data = crg_check(data);

% dimensions
ubeg = data.head.ubeg-d_suo;
uend = data.head.uend+d_euo;
if d_uinc, uinc = d_uinc; else
    uinc = data.head.uinc; end

vmin = data.head.vmin-d_svo;
vmax = data.head.vmax+d_evo;
if d_vinc, vinc = d_vinc; else
    vinc = data.head.vinc; end

% visualisation for each border mode
for i = bmode(1):bmode(2)
```

```
    disp(['Border mode: ', num2str(i)]);

    data.opts.bdmu = i;
    data.opts.bdmv = i;
    data.opts.rflc = 1;
    data = crg_check(data);

    % extrapolation > uend
    crg_show_road_uv2surface(data, [data.head.uend-20:uinc:data.head.uend+50], [vmin:vinc:vm

    % extrapolation < ubeg
    crg_show_road_uv2surface(data, [data.head.ubeg-50:uinc:data.head.ubeg+20], [vmin:vinc:vm

end
```

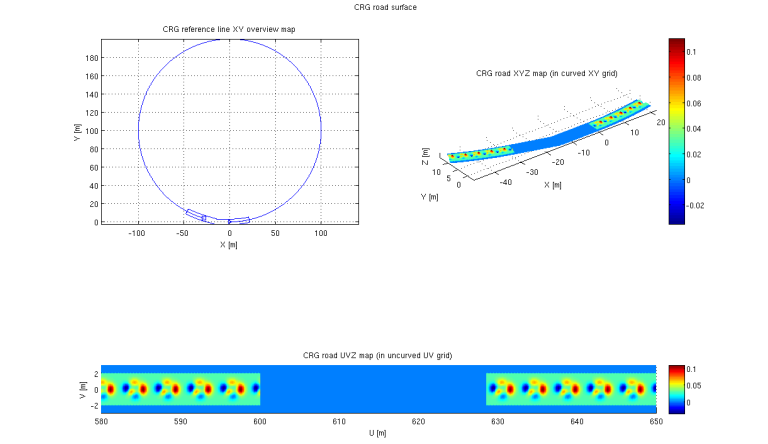
```
Border mode: 0
Border mode: 1
Border mode: 2
Border mode: 3
Border mode: 4
```





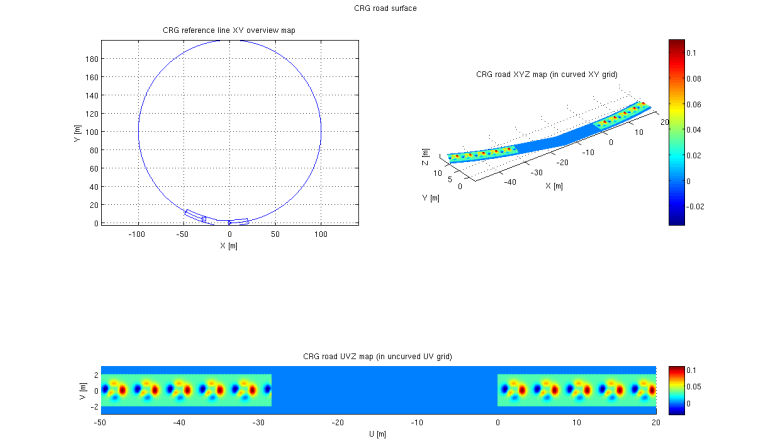
<unknown CRG file name>

2011-06-07 11:35:54



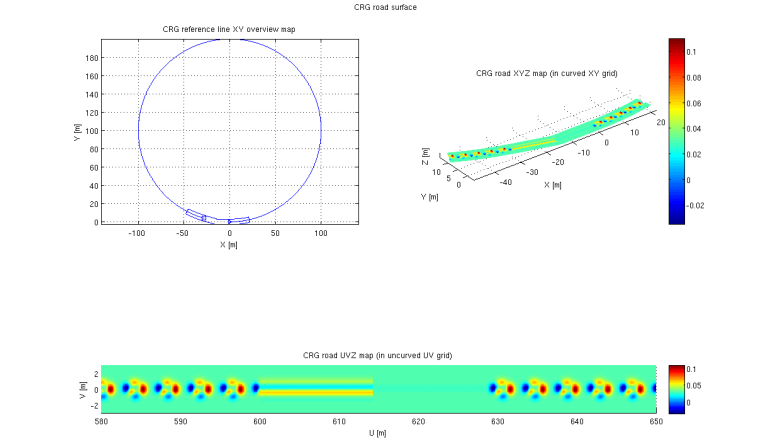
<unknown CRG file name>

2011-06-07 11:35:56



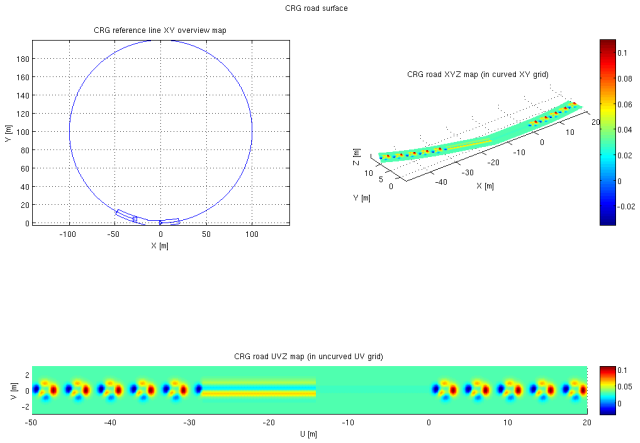
<unknown CRG file name>

2011-06-07 11:35:59

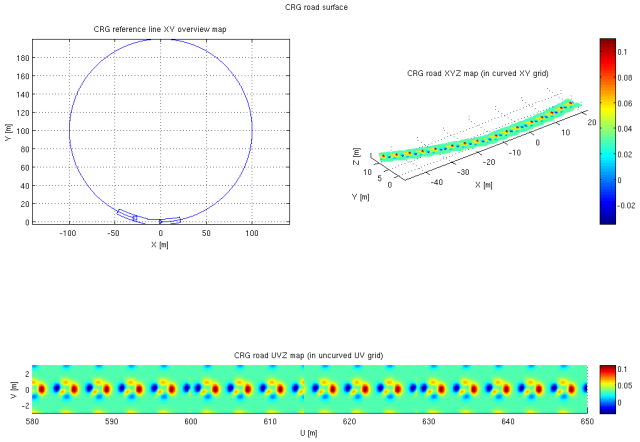


<unknown CRG file name>

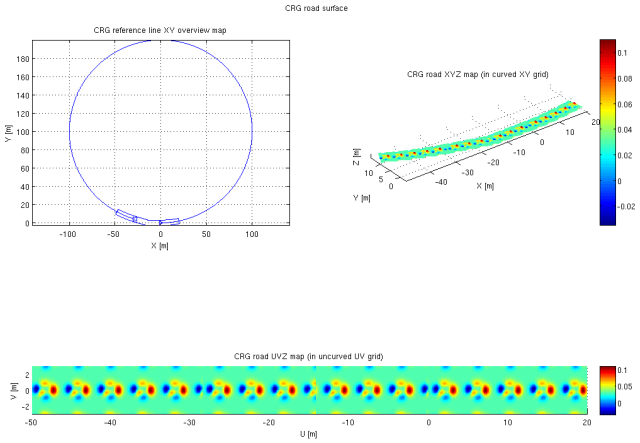
2011-06-07 11:36:03



<unknown CRG file name> 2011-06-07 11:36:05

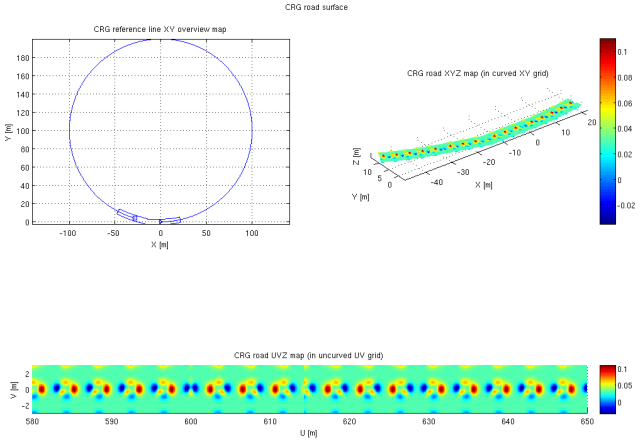


<unknown CRG file name> 2011-06-07 11:36:08



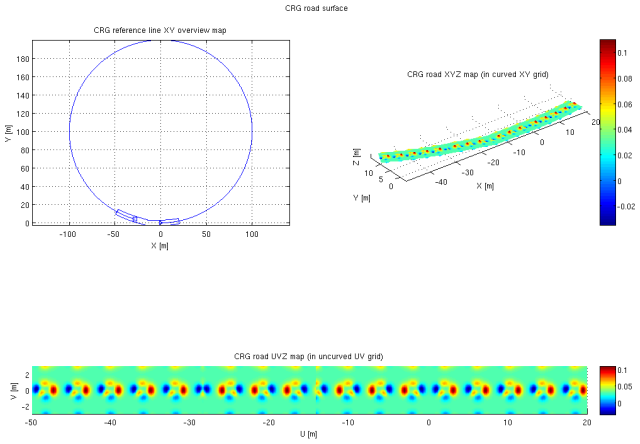
<unknown CRG file name>

2011-06-07 11:38:12



<unknown CRG file name>

2011-06-07 11:38:15



<unknown CRG file name>

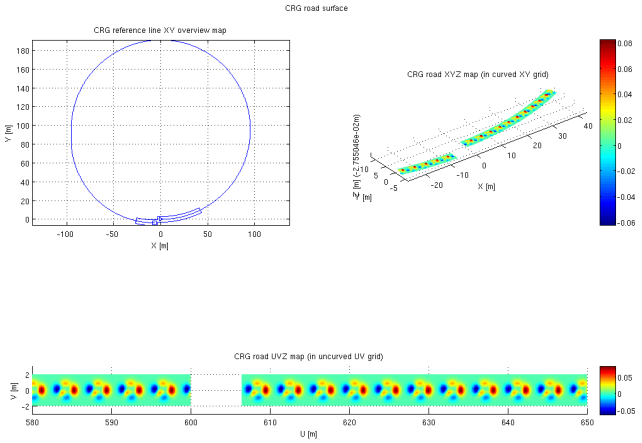
2011-06-07 11:38:18

Test2 (continues track - parallel)

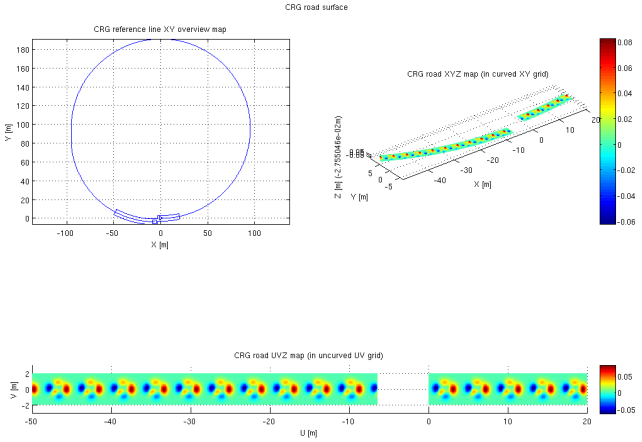
```
% gen synthetical track

ulength = 600;

LC1  = 450;  C1s  =  0.0;     C1e  =  3/2*pi;
LC2  = 10;   C2s  =  0.0;     C2e  =  0.0;
LC3  = 140;  C3s  =  0.0;     C3e  =  1/2*pi;

LS1  = 600; S1s = 0.0003;  S1e = 0.0;
LB1  = 600; B1s = 0.0;  B1e = 0.03;

c = {  LC1    { (C1e-C1s)/LC1 } ...
     ; LC2    { (C2e-C2s)/LC2 } ...
     ; LC3    { (C3e-C3s)/LC3 } ...
    };

s = { LS1   { (S1e-S1s)/LS1 } ...
    };

b = { LB1    { (B1e-B1s)/LB1 } ...
    };

if ~use_b, b = []; end
if ~use_s, s = []; end

data = crg_gen_csb2crg0([0.1,0.1], ulength, 2, c, s, b);

% z-values
[nu nv] = size(data.z);
nunv_max = ceil(nu/nv);

z = 0.01*peaks(nv);
z = repmat(z, nunv_max, 1);
data.z = single(z(1:nu, :));
data.z = data.z + 0.03;

data = crg_check(data);

mdop = struct();
mdop.rptp = 0;
data.mods = mdop;
data = crg_mods(data);

% dimensions
```

```
ubeg = data.head.ubeg-d_suo;
uend = data.head.uend+d_euo;
if d_uinc, uinc = d_uinc; else
    uinc = data.head.uinc; end

vmin = data.head.vmin-d_svo;
vmax = data.head.vmax+d_evo;
if d_vinc, vinc = d_vinc; else
    vinc = data.head.vinc; end

% visualisation for each border mode
for i = bmode(1):bmode(2)

    disp(['Border mode: ', num2str(i)]);

    data.opts.bdmu = i;
    data.opts.bdmv = i;
    data.opts.rflc = 1;
    data = crg_check(data);

    % extrapolation > uend
    crg_show_road_uv2surface(data, [data.head.uend-20:uinc:data.head.uend+50], [vmin:vinc:vm

    % extrapolation < ubeg
    crg_show_road_uv2surface(data, [data.head.ubeg-50:uinc:data.head.ubeg+20], [vmin:vinc:vm

end
```

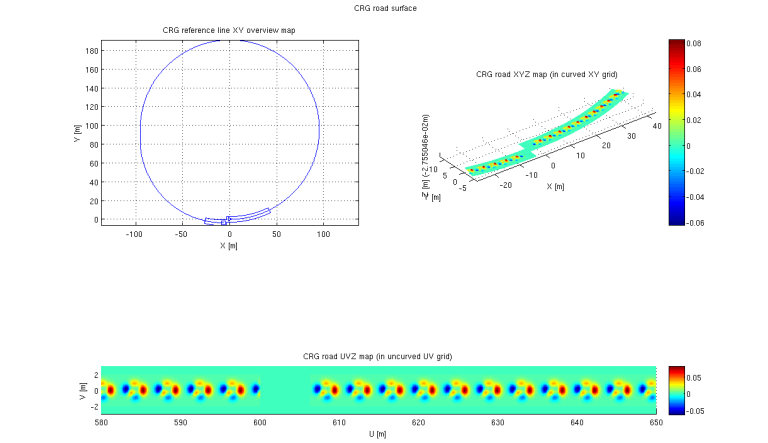
```
Border mode: 0
Border mode: 1
Border mode: 2
Border mode: 3
Border mode: 4
```



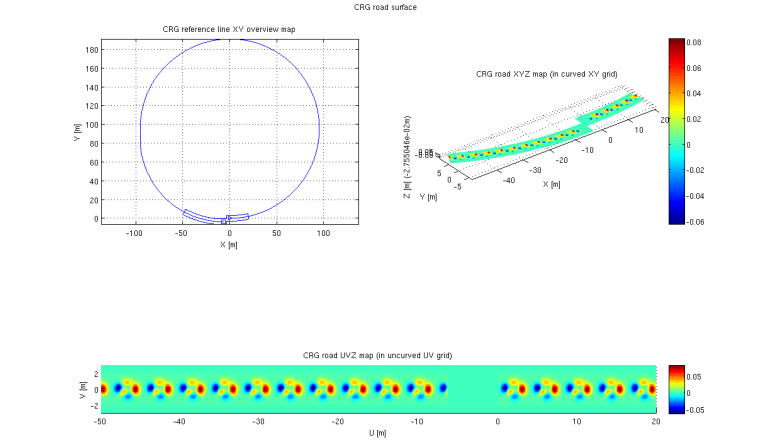
<unknown CRG file name> 2011-06-07 11:36:28



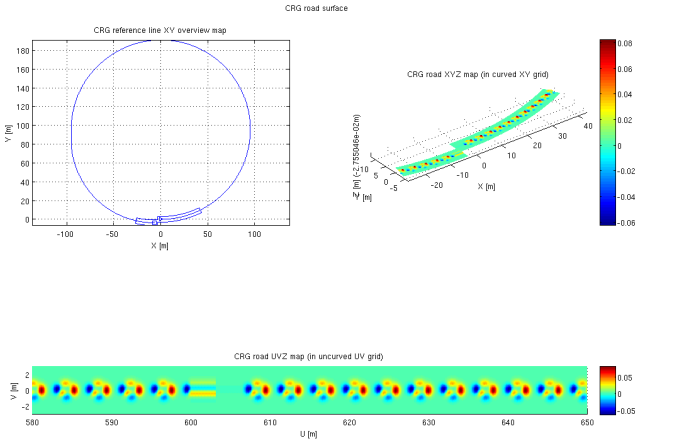
<unknown CRG file name> 2011-06-07 11:36:30



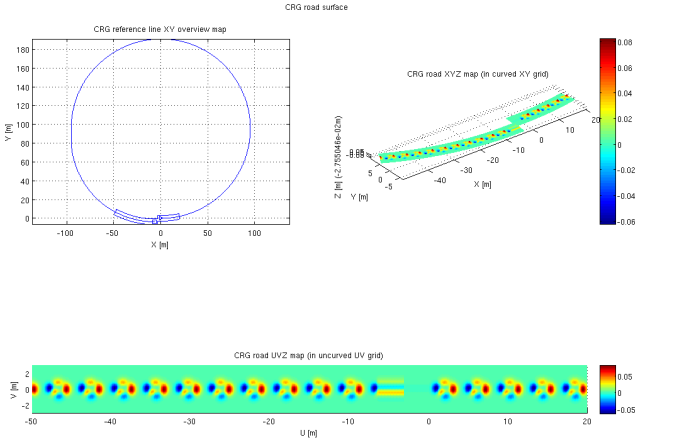
<unknown CRG file name> 2011-06-07 11:36:34



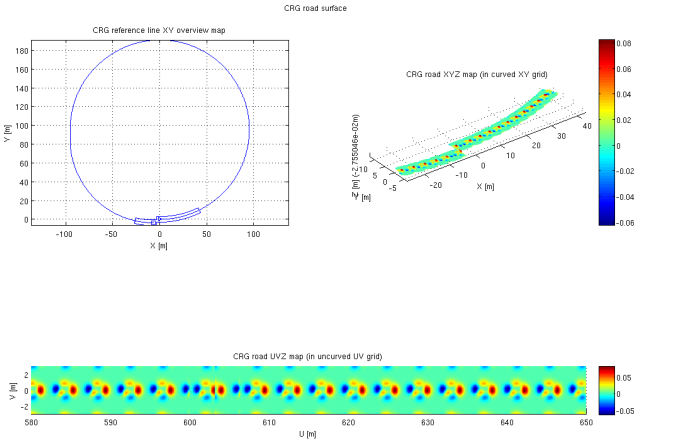
<unknown CRG file name> 2011-06-07 11:36:37



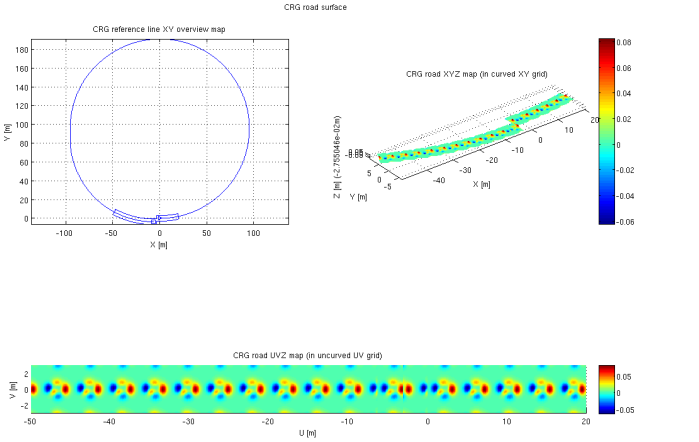
<unknown CRG file name> 2011-06-07 11:36:40



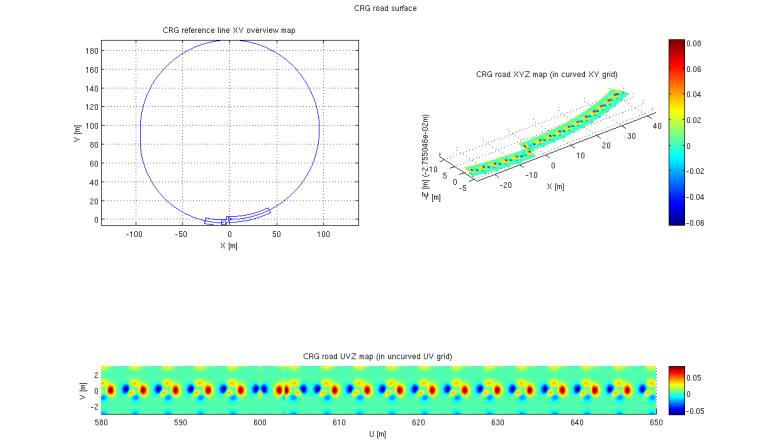
<unknown CRG file name> 2011-06-07 11:36:43



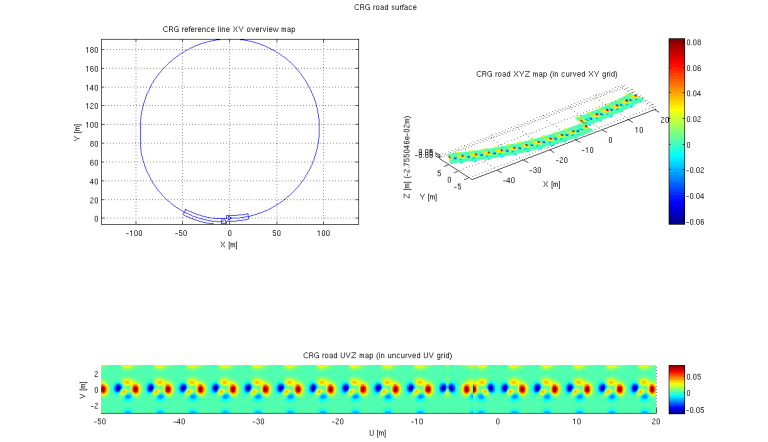
<unknown CRG file name> 2011-06-07 11:36:46



<unknown CRG file name> 2011-06-07 11:36:49



<unknown CRG file name> 2011-06-07 11:36:52



<unknown CRG file name> 2011-06-07 11:36:56