Tiago Padovan | Junichiro Kawaguchi

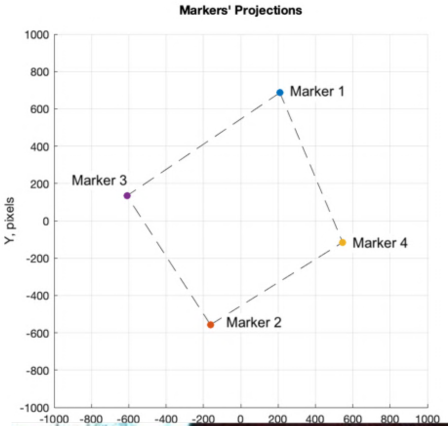
Simplified Motion Estimator Calling Script

Patchedconics has developed a simplified version of the Motion Simulator. It uses a static target and can be easily interpreted by the user. Because the target is static, the Estimator only returns correct direction cosine matrix (DCM) and position of the targets. Other parameters (angular velocity, angular momentum and moment of inertia) are provided with general values and have no meaning in such case.

The script is short, containing 87 lines which include comments, sections and a sequence to identify if the interpreter is MATLAB or Octave for using proper commands.

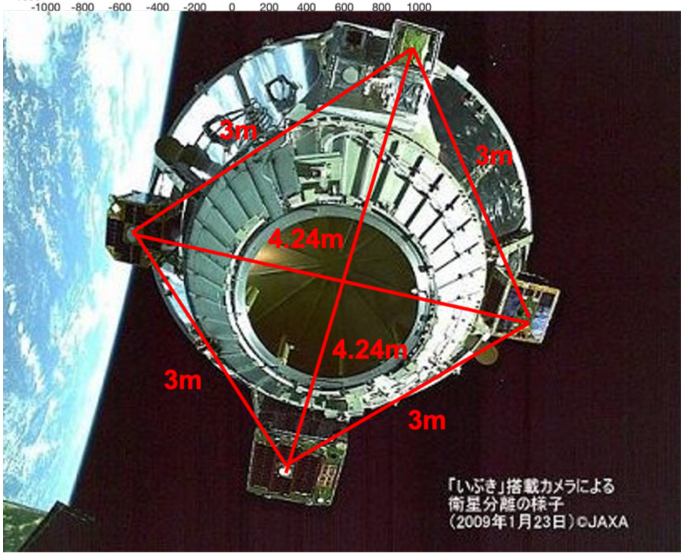
In the current version of the Assessment Package, a file must be provided to the Estimator defining the position of the markers with respect to the target body. Using a set of input parameters, we modified the markers positions definition to see how the Estimator behaves.

Input Parameters

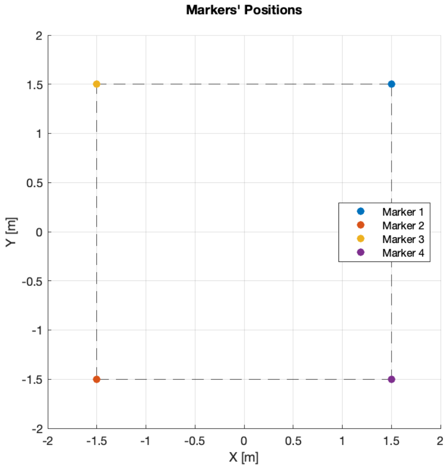


The input parameters preloaded in MSsimple.m file are referring to a real image and markers are assumed to be positioned in the 2000 x 2000 frame as follows:

```
Marker 1 [ 209.40   686.61]
Marker 2 [-162.39 -556.98]
Marker 3 [-609.69   133.91]
Marker 4 [ 545.58 -115.38]
```



Example 1:



On example 1, the makers are assumed to be on positions

```
[ 1.5  1.5 -1.5]
[-1.5 -1.5 -1.5]
[-1.5  1.5 -1.5]
[ 1.5 -1.5 -1.5]
```

w.r.t. the target body.

Using such markers configuration, the output DCM is as follows:

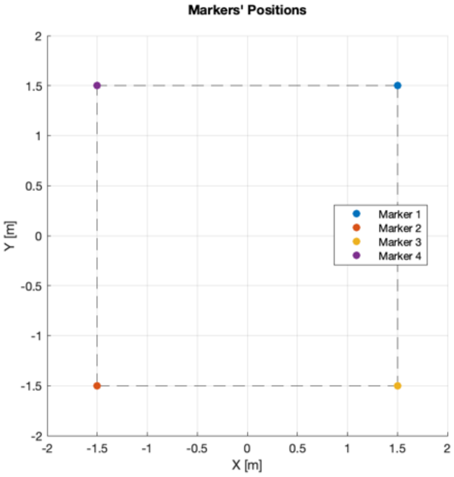
```
[ -0.43386      0.83244      0.49813
   0.83646      0.49841     -0.086239
  -0.3348       0.091411    -0.86098 ]
```

And the Euler angles are (yaw-pitch-roll sequence):

```
yaw = 117.53°  |  pitch = -29.88°  |  roll = -174.28°
```

Where yaw is the rotation around Z axis which is $X \times Y$, pitch around the Y axis and roll around the X axis.

Example 2:



On example 2, the makers are assumed to be on positions

```
[ 1.5  1.5 -1.5]
[-1.5 -1.5 -1.5]
[ 1.5 -1.5 -1.5]
[-1.5  1.5 -1.5]
```

w.r.t. the target body.

Using such markers configuration, the output DCM is as follows:

```
[  0.83244     -0.43386     -0.49813
   0.49841      0.83646      0.086239
   0.091411    -0.3348      -0.86098 ]
```

And the Euler angles are (yaw-pitch-roll sequence):

```
yaw = -27.53°  |   pitch = 29.88°  |   roll = 5.72°
```

Conclusion

Swapping the definition of Markers 3 and 4 has the same meaning to the estimator of rotating the target in 180º around the axis connecting Markers 1 and 2. And the results show the outputs are as expected. The Simplified simulator comes preloaded with example 1, the user may test example 2 by entering the new Markers definition.

APPENDIX I – MSsimple.m Script for Windows Platform

```
clear all; close all; clc

%% SEQUENCE TO IDENTIFY IF IS MATLAB OR OCTAVE -----------------------------
isMATLAB = false; LIC = license('inuse');
for elem = 1:numel(LIC); envStr = LIC(elem).feature;
    if strcmpi(envStr,'matlab'); isMATLAB = true; break; end
end

%% MARKERS CONFIGURATION --------------------------------------------------
mr   =    [ 1.5  1.5 -1.5; -1.5 -1.5 -1.5; -1.5  1.5 -1.5; 1.5 -1.5 -1.5];
mrk =    [mr(1,:)-mr(1,:) mr(2,:)-mr(1,:) mr(3,:)-mr(1,:) mr(4,:)-mr(1,:)];

%% INITIAL MARKERS' POSITIONS ON 2000 X 2000 FRAME ------------------------
M1   =    [ 209.40   686.61];       % Marker 1 X and Y coordinates
M2   =    [-162.39 -556.98];       % ...
M3   =    [ 545.58 -115.38];       % ...
M4   =    [-609.69  133.91];       % ...
M    =    [M1 M2 M3 M4];           % All Markers

%% ESTIMATORS PARAMETERS --------------------------------------------------
dte =    1;                        % Estimator Interval                [s]
B    =    10;                       % Estimator stored information      [s]
ST   =    0;                        % Steps stored by estimator before
                                    % first estimation starts           [s]
IL   =    1;                        % Number of times "I" is estimated
                                    % before "L" updates                [s]

%% PATHS -----------------------------------------------------------------
cdc      = pwd;                                           % Directory
ptME     = fullfile(cdc,'ME','1','Windows','ME');          % Motion Estmtor
ptNP     = fullfile(cdc,'ME','1','Windows','estdata.npz'); % Internal file
ptTE     = fullfile(cdc,'data','toestimator.csv');         % Input  csv file
ptTS     = fullfile(cdc,'data','tosimulator.csv');         % Output csv file
ptSP     = fullfile(cdc,'data',      'shape.csv');         % Marker csv file
ptDT     = fullfile(cdc,'data');                       % Directory for I/O
csvwrite(ptSP,mrk);

%% CALLING ESTIMATOR -----------------------------------------------------
% 1) Input file must be created
t        = 1;                                         % Time information
reset    = 1;                                         % Reset internal file
input    = [M, IL, ST, t, dte, B, reset];             % Input array
csvwrite(ptTE,input);                                 % Input array to csv
% 2) Motion Esitmator is called
command = sprintf('"%s" "%s" "%s" "%s" "%s"', ptME, ptTE, cdc, ptTS, ptSP);
system(command);

%% MOTION ESTIMATOR MUST BE CALLED AT LEAST TWICE -------------------------
t        = t + dte;                                   % Time information
reset    = 0;                                         % Keep internal file
input    = [M, IL, ST, t, dte, B, reset];             % Input array
csvwrite(ptTE,input);                                 % Input array to csv

system(command);     % Call executable with input file and arguments

%% READING OUTPUT FILE AND PLOTTING --------------------------------------
if isfile(ptTS)
  rslt     =    csvread(ptTS);                           % Read output csv file

  disp('Markers Positions [in meters, w.r.t. observer]:');
  M1r      = rslt(6,:); disp(['Marker 1: [',num2str(M1r),']']);
  M2r     = rslt(7,:); disp(['Marker 2: [',num2str(M2r),']']);
  M3r     = rslt(8,:); disp(['Marker 3: [',num2str(M3r),']']);
  M4r     = rslt(9,:); disp(['Marker 4: [',num2str(M4r),']']); disp(' ');

  DCM     = rslt(10:12,:); disp('Direction Cosine Matrix:')
  disp(['[',num2str(DCM(1,:))]);     disp([' ',num2str(DCM(2,:))]);
  disp([' ',num2str(DCM(3,:)),']']); disp(' ');

  disp('Estimated distances between markers [in meters]:');
  D12     = norm(M1r-M2r); disp(['D12: ',num2str(D12)]);
  D13     = norm(M1r-M3r); disp(['D13: ',num2str(D13)]);
  D14     = norm(M1r-M4r); disp(['D14: ',num2str(D14)]);
  D23     = norm(M2r-M3r); disp(['D23: ',num2str(D23)]);
  D24     = norm(M2r-M4r); disp(['D24: ',num2str(D24)]);
  D34     = norm(M3r-M4r); disp(['D34: ',num2str(D34)]); disp(' ');

  if isMATLAB
      [yaw, pitch, roll] = dcm2angle(DCM);
      yawd = rad2deg(yaw); pitchd = rad2deg(pitch); rolld = rad2deg(roll);
      disp('Euler angles: '); disp(['yaw: ',num2str(yawd)]);
      disp(['pitch: ',num2str(pitchd)]); disp(['roll: ',num2str(rolld)]);
      disp(' ');
  end
else; clc
    disp(['No outoput file was found. Estimator didn''t create ',ptTS]);
    disp(' '); end
```