Global Space Exploration Conference, Washington, D.C., United States. Copyright ©2012 by the International Astronautical Federation. All rights reserved.

TRON Tool: REPRESENTING A MOON LANDING SCENARIO IN TRON

Marco Sagliano
DLR German Aerospace Center, Institute of Space Systems,
Navigation and Control Systems Department,
Robert-Hooke-Str. 7, 28359 Bremen, Germany, Marco.Sagliano@dlr.de

Hans Krüger
DLR German Aerospace Center, Institute of Space Systems,
Navigation and Control Systems Department,
Robert-Hooke-Str. 7, 28359 Bremen, Germany, Hans.Krueger@dlr.de

Stephan Theil
DLR German Aerospace Center, Institute of Space Systems,
Navigation and Control Systems Department,
Robert-Hooke-Str. 7, 28359 Bremen, Germany, Stephan.Theil@dlr.de

Over the years, the interest towards new Moon missions has been growing in significant way. If Apollo program opened a new frontier in terms of space exploration, in future ever more countries will look at the Moon as scientific goal of new missions. JAXA, ISRO and ESA foresee manned missions over the next 15 years and several US missions are now under development. Given the costs of these missions, it is of course of vital importance to be able to test extensively the technology that will be used during the real missions in a testbed as close as possible to the inflight conditions. For this reason, DLR`s TRON (Testbed for Robotic Optical Navigation) represents a valid choice to prepare and test such a similar technology. The testbed, sizing approximately 13x3.5x4 m^3, and consisting of a 400W 2DOF lamp mounted on a three-DOF gantry, a KUKA robot with 7 DOF able to move until to 2 m/s, and a dSPACE station for their real-time control, is able to effectively simulate several mission scenarios, with a strong emphasis on the Moon scenarios. To do this, a dedicated Simulink library for the proper conversion of the moon landing trajectories has been developed, and a matlab-based tool has been created to automatize and make more intuitive the conversion process from a generic moon landing trajectory into its corresponding scaled TRON´s trajectory and the signal commands to be transmitted to the lab´s key elements. In this paper, a brief description of the hardware and its use will be provided, the Simulink library and its mathematical development for the trajectory conversion will be shown and the hardware commands derivation will be explained. Moreover, some results deriving from the preliminary results of simulations of lunar trajectories expressed in DCA and MCMF reference frames, in TRON laboratory coordinated, and finally the effective commands for the hardware will be illustrated.

I. ABBREVIATIONS

DOF	Degrees of Freedom
TRON	Testbed for Robotic Optical Navigation
MCMF	Moon-Centered Moon-Fixed Reference Frame
DCA	Downrange-Crossrange-Altitude Representation
L	Full-size Landscape Reference Frame
LS	Scaled Landscape Reference Frame
λ	Scale Factor
WRT	With Respect to

II. INTRODUCTION

The idea behind TRON is to provide an environment where new technologies, (with a particular emphasis on optical navigation technologies) can be deeply tested in controlled conditions. The first scenario that has been implemented is a Moon landing environment, in the framework of ESA`s Next Moon mission, so the problem to properly represent the trajectory in the laboratory has arisen. To do this a matlab-based tool and a dedicated library implementing the algorithms to convert a mission trajectory to TRON representation in order to test sensors and space technologies have been developed. The scale factor for the representation of the trajectory can potentially vary from real scale (1:1) to 1:50000, generating as a consequence a wide selection of opportunities in near and remote future.

Global Space Exploration Conference, Washington, D.C., United States. Copyright ©2012 by the International Astronautical Federation. All rights reserved.

III. TRON UNIT KEYS

TRON (Testbed for Robotic Optical Navigation) is a testbed located in Bremen, Germany, designed to test and validate new space technologies. Its size is approximately 13 x 5 x 3.5 cube meters.

Fig. 1: Overview of TRON

At the moment TRON consists of:

- A 6(+1)-DOF KUKA robot, able to move until 2 m/s and with a maximum payload of 16 kg. The extra degree of freedom is associated to the external axis that allow to move KUKA in TRON for an operative distance of 11 meters.
- A 3-DOF Gantry System that is also able to carry until 150 kg of payload, and can carry the model of an asteroid or a lamp, as in the case of the Lunar Landing Scenario.
- A 2-DOF 575Watt motorized lamp, able to rotate about 400 and 275 degrees (pan and tilt rotations respectively);
- A Laser Tracker system, used to verify the correct alignment of the components of the lab and to measure the movements during the simulations. Its accuracy is below 0.1 mm.
- A dSPACE station to control the key units in the lab. The lamp and KUKA can be controlled via Ethernet communication, while the Gantry system uses a Twinsync communication board, based on the proper set and control of three frequency converters.

IV. REPRESENTATION OF TRAJECTORIES

The realization of the first testing scenario foresees the possibility to have two common representations for a lunar descent trajectory. The former, intrinsically tridimensional, is the MCMF representation (Moon-Centered Moon-Fixed). The latter is the so-called Downrange-Crossrange-Altitude representation, based on a bidimensional "stretched" landscape strip. These two choices, in combination with the type of landscape that can be realized (flatted or with a curvature), give us the possibility to define four combinations:

- MCMF Representation → curve landscape
- DCA Representation → curve landscape
- MCMF Representation→ flat landscape
- DCA Representation → flat landscape

All the possibilities have been treated at simulation level, but the real commands for the hardware will be here derived for the last two, as the flat landscape have been installed in TRON. Let us start with the transformation from DCA to Flat Landscape;

IV.I DCA-FLAT LANDSCAPE

In this case let us consider the initial trajectory in DCA. To represent properly this trajectory it is necessary to apply a linear transformation, which involves a rotation, a translation and the scaling.

$$r_L = M_{DCA}^{L} \cdot r_{DCA} + r_{L\_DCA}$$
$$r_{LS} = \frac{r_L}{\lambda} \qquad [1]$$

where r_L is the full-size landscape reference frame, and r_{LS} is the position of the spacecraft in the scaled lanscape reference frame. λ is the scale factor (λ >=1), M_{DCA}^{L} represents the rotation matrix from DCA to L and r_{L_DCA} is the vector expressed in Landscape coordinates pointing towards the origin of DCA reference frame. For purposes regarding the alignment procedure, the lower-left corner of the landscape has been chosen as origin of landscape reference frame, but it can be obviously placed in any other point. Since the transformation between DCA and L is completely independent on TRON, a Landscape reference frame parallel to DCA has been chosen.

IV.II MCMF – DCA

In order to transform the trajectory from MCMF to LS we need an intermediate transformation. We want first to transform the trajectory from MCMF representation to its Downrange-Cross Range-Altitude components. Let us then consider the generic vector r_{MCMF}, with a particular emphasis on the first and last values (spacecraft at the beginning and at the end of descent phase), that we will call r_0 and r_L.

 Copyright ©2012 by the International Astronautical Federation. All rights reserved.

Let us transform these two positions into their spherical representation that will be useful for deriving the downrange, crossrange and altitude representation of the trajectory. The radius ρ, the right ascension θ (measured wrt the x axis) and elevation φ will be

$$\rho = \sqrt{x^2 + y^2 + z^2}$$
$$\theta = \tan^{-1}\left(\frac{y}{x}\right) \qquad [2]$$
$$\varphi = \tan^{-1}\left(\frac{z}{\sqrt{x^2 + y^2}}\right)$$

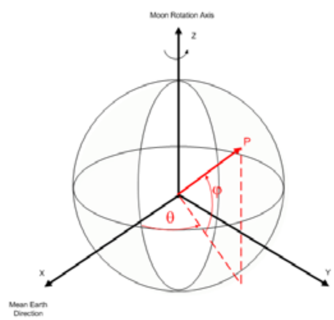

Fig. 2: Transformation to spherical coordinates system

This way we have obtained the values ρ_0, the right ascension θ_0 and φ_0, corresponding to the initial spacecraft position, and ρ_L, θ_L and φ_L, associated to the final position. Now let us compute the angle γ as follows: r_0 and r_L form a plane that is the trajectory plane. The angle between the trajectory plane and the XY plane of the MCMF reference frame is the angle γ we are interested to. So, in order to determine it,

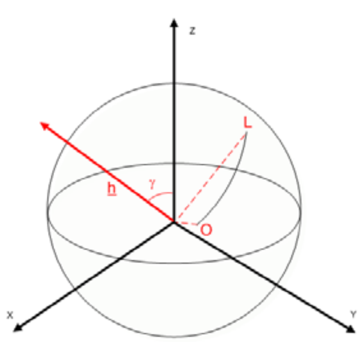

Fig. 3: Definition of angle γ

let´s compute the cross product between these vectors. We will get the normal vector <u>h</u>, forming the same angle with Z axis of MCMF reference frame.

$$\hat{h} = \frac{r_0 \times r_L}{\left|r_0 \times r_L\right|} \qquad [3]$$

We can then determine the angle γ as

$$\gamma = \cos^{-1}\left(\hat{h} \cdot \hat{Z}\right) \qquad [4]$$

We are now able to perform the 321 rotation from the xyz reference frame to x'y'z', when the angles are respectively θ_0, φ_0, and γ; this choice is justified by the definition of DCA reference frame, in which the downrange direction is determined by the trajectory plane of the spacecraft; in other words, the purpose of this transformation is to obtain a new Cartesian reference frame in which the spherical coordinates ρ', θ' and φ' associated to x', y' and z' will be directly proportional respectively to the Downrange, Crossrange and Height components; so, given a generic vector **r**, its representation in this new reference frame will be given by

$$r_{x'y'z'} = M_{321}(\theta_0, \varphi_0, \gamma) \cdot r_{xyz} \quad [5]$$

Where the matrix M_{321} is the well-known 321 rotation matrix

$$M_{321}(\theta,\varphi,\gamma) = \begin{bmatrix} \cos\varphi\cos\theta & \cos\varphi\sin\theta & -\sin\varphi \\ -\sin\theta\cos\gamma + \sin\gamma\cos\theta\sin\varphi & \cos\gamma\cos\theta + \sin\varphi\sin\gamma\sin\theta & \sin\gamma\cos\varphi \\ \sin\theta\sin\gamma + \cos\gamma\cos\theta\sin\varphi & -\sin\gamma\cos\theta + \cos\gamma\sin\varphi\sin\theta & \cos\varphi\cos\gamma \end{bmatrix}$$

[6]

Once derived x', y' and z', with the same transformation introduced in [5] let us compute the spherical reference frame coordinates ρ', θ', φ'; the transformation is perfectly identical to the previous one:

$$\rho' = \sqrt{x'^2 + y'^2 + z'^2}$$
$$\theta' = \tan^{-1}\left(\frac{y'}{x'}\right) \qquad [7]$$
$$\varphi' = \tan^{-1}\left(\frac{z'}{\sqrt{x'^2 + y'^2}}\right)$$

Once obtained this representation of the trajectory, we can observe that the transformation from this reference frame to DCA is given by:

Global Space Exploration Conference, Washington, D.C., United States. Copyright ©2012 by the International Astronautical Federation. All rights reserved.

$$
\begin{aligned}
D &= \rho'(\theta' + n2\pi) \\
C &= \rho'\varphi' \\
A &= \rho' - r_M
\end{aligned}
\quad [8]
$$

where r_M is obviously the radius of the moon, and ***n*** the number of entire orbits necessary to reach the final point. In this way it is possible to transform whatever trajectory from MCMF to the flat scaled Landscape simply building the chain from MCMF to DCA and from DCA to LS. The only limit to this conversion is given by the periodicity of the transformation from Cartesian to spherical coordinates. In case the treated trajectory is longer than one lunar orbit, a periodic coefficient has been added to the Downrange component.

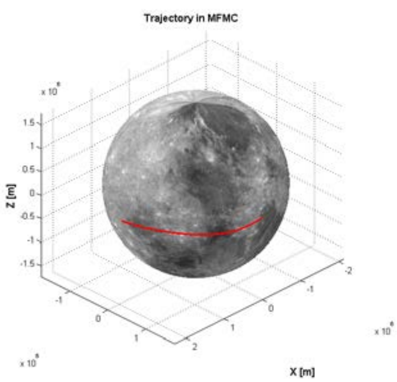

Fig. 4: Trajectory Representation in MCMF coordinates

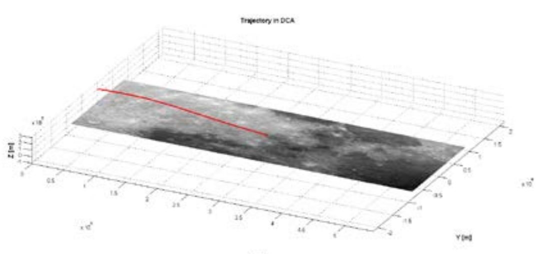

Fig. 5: Trajectory Representation in DCA coordinates

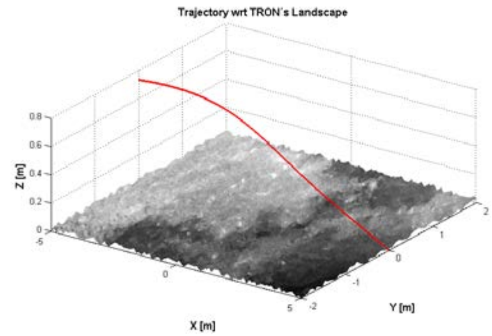

Fig. 6: Trajectory Representation in TRON´s landscape coordinates

IV.III <u>Tailoring the trajectories for TRON</u>

In more general case the trajectory cannot be represented as a single segment even with a scale factor of 50000. In this case the trajectory is then split into several segments that can be simulated and tested in TRON. The ensemble of the clusters of points can be easily generated with TRONTool, always guaranteeing that the single segment is compatible with the operative dimensions of TRON and that the footprint of the camera is on the landscape. All the information related with the segments of the trajectory are then automatically stored in TRONTool for further analysis or postprocessing operations.

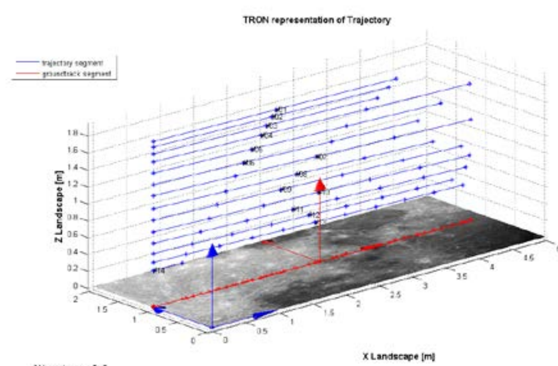

Fig. 7: Example of split trajectory on TRON´s landscape

V. <u>TRONTOOL</u>

In order to manage the trajectories, to generate TRON inputs, to store the output, and to visualize the segments of trajectories, a dedicated tool, TRONTool, has been developed. With TronTool the original trajectory can be analysed, split into its segments (as previously shown), and visualized.

Copyright ©2012 by the International Astronautical Federation. All rights reserved.

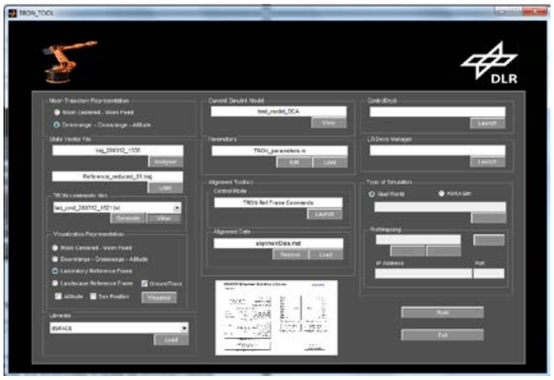

Fig. 8: TRONTool main interface

TronTool generates the segments, that can be visualized in the original reference frame (MCMF/DCA), in TRON reference frame, (used for example to control the lab) or in landscape reference frame. Moreover, it is possible to visualize the groundtrack of the trajectory, as well as the orientation of body reference frame.

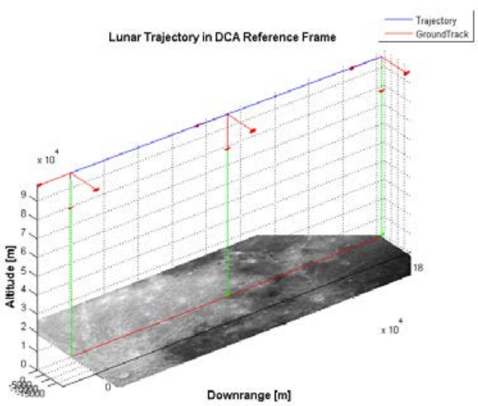

Fig. 9: Visualization of Trajectory in TRONTool – DCA reference frame

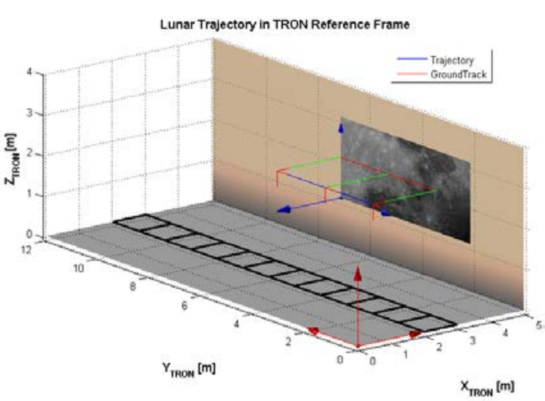

Fig. 10: Visualization of Trajectory in TRONTool – TRON reference frame

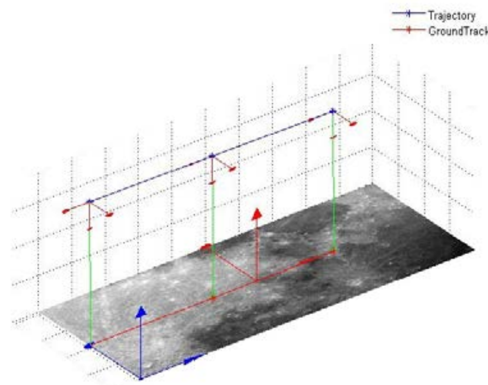

Fig. 11: Visualization of Trajectory in TRONTool – LS reference frame

VI. COMMANDS GENERATION

TRONTool is also able to generate the inputs for the key units of the lab, that are KUKA robot, the gantry and the lamp. Three ways for controlling the lab have been implemented, and consequently, three kinds of inputs can be generated.

- Pure units commands: in this case the proper commands for each unit are specified in terms of their internal reference frames. (E.g. pan and tilt angles for the lamp); in this case there is no explicit coupling between the units (e.g. the lamp and the gantry move independently);
- Lab commands: in this case all the commands are related to TRON reference frame, and are specified for spacecraft, sun position and footprint of the camera (there is coupling between gantry and lamp in order to be sure that the camera and the lamp are looking at the same point);
- Landscape commands: also in this case coupling is considered, but looking at the perspective of the landscape, more intuitive in terms of verification and analysis;

The commands are transmitted via XML interface and are executed in stop motion. This means that for each commanded position there is a pause of ~2 seconds before measurements are taken in order to be sure that all the vibrations will be properly damped.

In the figure below it is possible to see an example of commands for KUKA robot, where the commanded positions are the ones marked with circles and the continuous lines represent the effective movements of the robot.

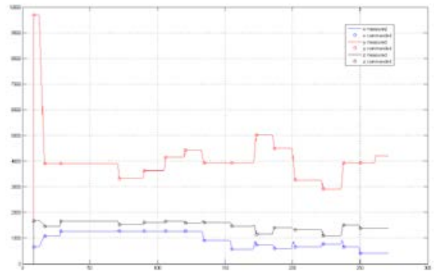

Fig. 12: Stop-Motion Commands example

VII. FURTHER WORK AND CONCLUSIONS

In this paper a brief description of DLR´s TRON laboratory has been given, and the algorithm to represent a generic lunar landing/ descending trajectory in TRON has been shown. An overview on TRONTool capability of analysis of trajectories and generation of coherent commands for the key units has been given as well. Further developments include the possibility to implement the continuous motion alternatively to the stop motion, and the process to tailor TRON to other scenarios, such as the Asteroid Landing. This can be for instance done in terms of hardware / laboratory building a 3d model representing the scaled asteroid to put in the lab, and in terms of software developing new libraries that allow to treat effectively such a similar scenario. Moreover, some features to automatically verify / update the correct alignment of the components of the lab is already in phase of implementation.

REFERENCES

[1] Wertz, J. R. (ed.), Spacecraft Attitude Determination and Control, Springer–Verlag, New York, 1978