AN APPLICATION OF THE SITUATIONAL SOLVER TO PROBLEMS CONTAINING CONDITION RELATED TO VISIBILITY OF THE MOON BY A SATELLITE

Atanas Atanassov

Space Research and Technology Institute – Bulgarian Academy of Sciences
e-mail: At_M_Atanassov@yahoo.com

Keywords: *Situational analysis; Orbital events; Moon-Earth shadow; Space mission analysis and design*

Abstract: *Computer simulations play an increasing role/importance in space mission design. Different conditions (visibility of the object, illumination from the Sun or the Moon, radiation background) determine the performance of different measurements with optical instruments. Various optical instruments make measurements when satellites are in the shadow of the Earth. In the absence of sunlight, it may be necessary to have no additional light pollution caused by reflected light from the Moon. This means imposing an additional condition that the Moon is below the visible horizon (behind the Earth) relative to a given satellite.*

Analogously to a model of the Earth's shadow concerning the Sun, a conical model of shadow was developed concerning the Moon as a source of reflected light. This geometric model is used for a situational condition for situational analysis purposes. Optimization has been made for application in multi-satellite analysisis.

ПРИЛАГАНЕ НА ПРОЦЕСОР ЗА СИТУАЦИОНЕН АНАЛИЗ ПРИ РЕШАВАНЕ НА ЗАДАЧИ СЪДЪРЖАЩИ УСЛОВИЕ СВЪРЗАНО С ВИДИМОСТ НА ЛУНАТА ОТ СПЪТНИК
Коничен модел на Лунно-Земна сянка

Атанас Атанасов

Институт за космически изследвания и технологии – Българска академия на науките
e-mail: At_M_Atanassov@yahoo.com

Ключови думи: *ситуационен анализ; орбитални събития; анализ и проектиране на космически мисии*

Резюме: *Компютърните симулации играят нарастваща роля при проектиране на космически мисии. Различни условия (видимост на обекта, осветяване от Слънцето или Луната, радиационен фон) са определящи за провеждането на различни измервания с оптични инструменти. Различни оптични инструменти извършват измервания когато спътници се намират в условията на земната сянка. При отсъствието на слънчева светлина, може да е необходимо да няма и допълнително светлинно замърсяване от отразена от Луната светлина. Това ще рече да се наложи допълнително условие Луната да се намира под видимия хоризонт (зад Земята) спрямо даден спътник. Алгоритъма е оптимизиран за прилагане при много-спътникови анализи.*

Аналогично на модел на земната сянка по отношение на слънцето е разработен коничен модел на земна сянка и по отношение на Луната, като източник на отразена светлина. Този геометричен модел се използва за ситуационно условие за целите на ситуационен анализ.

1. Introduction

Space mission analysis and design are essential at different stages of preparation and implementation of satellite missions [1]. An essential part of the space missions' design and implementation is the so-called situational analysis. The situational analysis deals with determining optimal time intervals, suitable for the execution of satellite operations related to measurements and real-time experiments. The determination of such time intervals demands knowing the satellite trajectories in consecutive moments and checking different situational conditions. The situational conditions are related to the objects of investigation (disposed on a planetary surface, celestial sphere or specific place of space), visibility and illumination of the objects. This type of analysis is applied to different stages of missions' preparation- starting with the conceptual study and preliminary analysis, going through missions' definition, design and development, and finishing with implementation.

Algorithms and calculation tools for space mission analysis and design are under development at the Space Research and Technology Institute at the Bulgarian Academy of Sciences. One such tool is a parallel solver for situational analysis [2]. Algorithms and program realization of the conical model of the Earth's shadow are discussed in the present work. Algorithms suitable for multi-satellite analysis are proposed.

2. Preliminary work

A parallel situation analysis solver was developed to determine time intervals when certain conditions of a geometric or physical nature are met. The goal was to solve a large number of situational problems, each with more than one independent condition. Some basic situational conditions have been developed, and the goal is to increase their number. This makes it possible to apply the situational processor in simulations of space missions using different scientific instruments and solving different scientific problems.

2.1. Situational task description model

A descriptor of situational tasks is a one-dimensional array whose elements are derived types containing the values of different attributes (parameters and constraints as well as results) of the conditions comprised in the task. The first (zero) element of the descriptor contains control information and results about the entire situational task. The following elements contain the values of different attributes (parameters and constraints as well as results) of the situational conditions.

2.2. Parallel situational analysis solver

A parallel solver for situational analysis was developed for this purpose. It is a processing program that consistently checks the feasibility of the conditions in a particular situational task. The parallelization is based on computational threads organized in a variant of the program model "pool of threads" [2]. In this variant, the threads are synchronized with each other while they receive situational tasks to solve (race condition). This excludes the solution of one situational task by more than one thread. The threads are also synchronized with the parent thread, which initiates the calculations and waits for them to complete at each step of the simulated time. Each thread takes one or more situational tasks for processing, according to the specified value of the parameter known as granularity.

2.3. Situational tasks designer

A dialogue editor of situational tasks is developed as an auxiliary tool. The compilation of a situational task is initiated, situational conditions are successively selected, and their respective parameters and restrictions are set with the help of dialogue controls. An already compiled situational task can be rejected or approved and saved as a template for future use. One already assembled situational task can be related to a particular satellite or all satellites of the respective multi-satellite system.

3. Conical model of the Lunar-Earth/earth's shadow concerning the moon

Usually, the "shadow of the Earth" is spoken of to the Sun, as the source of light. Similarly, we can consider a geometric shadow pattern when the moon is the source of reflected light (Fig. 1). Below, by "shadow of the Earth" we will understand a shadow caused by the Moon.

The lines $\overleftrightarrow{t_1}$ and $\overleftrightarrow{t_2}$ are tangent at the points $(\mathrm{T_{M,1}}, \mathrm{T_{M,2}})$ and $(\mathrm{T_{E,1}}, \mathrm{T_{E,2}})$ to the "Moon" and the "Earth " respectively in a plane defined by the vectors $\vec{R}_{ME}$ and $\vec{r}_{sat}$. From the similarity of the triangles $\Delta \mathrm{O}_M T_{M,1} H_M$ and $\Delta \mathrm{O}_E T_{E,1} H_M$ a formula can be derived for the length of the line segment $\overline{\mathrm{O_M H_M}} = \mathrm{H_M}$ which is analogous to the height of the conical shadow of the Earth caused by the Sun:

$$\mathrm{H_M} = \frac{\mathrm{R_E} . |\vec{R}_{ME}|}{\mathrm{R_M} - \mathrm{R_E}}.$$

In this formula, $\mathrm{R_E}$ denotes the Earth's radius, the distance between the Moon and the Earth **(**the magnitude of the vector $\vec{R}_{ME}$**)** varies during the synodic period in the interval from 363 104 km to 405 696 km**.** The length of the quantity $\mathrm{H_M}$, which is equivalent to the length of Earth's shadow and can be assumed to be a slowly changing quantity.

The next step is to determine whether the satellite is in the shadow of the Moon-Earth. One condition for this is that the sub-satellite point is located in the unlit part of the earth's surface (necessary condition). It is equivalent to checking if $\measuredangle \mathrm{H_M O_E T_{E,2}} < \measuredangle \mathrm{H_M O_E S}$:

$$\frac{\vec{R}_{ME}.\vec{r}_{sat}}{|\vec{R}_{ME}|\,.\,|\vec{r}_{sat}|} > \frac{\overline{\mathrm{O_M T_{M,2}}}}{\mathrm{H_M}},$$

because it compares the cosines of the indicated angles on both sides of the inequality without the use of arccosines.

To determine whether the satellite is in the shadow of the Moon-Earth, it remains to compare the segments $\overline{S'S''}$and $\overline{S'S}$ on the line $\overleftrightarrow{\mathrm{d}}$, perpendicular to the line $\overleftrightarrow{\mathrm{o}}$. From the similarity of the triangles $\Delta \mathrm{H_M} S'S''$and $\Delta H_M T_{M,2} O_M$ it follows:

$$\overline{S'S''} = \mathrm{R_M}.\frac{\mathrm{H_M} + R_{ME} + \overline{O_E S'}}{\overline{T_{M,2}H_M}} = \mathrm{R_E}.\frac{\mathrm{H_M} + R_{ME} + |\vec{r}_{sat}|.\cos(\measuredangle S'O_E S)}{\sqrt{\mathrm{H_M}^2 - \mathrm{R_M}^2}}.$$

The length of the line segment $\overline{T_{M,2}H_M}$, a leg in the right triangle $\Delta H_M T_{M,2} O_M$, is determined based on the length of the Moon-Earth's shadow $\mathrm{H_M}$ and the magnitude of the radius of the Moon $\mathrm{R_M}$ or finally:

$$\overline{S'S''} = \mathrm{R_M}.\frac{\mathrm{H_M} + R_{ME} + r_{sat}.\dfrac{\vec{R}_{ME}.\vec{r}_{sat}}{\mathrm{R_{ME}}\,.r_{sat}}}{\sqrt{\mathrm{H_M}^2 - \mathrm{R_M}^2}}.$$

or

$$\overline{S'S''} = \mathrm{R_M}.\frac{\mathrm{H_M} + R_{ME} - \vec{e}_{ME}.\vec{r}_{sat}}{\sqrt{\mathrm{H_M}^2 - \mathrm{R_M}^2}}$$

where $\vec{e}_{ME}$ is the unit radius vector of the Moon.

The length of $\overline{S'S}$ is determined by $\Delta O_E S'S$**:**

$$\overline{S'S} = |\vec{r}_{sat}|.\,sin\measuredangle S'O_E S$$

or

$$\overline{\mathrm{S'S}} = \mathrm{r_{sat}}.\sqrt{1 - \left(\frac{\vec{R}_{ME}.\vec{r}_{sat}}{|\vec{R}_{ME}|\,.\,|\vec{r}_{sat}|}\right)^2}.$$

The comparison between the sizes of the sections $\overline{S'S''}$ and $\overline{S'S}$ gives a final solution to the problem of the satellite falling into the Moon-Earth shadow.

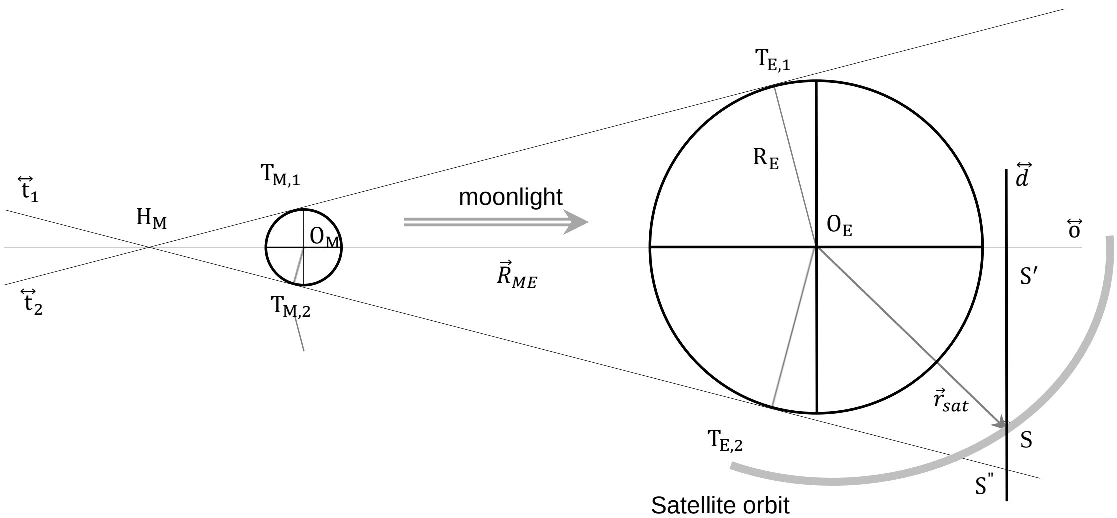

Fig. 1. Scheme of the Moon-Earth configuration and shadow zone

4. Program implementation

The checking of the situational condition is done in the Sit__76 subroutine written in the Fortran language (Appendix B). Calculations are performed on the model presented in chapter 3. To ensure parallel computations, all used local variables are declared in the **AUTOMATIC** statement.

In simulations of multi-satellite missions, some calculations about the geometry of the Moon-Earth's shadow are the same for all satellites. For this reason, the subroutine named **Preliminary_Calculations_Moon** is added to increase computational efficiency. It performs some calculations that depend on the Moon-Earth distance. Some subroutines (for determining the distance between the Earth and the Moon, as well as for a Julian day) are taken from [3]. The subroutine **If_Flag** has control functions and is called by all situational conditions functions.

5. Example of application

As an illustration of the conic model under consideration, figure 2 shows a timing diagram of the passage of a satellite through the shadow of the Earth with the source Sun and the shadow with the source Moon. Also shown is a passage over a circular area of the earth's surface with a latitude of the center 20° and an angular radius 10°. The orbit of the satellite has a semi-major axis a= 7200 km, eccentricity e= 0.001 and inclination i= 45°.

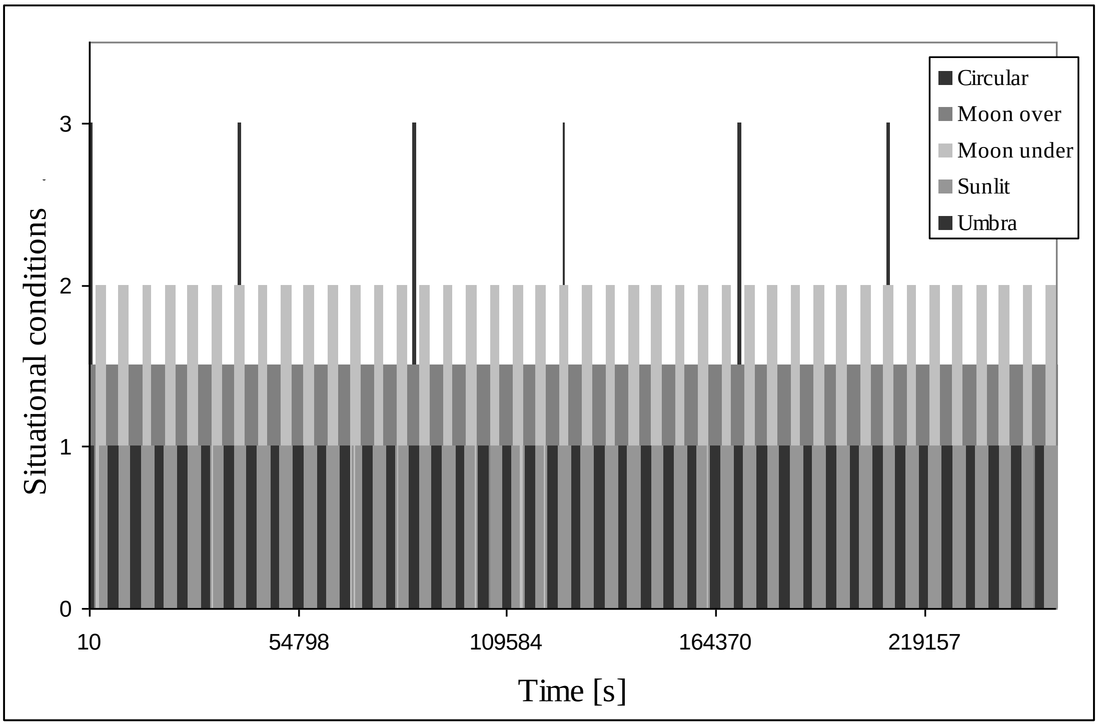

Fig. 2. Time diagram of the situational conditions. “Moon under” the earth's horizon; “Moon over” the earth's horizon relative to the satellite; the satellite is over the “Circular” region, the satellite is in “Umbra” and the satellite is in the “Sunlit“ region around the Earth.

6. Conclusion and future work

Based on the Moon-Earth shadow model, a situational condition has been developed for the purposes of situational analysis. Incorporating this condition into situational tasks along with other situational conditions is practically important for solving remote sensing tasks as well as other areas of space exploration. Optimization methods of situational analysis are under progress for such tasks that contain more conditions.

References:

1. Wertz, J. R., Larson W. J., Space Mission Analysis and Design. Microcosm Press, Kluwer Academic Publishers, third ed., 1999.
2. Atanassov, A. M., 2016. Parallel satellite orbital situational problems solver for space missions design and control, *Advances in Space Research*, v. 58, 9, pp. 1819–1826.
3. Vallado, D. A., 2013 Fundamentals of Astrodynamics and Applications. Microcosm Press, third ed.

Appendix A. Situational condition descriptor

```
type      SitCond
  integer        sit_code      ! code of the situation condition; every situation have some code
  integer        sat_num      ! which satellite concern this situation task
  logical        flag            ! satisfaction of sit.cond: .false. or .true.
  logical     begin_sit        ! local sit.cond parameter
  logical     fl_rezults       ! if .true. - flag za kraj na situacija i gotovi rezultati
  real*8      t12(2,3)         ! Start and final times when the condition is meet
  real       duration           ! duration of a current situacional condition/event
  real          dt_sit           ! acumulates "duration" of sit.cond before ending
  real      t_cond_total       ! acumulates total durations for the hole observational period
 UNION
 …
   MAP                         ! Sit_76: The Moon is under horizon
      logical     Moon_under       ! Moon's "umbra"
      logical     Moon_horizon      ! Moon on horizon (rise/set)
      logical     Moon_over        ! Moon's light
   END MAP
 …
 END UNION
end type  SitCond

TYPE     sit_task
 UNION
  MAP                         ! Only for sit.tasks control- contain number of situation conditions
     integer     SP_code          ! Contains serial number of the situation
     integer     SP_type          ! Contains unique number of situation task type
     integer     max_cond        ! Contains the number of conditions in the task
     logical    requirement       ! satisfaction of sit.task: .false. or .true.
     integer     opt_level        ! Optimization algorithm: 0- none, 1/2/3
     logical    begin_sit
     logical    fl_rezults        ! if .true. - end of interval
     real*8      t1,t2              ! Start and final times when the task is meet
     real       duration,dt_sit   ! duration of the time interval
     real      t_task_total       ! accumulates total durations for the hole simulation period
     integer     aim_code          ! kod na cel za situacijata
  END MAP
  MAP
    type (SitCond) sit_cond
  END MAP
 END UNION
END TYPE  sit_task
```

Appendix B. Program code of the situational condition

```
FUNCTION     Sit__76(t,dt,xv,Moon_under,Moon_horizon,Moon_over,fl_rezults, &
                                                duration,begin_sit,dt_sit,t12)
 USE  DFlib
 logical     Sit__76, Moon_under,Moon_horizon,Moon_over,fl_rezults, begin_sit
 real                                          duration,                t12*8(2,3)
 real*8            t,dt,xv(3)
!_______________________________________________________________________________
 logical    flag
 real*8      Re/6378.388D3/,Rm_Moon/1737.4D3/
 real*8           Rm_M,cd_M(3) ! modul na rad.vek.; cosinus directions
 real*8       R_EM,H_shad,cos_psi,cos_psi0,S1S,S1S2,p
 real*8                xyz_Moon(3)!,rectasc,decl,JD,JD_frac
 common     /cMoon_Vall/xyz_Moon
 AUTOMATIC     flag,cos_psi,cos_psi0,p,S1S,S1S2
```

```
!.........................................

  r_sat= SQRT(xv(1)**2 + xv(2)**2 + xv(3)**2) ! a module of |radius - vektor| of the satellite
       cos_psi = (     xyz_Moon(1)*xv(1) + xyz_Moon(2)*xv(2) + xyz_Moon(3)*xv(3))/ &
            (SQRT(xyz_Moon(1)**2    + xyz_Moon(2)**2    + xyz_Moon(3)**2 )*r_sat)
a1:IF(cos_psi.LT.cos_psi0) THEN ! necessary condition checking
            p = -r_sat*cos_psi;
        S1S2= (R_EM + H_shad +  p )*Rm_Moon/SQRT(H_shad**2 - Rm_Moon**2)
        S1S= SQRT((r_sat - p)*(r_sat + p));
  a2:IF(S1S.LT.S1S2) THEN; flag=.true.
                                    Moon_under=.true.;  Moon_over=.false.
                                     ELSE;  flag=.false.
           ENDIF a2
                Moon_under=.false.; Moon_over=.true.
                  ELSE; flag=.false.
          Moon_under=.false.; Moon_over=.true.
      ENDIF a1;

      CALL  If_Flag(dt,Sit__76,flag,fl_rezults,duration,begin_sit,dt_sit,t12)

RETURN
ENTRY   preliminary_calculations_Moon()

  R_EM  = Re     *SQRT(xyz_Moon(1)**2 + xyz_Moon(2)**2 + xyz_Moon(3)**2)  ! Earth-Moon distance
  H_shad= Rm_Moon*R_EM/(Re - Rm_Moon);
cos_psi0= Rm_Moon/H_shad ! Neobhodimo uslovie

END FUNCTION  Sit__76
```