SITUATION ANALYSIS BASED ON SPATIOTEMPORAL SITUATION CONDITIONS

Atanas Atanassov

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

Keywords: *satellite situation analysis, situation condition, constraints satisfaction problems*

Abstract: *The situation analysis is important tool applied at different stages of preparation and implementation of space missions. The situation analysis establishes the time/temporal intervals, when specific conditions/constraints, (which are very important for every particular experiment, measurement, solving scientific or technological problem) are met.*

Spatiotemporal conditions are involved in the present work for more adequate definition of situation conditions, depending on specific characteristic temporal intervals, related to nature of investigated processes and phenomenons.

МНОГОСПЪТНИКОВ СИТУАЦИОНЕН АНАЛИЗ ВКЛЮЧВАЩ И ВРЕМЕТО КАТО ДОПЪЛНИТЕЛНО ОГРАНИЧЕНИЕ

Атанас Атанасов

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

Резюме: *Ситуационния анализ е важно средство на различни етапи от подготовката и провеждането на космически експерименти. С него се установяват времевите интервали в рамките на които се изпълняват условия, съществени за провежданите експерименти и измервания и решаваните научни и технологични задачи.*

В настоящата работа се въвеждат пространствено-времеви ситуационни условия, чрез които се цели по-добро дефиниране на условията за провеждане на експерименти и измервания, в зависимост от времевите характеристики и природата на изучаваните процеси и явления.

Introduction

Situation analysis is substantial on different stages of space mission preparation and realization [1]. Analysis of potential orbital events with participation of satellites from different space missions can be very useful [2, 3]. Such analysis could be useful and after completion of various space missions, when data measured from instruments disposed on different satellites are collated.

Program system for analysis and simulations of multi-satellite space experiments is under development at branch of Space Research and Technology Institute, in Stara Zagora [4]. Parallel processor for situation problems solving is incorporated in frames of this system [5]. Possibilities for space-time situation conditions definition are shown in the present paper.

Essence of situation analysis

Situation analysis aims at finding of time intervals, when specific conditions are met, which are suitable for conducting observations, measurements, experiments and in general, satellite operations.

Situation conditions represent different constraints related to:

– observation conditions for the measurements; evaluation of possible side effects of disturb factors – Sun, Moon, radiation background.

– objects of observation, which can be (relatively) static or moving on Earth's surface.

– Измервания на параметрите на средата (атмосфера, йоносфера) (in-situ или с дистанционни методи).

– measurements of environmental parameters (of the atmosphere, ionosphere, magnetosphere) in-situ or by remote sensing.

In most cases, simultaneous realization of several conditions is necessary for one measurement related to specific scientific problem. These conditions can be presented through logical predicate functions and so, the check can give false or true. Then we can write down for general expression of situation problem:

$$\mathrm{SP} = \mathrm{s}_1(\alpha_1, \beta_1, \mathrm{t}) \vee \mathrm{s}_2(\alpha_2, \beta_2, \mathrm{t}) \vee \ldots \vee \mathrm{s}_\mathrm{N}(\alpha_N, \beta_\mathrm{N}, \mathrm{t}),$$

where $\{\alpha\}$ and $\{\beta\}$ are sets of parameters and constraints for each of situation condition participating in the situation problem SP.

The aim of the situation analysis is determination of time intervals optimal for space experiments and measurements execution. The optimal time intervals are determined according to specific situation conditions having geometrical or physical nature and their appropriate parameters.

Sometime it is possible one object to be observed with instruments disposed on more than one satellite. Even satellites can be of different missions, which implementation coincides in time. The same measurements can be performed with different kind of instruments.

Some instances about introduction of time in situation conditions

Usually in multi-satellite situational analysis situational conditions for different satellites are checked against a single point in time. This is strong restriction, in some cases when the processes are relatively slow. It may be possible to compare data from different satellite missions obtained in the framework of a specific interval of time, during which the studied subject is changing slowly.

The following instances of insertion of the time in situation conditions are possible:

– two satellites pass over the same point on the Earth's surface (crossing of satellites traces) in frame of time interval $\tau = \left|\mathrm{t}_{\mathrm{s}_1} - \mathrm{t}_{\mathrm{s}_2}\right|$ shorter than some characteristic time τ^*.

– two satellites pass over circular area of Earth's surface, defined with center coordinates (φ,λ) and angular radius θ, one of the two satellite pass later with delaying τ.

– two satellites pass over two different circular areas on Earth's surface defined with geographic coordinates of their centers and angular radii $(\varphi_1, \lambda_1, \delta_1; \varphi_2, \lambda_2, \delta_2)$ in the term of time interval τ^*. This situation condition can be connected with geographic area, which are interesting from geophysical point of view (like areas over north and south cusps).

– traces of two satellites are close enough, so that the satellites pass over close points, at angular distance $\varphi<\varphi^*$ in the frame of time interval $\tau < \tau^*$.

The follow definition is possible to be writing down for the last example:

$$\gamma_1 = \left| \begin{array}{l} \varphi_{t_1,t_2} < \varphi^* - \text{without delaying} \\ \min(\varphi_{t_1,t_2-k.\Delta t}, \varphi_{t_2,t_1-k.\Delta t}) < \varphi^*; \mathrm{k}_{\max} = \dfrac{\tau^*}{\Delta t}; \mathrm{k} = \overline{1, \dfrac{\tau^*}{\Delta t}}; \varphi_{t_1,t_2} > \varphi^* - \text{with delaying} \end{array} \right.$$

Where $\mathrm{t}_1 - \mathrm{k}.\Delta\mathrm{t}$ and $\mathrm{t}_2 - \mathrm{k}.\Delta\mathrm{t}$ are series of sequential moments for the two satellites respectively, φ^* is maximal angular distance, which appear as angular constraint for the particular situation condition. Applying "delaying" in the time, angular distance φ_{tt} (where t- current time) is larger than threshold constraint angular distance φ^*. Δt is step in time and τ^* maximal delaying time for one of the two satellites.

Specifics in development of space-time situation conditions

These conditions need memory storage, where values of appropriate variable in consequent moments in time are stored. These values are analyzed at every step in the time. A lot of situation problems may contain conditions, which have same template but different parameters and constraints. All of these parameters and constraints are stored according to situation condition definition model. Characteristic time τ^* is also specific for each space-time situation condition. This time determines the size of allocated array. Dynamic storage with appropriate size is allocated for each spatial-timed situation condition in one situation problem. The address and size of this memory are stored in two specific parameter of the situation condition (fig. 1b). The working data necessary for one situational condition included in one situational problem are placed in specific place (address) in dynamic storage, which is allocated from operation system after request, at suitable place in the program.

```
type     sit_task
 UNION
  MAP              ! Only for control- contains number of situation conditions
    …              !
  END MAP
  MAP
    type (SitCond) sit_cond     ! situation conditions description templete- parameters
  END MAP
 END UNION
end type sit_task                                                          (a)
```

```
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          ! satisfied sit.cond: .false. or .true.
  logical     begin_sit     ! local sit.cond parameter
  logical     fl_rezults    ! if .true. - flag for end of situation and ready results
  real*8      t12(2,3)      ! contains the last time interval when the sit.cond was satisfied
  real        duration      ! duration of a current situational condition/event
  real        dt_sit        ! local sit.cond parameter- accumulates duration of sit.cond before ending duration
  real        t_cond_total  ! local sit.cond parameter- accumulates total durations for the hole observational period
 UNION
  MAP                 ! Sit__1: Pass over circular region of Earth surfase with centre (lati,longi)
   …
  END MAP
   …
  MAP                 ! Sit__15: two satellites pass over close points on the Earth surface with delaying τ
   integer    sec_sat_4           ! secondary satellite
   real          lati_4           ! latitude of circular region
   real         longi_4           ! longitude
   real         angle_4           ! angle of region
   Integer    address_4,   num_4  ! local storage for condition and length (moments stored)
   real        angle_4,delay_t_4
   logical       flag_4
  END MAP
  MAP                    ! another situation condition descriptions
   …
  END MAP
 END UNION                                                                     (b)
end type  SitCond
```

Fig. 1. a). general template for situation condition; b) attributes of a presented situation condition

Source code of situation condition

Situation condition related to closeness of satellites' trajectories (traces on Earth's surface) is explained in the present paper. Description model of situation condition and program realization, containing different attributes (parameters and constraints), are shown. Other situation conditions containing a time as parameter could be developed in a similar manner.

Figure 1a illustrates general template for definition of situation conditions [4]. Figure 1b illustrates template of the given situation conditions. The template contains attributes general for all

```
FUNCTION     Sit__15(t,dt,xv1,adr_xv2,angle,delay_t,address,num,fl_rezults,duration,begin_sit,dt_sit,t12)
 USE  DFlib
  logical        Sit__4,                                    fl_rezults,         begin_sit
  integer                   adr_xv2,              address
  real                                  angle,delay_t,                    duration
  real*8              t,   xv1(3),xv2(3),                                               t12(2,3)
!_______________________________________
 logical    flag
 integer    addressx1,addressx2,addresst1,addresst2
 real*8    x1(3,num),x2(3,num),t1(num),t2(num),t1_min,t2_min,dt,dt_min, mul,a1,a2,a3,b1,b2,b3,  a(3),b(3)
 real*8,parameter ::  Rz=6371.D3,pi=3.141592654,  grrad= pi/180.D0
 AUTOMATIC  cos_angle_s1,cos_angle_s2,cos_angle,t_min1,t_min2,flag,cosmax1,cosmax2
  POINTER(addressx1,x1); POINTER(addresst1,t1); POINTER(addressex,index)
  POINTER(addressx2,x2); POINTER(addresst2,t2); POINTER(addressek,indek); POINTER(adr_xv2,xv2)
    mul(a1,a2,a3,b1,b2,b3)= a1*b1 + a2*b2 + a3*b3
        addressx1= address
        addressx2= address    + 8*3*num;     addresst1= addressx2 + 8*3*num
        addresst2= addresst1 + 8     *num;     addressex= addresst2 + 8     *num;          addressek= addressex + 4;
L1:  IF(index.EQ.num) THEN
      x1(:,1:num-1)= x1  (:,2:num); x2(:,1:num-1)= x2 (:,2:num); t1(  1:num-1)= t1 (  2:num); t2(  1:num-1)= t2 (2:num)
      x1(:,     index)= xv1(:);           x2(:,    index)= xv2(:);           t1(      index)= t;                 t2(       index)= t
 ELSEIF(index.LT.num) THEN;
     index=             index + 1; indek= index
      x1(:,     index)= xv1(:);           x2(:,    index)= xv2(:);           t1(  index)= t;                    t2(      index)= t
 ENDIF  L1; flag=.false.
              i= index; cos_angle_s2= COS(angle*grrad); t_min2= t; cosmax1=.0
L2:  DO  j=indek,1,-1
       cos_angle= (mul(x1(1,i),x1(2,i),x1(3,i),x2(1,j),x2(2,j),x2(3,j)))/   &
            (SQRT(mul(x1(1,i),x1(2,i),x1(3,i),x1(1,i),x1(2,i),x1(3,i)) * mul(x2(1,j),x2(2,j),x2(3,j),x2(1,j),x2(2,j),x2(3,j))))
        IF(cos_angle.GT.cos_angle_s2.AND.ABS(t-t2(j)).LE.delay_t) THEN
                          cos_angle_s2=   cos_angle; t_min2=   t2(j); flag=.true.
          ENDIF
    END DO  L2
      j= indek; cos_angle_s1= COS(angle*grrad); t_min1= t
L3:  DO  i=index,1,-1
       cos_angle= (mul(x1(1,i),x1(2,i),x1(3,i),x2(1,j),x2(2,j),x2(3,j)))/   &
           (SQRT(mul(x1(1,i), x1(2,i),x1(3,i),x1(1,i),x1(2,i),x1(3,i)) * mul(x2(1,j),x2(2,j),x2(3,j),x2(1,j),x2(2,j),x2(3,j))))
        IF(cos_angle.GT.cos_angle_s1.AND.ABS(t-t1(i)).LE.delay_t) THEN
                          cos_angle_s1=   cos_angle; t_min1=   t1(i); flag=.true.
          ENDIF
    END DO  L3
L3:IF(flag) THEN
      IF(.NOT.begin_sit) THEN
                        begin_sit=.true.; fl_rezults=.false.
                   IF(cos_angle_s2.GT.cos_angle_s1) THEN
                        begin_sit=.true
                          t12(1,1)= t_min2; t12(1,2)= t_min1; t12(1,3)= t_min2;
                                   ELSE
                          t12(1,1)= t_min1; t12(1,2)= t_min1; t12(1,3)= t_min2
                     ENDIF; dt_sit=.0
                            ELSE
                          t12(2,1)= t; t12(2,2)= t_min1; t12(2,3)= t_min2
        ENDIF
         Sit__4=.true.; dt_sit= dt_sit + dt
           ELSE
     IF(begin_sit) THEN
         begin_sit=.false.;  fl_rezults=.true.; duration= dt_sit-dt; dt_sit=.0
                ELSE
                            fl_rezults=.false.; duration=.0;
        ENDIF
         Sit__4=.false.;
     ENDIF  L4
END FUNCTION  Sit__15
```

Fig. 2. Source code of the subroutine Sit__15 for calculation situation condition

situation conditions as well as specific one. Such specific attributes are identification numbers of satellites, addresses of allocable storage containing coordinates and angular distance between points of the traces (look above).

Allocated storage contains coordinates of the two satellites for series of last M moments in the time. For the number of moments M we may write down:

$$M = \frac{\tau}{\Delta t},$$

where τ is delay time, and Δt is step in the simulation time. The dialogues for definition of situation and dynamic problems as well as step in simulation time are two mutually independent acts. Because of that, determination of the values M_i for all situation problems, which contain delayed situation conditions, must be made before start of simulation.

Figure 2 illustrates source code about presented situation condition. Operators from type pointer are used for connecting working data to specific addresses. IF - THEN construction, labeled with L1, is used for accumulation of working data. DO operators, labeled with L2 and L3, check situation condition on series of data for each of satellites $\min(\varphi_{t_1,t_2-k.\Delta t}) < \varphi^*$ or $\min(\varphi_{t_2,t_1-k.\Delta t}) < \varphi^*$ The last IF – ENDIF construction, labeled with L4, is used for lock - unlock situation analysis control and fixation of initial and final moments of time.

Conclusion

The introduction of the time as a parameter in the situation analysis and its connection with temporal parameters reflecting dynamical characteristics about the nature of the studied objects is an interesting problem. Such an analysis would increase the possibilities as regards the experiment (searching and finding of the additional time intervals appropriate for the measurement, the extraction of further information), and in terms of a fuller use of data from completed missions.

Joint measurements made with instruments located on different satellites can be facilitated by the proposed analysis. Other situation conditions containing time constraint from types listed above will be realized. Widening of the class of space-time situation conditions from geometric to physical nature is upcoming.

References:

1. Wertz, J. R., Larson, W. J., 1999. Space Mission Analysis and Design, third ed. Microcosm Press, Kluwer Academic Publishers.
2. Nazirov, R., & Prokhorenko, V., 1996. The Multi-Satellite Space Missions and Campaigns Long-Term Planning. In *Space Mission Operations and Ground Data Systems-SpaceOps' 96* (Vol. 394, p. 441).
3. Nazirov, R. R. and Prokhorenko, V.I., 1998. Situation analysis in the problems of space physics, *Kosm. Issl.*, vol. 36, no. 3, pp. 311–322. [*Cosmic Research*, pp. 292–303].
4. Atanassov, A., Program System for Space Missions Simulation – First Stages of Projecting and Realization, In Proceedings of SES 2012, 209-214, 2013.
5. Atanassov, A. M., Parallel Satellite Orbital Situational Problems Solver for Space Missions Design and Control. Adv. Sp. Res., v. 58, № 9, p. 1819-1826.