

DEVELOPMENT OF TOOLS FOR COMPUTER MODELING AND SIMULATION OF SATELLITE ENERGY SUBSYSTEM FUNCTIONING

Atanas Atanassov¹, Lilia Atanasova²

¹Space Research and Technology Institute – Bulgarian Academy of Sciences

²Medical University – Sofia

e-mail: At_M_Atanassov@yahoo.com

Keywords: space mission analysis and design, multi-satellite systems, satellite energy systems, models, computer simulation

Abstract: Dialogue tools have been developed to create models of satellite energy subsystems. These tools are aimed at multi-satellite systems modelling and simulation. When compiling a model of an energy system, it is linked to the model of a system of satellites. At this stage, an energy system includes a system of solar panels and batteries. One model can be applied to all satellites in the system; it is possible to create a different model for each satellite.

The main attributes, with which models of the panels and batteries of an energy subsystem and their interfaces with other models, as part of the architecture of a system for simulation of multi-satellite systems, are presented. A descriptor of the model of energy systems is considered. Each current descriptor is stored in an array of descriptors. When compiling simulation models, different multi-satellite systems can be created, each with its own system of energy subsystems. After entering the panels and batteries model attribute values, a functional processor is created to simulate the operation of the energy systems of the satellites. A separate functional processor is created for the power systems of each multi-satellite. Each functional processor is implemented by a different pool of threads. The simulation of the electric power yield from the panels and the charging of the batteries is performed with a specific command of the functional processor.

Examples of computer simulation of the operation of satellite panels are shown.

РАЗРАБОТКА НА СРЕДСТВА ЗА КОМПЮТЪРНО МОДЕЛИРАНЕ И СИМУЛАЦИЯ НА ФУНКЦИОНИРАНЕТО НА СПЪТНИКОВИ ЕНЕРГИЙНИ СИСТЕМИ

Атанас Атанасов¹, Лилия Атанасова²

¹Институт за космически изследвания и технологии – Българска академия на науките

²Медицински университет – София

e-mail: At_M_Atanassov@yahoo.com

Резюме: Разработени са диалогови средства за създаване на модели на спътникови енергийни подсистеми. Те са насочени към мултиспътникови системи. При съставянето на модел на енергийна система той е обвързан с модела на система от спътници. На този етап една енергийна система включва система от панели и акумулатори. Един модел може да се отнесе към всички спътници от системата, но е възможно да бъде отнесен и само към един от тях

Разгледан е дескриптор на енергийни системи. Тези дескриптори се записват в различни типове масиви. При съставяне на симулационни модели могат да се създадат различни многоспътникови системи всяка със своя система от енергийни подсистеми. След въвеждане на стойностите на панелите и акумулаторите се създава актуален функционален процесор за симулиране на работата на енергийните системи на спътниците. За енергийните системи на всяка многоспътникова система се създава отделен актуален функционален процесор. Всеки отделен функционален процесор се реализира чрез отделен пул от нишки.

Симулацията на добива на енергия от панелите и зареждането на акумулаторите се осъществява с подпрограма извиквана от функционалния процесор. Представени са основните атрибути с които се създават модели на панелите и акумулаторите на енергийна система, интерфейсите им с други модели като част от архитектурата на система за симулация на многоспътникови системи.

Показани са примери от компютърна симулация на функционирането на спътникови панели.

Introduction

With the development of space technology, the possibility of saturating outer space with multi-satellite systems is becoming a reality. Concepts based on different classes of satellites (mini, nano-, pico-, femto-, chip) are under development [1–4].

The satellite equipment is determined according to tasks to be solved. In all cases, it is necessary to have an energy system that must meet the needs of the used service systems and scientific-applied tools. The application of computer simulation programs is a powerful tool for optimizing the subsystems of individual satellites and multi-satellite systems. Particular attention is paid to the capacity and reliability of energy systems for the successful implementation of satellite missions.

Development of tools for a definition of satellite energy system models and calculation instruments are under development at Space Research and Technology Institute at the Bulgarian Academy of Sciences.

Previous work

The development of a software system for space mission simulations was initiated [5]. Initially, it included tools for defining models of the motion of the multi-satellite systems and an integrator of equations of motion [6]. In addition, tools have also been developed to describe the definition of situational tasks and to determine the time intervals in which various orbital events take place (based on specific parameters and constraints). The developed parallel processor for situational analysis includes computational models for different situational conditions [7]. Models and computing tools have also been developed to visualize 3D scenes involving the Earth and satellite systems [8]. Other tools, useful in developing multiphysics simulation applications, have also been developed [9, 10].

Statement of the problem

Different variants of energy systems are known depending on the energy source and the means for its accumulation [11–13]. Here we will consider the case of solar panels and rechargeable batteries. Solar panels transform solar energy into electricity in the illuminated part of the orbit. Some of this

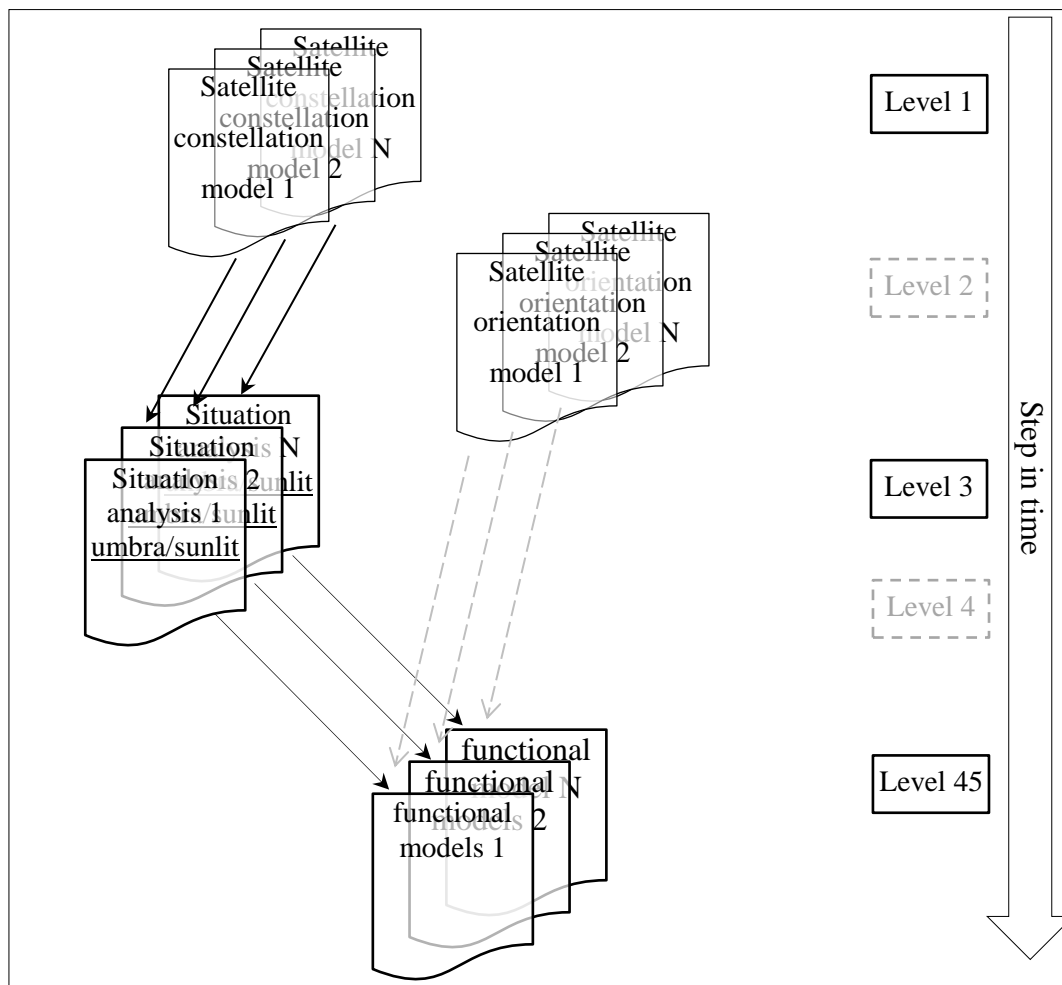


Fig. 1. Developed models, distributed by levels, with connections between them, are shown

energy is consumed by the satellite service subsystems. Another part of the energy is used by scientific instruments, depending on the type and way of their work and the scientific tasks related to specific orbital events, which are subject to situational analysis. The rest of the generated energy charges rechargeable batteries.

Simulating a satellite power subsystem is not a stand-alone task. It is performed together with tasks such as simulation of the orbital motion of the satellite and its orientation and situational analysis to determine the conditions for the operation of the energy subsystem. These conditions are related to the positioning of the satellite in the illuminated part of the orbit or in the umbra, as well as specific conditions for conducting measurements when instruments consume energy.

At this stage, the task is simplified, as we do not deal with the orientation of the satellites, respectively the solar panels. This task is generally easy to solve, depending on the specific design of the satellite and the structural configuration of its main elements. The focus is mainly on developing an interface between some models of the satellite orbital dynamics, the situational analysis and the energy system, within a multiphysics simulation system. Figure 1 illustrates the connections between these models.

Development of tools for modelling of satellite solar arrays and batteries

The model of an energy system based on solar panels and batteries is drawn up in a dialogue form, which includes controls for entering the values of various attributes. The solar array model includes some attributes as the number of the panels of solar array, area of the solar panels and efficiency. A possibility to choose a law of ageing as a result of the hostile space environment (charged particles, vacuum, electro-discharging) is insured.

The model of a battery comprises the next set of parameters: number of modules, maximal capacity, lower and upper voltage limits (depth of discharge), charging mode, the maximal number of cycles and ageing model. Ageing of the battery is a result of the hostile space environment. The developed dialogue tools provide an opportunity to choose when defining the parameters of both models – of the panels and the batteries, whether to apply them to one satellite or to the entire constellation.

The creation of an energy system model begins with choosing a satellite mission model. The names of the already created missions are taken from their descriptors. These descriptors are located in an array whose address is in named common area **c_IVPs** (/c_IVPs/num_IVPs,IVPs_descriptor_adr).

The descriptor of the selected mission must contain the number of the situational descriptor if models for situational tasks have been created in advance. The identification number of the situational task that checks if the satellite is on the illuminated part of the trajectory or in the umbra is established and stored.

After entering values of the various attributes of the energy system model, parameters of the computational tool with which simulation of the functioning of the energy system is performed are also entered. This is how it is created a pool of NumThreads threads with a subroutine for creating pools [14]. The instrument descriptor is written to an array of the descriptors (Fig. 2a).

Fig. 2a shows the creation of an actual calculation instrument. In the first place, a memory for different parameters of the pool is allocated by procedure MALLOC. The variables containing these parameters are saved in one array of the storage. The subroutine SolarArray_SimulatorUPC is an initial subroutine for the calculation instrument. It is a buffer subroutine used to transmit data to the threads of the computational tool.

Fig. 2b shows the descriptor of the energy system. It is used by the interpreter of the simulation model. With the help of some of the parameters of this descriptor, a call to the functional processor is made, which simulates the functioning of the energy system. Figure 2c shows the ES_Payload descriptor of the power system for starting the simulation from the simulation control subroutine. For this purpose, this descriptor is added to other descriptors of level "level_45". This is done with the "ad_object" subroutine. The energy system descriptor contains the name and the serial number of, the descriptor of the multi-satellite system (among the recorded objects for "level_45") as well as data addresses. Thus, a connection is made between the different models within a multiphysics simulation model (Fig. 1).

Fig. 2d show the second type descriptor payload_descriptor, which serves for transferring information toward actual simulator threads using named common area /c_payloads/. This common area is contained in the buffer subroutine SolarArray_SimulatorUPC.

Each simulation model may include one or more multi-satellite systems. A model of the energy subsystem is created for each satellite in one system. The energy systems of a multi-satellite system may be the same or a specific one may be defined for each individual satellite.

```

External      SolarArray_SimulatorUPC
POINTER(PoolParam_adr,ES_pool_par);   POINTER(ES_handler_adr,ES_handler);
...
memory= MALLOC(4*4*NumThreads+4+4)
ES_GlbCount_adr= memory;   !
ES_handler_adr = memory + 4;
PoolParam_adr = memory + 8; !

CALL CreatePoolThreads(SolarArray_SimulatorUPC,NumThreads,ES_pool_par,ES_handler);

a).

AS_pool_par%AI%num_threads = NumThreads      ! num_threads
AS_pool_par%AI% ha_race     = ES_handler_adr  ! handler for cinchronization
AS_pool_par%AI%counter_adr  = ES_GlbCount_adr ! global counter of the pool
AS_pool_par%AI%thread_par_adr= PoolParam_adr  ! parametrs of the pool
AS_pool_par%AI%granulation  = Granularity;    ! granularity

CALL add_object(num_Als,Als_descriptor_adr,Als_descriptor_adr,AS_pool_par)

(b).

type level_45 ! Description of satellite subsystems & payload
integer kind_payload
character name*28
integer IVP_index ! connection to IVP - this is base object
integer AS_index ! (A)ctual(S)olver of (E)nergy(s)ystem
integer adr_panels
integer adr_battery
end type level_45
type(level_45) ES_Payload
...
ES_Payload%kind_Payload= 2 ! 1 - for (E)nergy(S)ystem
ES_Payload%name = IVP_par(kod_IVP)%name ! take the name of IVP
ES_Payload%IVP_index = kod_IVP ! number of IVP - for coordinates of satellites
ES_Payload% AS_index = num_Als ! parametrite na solvera - za upravljenieto mu
ES_Payload%adr_panels = adr_panels; ES_Payload%adr_battery = adr_battery
CALL add_object(num_payloads,Payload_descriptor_adr = ES_payload_adr, &
Payload_descriptor_adr_new= ES_payload_adr, &
Level__45= ES_Payload)

(c).

common /c_payloads/num_payload,payload_descriptor_addr

payload_descriptor% kind_payload = 2 ! 2 for (E)nergy(S)ystem
payload_descriptor% t_adr = IVP_par(kod_IVP)%IVP% t_adr !
payload_descriptor% dt_adr = IVP_par(kod_IVP)%IVP% dt_adr !
payload_descriptor% num_sat = IVP_par(kod_IVP)%IVP%num_objects !
payload_descriptor% xvn_adr = IVP_par(kod_IVP)%IVP%xvn_adr !
payload_descriptor% xvk_adr = IVP_par(kod_IVP)%IVP%xvk_adr !
payload_descriptor% num_sit_tsk = StPrb(iSitProb)%SitProb%num_sci_task
payload_descriptor% max_num_con= StPrb(iSitProb)%SitProb%max_num_sit !
payload_descriptor% sit_task_adr = StPrb(iSitProb)%SitProb%addr_sit_prob
payload_descriptor% panels_adr = adr_panels;
payload_descriptor%batteries_adr = adr_battery
Payload_descriptor_addr = %LOC(Payload_descriptor);

(d).

```

Fig. 2. a). A pool of threads creation; b),c). Assignation values to attributes of the descriptors and adding them to the corresponding layers; d). A descriptor that transmits information to the threads of a functional processor.

Development of processor for functional simulation of satellite subsystems

Parallel calculations are applied when many energy subsystems (corresponding to the number of satellites) are simulated. Earlier developed approach related to other parallel computational tools, used in the program system for space mission simulation, is applied in this case. The "Parallelism across the system" approach is applied. This means that calculations for a single energy subsystem are performed on a separate processor core (physical or logical).

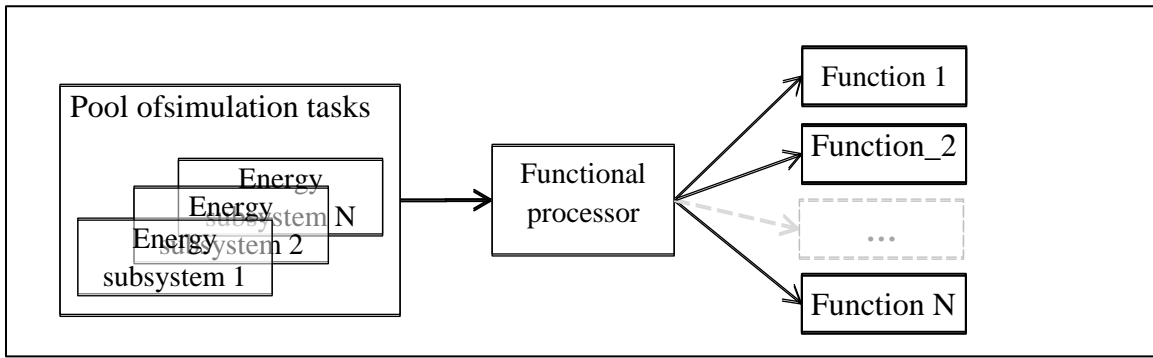


Fig. 3. Schema of the parallel functional processor

Fig. 3 illustrates the simulation of the functioning of different satellite subsystems and sensors. As mentioned above, the functional processor is a pool of threads. The threads are synchronized with each other in order not to process the same tasks (EP) (race condition). The threads are also synchronized with the parent thread, which starts their work and sets the end of the calculations for each step of the simulation time. All developed computational instruments use the same subroutine "Pool_threads_Control_1" for thread management [14].

The functional processor can call various functions (Fig. 4) that simulate the operation of different subsystems. Each of the tasks contains an identification code of the function that the processor calls for simulation.

Examples

The computer simulations are based on a model of satellite circular orbit with a radius 7×10^3 km. The Sun's position is close to the autumnal equinox and ascending node $\Omega = 60^\circ$. The numerical integration of the satellite motion is performed by a step of 10 sec. Passing through the penumbra is neglected due to the short time. Simulations for different inclinations of the orbital plane are shown in Fig. 5.

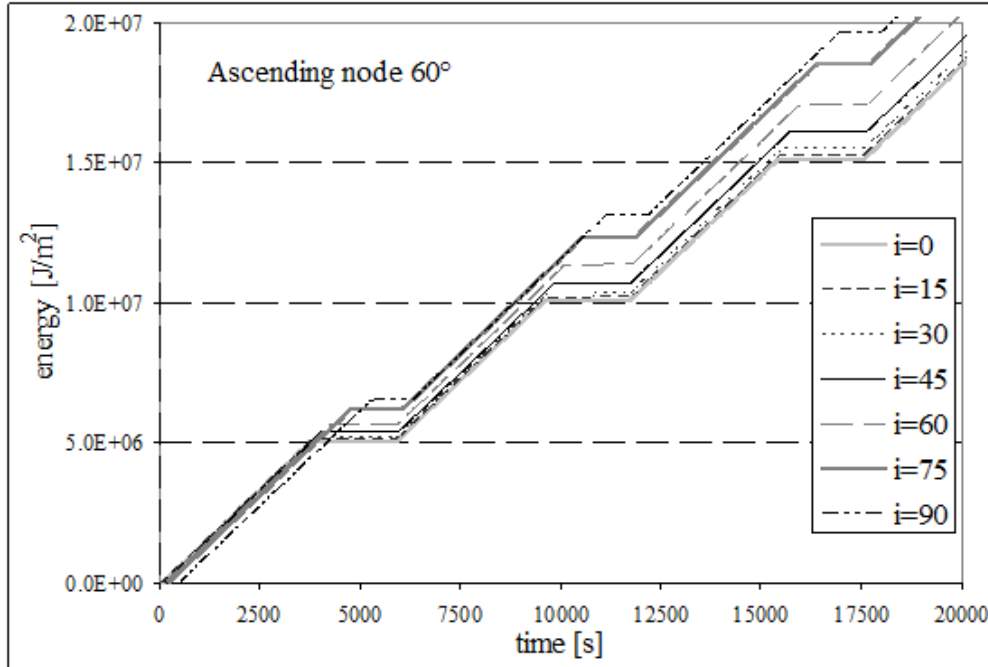


Fig. 5. The solar energy falling on a unit area of the illuminated part of the satellite's orbit, depending on the inclinations

The linear section of the graphics is for the case when the normal to the solar panels is directed towards the sun of the entire illuminated area of the orbit. When the solar panels are firmly attached to the satellite during its reorientation to track targets on the earth's surface, the direction of the normal changes. In this case, the energy produced will also change. The efficiency of the panels also depends on their temperature.

```

SUBROUTINE FuncProcessor_UPC(task_count,payload_descriptor_adr,num_tasks)
USE FuncMod,Only:satellite
  integer task_count
  type(satellite) payload_descriptor
  real*8 t,dt
  logical Func__1,Func__2

POINTER(payload_descriptor_adr,payload_descriptor)
POINTER(t_adr,t);POINTER(dt_adr,dt)

  t_adr = payload_descriptor% t_adr;
  dt_adr = payload_descriptor% dt_adr;
  num_sat = payload_descriptor%num_sat

  IF(payload_descriptor%kind_payload.EQ. 1) THEN ! Orientation & stabilization
    icode= func__1(t,dt,task_count,payload_descriptor_adr,num_sat)
  ELSEIF(payload_descriptor%kind_payload.EQ. 2) THEN ! (E)nergy(S)ytem
    icode= func__2(t,dt,task_count,payload_descriptor_adr,num_sat)
  ENDIF

END SUBROUTINE FuncProcessor_UPC
(a).

FUNCTION func__2(t,dt,task_count,adr_satellite_descriptor,num_sat)
USE FuncMod, ONLY: solar_panels,satellite_battery,satellite
USE RN,ONLY:sit_task
  logical func__2
  integer task_count,adr_payload_descriptor
  real*8 t,dt
  type(sit_task) situation ,sit(2) !(0:max_num_con,num_sit_tasks)
  type( solar_panels) panel,panels(num_sat),panels2(2)
  type(satellite_battery) battery,batteries(num_sat),satellite_battery2(2)

  num_sit_tasks = satellite_descriptor%num_sit_tsk; adr_panels = satellite_descriptor%panels_adr
  max_num_con = satellite_descriptor%max_num_con; adr_batteries= satellite_descriptor%batteries_adr
  i_sit_adr = satellite_descriptor%sit_task_adr

  length_sit = LOC(sit(2)) - LOC(sit(1))
  length_panels = LOC(panels2(2)) - LOC(panels2(1))
  length_battery= LOC(satellite_battery2(2)) - LOC(satellite_battery2(1))

  num_sit_task= panels(task_count)%id_sit_cond
  sit_adr = i_sit_adr + (1+max_num_con)*length_sit*(num_sit_task-1) + length_sit

  IF( situation %sit_cond%flag) THEN
    area_panel = panels(task_count)%size*panels(task_count)%modules
    efficiency = panels(task_count)%efficiency;
    Gen_Energy= efficiency*area_panel*solar_constant
    Gen_current= Gen_Energy/batteries(task_count)%granici(1,batteries(task_count)%komutator1)
    Sol_energy = Sol_energy + solar_constant*dt
    batteries(task_count)%level(batteries(task_count)%komutator1)= &
      batteries(task_count)%level(batteries(task_count)%komutator1) + Gen_Energy*dt
    IF(batteries(task_count)%level(batteries(task_count)%komutator1).GT. &
      batteries(task_count)%capacity(batteries(task_count)%komutator1)*3600.*1000) THEN
      batteries(task_count)%komutator1= batteries(task_count)%komutator1 + 1
      IF(batteries(task_count)%komutator1.GT.batteries(task_count)%modules) THEN
        batteries(task_count)%komutator1= 1
      ENDIF
    ENDIF
    ELSE ! In Earth's shadow

  ENDIF
  func__2=.true.

END FUNCTION func__2 ! Sim_SolarPanels
(b).

```

Fig. 4. a). part of the subroutine, which calls specific functions for simulation of different subsystems functioning;
b). part of the subroutine, which simulates energy system functioning

Conclusion and future work

The concept and development of tools for modelling satellite energy systems and tools for computer simulation of systems from multi-satellite systems are presented.

Further development of the models of stabilization and orientation of the satellites and the solar panels, and the inclusion of appropriate calculations correcting the quantity of solar energy falling on the panels is forthcoming. The task of self-shading of the solar panels from the structural elements of the satellite will also be included. An interesting problem is related to the optimization of the self-shading task.

Reference:

1. Selva, D., A. Golkar, O. Korobova, I. L. I. Cruz, P. Collopy, & O. L. de Weck, Distributed earth satellite systems: What is needed to move forward?. *Journal of Aerospace Information Systems*, 14(8), 2017, pp. 412–438.
2. Barnhart, D. J., T. Vladimirova, & M. N., Sweeting, Very-small-satellite design for distributed space missions. *Journal of Spacecraft and Rockets*, 44(6), 2007, pp. 1294–1306.
3. Barnhart, D. J., Vladimirova, T., Baker, A. M., & Sweeting, M. N. (2009). A low-cost femosatellite to enable distributed space missions. *Acta Astronautica*, 64(11-12), 1123–1143.
4. Brown, O., P. Eremenko, & P. Collopy, Value-centric design methodologies for fractionated spacecraft: Progress summary from Phase I of the DARPA System F6 Program. In *AIAA Space 2009 Conference & Exposition*, 2009, p. 6540.
Available on: <https://digitalcommons.usu.edu/cgi/viewcontent.cgi?article=1263&context=smallsat>
5. Atanassov, A. M., Program System for Space Missions Simulation – First Stages of Projecting and Realization. *Proceedings of SES 2012, 2013*, pp. 209–214.
6. Atanassov, A. M., Parallel, adaptive, multi-object trajectory integrator for space simulation applications, *Advances in Space Research*, v. 54, 8, 2014, pp. 1581–1589.
7. Atanassov, A. M., Parallel satellite orbital situational problems solver for space missions design and control, *Advances in Space Research*, v. 58, 9, 2016, pp. 1819–1826.
8. Atanassov, A., Atanasova, L., Development of New Functionalities for a Space Missions Simulation Program System, *Proceedings SES 2019, 2019*, pp. 70–74.
9. Atanassov, A., Method of Thread Management in a Multi-Pool of Threads Environments, *Proceedings of SES*, 2014, 2015, pp. 241–246.
10. Atanassov, A. M., Development Classes of Objects' Descriptors for Space Missions Simulation. *Proceedings SES 2015, 2016*, pp. 51–55.
11. Patel, M. R. *Spacecraft power systems*. CRC press, 2004.
12. Hyder, A. K., Sabripour, S., Flood, D. J., Halpert, G., Wiley, R. L., & Hyder, A. K. *Spacecraft power technologies* (Vol. 1, p. 498). London: Imperial College Press, 2000.
13. Chen, Q., Liu, Z., Zhang, X., & Zhu, L., *Spacecraft Power System Technologies*. Springer Singapore, 2020.
14. Atanasov, A., Razработка na metodi, algoritmi i programni sredstva za analiz i proektirane na kosmicheski misii, PhD thesis, 2019 (bul).
15. Atanassov, A. ., Atanassova, L. ., Development of Tools for Models' Design of Systems of Multi-Satellite Systems. *Proceedings SES 2020, 2020*, pp. 110–115.
16. Wertz, J. R., *Orbit & Constellation Design & Management: Spacecraft Orbit and Attitude Systems*. Microcosm Press. 2001