LOCOMOTION SYSTEM OF THE IARES DEMONSTRATOR FOR PLANETARY EXPLORATION

A. BALÁZS1, J. BIRÓ1, I. HIERNYES1, I. HIORVÁTH1, S. SZALAI1,
A. GRINTCHENKO2, V. KACHIRINE3, G. KOZLOV3, S. MEDVEDEV3,
V. MICHKINIOUK2 and L. MARECHAL3

1FKI RMKI, Budapest, Hungary, 2VINIITRANSMASH, Saint Petersburg, Russia and 3Centre
National d’Etudes Spatiales, Toulouse, France

Abstract The Eureka project no. 969 for the development of an earth robot called IARES (illustrateur autonome de robotique mobile pour l'exploration spatiale) was initiated in 1993 in the frame of an international cooperation associating French laboratories and industrial companies to foreign partners (Spanish, Russian and Hungarian). This demonstrator will show the feasibility of a planetary robot by testing in a realistic environment the robotic axes studied for several years in French and European research centres. After a short description of the IARES project and demonstrator, this paper presents in detail the IARES locomotion subsystem developed under the responsibility of VINIITRANSMASH (Russia) with the cooperation of FKI RMKI (Hungary) © 1998 Published by Elsevier Science Ltd. All rights reserved

GENERAL DESCRIPTION

The management of the project was delegated to CNES, by the Eureka steering committee. The division of the activities is presented in Fig. 1. The main objectives of this project are to demonstrate the feasibility of a planetary mission and to set information to be able to evaluate the characteristics and performances of a future flight vehicle.

The tests of the demonstrator are performed in an environment as close as possible to Mars and/or the Moon. The functions are the same as for a planetary rover in order to evaluate the performances. Standard ground equipment is used if the technologies are compatible with later space qualification. The global mass of the robot is the same as for a real mission. (The highest gravity on the Earth is compensated by more powerful actuators.) The electrical generation/distribution system and the computers are not representative, but are sized for the mobile robot.

The reference mission taken into account to define the demonstrator was, firstly the exploration of Mars. The delay of Mars missions and the new interest in the Moon (ESA proposal for the LEDA demonstration mission on the Moon) led us, by mid-1994, to modify our reference mission towards moon exploration. Therefore, we decided to build an IARES demonstrator for lunar exploration functionally equivalent to the LEDA mission requirements [1, 2].

The chassis was delivered to CNES mid-1996 and the electrical architecture has now been integrated. The locomotion control software, the perception equipment, the path generation and supervision software and the localization unit are under development. The demonstrator will be integrated by CNES. The validation tests of the IARES demonstrator are foreseen to be complete by May 1998.

The mobile robot has a global mass of 150 kg and a mean power of 400 W. The important subsystems constituting this robot are as follows.

Locomotion. An advanced 6 × 6 locomotion chassis, the object of this paper. It is described in detail in the following paragraphs.

Localization. The inertial unit comprises three accelerometers and three gyroimeters mounted along the three axes of the measurement frame. The position is updated every 2 mm; during 10 s at “null velocity”. The performances are: relative position 3%, attitude 0.1°, direction accuracy < 1°.

Perception. The functions required from this subsystem are image acquisition synchronously to the inertial unit data. This device includes one pair of digital cameras and the suitable correlation algorithms to produce the three-dimensional points from the images. The stereo basis is 0.20 m. The stereo reference (point of view) is changeable between 1500 and 2100 mm from the ground.

Path generation. From the three-dimensional points produced by the perception, this function rebuilds the digital terrain model (DTM) that will be used to navigate. Two types of planners will be provided: the “simple path planner” that generates a straight-line trajectory per step and the “two-dimensional path planner” which can be used in the usual sequence “perception when rover stops—path generation—trajectory execution” repeated until the final goal is reached or in the “continuously planning autonomous navigation” with perception and path generation during trajectory execution.
Manipulation. The manipulation device comprises a 3 d.o.f. robot arm and a 3 d.o.f. gripper [2]. The robot arm and gripper have the following capabilities:

- sample mass and volume: \( \leq 0.5 \) kg and \( \leq 70 \times 70 \times 70 \) mm\(^3\);
- force/torque: 25 N and 1.5 Nm;
- position and orientation accuracy with video feedback control: 1 mm and 0.5\(^\circ\);
- maximum and minimum speed with samples: (0.005 m/s; 0.5\(^\circ\)/s) and 0.003 m/s, 0.25\(^\circ\)/s);
- CCD camera: \( f = 6 \) mm; Laser emitter;
- global mass \( \leq 20 \) kg; Manipulation power: \( \leq 120 \) W; length: 1.10 m.

The manipulation work space is defined as a circle of 0.80 m radius from arm basis.

Data processing hardware architecture consists of:

- an onboard VME rack with a Power PC 604 board running under Lynx-OS operating system and input/output boards for analog/digital acquisitions or serial (RS232, Ethernet) and transputer link communications;
- a T805 transputer module for locomotion [3];
- an IOT 336 Transputer module based on T805 for Manipulation [4];
- a specific computer board for localisation inertial unit.

The functional control modes of the demonstrator are described in more details below.

Autonomous navigation mode. In this mode, the operator selects a goal that has to be reached by the mobile robot according to the mission objectives. For the goal and planner selection, the operator uses a global camera image that has been transmitted by the on-board vision system. The robot moves autonomously toward the goal. Communications are then used only for monitoring or when the supervisor decides to stop the rover and generate a call to Earth.

The following sequences are realized: a DTM is generated with the images acquired by the stereo vision subsystem. The terrain areas are labelled as navigable or dangerous according to the locomotion capacities. The maximal path length in the direction specified by the operator avoiding dangerous and unknown areas is then computed taking into account the locomotion uncertainties and safety margins. The computed trajectory is then executed.

In the continuous navigation mode, stereo image acquisition and up to path generation are performed during execution of this trajectory. This is then replaced by the new path, before reaching completion. Perception during the trajectory execution increases the anticipation capacities of the rover related to obstacle avoidance. Furthermore, in this continuously planning mode, the path generation computation time is completely masked.
Teleoperated navigation mode. This mode under the name TESA (acquired synthetic environment teleoperation) belongs to the class of teleoperation modes with predictive display. For rover control, the ground is supplied with DTM that are periodically generated onboard from the stereo images taken during the move. The operator is provided with a three-dimensional knowledge of the environ-
ment in a short range beyond the vehicle (8–10 m). Furthermore, the operator is supplied with a periodic global image at low rate (1 per min) in order to be able to determine the long term path. The IESMA mode compensates the time delay by displaying to the operator the predicted position of the vehicle in the DTM. Using that technique, the operator can drive the vehicle continuously since he perceives immediately the effect of his commands (direct predictive teleoperation implementation) or defines continuously the vehicle objective (predictive teleoperation with shared control implementation).

Autonomous manipulation mode. The arm is used to realize manipulation tasks such as sample collection, soil hardness measurement or rover visual inspection. In the autonomous manipulation mode, the operator selects either a set of samples or a set of soil points. The inspection points are predefined, but their accessibility is computed taking into account the current mobile robot configuration. For the sample or soil point selection, the operator analyses the four images of the arm working area that have been acquired by the on-board gripper camera with a height of 800 mm. Then, the manipulation task (COLLECT, MEASURE or INSPECT) with its set of goals is sent to the mobile robot where it is executed and managed by the supervisor. Communications are then used only for monitoring or when the supervisor decides to stop the task and generate a call to Earth. For the navigation modes, the path generation function sends the trajectory as a set of coordinates to the locomotion subsystem which is described hereafter.

CHASSIS DESCRIPTION

The chassis, developed in VNIITRANSMASH [5], represents the six-wheel vehicle with rigid wheels and the articulated frame (Figs 2 and 3). The chassis has some redundancy, so simple modes of movement (turning, for instance) can be executed in several slightly different ways. This redundancy allows us to carry out a number of new, more sophisticated functions, increasing practicability and manoeuvrability. In Table 1 technical characteristics of the developed chassis are given. The mass of the chassis (in view of the requirement of operation in conditions of the earth gravitation) is 84 kg. Total mass of the mobile robot is 150 kg.

For the IARES-L chassis the fulfillment of a high number of functions is provided by the mechanism with 19 degrees of freedom. Seventeen electric drives are used for management of elements of the chassis, from which six drives are traction. Six electric drives are intended for control of turn of wheels, and the management of each drive of turn is carried out independently. It allows us to set the centre of a turn in a wide range of situations with possible displacement in transversal and longitudinal directions.

New qualitative function—the opportunity of a movement by tack is entered. It allows us to solve two serious problems. The first is overcoming a tack of separate rise, when a movement on a rise on a line of the maximum slope in a rolling mode is limited by the maximum spinning of the wheels. The second is countering the unwanted action of a hill-side on the trajectory of movement of the robot across a slope.

Blocking a frame allows the robot to negotiate trenches and cracks of large width. The longitudinal stabilization of the frame in a vertical plane allows us to reduce non-uniformity of distribution of loading on the wheels at a movement on rises and hillsides. Moreover, this increases the margin of stability on overturning. Despite the rather small size, the chassis is capable of negotiating separate obstacles of heights up to 0.6 m and to move on complex obstacles, representing a combination of jut and trench with difference of heights of up to 0.3 m.

<table>
<thead>
<tr>
<th>Table 1. Technical characteristics of the IARES-L locomotion subsystem</th>
</tr>
</thead>
<tbody>
<tr>
<td>IARES-L chassis mass/payload and equipment mass</td>
</tr>
<tr>
<td>80/70 kg</td>
</tr>
<tr>
<td>Supply voltage</td>
</tr>
<tr>
<td>28 V</td>
</tr>
<tr>
<td>Maximum power consumption of the chassis</td>
</tr>
<tr>
<td>390 W</td>
</tr>
<tr>
<td>Chassis control system</td>
</tr>
<tr>
<td>Digital, multiprocessing</td>
</tr>
<tr>
<td>Locomotion control modes (telecommanding or manual box)*</td>
</tr>
<tr>
<td>1. Low level commanding*</td>
</tr>
<tr>
<td>2. Movement commanding*</td>
</tr>
<tr>
<td>Trajectory commanding</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>Number of chassis modules</td>
</tr>
<tr>
<td>1/2</td>
</tr>
<tr>
<td>Number of payload containers/number of external payload mounting places</td>
</tr>
<tr>
<td>1. Wheel mode (two speeds)</td>
</tr>
<tr>
<td>2. Walking mode</td>
</tr>
<tr>
<td>Mode of turning</td>
</tr>
<tr>
<td>Kinematic</td>
</tr>
<tr>
<td>Maximum speed (flat and hard surface no obstacles)</td>
</tr>
<tr>
<td>0 m/s</td>
</tr>
<tr>
<td>Tractor</td>
</tr>
<tr>
<td>0.35 m/s</td>
</tr>
<tr>
<td>Centre mass height</td>
</tr>
<tr>
<td>265 mm</td>
</tr>
<tr>
<td>Wheel diameter/wheel width</td>
</tr>
<tr>
<td>380/152 mm</td>
</tr>
<tr>
<td>Nominal/maximum wheeled base</td>
</tr>
<tr>
<td>855/1333 mm</td>
</tr>
<tr>
<td>Chassis size</td>
</tr>
<tr>
<td>1235 x 1200 x 720 mm</td>
</tr>
</tbody>
</table>

*Estimation on the basis of experimental mock-up test data
and extents 0.1 ... 0.5 m. This promotes chosen kinematics of suspension directing mechanisms of wheels and mechanism of an articulation of a frame in a vertical plane.

The control system of the chassis allows the regulation of the speed of rotation individually for each wheel. Also, the appropriate angle of its turn can be given for each wheel. Fulfillment of these functions allows us to move the chassis with various fixed speeds of rotation of the wheels as at a rectilinear movement, and at fulfillment of turns. Moreover, coordination of speeds of rotation of wheels and their movement in peristaltic mode is provided for overcoming complex obstacles.

The chassis has support tractive rollers, on the bottom of the levers of the directing mechanisms of the suspension. The drive for the rollers is provided through the two-seated hinges from tractive electric motors. Rollers on the bottom prevent the chassis from getting stuck as it moves over ledges. The size of the wheels allows the robot to move on ground with physical properties similar to those on the Moon, with the settlement on factor of roll resistance $f = 0.08$. This is a little better than average
factors of roll resistance obtained in the operation of the chassis “Lunokhods 1 and 2”.

Rigid wheels are used in spite of the fact that the factor of roll resistance of rigid wheels in a normal case is more than with elastic wheels. Elastic wheels are not used because when a block occurs more force is required. Essentially, the effort created with rigid wheels in a wheel-walking peristaltic mode of a movement will be much greater than with elastic wheels. This is of basic importance for a planet rover in a critical situation.

**LOCOMOTION SUBSYSTEM**

The locomotion subsystem performs low-level motion control: it receives its path as a set of coordinates, generates the necessary low level commands and distributes them to the engines of the different wheels. At the same time, it collects data from numerous sensors, which are—together with the actual coordinates received from the localization subsystem—used to control the correct movement along the path. If necessary, the locomotion subsystem is able to modify the working of the engines to ensure the correct execution of the path. In accordance with the initial requirements of the locomotion subsystem, the declination from the specified path will be less than 10 cm.

**Hardware architecture**

The rover chassis has a kinematically complex mechanical structure, which contains 17 electric actuators: this has a definite impact on the locomotion subsystem. First of all, this is demonstrated on its structure, which has been implemented as a distributed multiprocessor system, in which independent mechanical units of the chassis are controlled by separate processor units, enabling the implementation of electrical and logical interfaces with minimized number of cables. So, the locomotion subsystem is a distributed control system, containing a T800 and a T225 transputer on the top level of the motion control and three other so-called wheel-controller units on the lower level (Fig. 4).

Calculations about the optimal distribution of the controlling hardware elements and computing power showed that each wheel-pair should have one wheel controller, so one controller supervises two locomotion wheels on the same axis, and several smaller engines of the above mentioned tasks. Because the frame can be transformed, and also in order to decrease the amount and weight of electric cables, each wheel controller is placed near to its wheels as possible. It was desirable to choose some method of communication between the locomotion transputer and the wheel controllers fulfilling these requirements. We selected a serial bus to decrease the bending force of the cable. The MIL-STD-1553B bus has been developed for US military avionics, but also widely used in space electronics (Cassini, SRG space probes). This bus uses low wire count cables, has medium speed and low noise sensitivity, features galvanic isolation and also supports fault tolerant systems.

The transputers and the wheel-controllers will communicate on the serial bus (MIL-STD-1553B). The transputers will receive the path from a higher level, decompose it into elementary commands and distribute them to the wheel controllers. The T800 will also collect sensor data from the wheel controllers and use them for the correct execution of the commands.

For testing purposes, there is a technological connector on the rover by which one or more external device can be connected to the bus. For instance, a PC equipment with an MIL-STD-1553B interface can be used to monitor the bus traffic, to debug the wheel controller software or to download new software versions via this line.

The transputer was selected for as a real space mission processor: it is a high-performance microcomputer designed to facilitate interprocess and interprocessor communication. The transputer comprises a 32-bit RISC processor with 2–4 kbytes of fast static RAM, process scheduling in hardware with a submicrosecond context switch, external memory controller, and four high-speed serial links. A single transputer is a powerful processor in its own right: the serial links allow an architecture in which transputers are arranged in an array, each communicating with its four nearest neighbours. With suitable algorithms, this permits high performance on picture processing problems. It is a successful low-cost approach to achieving a high parallel-processing rate. The serial links minimize a failure propagation. (Another transputer—see Fig. 4—will serve as a control for the manipulator arm, which will be made in IKERLAN, Spain, although this second transputer does not belong to the locomotion subsystem logically.) The transputer module used in the locomotion subsystem is a standard T800 factory made module with a 20 MHz clock and 2 Mbytes RAM. The interface toward power-PC is one of its transputer links (equipped with RS-422 drivers). Between the main locomotion transputer and the MIL-STD-1553B bus a second T225 transputer chip is installed, which deals with the administration of the bus.

The three wheel controllers have an identical structure. One Controller has three PCBs assembled on a frame and is put in a box together with engine control electronics. The size of the controller is 110 x 79 x 50 mm. The controller has its own switched mode power supply having galvanic isolation and a noise filter and this converts the on-board 28 V DC supply voltage to the required inner voltages. The CPU is an AMD 80C86 microprocessor with 16 bit inner structure, but 8 bit wide outer bus. The required 32 kbyte program memory and
Locomotion system of the IARFS demonstrator for planetary exploration

32 kbyte working RAM requires only two chips. To save room and power, a configurable logical array chip is working on the board as a clock generator, several processors support function, a real-time clock, a two channel DMA controller and Mil-Std-1553B bus interface.

Another similar chip is configured as an I/O port to control the engines, but also as a two-channel counter/timer to determine the actual speed of the engine. The engines have been built in pulse generators, the frequency of which is proportional to the rotation speed of that engine. Controlling the engine speed is important in determining the travelling speed of the rover itself, but also to detect a possible slip or speed difference among wheels. There is also an eight channel 8 bit analog/digital converter for reading the angle sensors built into the chassis (steering position and frame transformation) and the locomotion engine current is also monitored to detect if the robot is stuck. A two channel digital/analog converter is used to control the speed of the locomotion engines because the electronic box of these engines requires an analog voltage. At an earlier stage a pulse width modulation control method was also considered.

For galvanic isolation of the actuator signals from high-power circuits, a bank of optical isolators is used. There are simple digital signal isolators, but the analog speed control signals and the engine current sensors have analog isolator circuits. For optical isolators low current high transfer ratio optocouplers are used.

To protect the system against crashing the controller has a watchdog circuit that must receive pulses from the software, otherwise it applies a reset to the controller. Not only the software, but also the hardware circuits are included in this scheme so if either a hardware element deconfigures accidentally or the software crashes for some reason the controller restarts, reloads the ICA chips and starts the system software. With suitably organized software on this, and the higher level, and with the features of Mil-Std-1553B bus system, the higher level can detect an accidental dropout of a controller and can re-establish control over it.

As a program store an EEPROM is used. This makes software development easier in comparison with use ROMs and makes field changes possible. An area in this EEPROM is protected (by hardware) so that field changes cannot accidentally erase the so-called system area. The other application program area can be written and modified without disassembling the mechanism. This reprogramming procedure requires compiling the new user program; then, with the help of a PC equipped with a Mil-Std-1553B interface, the program can be down loaded to the controller. This new program will then be non-volatile until overwritten. To prevent accidental overwriting the reprogramming possibility of the EEPROM can be disabled by hardware and also by the EEPROM's "secure" mode feature, which when activated by a code sequence blocks further writing into it.

Software

As the locomotion subsystem's two major programmable hardware components have been implemented on different hardware platforms, the corresponding software components have been developed independently, although they have a lot of similar features. Their basic software architecture is the same: an operating system performs low-level tasks, while an application program is responsible for the locomotion control. Also, as the whole locomotion subsystem is an embedded system, special tools and utilities (loaders, monitoring and debugging tools) have been developed to help the development cycle.

The transputer system software is a redesigned version of an earlier system that has been developed for the Russian Marsokhod (in the frame of the Mars-98 project). The operating system must conform to three major requirements:

- concurrent operation of the application processes;
- real-time requirements; and
- software reliability.

Concurrent operation is largely supported by the transputer hardware. The chip has a hardware scheduler which makes the process of software switching very fast. The channel concept offers a consistent model for interprocess communication and I/O access.

The most crucial real-time requirements concern the low-level motion control and the safety of motion. The software must recognize if the rover gets into a critical situation (for instance sliding down a slope) and must react immediately so as to avoid a disaster. It also must reply to some commands in a specified time.

Rigorous software specification, the use of languages with support for data abstraction and modularity, use of proven algorithms and methodologies, etc. can significantly improve software quality and eliminate many errors. Intensive and extensive system testing can also detect and remove still existing faults. Although these methods can help substantially certain errors (such as hardware errors, deadlocks, floating point errors, even remaining software bugs, etc.) may occur during operation. Therefore, these errors must be handled by the software. In other words: fault tolerance is also to be used.

In the following, we briefly describe the internal architecture of the operating system and its functions. The system is capable of running several application processes concurrently. Furthermore, it provides different communication interfaces to the application processes: interprocess communication and I/O communication.
All interprocess messages go through the system. The communication scheme supported is similar to the OCCAM channel concept. It is strictly synchronized, i.e. the processes cannot continue their operation until the exchange of data is really performed. Alternative input is also supported, i.e. processes can wait for messages from more than one source and react to the first message that arrives. Library functions support the transfer of different data types. I/O operations are managed by special operating system processes (I/O handlers). An I/O handler must be reserved before and released after being used. If the handler is already reserved, the request is refused. If the reservation is successful, the application processes can directly access the I/O handlers. The low level message protocol of the I/O units is uniform. As the nature of the I/O units can be totally different, there is an extensive set of library functions to support byte or block oriented I/O, time-out possibilities, setting internal registers and so on. More specific protocols can be built above this level by the application processes if necessary.

The system also has background processes. Some of them perform regular self tests (checking the local and global RAM, I/O units, other hardware elements). If these tests or other processes find an error then an error handler process tries to cope with it and restore the normal operation.

A special operating system process collects certain diagnostic data and transfers them to a monitoring PC via the service link. This monitoring process is invisible to the application processes and does not cause any logical change in their operation. (Timing relations can be different of course.) The services of the monitor process fall into three groups: memory dump; process scheduling information (the state of process queues, the timer queues, processes that stopped because of setting the error flag, etc.); and internal status information (information about processes waiting for interprocess communication or for I/O operation). The second and third groups usually provide complete information about the status of the different processes. Dead-locks and erroneous processes can be detected this way.

The wheel controller software consists of two major components: the operating system and the application software. The operating system is in charge of initialising the processor and other hardware elements, loading and starting the application software (developed by VINIITRANSMASH) and providing different system services to the application software. The services include timing, MIL-STD-1553B operations and monitoring support. Certain PC utility programs communicate with the operating system via the MIL-STD-1553B bus and enable the user to access its services.

The operating system provides four groups of services after the application software is started: bus handling, I/O support, timing services and monitoring. The first group of services is that the operating system handles the communication via the MIL-STD-1553B bus. I/O port access (the second group of services) is transparent: the application can read and write any port freely. (In certain cases—the A/D converter, etc.—the system provides additional library functions and macros to handle the conversion of acquired data.) Timing services can be very important for the application due to the real-time requirements of the locomotion control. The system provides global variables that are incremented (or decremented) by a regular system interrupt (at a frequency of 100 Hz). These variables can be used by the application as an overall clock or as an alarm-clock. If this is not sufficient, an application to interrupt routines can be registered and invoked by the system in certain conditions.

The last important component of the operating system is the monitor. It provides testing and debugging support to the programmers of the application software. A special set of commands (called MONITOR COMMANDS) and captured by the monitor. By default, the operating system is in monitoring mode: it enables the user to investigate and change the content of any memory location or I/O address. Further monitor commands are available, if the application has been started in a special mode: breakpoints can be defined, single stepping of the application program can be requested and the content of the registers can also be viewed. A utility program can be used on the PC to exploit the possibilities of the monitor. All monitor commands are handled exclusively by the operating system, thus being transparent to the application software.

CONCLUSION

In less than 1 year, the IARES demonstrator will have executed several missions on the CNES test site GEROMS (groupement pour les essais de robots mobiles spatiaux). These tests will provide Europe with an important data base on mobile robots with an advanced locomotion chassis and all the equipment and functions needed for autonomy and teleoperation of the vehicle, and the manipulation arm. This will show the capability of Europe to design, develop, integrate and validate a robot for a future planetary mission. This project is a good example of successful international cooperation and could be the origin of a European planetary exploration program (Moon or Mars).

REFERENCES


