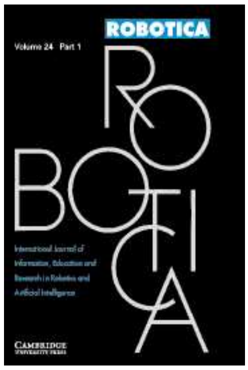

Control and navigation aspects of a group of walking robots

Teresa Zielinska

Robotica / Volume 24 / Issue 01 / January 2006, pp 23 - 29
DOI: 10.1017/S0263574705001840, Published online: 31 October 2005

Robotica (2006) volume 24, pp. 23–29. © 2005 Cambridge University Press
doi:10.1017/S0263574705001840 Printed in the United Kingdom

Control and navigation aspects of a group of walking robots

Teresa Zielinska

Warsaw Univ. of Technology (ITLIMS), Nowowiejska 24, 00-665 Warsaw (Poland)
E-mail:teresaz@meil.pw.edu.pl

(Received in Final Form: March 10, 2005)

SUMMARY

The properties of the software part of a control system implemented in walking robots are described. The paper presents also the navigation method elaborated for a family of hexapods. The general structure of the programming system is given together with the functional description of the system modules. The event-based action scheme of the central part of the system responsible for data distribution and high level navigation is addressed in detail.

KEYWORDS: Walking robots; Control; Navigation; Group.

I. INTRODUCTION

In prototyping works the type of applied hardware and its architecture influences the functions of programmed control tasks and their distribution over the hardware resources (functional architecture of the control system). On the other hand, within the same hardware, the functional decomposition of control tasks and their implementation method can be different. With chosen software architecture (e.g. with hierarchical dependencies of control tasks) motion can be produced using different rules-behavioral, reflexive or any others. Therefore the evaluation of the control system is not simple, and the structural and functional choices are not unique. The design target in the control systems development should be flexibility for modifications. The system must be designed in this way that its mainframe remains unchanged with changes of hardware and modifications or developments of the prototype.

Many works from the area of multi-legged walking machines are dedicated to the mechanical design and motion generation principles considering single prototypes. Discussing the control aspects of walking machines researchers usually relate it to information processing and motion evaluation methods (e.g. 1–5, etc.). Only a limited number of publications[6–13] consider the realization problems of control systems introducing them in a general way. Works on navigation for groups of walking machines are in the initial stage. In this paper we introduce the structure of real-time control system implemented in walking machines with different number of legs. The most advanced version was elaborated for a group of hexapods. The system was elaborated on the assumption that the machines can move together without collisions and following geographic coordinates sent by the operator from a remote central station. The machines are locally self-navigated avoiding obstacles and changing the gaits according to the surface conditions. The first version of elaborated system was tested on a rickshaw and quadruped using a different controller from that of hexapods. The paper discussing the mainframe of the control software discussing the task distribution and the method used for real-time task coordination. A navigation method is shortly introduced.

II. CONTROL SYSTEM DEVELOPMENT

The main kernel of the control software was developed step by step starting with the control of a two-legged small rickshaw, with a later transfer into a small quadruped[14] and the final implementation of the extended system into the hexapods. Kinematic parameters and data exchange schemes used inside the software parts were defined assuming that the change in size and number of legs did not require any changes in the mainframe of the software. In the rickshaw 6 encoders were used for internal sensing, in the quadruped we had 12 encoders, and for the laboratory tests of force-controlled motion we had one three component leg-end force sensor.[15] In the rickshaw and quadruped prototypes we used off board-control hardware, in the hexapods on board control system was applied.

Considering the differences of build prototypes and possible differences in their low level control hardware it was assumed that the higher level control will be implemented in the PC104 (industrial PC) computer keeping the same real-time operating system.

The most critical demands in the sense of calculation speed and coordination of control tasks were from the hexapods. Each hexapod had 18DOF (3DOF in each leg), with 18 digital encoders. Leg-ends were equipped with contact sensors (6 sensors). For the obstacle avoidance four ultrasonic proximity sensors with an angular range of 35° and the distance range starting from 0.4 m with the chosen maximum range up to 3 m were used. On-board inclinometers were used for measurements of roll and pitch angle. The prototypes were build as self-navigated but remotely guided. Therefore each hexapod was equipped with a magnetic compass for azimuth angle evaluation (1 sensor), a GPS system (interpreted as 1 sensor) and a radio transmitter-receiver for contact with remote operator station (interpreted as 2 sensors). Complementary to the power supply switch was added a radio-monitored plunger used as the remotely operated initializer of the work of the control system (1 sensor). Each hexapod had a total of 34 sensors.

In hexapods, for low level control, Motorola MC68332 micro-controllers with two processors, Central Processing

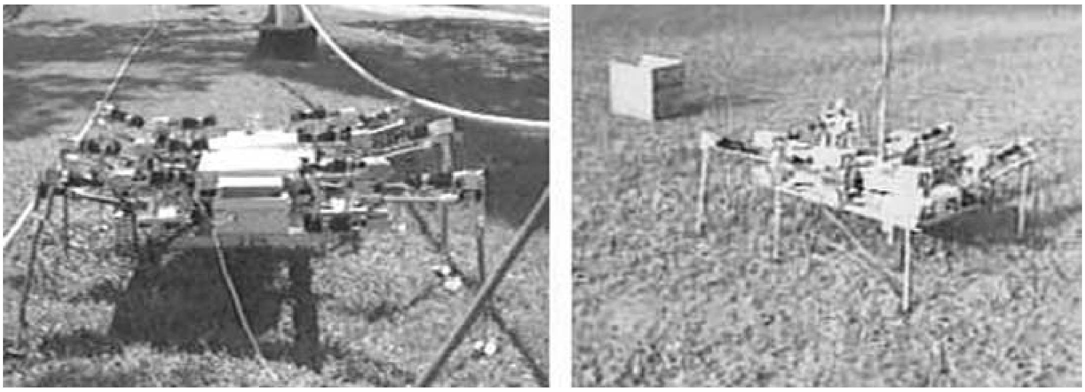

Fig. 1. GROVEN hexapods during self navigated motion.

Unit and Time Processor Unit (CPU and TPU), were applied. Those controllers are so-called Multi Purpose Controllers – MPC, as they can be used for the control of different devices. In our application each MPC controlled three DC motors located in one leg using the feedback from the incremental encoders. Computer PC 104 as the host unit communicated with all micro-controllers. Two-way communication (sending and receiving) was provided through 16 bits (1 single word) input-output (I/O) ports, one port served one controller. The sending to the MPC or receiving were handled by read and write permission bits of the status registers. Micro-controllers were programmed in assembly language. A PID based control algorithm (with modification, including fuzzy logic rules) was used for the generation of the motor control signals. Digital pulse width modulation (PWM) was used for motors speed steering.

To customize the PC–MPC communication non-commercialized protocols were used but the adequate software was written for PC and for MPC. This software served not only for communication but also included error handling and debugging functions.

III. MOTION PRINCIPLES

The built hexapods (Figure 1) GROVEN I,II,II,IV with a weight from 250 to 450N had small variations in size; their body length was in the range 0.8 m and width – 0.4 m. With the applied mechanical structure it was very easy to change the body height, but the typical body height kept during walk was 0.3 m for the convenience of walking. The typical step length used by in the majority of gaits was equal to 0.28 m.

The motion path was remotely specified but not determined by the operator (for the details of body path planning considering the terrain map see reference [2]). The operator sent the path description giving the list of geographical coordinates (geographical altitude and attitude of each target point). Using the radio link, the operator initialized the motion, and at any moment he was allowed to send an emergency stop command. During the motion to the specific point, it was possible to modify the next (update, or cancel). The operator received the information about the coordinates of the actual point being on target, the currently passed distance from the previous point and the information about the detected obstacles. During the walk towards the target each machine avoided obstacles, the detailed shape of the route resulted from the terrain conditions (distribution of the obstacles). The operator was not involved in the gait specification. The current path was locally modified by the on board control system basis on the information obtained from compass, inclinometers and proximity sensors. The device moved from one target point to the next along the shortest path which was a straight line if possible, if it was not (due to the obstacles distribution), the motion trajectory was kept inside a certain *corridor* (Figure 2). When the obstacle was too wide comparing the acceptable corridor width the device stopped and informed the operator (Figure 2 – right).

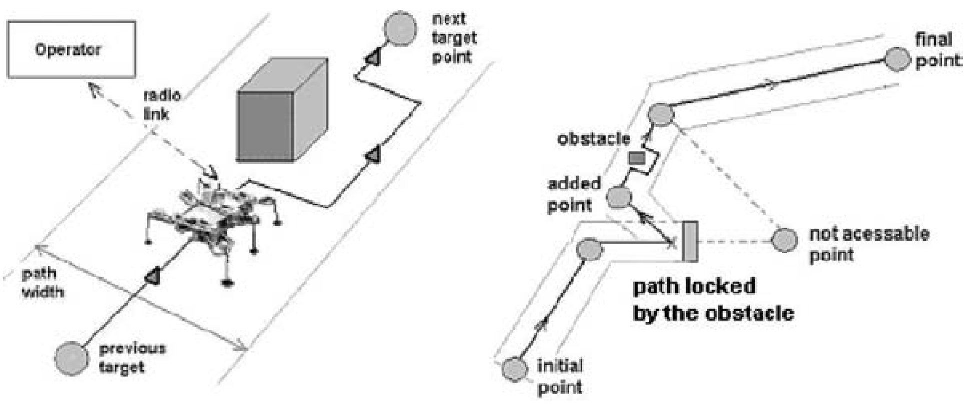

Fig. 2. Illustration of the navigation method.

He could then either modify (change) the actual target point or stop the motion. With the successful avoidance of the obstacle, the device was returning to the central line of the path (Figure 2). The actual gait pattern was chosen by the on-board control system (from the available set of gaits), considering the environment conditions sensed by inclinometers (changes of pitch and yaw angle during walk).

When, using a correction, a change of motion direction was needed and the turning gait was activated. The side gait to the left, or to the right was used for obstacle avoidance until the obstacle went out of the motion direction towards a current target (Figure 2 – left). The fastest tripod gait was used for a smooth surface detected by the small changes of machine roll and pitch angle. In a tripod gait at anytime the body is supported by three legs and another three are in transfer; it is the fastest statically stable hexapods gait. When, for several consecutive walking steps, the inclinometers registered fluctuations of inclination angles (encountering rough terrain), the five legged gait – also called the wave gait – with better support conditions was used. This gait is slowest but offers the *most stable* walking conditions because in every moment the body is supported by five legs, and only one is in transfer. Using experiments we formed that for our machines walking on natural grass with the variance of inclination being in the range of 6° (around zero), and with inclination not greater (in absolute value) that 8° stable motion by a tripod gait is possible, one of those parameters overcoming this acceptable range machine transferred from tripod to the wave gait. The machine will switch back from wave to the tripod gait when for the last several steps the variance of inclination drops back again to below 6° and peaks of inclination are below 8°. Figure 2 illustrates the motion path on a *macro-scale*; on a *micro-scale* heading corrections are shown (Figure 3).

The unevenness of terrain can create an uneven leg support for those producing a body rotation around the vertical axis (yaw angle), and this is registered by the compass as a change of motion direction. To avoid the constant and not needed correction of the direction it was assumed that the motion direction is not corrected with the difference between the expected and actual direction remaining in the range of 10°. This value was chosen based on observations. In Figure 3 the machine moves from point P_1 to P_2. When the absolute value of the difference between expected and current heading (angle α) does not exceed 10° the machine will move without a direction correction until the point B where the angle between line BP_2 and the current motion direction (angle β) reaches (or exceeds) 10°. At this point the machine will correct the motion direction as it is illustrated. When inclinometers register the inclination towards the motion direction (small hill or valley with inclination in our experiments – up to 15° uphill or downhill) hexapods transferred from the tripod gait to the proper wave gait (wave gait with the leg-ends motion range adopted to the inclination). The adequate to the inclination shift of the leg-ends motion range in relation to the hip stabilized the posture. The mentioned motion range was shifted backwards when moving uphill, and shifted forwards when moving downhill.

The maximum range of proximity sensors was set to 1 m for side sensors and 2 m for front sensors. The path width was equal to 3 m, the range for positioning the body central line. Considering the body width and side sensors range meant that the device was able to avoid the obstacle blocking up to 2 m of the acceptable corridor. When the front obstacle was detected for the body located symmetrically on the corridor central line, avoiding by the right side was preferred (Figure 2). Only when the motion to the right resulted in crossing the corridor boundary (due to the permanent detection of the obstacle) the device moved to the left until the end of the obstacle or the corridor boundary. In the last situation the information about the blocking condition was sent to the operator. This message was accompanied by the actual coordinates of the machine and the position of the obstacle (where it is located). Two possible actions were to modify (change) the current target point, or to stop the motion.

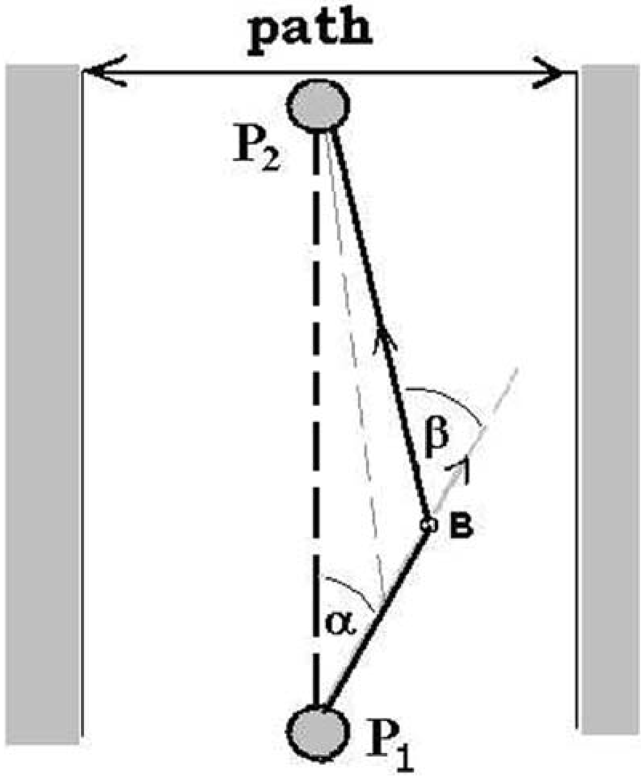

Fig. 3. Heading correction.

The limitation of motion area to a given corridor prevented the collisions between the hexapods when they move in a group. The operator communicated with every robot.

IV. NAVIGATION PROBLEMS

The accuracy of the applied commercial GPS system was limited to 5 m; the walking distances were not large – several hundreds of meters in total Considering this we used the GPS data only for the evaluation of the starting position. During the motion, the actual position was calculated incrementally, considering the changes of motion direction and the gait step length. According to our observations, incremental navigation was not worse than GPS accuracy for considered walking distances; in our tests for motions with different trajectories the final positioning error was never greater than 5 m.

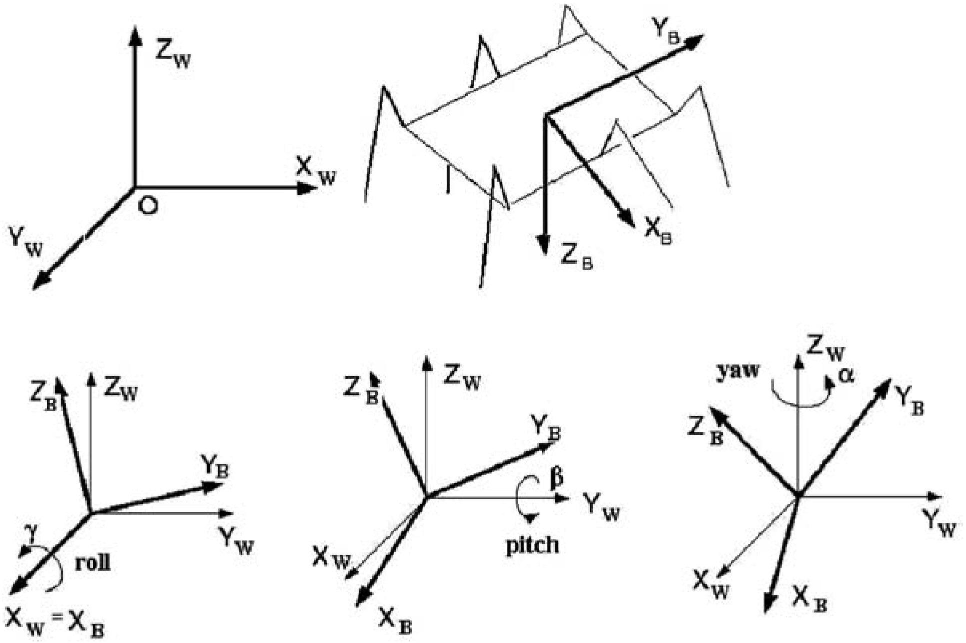

Fig. 4. Coordinate frames and transformations (γ–roll, β–pitch, α–yaw).

An incremental navigation formula used the information about 3 possible rotations of the body in relation to a non-moving world frame. Yaw (rotation around Z_W – vertical axis) results from changes of body heading (see Figure 4) and for smooth terrain can be found using the information about the realized gait. The real value is confirmed by the compass measurements. Pitch (rotation around the Y_W axis) and roll (rotation around the X_W axis) are the result of surface properties and were obtained from inclinometers.

The control system generates coordinates of leg-end trajectories expressed in hip frames H_k; it is easy to transform it to the body frame B as the hip frames are only translated.

The homogeneous matrix expressing the body B frame localization in relation to the world frame W is equal to:

$$
{}^{\mathbf{W}}_{\mathbf{B}}\mathbf{T} = \begin{bmatrix} c\alpha\ c\beta & c\alpha\ s\beta\ c\gamma - s\alpha\ c\gamma & c\alpha\ s\beta\ c\gamma - s\alpha\ s\gamma & {}^{W}_{B}p_x \\ s\alpha\ c\gamma & s\alpha\ s\beta\ s\gamma + c\alpha\ c\gamma & s\alpha\ s\beta\ c\gamma - c\alpha\ s\gamma & {}^{W}_{B}p_y \\ -s\beta & c\beta\ s\gamma & c\beta\ c\gamma & {}^{W}_{B}p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

where $s\alpha = \sin(\alpha)$, $c\gamma = \cos(\gamma)$, and so on.

The last column of homogeneous matrix describes translation from the origin of frame W to the origin of frame B (expressed in frame W). Let us assume that the k-th leg coordinates are described in the body frame by:

$$
{}^{\mathbf{B}}\mathbf{X_k} = \begin{bmatrix} {}^{B}x_k \\ {}^{B}y_k \\ {}^{B}z_k \\ 1 \end{bmatrix}
$$

It is easy to express the leg-end position and orientation in frame W:

$$
{}^{\mathbf{W}}\mathbf{X_k} = {}^{\mathbf{W}}_{\mathbf{B}}\mathbf{T}\ {}^{\mathbf{B}}\mathbf{X_k} \tag{1}
$$

For the evaluation of body frame localization as well for the evaluation of ${}^{\mathbf{W}}\mathbf{X_k}$ (if it is needed), the actual value of matrix ${}^{\mathbf{W}}_{\mathbf{B}}\mathbf{T}$ must be known, for it is used in the incremental formula. The increment is one sequence of leg transfers. Let ${}^{\mathbf{W}}_{\mathbf{B(i-1)}}\mathbf{T}$ describes the homogenous matrix after $i-1$-th sequence, and $\Delta^{\mathbf{B(i-1)}}_{\mathbf{B(i)}}\mathbf{T}$ describes the body rotations and translation realized during the i-th sequence. After this sequence, the matrix ${}^{W}_{B(i)}T$ describing B in frame W is equal to:

$$
{}^{\mathbf{W}}_{\mathbf{B(i)}}\mathbf{T} = {}^{\mathbf{W}}_{\mathbf{B(i-1)}}\mathbf{T}\ \Delta^{\mathbf{B(i-1)}}_{\mathbf{B(i)}}\mathbf{T} \tag{2}
$$

Matrix $\Delta^{\mathbf{B(i)}}_{\mathbf{B(i-1)}}\mathbf{T}$ is expressed by:

$$
\Delta^{\mathbf{B(i)}}_{\mathbf{B(i-1)}}\mathbf{T} = \begin{bmatrix} c\alpha_i\ c\beta_i & c\alpha_i\ s\beta_i\ c\gamma_i - s\alpha_i\ c\gamma_i & c\alpha_i\ s\beta_i\ c\gamma_i - s\alpha_i\ s\gamma_i & \Delta p_{xi} \\ s\alpha_i\ c\gamma_i & s\alpha_i\ s\beta_i\ s\gamma_i + c\alpha_i\ c\gamma_i & s\alpha_i\ s\beta_i\ c\gamma_i - c\alpha_i\ s\gamma_i & \Delta p_{yi} \\ -s\beta_i & c\beta_i\ s\gamma_i & c\beta_i\ c\gamma_i & \Delta p_{zi} \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

where α_i, β_i, γ_i are the body rotations realized in the i-th sequence and Δp_{xi}, Δp_{yi}, Δp_{zi} describes the translation of body frame B_i origin in relation to frame B_{i-1} during this sequence. Translations are obtained from the kinematics of gait pattern. In turning gaits the value of rotation (yaw α_i) in one sequence of leg transfers is also know from the gait descriptio.[1] In a real situation the value of α_i was confirmed by the compass measurements, and we observed good agreement between the measured and the pattern value. Angles γ_i, β_i (roll, pitch) are obtained from inclinometers. Considering the relation (1) and (2), the leg-end position expressed in frame W after i-th gait sequence is equal to:

$$
{}^{\mathbf{W}}\mathbf{X_{k(i)}} = {}^{\mathbf{W}}_{\mathbf{B(i)}}\mathbf{T}\ {}^{\mathbf{B}}\mathbf{X_{k(i)}} \tag{3}
$$

V. ARCHITECTURE OF CONTROL SYSTEM

As it was stated, the lowest–actuating level of the control software was implemented in Multi Purpose Controllers (MPC) performing the motors motion control. The on-board host unit (PC104) was responsible for the communication with operator, receiving the information from external sensors and controlling the motion with sending the reference data to the MPCs. The higher level control was performed by the concurrent processes (tasks) created using the real-time operating system. Despite the specific realization the main feature of every real-time system is that its processes (tasks) exchange data according to time schemes, depending on the system state, and according to other conditions specified by the programmer.[13,16,17] The functional architecture of PC104 implemented control software resembled the hierarchical architecture. The difference between an implemented system and a system with traditional hierarchy (very popular in industrial manipulators) lies in the task functions and task dependencies. In our system, the highest level sends to the lower – the demand (walk to the specific point), and the lower level uses all his tools (sets of available gaits) trying to fulfill the demand; only when it is not possible, the higher level is informed. In typical hierarchical systems the decisions of higher levels are obligatory, and in walking machines motion is performed according to pre-planned paths build using the environment models. In our prototypes the exact motion path was not pre-specified; changes were possible according to the environmental conditions and operator plans. We named our system structure *externally driven soft hierarchical structure* because system actions are driven in real-time by the external conditions and by an operator's directives.

In the system implementation there are three processes responsible for main functions; it is the Brain, Leg and Driver process (Figure 5). The so called Brain process monitors the motion, calculating (in proper moments) the distance and direction to the next motion target point. This utilizes the coordinates of the already reached point and the next point specified by the operator. Motion distance is expressed with a resolution of centimeters, motion direction is expressed by the angle measured in relation to the geographic north (azimuth). The demanded motion distance and heading is transferred to the Leg process which uses all its "tools" to fulfill the Brain demand. Those tools are gaits chosen accordingly to the motion conditions. The Leg process is responsible for the path and gait evaluation considering the detected external conditions and the Brain demand. This process calculates the gait patterns in the slow motion mode or evaluates the gait by specifying the gait code in the fast mode. The Leg controls the obstacle avoidance according to the summarized principles and governs the local incremental navigation considering the gait steps length and heading information (as already described).

The Driver process is the last (lowest) level of the host software; it communicates with the MPCs using the communication protocol described in Part One.

The reference values produced by the Leg process are sent to the Driver. In the faster mode it is the gait code. In the slower mode, for the chosen gait, the inverse kinematics problem is solved and the Leg sends to the Driver the leg joints angular position. Using the stored information

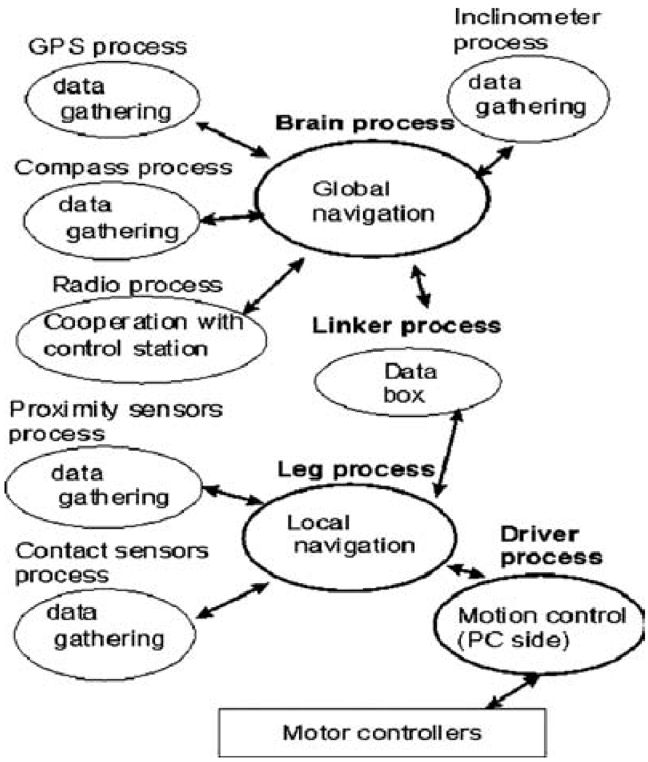

Fig. 5. Block diagram of control software.

about the actually realized angles the Driver calculates the angle increments and express it in the increments of encoder counts. For each set of reference values produced by the Leg, the calculated increments must be realized in one control step. The feedback devices are incremental encoders. The values are sent to the motion controllers (MPCs) using the communication protocol described in Part One. For the change of low level hardware, for example from MPC to MC (dedicated motion controllers) having their control commands and communication frame, the Driver remains unchanged. Only the different library of subroutines responsible for communication with motion controllers must be included. It was done for the LAVA quadruped.

For gathering the sensory information and for communication by radio link with an operator additional processes are used. They are the servers to the Brain and the Leg. They collect the sensory information with filtering, averaging and actualizing the readings. The processes (tasks) serving the sensors are always keeping the latest delivered information. The frequency of the readings is not important and not evident by watching the system actions.

In the system structure presented in Figure 5, an additional process called Linker is specified. This process is used only as the post box. The Leg or Brain deposits the addressed data or messages at the Linker. The receiver (given in the address) collects the data only when they are needed. The data needed for the motion navigation by the Leg are kept in the Linker. With this structure the latest possible valid readings are available without delay. Communication through the Linker prevents the waiting for requested information when the information deliverer is busy with the other tasks. For example, when the Brain is busy receiving the commands from the control station (which is a totally an asynchronous event) the Leg is able to collect the data from the compass stored in the Linker.

The Linker keeps also the information about currently passed distance, heading and the status of obstacle avoidance produced by the Leg. It can be used by the Brain when sending the data for a wish of an operator.

VI. THE MAIN PART OF CONTROL SYSTEM: BRAIN

For the implementation of the above multi-processes structure the real time QNX operating system and *Watcom C* programming language were used. The inter-process communication is performed according to the typical client-server pattern.[14,18] Figure 6 illustrates the simplified state diagram of the Brain. For a clearer illustration, only the main actions are considered. After the system initialization the Brain process waits for the special command sent by the operator through the radio link. If this command is received, the process serving the radio link (Radio process) will transfer the information to the Brain. Then Brain is in the waiting state expecting the coordinates of target points sent by an operator through the radio link. After receiving those data (and confirming it) the Brain goes into the motion loop. The first action distinguished in our diagram for this loop is the check for the new readings from sensors or radio (contacting adequate processes). It must be noted that for the very first target point the sensory reading are obligatory for starting the motion loop (in the drawing it is neglected for simplicity). With this motion loop scheme the actual sensory data is always present and used for evaluation of motion distance and motion direction. The processed sensory data is sent to the Linker process with the address of Leg as the receiver together with motion or motion continuation request. The *Watcom C* sending mechanism applied by us is combined with the automatic receiving, excluding the situation when the motion state readings are for a special operator's request (announced by the Radio to the Brain and transferred to the Linker – *Radio process: data request*. After sending the motion or motion continuation (addressed to Leg, transmitted to the Linker), the Brain receives the deposited motion state (heading and passed distance). If the Radio process produces a request for the current state, it will be sent using the data lastly received by the Brain. If there are no new sensors data and the motion is in progress, the Brain ensures the Leg by sending only the information that the motion should

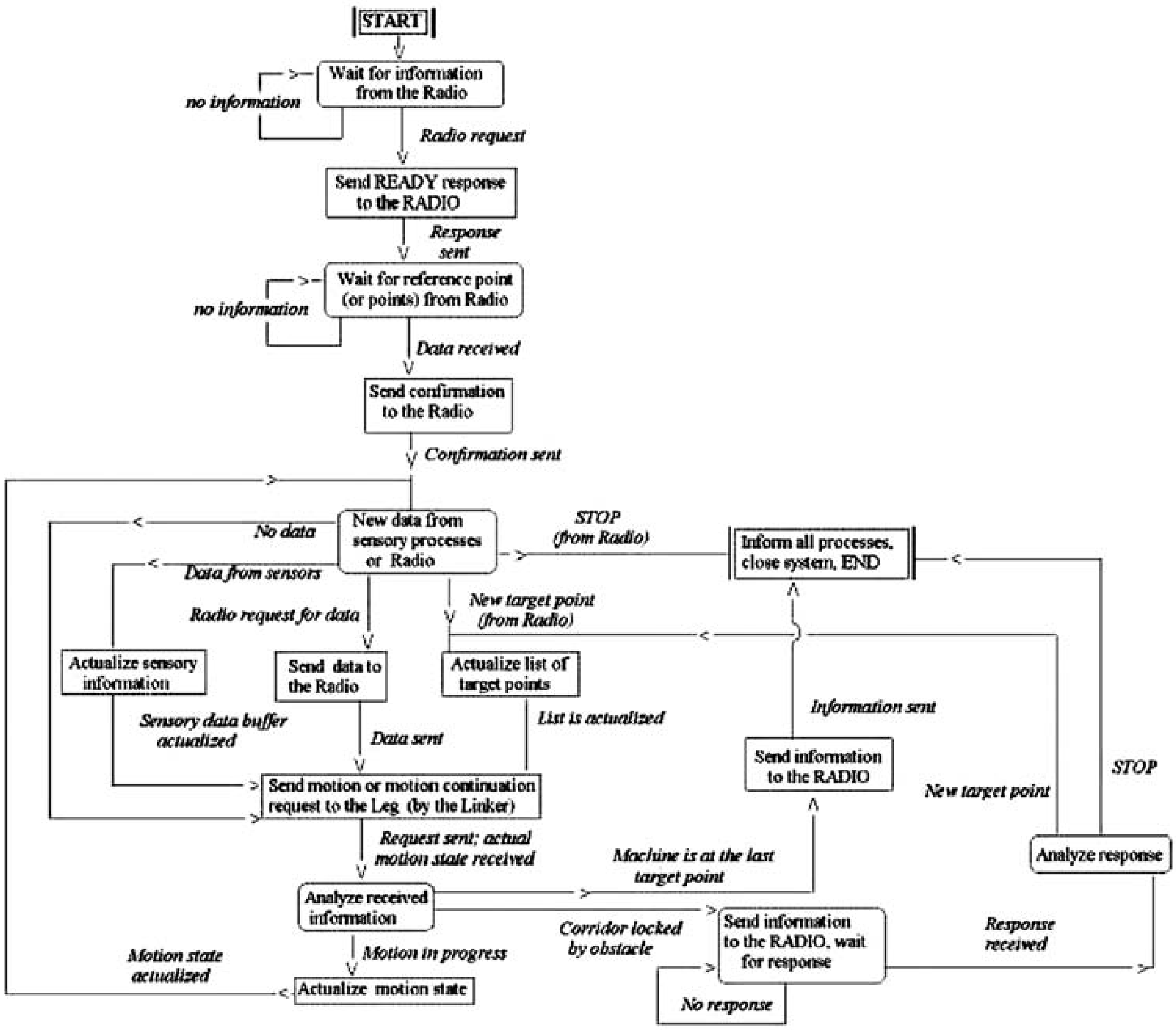

Fig. 6. Block diagram of Brain action.

be continued. If the new data were received (e.g. azimuth angle) the continuation request will be accompanied by the current motion direction. It will be used by the Leg for the test, if to correct the direction or not. If the route new point is added or changed by the operator, the Brain will actualize the targets list (in Figure 6 – *actualize the motion plan*). It is not possible to modify the coordinates of the point to which the machine currently moves. If the path is totally locked by the obstacle, the information is sent to the Brain and next to the operator through the Radio process. In this situation, the Leg automatically stops the motion waiting for an operator's decision. The system will finish its action when the command STOP is received by the radio link. The system also stops when the last target point is reached. The last point sent by an operator in a route description is specially marked; this prevents the system to wait for the next points.

VII. CONCLUSION

For the proper interpretation of presented results it must be noted that the values listed during the discussion of relation between inclination changes and gait choices were identified experimentally by considering the specific terrain and devices with a concrete weight and size. For the different surface and machines with different size and built those values cannot give a good effect.

The design target of complex control systems for novel robots with advanced motion abilities is to produce the controllers (hardware and software) which is easy to program, re-program and debug. Those systems must be enabled to implement different motion principles with different hardware extensions (actuators, sensors, etc.). Despite the current interest in robot control systems, most of the described applications concern manipulators or mobile robots. The majority of autonomous walking robots moves with on-board motion generation but without remote operator guidance, and they may not follow the reference paths.

The author's aim was to show how to build a control system which is flexible for changes and effective in work considering motion control of walking machines or mobile (wheeled) robots. The described structure of the control system and the logic scheme of actions (Brain) can be applied as a mainframe control software for other mobile devices with different methods of motion generation.[19] Presented navigation rules can be used for controlling the motion of a group of robots under distant operator guidance.

Acknowledgments

Work on development of rickshaw and quadruped was conducted at the Robotics Research Centre, Nanyang Technological University (Singapore) under the management of Dr. G. Seet. Work on hexapods was conducted by a team of mechanical, software and control engineers under the management of K. C. Choong (GROVEN). The main contribution to the mechanical design was made by J. Heng. Low level software was implemented by V. Wong. T. Zielinska evaluated the motion methods, designed and implemented the higher level control and communication protocol.

References

1. S. Bai, K. H. Low and T. Zielinska, "Quadruped free gait generation for straight-line and circular trajectories," *Advanced Robotics* **13**(5), 513–538 (1999).
2. S. Bai and K. H. Low, "Terrain evaluation and its application to path planning for walking machines," *Advanced Robotics* **15**(7), 729–748 (2001).
3. M. Likhachev and R. Arkin, "Spatio-temporal case-based reasoning for behavioral selection," in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)* (2001) pp. 1627–1634.
4. R. A. Brooks, "A robust layered control system for a mobile robot," *IEEE J. of Robotics and Automation* **RA2**(1), 14–23 (1986).
5. T. Zielinska and K. H. Heng "Mechanical design of multifunctional quadruped," *Mechanism and Machine Theory* **38**, No. 5, 463–478 (2003).
6. K. Berns, W. Ilg, M. Deck, J. Albiez and R. Dillman, "Mechanical construction and computer architecture of the four-legged machine BISAM," *IEEE Trans. on Mechatronics* **4**(1), 1–7 (1999).
7. M. C. Birch, R. D. Quinn, G. Hahm, S. M. Phillips, B. T. Drennan, A. J. Fife, R. D. Beer, X. Yu, S. L. Garverick, S. Laksanacharoen, A. J. Pollack and R. E. Ritzmann, "Cricket-Based Robots; Introducing an Autonomous Hybrid Microrobot Propelled by Legs and Supported by Legs and Wheels," *IEEE Robotics and Automation Magazine* **9**, No. 4, 20–30 (2002).
8. P. Gonzalez de Santos, J. A. Galvez, J. Estremera and E. Garcia, "SIL04: A true walking robot for the comparative study of walking machines techniques," *IEEE Robotics and Automation Magazine* **10**, No. 4, 23–32 (2002).
9. L. Minati and A. Zorat, "A tree architecture with hierarchical data processing on a sensor-rich hexapod robot," *Advanced Robotics* **16**, No. 7, 595–608 (2002).
10. F. Pfeiffer, J. Eltze and H. J. Weidemann, "The TUM-Walking Machine," *Intelligent Automation and Soft Computing* **1**, 307–323 (1995).
11. F. Pfeiffer and J. Steuer, "Design of Walking Machines – Control Aspects," *14th Triennal World Congress*, Bejing, P. R. China (1999) pp. 413–418.
12. N. G. Shakev, J. K. Tombakov, I. Kalvatchev, A. V. Topalov, P. E. Pavlova and H. L. Akin, "Control system architecture for an autonomous quadruped robot," *5th Int. Conf. On Climbing and Walking Robots (CLAWAR'2002)* (2002) pp. 867–874.
13. C. Zielinski, "Motion Generation in MRROC++ Based Robot Controller," *ROMANSY 14, Theory and Practice of Robots and Manipulators. CISM Courses and Lectures No. 438* (Springer Verlag 2002) pp. 299–306.
14. T. Zielinska and K. H. Heng, "Multifunctional Walking Quadruped," *ROBOTICA* **20**, Part 5, 585–593 (2002).
15. T. Zielinska and K. H. Heng, "Development of a walking machine: Mechanical Design and Control Problems," *Mechatronics: The Science of Intelligent Machines* **12**, No. 5, 737–754 (2002).
16. D. Henrick and T. Honiger, "Parallel processing approaches in robotics," *IEEE Int. Symp. on Industrial Electronics (ISIE'97)* (1997) **Vol. 2**, pp. 702–707.
17. W. A. Halang and K. M. Sacha, *Real-time Systems*. (World Scientific, 1992).
18. H. S. Dulimarta and A. K. Jain, "A Clinet Server Control Architecture for Robot Navigation," *Pattern Recognition* **29**, No. 8, 1259–1284 (1996).
19. C. Zielinski and T. Zielinska, "Sensor-Based Reactive Robot Control," *Proc. 10-th CISM-IFToMM Symp. on Theory and Practice of Robots and Manipulators* (Springer Verlag, 1995) pp. 315–322.