Wheeeler – Hypermobile Robot*

Grzegorz Granosik[1], Krzysztof Mianowski[2], and Michał Pytasz[1]

[1] Technical University of Lodz, Institute of Automatic Control, Stefanowskiego 18/22, 90-924 Lodz, Poland
granosik@p.lodz.pl, mpytasz@wpk.p.lodz.pl
[2] Warsaw University of Technology, The Faculty of Power and Aeronautical Engineering, Nowowiejska 24, 00-665 Warsaw, Poland
kmianowski@meil.pw.edu.pl

Abstract. We have designed and built the prototype of hyper redundant, articulated mobile robot propelled on wheels and therefore called - Wheeeler. In this paper we present progress in our project, focusing on modeling and prototyping phase. We show model of the robot built and verified in 3D simulator and proof-of-concept 3-segment device. Wheeeler is designed to operate in a rough terrain and fulfill tasks such as climbing up or down the stairs, going through trenches, avoiding or climbing over obstacles, operating in narrow, limited spaces like ventilation shafts. The major difficulty of control of hypermobile robots is synchronization of multiple actuators. Design of the high level control system, which can help human operator to intuitively steer this robot is the main goal of our project. In the further part of this paper we introduce communication and control architecture.

Keywords: Hypermobile robot, teleoperation.

1 Introduction

Hypermobile robots are a group of articulated body, serpentine robots; however, in comparison to them they introduce actuated wheels, legs or tracks. Such machines can travel in rough terrain and overcome obstacles much higher than robot itself. In addition to high mobility, due to its slender body, robot can crawl into narrow spaces or pipes for inspection or search and rescue actions. These extended capabilities of hypermobile robots caught attention of researchers in a few laboratories, comparing to the vast number of laboratories working on mobile robots in general. In spite of the fact that design of hypermobile robots is difficult and resource consuming (many identical segments and joints have to be built), there are several working robots and a few practical applications of these robots shown already [4].

The first practical realization of a hypermobile robot, called KR-I, was introduced by Hirose and Morishima from Tokyo Institute of Technology [5] and later improved with version KR-II [6]. This first serpentine robot was large and heavy, had a train-like

* This work was supported by The Ministry of Science and Higher Education under Grant No. 3 T11A 024 30.

A. Gottscheber et al. (Eds.): EUROBOT 2008, CCIS 33, pp. 68–83, 2009.

appearance comprising of multiple vertical cylindrical segments on powered wheels (tracks in KR-I).

There are a few projects of sewer inspection robots realized in Germany [9, 16, 17]. Robots are usually richly equipped with sensors: infrared distance sensors, torque sensors, tilt sensor, angle sensors in every segment, and a video camera in the head segment. Some of them present semi-automatic or automatic driving abilities. Recently developed MAKROplus robot is commercially available [7].

In Japan, country the most suffering from earthquakes, the International Rescue System Institute was established in 2002. One of the goals of this organization is promoting and supporting developments of search and rescue robots. Among many designs, four hypermobile robots have been presented to date. They are Souryu and IRS Souryu, first introduced by Takayama and Hirose [18], 4-segment pneumatically driven Moira [12] and 8-segment Kohga designed by Kamegawa et al. [8].

Researchers from the Mobile Robotics Lab. at the University of Michigan designed the whole family of hyper mobile robots called Omnis [3]. In the OmniPede, the first one, they introduced three innovative functional elements: (1) propulsion elements (here: legs) evenly located around the perimeter of each segment; (2) pneumatic power for joint actuation; and (3) a single so called "drive shaft spine" that transfers mechanical power to all segments from a single drive motor [10]. Further study led to the development of the far more practical OmniTread, which offers two fundamentally important advantages over its predecessor and, in fact, over all other serpentine robots described in the scientific literature to date. These features are: maximal coverage of all sides of all segments with propulsion elements and joint actuation with pneumatic bellows [2].

The newest construction from NREC (National Robotics Engineering Center) is Pipeline Explorer – robot designed and built for inspection of live gas pipelines [15]. This robot has a symmetric seven-element articulated body containing: locomotor modules, battery carrying modules, and support modules, with a computing and electronics module in the middle. It is fully untethered (battery powered, wirelessly controlled) and can be used in explosive underground natural gas distribution pipelines.

An example of reconfigurable hypermobile robot was developed by Zhang et al. [19]. The JL-I system consists of three identical modules having a form of crawlers with skid-steering ability. Each module is an entire robotic system that can perform distributed activities but a docking mechanism enables adjacent modules to connect or disconnect flexibly and automatically. This system with several identical modules which can work separately or simultaneously when assembled, uses hierarchical software, based on the multi-agent behavior-based concept. Robot showed ability to climb steps, span gaps and recover from any rollover situation.

2 Wheeeler Design

Our project of Wheeeler is focused on design of a high level control for hypermobile robots. As we observed from the literature review, most of the hyper mobile robots presented to date lack the autonomy or intuitive teleoperation. Although, every robot has some control system but in most cases they employ multi degree-of-freedom (DOF) joysticks [12] or sophisticated user interfaces [1], or require more then one human operator. Our goal is to simplify teleoperation of these robots and increase their applicability. We start with precise modeling of Wheeeler and designing the most intuitive user interface to control it. Then we show some mechanical details of suspension system and proof-of-concept prototype containing 3 identical segments built with many off-the-shelf components used in RC models technology.

2.1 Simulation

For the modeling phase of the project, the Webots 5 PRO simulator was selected [11]. It is a mobile robot modeling environment which uses VRML model description, interpreted during simulation by ODE physics and OpenGL presentation layers. The modeling scene may consist of multiple elements which can be general solid objects, lights and robots. Each robot can have own controller implemented using C, C++ or Java programming language. Communication between robots is possible through robot nodes simulating wireless communication devices with predefined link parameters.

2.2 Concept of Wheeeler

Structure of described robot is modular. It consists of seven, geometrically identical segments shown in Fig. 1. Each segment has an actuated axle with two wheels and a passive suspension. On each of two ends of a segment, there is a 1DOF actuated joint, to be connected to the following segment, or in case of robot ends – to attach a camera. Assembled robot has 2DOF articulated joints between each two segments, 1DOF joints controlling cameras, and actuated wheels. This gives a total of 3 control variables per segment, and 21 in total.

We assumed position feedback from joints and vision feedback from two cameras mounted on both ends of robot. With these two cameras robot will have advantages similar to Kohga robot providing operator with view from the nose camera and perspective view from behind and above the robot when tail of Wheeeler is lifted in scorpion-like manner (see Fig. 2).

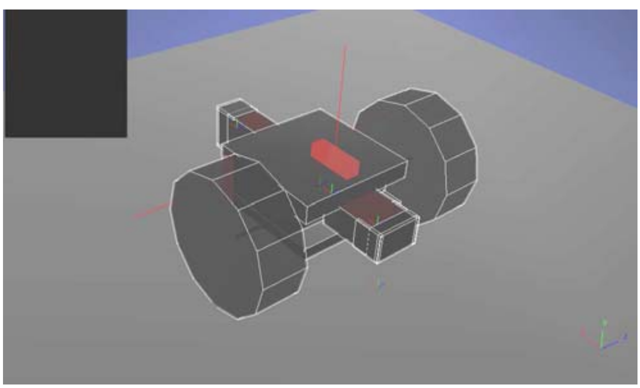

Fig. 1. Segment of Wheeeler

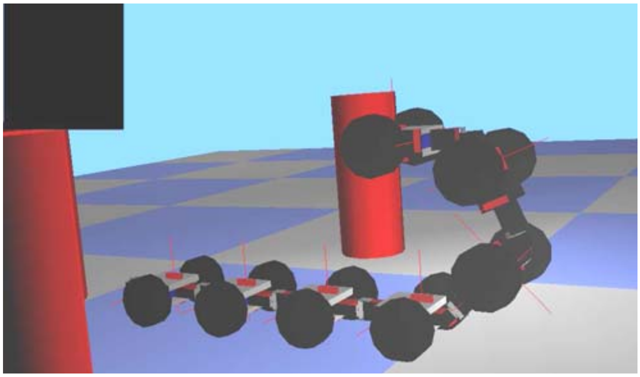

Fig. 2. Scorpion-like pose of Wheeeler

We assume that robot is able to raise two front segments, and therefore climb obstacles with height at least 1.5 times higher than the robot itself, (see Fig. 3). Inter-segment joints working in vertical direction have a range of movement close to ± 90, while in horizontal direction it is a little over ± 45.

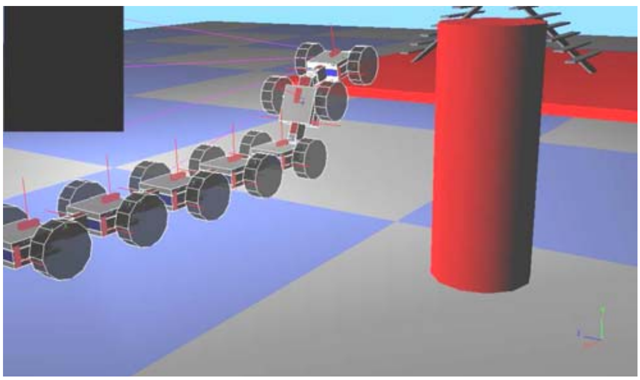

Fig. 3. Wheeeler climbing an obstacle in virtual environment

These ranges combined with short segments and zigzag posture of robot (e.g. as shown in Fig. 4) can compensate for lack of all side tracks (known from Moira or OmniTread). When rotation of upper wheels is opposite to the lower wheels robot is able to enter pipes, shafts, or low ceiling environments.

Each segment is equipped with four distance sensors, 3 axes accelerometer, a gyroscope, and a potentiometer to measure position of suspension mechanism. This gives a total of 9 sensors per segment, 63 sensors in total to be processed.

In current stage of the project, all distance sensors, potentiometers, accelerometers and cameras are implemented. These sensors are available in simulation environment. All information read from them is processed by robot controller and streamed to the client-operator. Gyroscopes are currently being implemented, since in the Webots software they are not supported. However, the functionality of the simulator can be extended through physics plug-in written by user.

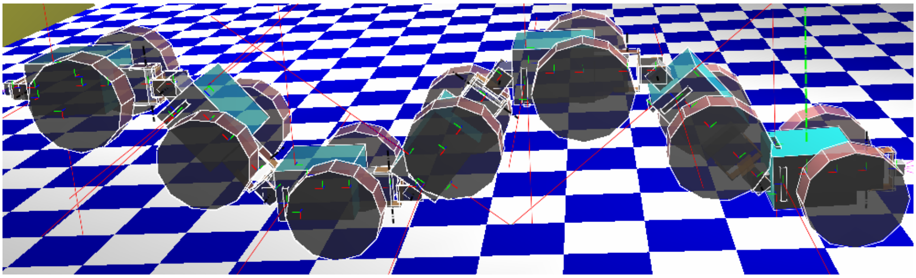

Fig. 4. Wheeeler in zigzag configuration

2.3 Mechanical Details

For precise mechanical design we have been using both AutoCAD and ProEngineer. The 3D CAD models helped us to visualize structure of the transmission and prepare suspension design as shown in Fig. 5.

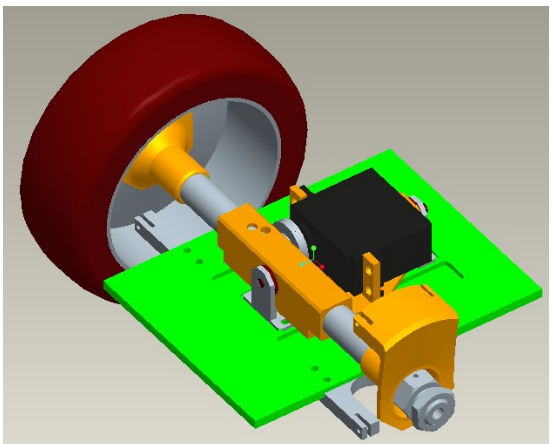

Fig. 5. 3D model of driving system of Wheeeler with passive suspension (springs not shown)

Earlier designs of robots of this kind, very often did not take into account a fact, that the useful force generated by the robot (it's segment), which propels robot and/or generates reaction on hindrance is dependent on the torque/force between considered segments and the ground. This reaction is dependent not only on the gravity component of the mass of considered segment but is dependent on reaction forces between this segment and the neighbor segments of snake-like robot structure. Typical design of articulated mobile robot consists of the number of segments connected by passive and/or active joints, driving systems containing motor with wheels for generating driving forces, and what is essential, driving system is rigidly connected to the body of the segment.

In our design, as shown in Fig. 5 and Fig. 6, the driving module with gear and wheels is connected to the body of segment by the rotational joint. Driving module consists of motor with gear 200:1 and additionally conical gear 1:1 for changing the direction of rotational motion. This module is mounted in the base of the segment of the robot rotationally in such a way, that it can rotate ±10° over longitudinal axis of the robot. This angle is measured with rotary potentiometer. Axis of wheels is supported by two ball-bearings in the funnel, which at ends is connected with the body of segment using springs.

This solution assures enlargement of the angle between the surface of junction with the ground of one segment in the relation the neighbor segment (or two segments) as can be seen in Fig. 7. Beside of this, driving wheels with motor are connected to the body of segment with springs, that additionally improves the reaction forces between the wheels and the ground. Therefore, friction forces between wheels and the ground are greater, what improves performances of the robot.

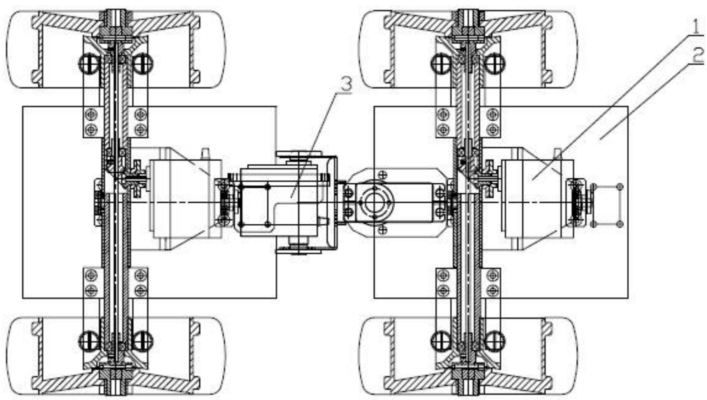

Fig. 6. Basic components of the robot: 1. – driving motor (module contains bevel gears), 2. – base of the segment, 3. – bending motor of the joint

Robot's segment includes driving part and two actuators for changing (controlling) angles of orientation in the relation to two neighbor segments. Obtained solution is characterized by very good integration of all mechanical and/or electrical/electronic components. Current design ensures space for all control/sensory system and a complete equipment for typical mission as was described in previous section.

We are very pleased with the behavior of Wheeeler's passive suspension, which helps to travel in an uneven terrain, as shown in Fig. 7, preserving continuous contact with ground for all wheels of the robot. Even for obstacles as high as half of wheel's diameter springs allow each segment to conform to the ground conditions and provide good grip for all tires.

In Fig. 8 we can see the behavior of springs depending on the position of wheel with respect to the floating platforms. If wheel is lifted up by an obstacle springs extend as shown on the left part of Fig.7, while springs on the wheel which is lower are compressed (right part).

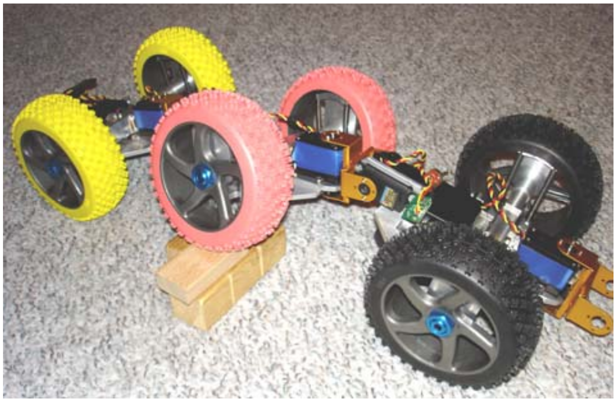

Fig. 7. Prototype of Wheeeler on an uneven terrain – 3 segments (out of planned 7 segments) connected with 2 DOF joints

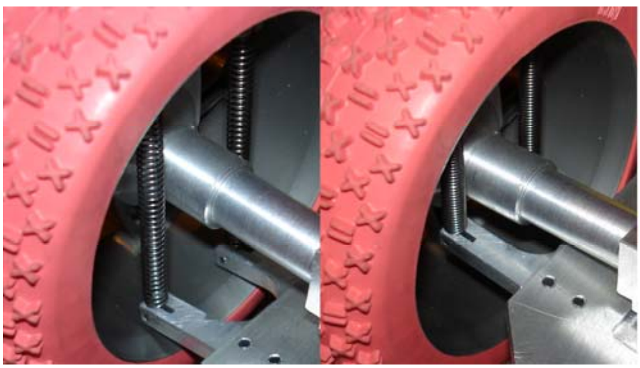

Fig. 8. Behavior of springs in passive suspension during riding over obstacle

3 Control

At first, basic teleoperation with only a visual feedback was introduced. A simple, IP network, socket based client-server application was built, featuring ability to send elementary commands to the robot. Communication was unidirectional, allowing client (operator) send one of the following instructions:

- new angular velocity of the axle of specified segment,
- new position of the horizontal or vertical joint of a segment,
- stop all segments.

This form of control would be very inconvenient in a real application; therefore a simple propagation algorithm for angular position of joints was introduced.

3.1 Follow the Leader Approach

To clearly explain this method, let us draw segments as presented in Fig. 9. To simplify, we assume that maximum angle of turn does not exceed the range of a single, horizontal joint of a segment. In an instant of time t_0 (left part of Fig. 9) robot started to turn and the angle of the turn is α_0. In a next simulation step (right part of Fig. 9), robot moved by Δx. In this case we can express the angle of turn as a sum of α and β. Length of a segment is represented by l, and it is a distance between two following segment joints working in the same direction. This lets us calculate joint variables in the next step of simulation according to equation (1).

$$\alpha = \arcsin\left(\frac{l-\Delta x}{l}\cdot\sin(\alpha_0)\right) \tag{1}$$

The equation (1) assumes that the linear velocity of all segments which have not started to turn is known and common. We have to remember that the velocity does not have to have the same value as the linear velocity of segments which turned already. Different situation can be seen in case of wheels on the turn curve. Their angular velocity is not very significant since in this robot, because of its structure, a slip on the turn curve cannot be avoided, therefore this variable will not be analyzed here. If we would represent the displacement of segments after the turn by Δy, according to the law of cosines (2),

$$\Delta y^2 = (l-\Delta x)^2 + l^2 - 2\cdot(l-\Delta x)\cdot l\cdot\cos(\beta) \tag{2}$$

we can see that displacement Δy according to (3)

$$\Delta y^2 = \Delta x^2 + (2\cdot l^2 - 2\cdot l\cdot\Delta x)\cdot(1-\cos(\beta)) \tag{3}$$

can be equal to Δx only if $cos(\beta) = 1$ or when $\Delta x = l$.

Both situations may not occur due to logical or physical limits. This led us to conclusion, that when the velocity of a robot would be set globally, it could only mean a value for the single, specified segment. If a robot is changing horizontal direction of move, linear velocities of other segments have to be calculated accordingly.

Of course this robot operates in three-dimensional space; therefore the algorithm for propagation of angles may be applied to the vertical joints between segments, with respect to the physical limits derived from segments geometry.

In the above it was assumed, that the robot moves in one, specified direction, first segment is defined by the direction of motion. This robot has to be operated in a user friendly way, therefore for the control purposes a mapping of segment addresses algorithm analyzing current linear velocity is required. The angular velocity of an axle, and therefore wheels is not the only parameter required to find the linear velocity of the robot. The other parameter known through measurement is the gravity direction measured by accelerometers. For the convenience of the operator the control application has to adapt to present conditions, letting one control the first segment, however segment numbering depends on real direction of motion. Currently – in the model – angular velocity of wheels and the pose of the first segment define the direction of robot used for remapping of axes, however in the real case more variables will have to

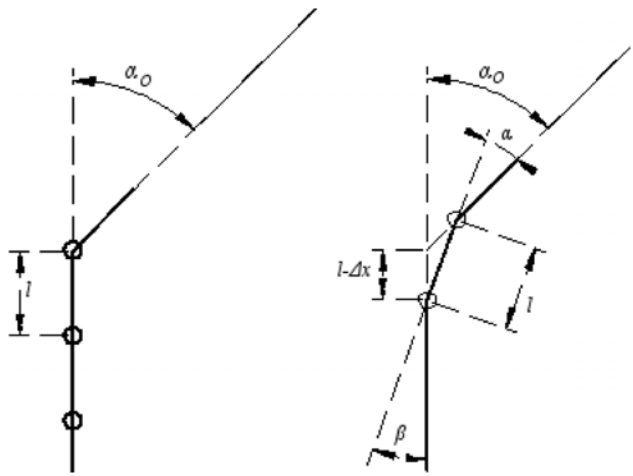

Fig. 9. Kinematics of the horizontal turn

be analyzed. A single sensor can always break, as well as some other distortions may appear, therefore in a real application a sensor fusion mechanism needs to be employed. Fusion of sensor may appear to be even more serviceable in case of attempts to estimate, for example, slippage of the wheels, or other parameters of motion, which cannot be measured in a straight forward way.

The functionality discussed above introduced new control commands to be sent to the robot:

- set/reset auto mode,
- cameras control commands – since in automatic mode we do not control other joints individually.

3.2 Sensor Fusion

To improve the behavior of propagation algorithm in a rough terrain – such as debris, where wheels may bounce on the surface, a basic sensor fusion algorithm was proposed, as schematically shown in Fig. 10. This algorithm assumes that if some of the following segments form a straight line, measurements obtained from these segments should be comparative, therefore they are grouped and the sensors' readings are averaged in a presented way. In the situation when groups are degraded to one segment and acceleration readings are above threshold we assume larger belief to encoder readings (70%). Algorithm shown in this figure is simplified; iterations of calculations for segments or segment groups were omitted.

3.3 Communication Layer

With the further development, operator–robot communication framework was changed. CORBA communication was introduced. The choice was made because of its portability and flexibility and detailed explanation can be found in [13]. With robot development and sensory suite extension the larger amount of data had to be transferred over network, including:

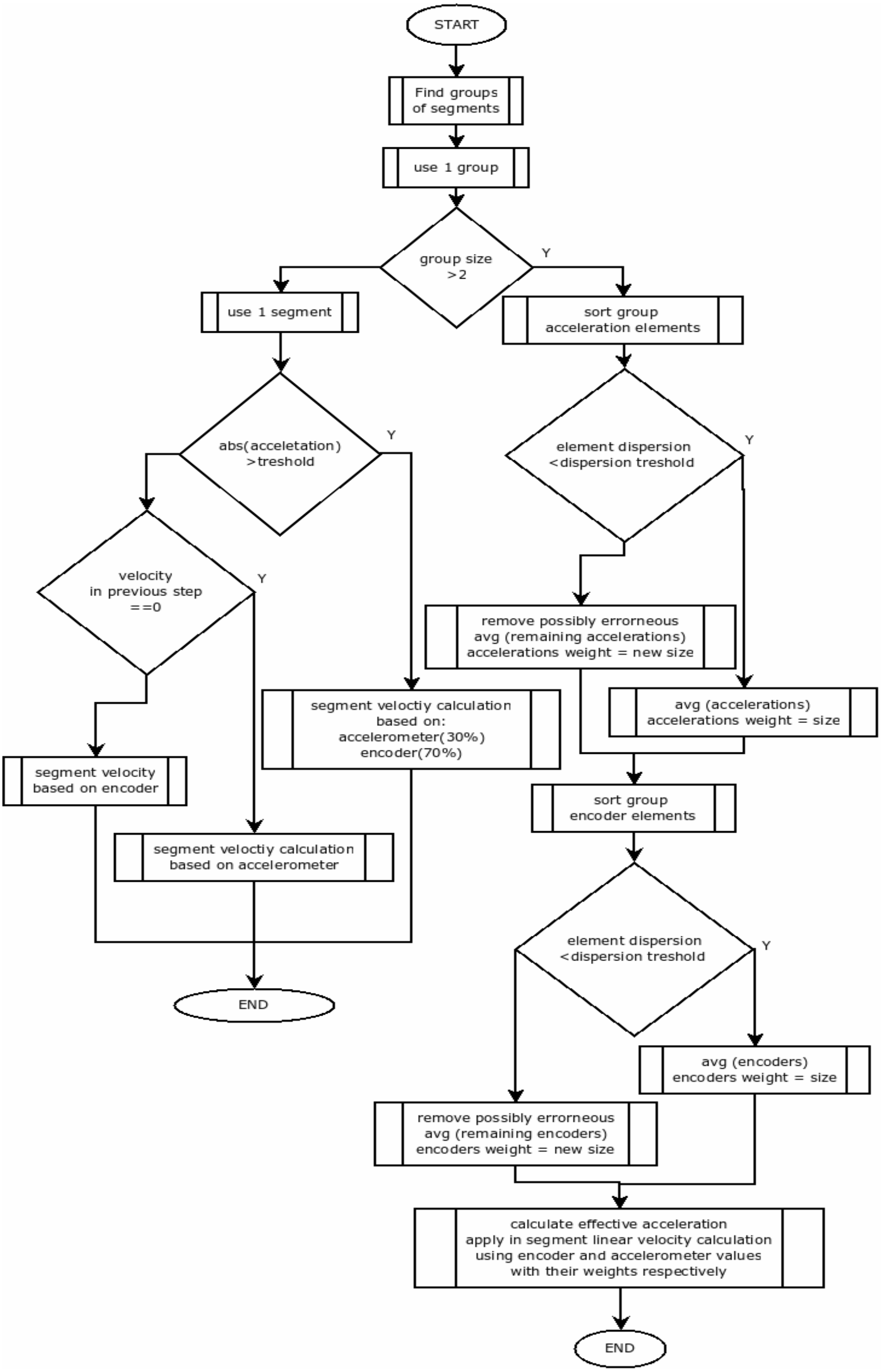

Fig. 10. Basic sensor fusion algorithm

- control commands to robot,
- sensor data from robot,
- video streaming (future plans).

The selected mechanism allows for easy extension of communication features, decreasing probability of programming errors to occur.

In our robot, communication can be divided into 2 sections: control data (shown in Fig. 11) and sensor data (see Fig. 12).

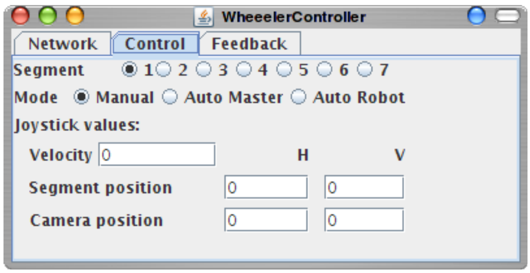

Fig. 11. Control information tab

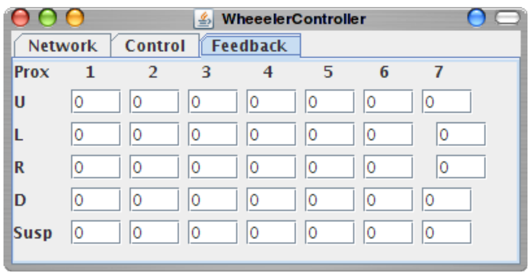

Fig. 12. Sensor information tab

Let us present the current stage of the network interface. The WheeelerControl interface, presented in Fig. 13 is a set of remote methods included in the robot-side servant to be executed by the operator part of control application (details can be found in [14]).

`WheeelerSensors` interface is a servant implemented on the operator side of application. These methods are supposed to be executed by the robot to deliver sensor data to the operator. This interface will be extended with video streaming features as soon as they are implemented in the real robot. Additional set of methods in the `WheeelerControl` interface to access sensors can be considered redundant, however, its real purpose is debugging of application in case of instability in the development phase of the project. In the interfaces mentioned, `set` and `get` methods can be seen, names of which are self explanatory also providing the direction of data flow. Methods starting with prefix `send` are supposed to provide the robot with the references to the remotely available objects on the client side.

```
WheeelerControl
----------------
+setPosition(in servo:octet,in position:float): void
+setHCamPosition(in position:float): void
+setVCamPosition(in position:float): void
+setSpeed(in segment:octet,in speed:float): void
+setModeAuto(in mode:boolean)
+stopNow(): void const
+getProximitySensor(in addr:octet): long
+getSegmentAngle(in addr:octet): float
+getAccenerometer(in addr:octet): float
+getGyroscope(in addr:octet): float
+getForce(in addr:octet): float
+getServoCurrent(in addr:octet): float
+setSensorStreaming(in mode:boolean): boolean
+setVideoStreaming(in mode:boolean): boolean
+sendSensorsReference(in ws:WheeelerSensors): boolean
+sendVideoReference(in wv:WheeelerVideoStreaming): boolean
```

```
WheeelerSensors
----------------
+setProximitySensor(in address:octet,in value:long): void
+setSegmentAngle(in segment:octet,in value:float)
+setAccelerometer(in segment:octet,in value:float): void
+setGyroscope(in segment:octet,in value:float): void
+setForce(in addr:octet,in value:float): void
+setServoCurrent(in addr:octet,in value:float): void
```

Fig. 13. Declaration of CORBA network interfaces

3.4 Controllers

In order to control all the sensors envisioned in Wheeeler and in order to simplify mechanical fit into the segment we have designed and built specialized controller based on the AT90CAN128 (Atmel), as shown in Fig. 14. We have chosen this processor for the fast AVR structure and relatively high processing power, as for 8-bit controllers. Additionally, in-system programming from Atmel offers reprogramming of each processor of the system directly through CAN bus. This will simplify development procedure of the robot's lowest level software. Local controllers are augmented with all necessary peripherials: 3-axis accelerometers LIS3LV02DL (STMicroelectronics), single axis gyroscope ADIS16100 (Analog Devices), quadrature counters LS7366R (LSI/CSI) and IR distance sensors GP2D120 (Sharp). Functionality of controller can be further extended through serial communication interfaces: CAN, SPI, I2C and RS232. CAN bus is used as a main communication means for data acquisition and control.

Local controllers are daisy-chained along robot's body and connected to the main controller realized on PC104 computer. This main controller gathers data from robot and forwards to operator's station via wireless link. It will also be used to transfer video signal. In the opposite direction, control orders come from operator; they are being analyzed in main controller and distributed to local ones. We are planning that main controller will be also responsible for basic autonomous behavior of Wheeeler.

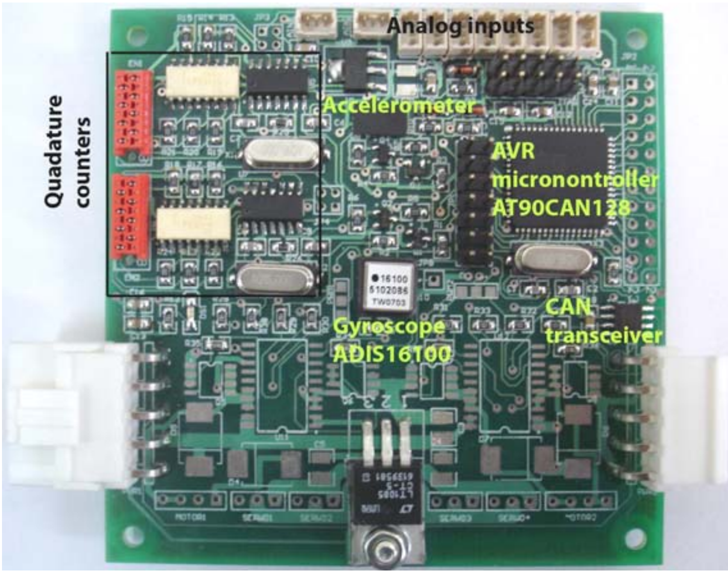

Fig. 14. Local controller mounted on each segment of Wheeeler

4 Experiments

First experiments have been made in the simulation environment. To verify the correctness and efficiency of a suggested control algorithm for this robot a test group of children aged 8-12 was asked to try to fulfill a set of simple tasks (following predefined paths). In the basic one, there was a need to drive through a narrow, straight corridor and traverse debris. In the extended task - climbing up and down the stairs and traversing a trench was required.

After a brief introduction to robot control and some preliminary practice, each member of a test group was asked to make several attempts of fulfilling given tasks with various controller features enabled or disabled. After each attempt the experiment participant was asked about opinion on robot behavior and difficulties in its operation. During all attempts robot's path, all variables concerning active joints and forces vectors were recorded. Also, after the use of the most basic version of the controller, group members were asked to form a team to drive the robot manually controlling each joint variable and each axle velocity.

In the experiment it was proven, that this robot would require a lot of practice to be team-controlled. After one hour training the group was unable to fulfill any of the tasks due to joint variables and velocity synchronization problems. With the most basic controller which included joint variable propagation algorithm it was always possible, after 5 minutes of training, to complete the most basic task, however, propagation accuracy was lost on the debris. It was proven that even with the most basic algorithm it is possible to climb up and down the stairs or traverse a trench; however, the operator must have the ability to enable or disable vertical propagation of joint variables. It has to be stated, that for teleoperated stairs climbing it is required to use a scorpion-like pose, preferably with front camera able to be moved vertically.

After building the proof-of-concept version of Wheeeler, consisting of 3 segments, we have made some preliminary tests to verify robot's behavior, power consumption and performance. Results are presented in the following tables. Table 1 compares speed of the robot on the flat terrain (carpet) and supply current measured for three levels of supply voltage. Table 2 shows the current consumption during driving on the inclined steel flat surface (measurement for two inclinations and three voltage levels).

Table 1. Performance of 3 segment Wheeeler on the flat terrain

Supply voltage [V]	Speed of robot [cm/s]	Starting current [A]	Nominal current [A]
5	32.2	1.7	0.7
6	40.0	1.7	0.7
7	46.0	2.3	0.9

Table 2. Power consumption of 3 segment Wheeeler on inclined surface

Inclination [deg]	Supply voltage [V]	Starting current [A] Going up (down)	Nominal current [A] Going up (down)
16.2	5	2.6 (1.5)	1.4 (0.3)
	6	2.7 (1.6)	1.5 (0.3)
	7	2.8 (1.6)	1.7 (0.5)
21.3	5	2.6	2.0 (0.2)
	6	2.7	2.0 (0.3)
	7	2.8	2.0 (0.4)

5 Conclusions

Hypermobile robots, which are a subcategory of articulated, snake-like robots, are a very interesting field of research. There is a need for robots which are capable of operating in a rough terrain - natural or urban. Typical tasks to be considered are:

- climbing or avoiding obstacles,
- going through trenches,
- moving in narrow spaces such as tunnels, ventilation shafts, ruins inspection.

The common features of hypermobile robots are actuated wheels or tracks, multiple joints on the robot's body allowing it fit into narrow spaces, and relatively large length compared to its width (or field of its cross-section).

The most significant difficulty in designing a hypermobile robot is the need of synchronization of every actuated element during the robot movements. This issue is the main objective of presented project of Wheeeler.

We expect that currently implemented control algorithms let the above requirements to be fulfilled. Nevertheless, for the convenience of the operator and improved safety and reliability of robot operation several improvements should be implemented:

- robot pose should be verified – whether it has not overturned. In such case the robot should be able to restore its default pose automatically, if only surrounding objects do not make it impossible,

- image recognition could be implemented to increase robot autonomy in case of loss of communication with the operator or just to improve teleoperation convenience,

Also a very important feature to be implemented in the near future is video streaming, through CORBA. It is supposed not only to provide a visual feedback, but also supply additional information about the environment through further video processing.

References

1. Baker, J., Borenstein, J.: The Joysnake A Haptic Operator Console for High-Degree-of-Freedom Robots. In: 2006 International Joint Topical Meeting: "Sharing Solutions for Emergencies and Hazardous Environments", Salt Lake City, USA, February 12-15 (2006)
2. Granosik, G., Hansen, M., Borenstein, J.: The OmniTread Serpentine Robot for Industrial Inspection and Surveillance. Industrial Robots International Journal, Special Issue on Mobile Robots IR32-2, 139–148 (2005)
3. Granosik, G., Borenstein, J., Hansen, M.G.: Serpentine Robots for Industrial Inspection and Surveillance. In: Huat, L.K. (ed.) Industrial Robotics. Programming, Simulation and Applications, pp. 633–662. the plV pro literatur Verlag, Germany (2007)
4. Granosik, G.: Hypermobile robots. In: Aschemann, H. (ed.) New Approaches in Automation and Robotics, pp. 315–332. I-Tech Education and Publishing, Vienna (2008)
5. Hirose, S., Morishima, A.: Design and Control of a Mobile Robot With an Articulated Body. The International Journal of Robotics Research 9(2), 99–113 (1990)
6. Hirose, S., Morishima, A., Tukagosi, S., Tsumaki, T., Monobe, H.: Design of Practical Snake Vehicle: Articulated Body Mobile Robot KR-II. In: Fifth Int. Conference on Advanced Robotics, 'Robots in Unstructured Environments', vol. 1, pp. 833–838 (1991)
7. INSPECTOR SYSTEMS Rainer Hitzel GmbH, `http://www.inspector-systems.com`
8. Kamegawa, T., Yamasaki, T., Igarashi, H., Matsuno, F.: Development of the Snake-like Rescue Robot KOHGA. In: Proc. IEEE Int. Conference on Robotics and Automation, New Orleans, LA, April 2004, pp. 5081–5086 (2004)
9. Klaassen, B., Paap, K.L.: GMD-SNAKE2. A Snake-Like Robot Driven by Wheels and a Method for Motion Control. In: Proc. Of IEEE International Conference on Robotics and Automation, Detroit, MI, May 10-15, pp. 3014–3019 (1999)
10. Long, G., Anderson, J., Borenstein, J.: The OmniPede: A New Approach to Obstacle Traversion. In: Proc. IEEE Int. Conf. on Robotics and Automation, USA, pp. 714–719 (2002)
11. Michel, O.: WebotsTM: Professional Mobile Robot Simulation. International Journal of Adavnced Robotic Systems 1(1), 39–42 (2004)
12. Osuka, K., Kitajima, H.: Development of Mobile Inspection Robot for Rescue Activities: MOIRA. In: Proc. IEEE/RSJ Int. Conference on Intelligent Robots and Systems, Las Vegas, Nevada, pp. 3373–3377 (2003)
13. Pytasz, M., Granosik, G.: Applying CORBA technology for the teleoperation of Wheeeler. In: Kozłowski, K. (ed.) Robot Motion and Control 2007 in the "Lecture Notes in Control and Information Sciences", pp. 311–318. Springer, London (2007)
14. Pytasz, M., Granosik, G.: Object oriented network control of Wheeeler – the hyper mobile inspection robot. In: CD Proc. of IEEE International Workshop on Safety, Security, and Rescue Robotics, Rome, Italy, September 27-29 (2007) ISBN: 978-1-4244-1569-4

15. Schempf, H., Mutschler, E., Goltsberg, V., Skoptsov, G., Gavaert, A., Vradis, G.: Explorer: Untethered Real-time Gas Main Assessment Robot System. In: Proc. of Int. Workshop on Advances in Service Robotics, ASER 2003, Bardolino, Italy (2003)
16. Scholl, K.U., Kepplin, V., Berns, K., Dillmann, R.: Controlling a multi-joint robot for autonomous sewer inspection. In: Proc. IEEE Int. Conference on Robotics and Automation, ICRA 2000, vol. 2, pp. 1701–1706 (2000)
17. Streich, H., Adria, O.: Software approach for the autonomous inspection robot MAKRO. In: Proc. IEEE Int. Conference on Robotics and Automation, New Orleans, LA, USA, pp. 3411–3416 (2004)
18. Takayama, T., Hirose, S.: Development of Souryu-I connected crawler vehicle for inspection of narrow and winding space. In: 26th Annual Conference of the IEEE Industrial Electronics Society, IECON 2000, vol. 1, pp. 143–148 (2000)
19. Zhang, H., Wang, W., Deng, Z., Zong, G., Zhang, J.: A Novel Reconfigurable Robot for Urban Search and Rescue. International Journal of Advanced Robotic Systems 3(4), 359–366 (2006)