TIGRE - An autonomous ground robot for outdoor exploration

Alfredo Martins, Guilherme Amaral, André Dias, Carlos Almeida, José Almeida, Eduardo Silva
INESC TEC Robotics Unit
ISEP - School of Engineering Polytechnic Institute of Porto
Email: aom, gsilva, adias, c.almeida, jma, eaps@lsa.isep.ipp.pt

***Abstract*—In this paper we present an autonomous ground robot developed for outdoor applications in unstructured scenarios. The robot was developed as a versatile robotics platform for development, test and validation of research in navigation, control, perception and multiple robot coordination on all terrain scenarios. The hybrid systems approach to the control architecture is discussed in the context of multiple robot coordination. The robot modular hardware and software architecture allows for a wide range of mission applications. A precise navigation system based on high accuracy GPS is used for accurate 3D environment mapping tasks. The vision system is also presented along with some example results from stereo target tracking in operational environment.**

I. INTRODUCTION

In this work we present an autonomous ground vehicle for outdoor exploration. The robot was designed in order to provide a versatile platform for multiple application robotics research in outdoor land scenarios.

A large volume of research has addressed the applications of unmanned ground vehicles in outdoor scenarios and the problems posed are as diverse as motion control, localization, mapping, planning, perception or decision making or artificial intelligence oriented ones [1]. Outdoor land robots can be useful in safety and security applications (both civilian and military), in surveillance and patrolling tasks, in reconnaissance, agriculture, exploration and mapping, for cargo, human transport and logistics support, in establishing communication links or in search and rescue operations.

These applications have different sets of requirements leading to the existence of multiple robotic dedicated solutions. Many mobile research platforms ranging from commercial solutions [2] to custom research lab developed ones [3], [4], [5] are available.

One area of active research and with strong impact is search and rescue applications [6]. Multiple robot approaches with heterogeneous capabilities [7] have been proposed, leading to developments in multi-robot coordination.

In the outdoor field robotics scenario, the European Land Robot Competition (ELROB) [8] has been fostering the development of outdoor mobile robots mainly for security in surveillance and patrol tasks and for transport support.

For (ELROB) senarios (outdoor natural environments) UGVs are developed based on medium or full sized all terrain vehicles [3], [7], [4] in order to operate in the relatively large operation areas and be able to carry a suitable set of sensors.

The TIGRE (Terrestrial Intelligent General purpose Robotic Explorer) robot is a vehicle of this class. It is also based on an all terrain vehicle and combines autonomous drive robot capabilities, such as GPS based navigation, road and terrain classification for motion planning, vision and laser rangefinder obstacle avoidance with outdoor manoeuvrability and specific surveillance sensors such as infra-red vision. Three main guidelines structured the development: capability of operation in medium size areas, to act as research platform in multi-robot coordination in outdoor environments and to support robotic research in particular areas of field robotics such as underground navigation or precise 3D environment modelling.

The navigation system uses high precision GPS for outdoor localization, with particular relevance in missions for precise 3D modelling and the system has also the possibility of using additional higher quality INS sensors for operation in GPS deprived areas. A high precision 3D LIDAR can also be incorporated for modelling tasks allowing the test and development of new modelling and navigation solutions.

In the following sections the TIGRE hardware is described followed by the software architecture.Guidance and control aspects are addressed in section IV. Next, the localization system is described followed by the vision system architecture. Some results from missions in target detection and localization are presented in section VI followed by concluding remarks and future works.

II. HARDWARE

This system is based on a electric propulsion all-terrain vehicle equipped onboard processing (Intel i5 based single board computer,), wireless communications (IEEE 802.11a Ubiquiti Bullet 5GHz access point), infra-red pan&tilt thermographic camera (L3 ThermoEye 5000), laser rangefinder (SICK LMS-200), a visible spectrum camera pair (Basler acA1300-30gc), precision GPS receivers (Septentrio PolaRx2e and Novatel Smart Antenna) and inertial sensors (Microstrain 3DM).

Fig. 1. TIGRE UGV

Traction is achieved through a brushless DC motor physically connected to the rear axle. The direction is also electrically actuated and uses the Ackerman trapezium geometry. A magnetic encoder provides the absolute direction angle.

Four LiFePO4 batteries are used providing a minimum 4hurs of autonomy time assuming a continuous usage at $1ms^{-1}$ vehicle speed. The vehicle is depicted in figure 1. An aluminium frame with a tower was fixed in the vehicle to support all the sensors. The color GigE cameras where positioned at the tower's top in order to provide a stereo vision (with external synchronized trigger control). The thermographic pan & tilt unit is also fixed on top between the stereo pair along with the the IMU. Both the GPS and wireless communication antennas are located at the rear of the tower. At the front of the robot was set the laser range finder unit. The main system electronics is located in a watertight enclosure.

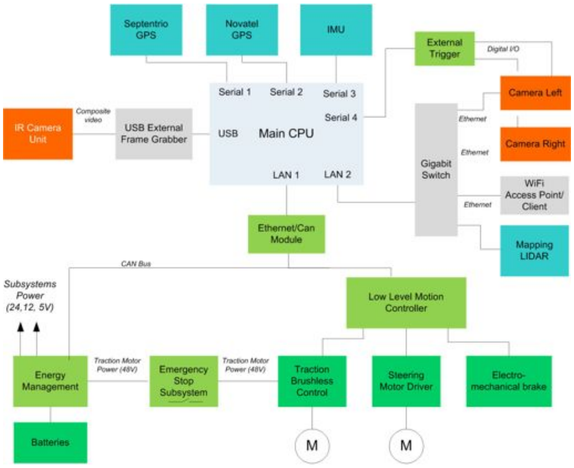

Fig. 2. System architecture

A set of custom made low level vehicle control subsystems (power system control, direction control and traction controller) are connected in a CAN bus. A custom developed Ethernet/CAN interface is used in one of the CPU ethernet ports to provide access to the vehicle CAN bus (see figure 2). In addition, a separate emergency module with a dedicated RF remote is used to cut the traction motors power remotely and/or actuate the mechanical brakes with a small electric actuator.

III. SOFTWARE ARCHITECTURE

The vehicle software architecture follows a modular and hierarchic structure (see Fig. 3).

Lower level modules provide interface with the sensors and actuators (on top of the various Linux device drivers). The image processing is performed in a pipeline structure with increased abstraction and reduced information [9] at the later stages. In *Multi-camera target detection* identified image targets are combined to produce 3D target candidates. The *Target Selection* phase performs tracking and selection of relevant targets. These are then, either used in the *Guidance control* (for ex: in target following) and/or published to other robots in the team for multi-robot operations.

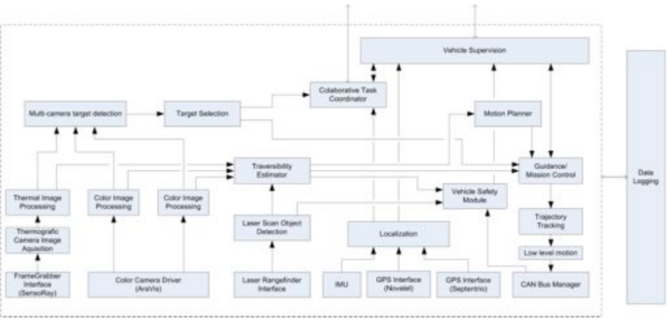

Fig. 3. Software architecture

Image information (color based classification) is combined with detected obstacles with the laser rangefinder to determine the terrain traversability in front of the vehicle. This information is used both in the guidance control implementing the current motion maneuver and in the *Vehicle Safety Module* for emergency stops.

The localization produces a current estimate of the vehicle pose in the world reference frame, from the navigation sensors, namely the IMU and the GPS. This information is used in all the motion control levels (both in *Trajectory Tracking* and *Mission control*) and also sent to other robots in a cooperating scenario. Data logging and remote telemetry can be performed in all the hierarchic levels and software modules.

Motion control is performed by an hybrid system executor (*Mission control*) responsible for generating vehicle trajectories for the *Trajectory tracking*. This module produces vehicle low level velocity and position (steering angle) references sent to the hardware low level motion control trough the CANbus.

The software is implemented in multiple processes and runs on the GNU/Linux operating system.

The ROS framework [10] is used, providing both interprocess communications, adding modularity and maintainability and providing set of useful development and implementation tools. There is no direct 1 to 1 mapping in the architecture represented in Figure 3 and the ROS node graph implementation since some modules in figure 3 are implemented in multiple nodes (such as image processing or localization) and parts of the software are not implemented directly in ROS.

The ROS framework still has large restrictions for multi-robot scenarios and also some limitations on the communications latency and overhead. Some solutions have been provided in order to alleviate the problem posed by the need of existence of a centralized ROS master node. These solutions establish some form of communication between the masters running on each robot. But, the intermittent robot connections (usual large areas of operation with robots entering and leaving the team) are not well supported. To solve this problem for multi-robot communications we use the LCM communication middleware [11] in order to publish/subscribe in another robot the topics of the other robots.

Also taking in mind both the multi-robot applications and the need for accurate time synchronization of data, the Linux

Chrony clock-synchronization daemon is used with PPS (pulse per second) information from GPS. This process allows to synchronize the system clock to GPS time and in addition provides multiple robot clock synchronization.

Additionally video streaming (from any of the cameras) is performed by the VLC streamer with MPEG4v codec.

IV. NAVIGATION GUIDANCE AND CONTROL

A. Control

The vehicle control architecture follows an hybrid systems approach [12] allowing both continuous time control with discrete state evolution. The basic vehicle control maneuvers can be defined as hybrid automata [13]. More complex motion control maneuvers can be obtained by the hierarchical composition of simpler ones. Lower level reactive control can be combined with higher level deliberative planning both in continuous time and by the discrete state transition events occurring (or generated) at the various levels of hierarchy in the maneuvers.

The path tracking used in the basic path following maneuvers (continuous time part) is a simple line of sight controller and is the same used in [14]. The method is a nonlinear feedback function of the cross track error e_p measured from the center of the front axle to the nearest path point (c_x,c_y).

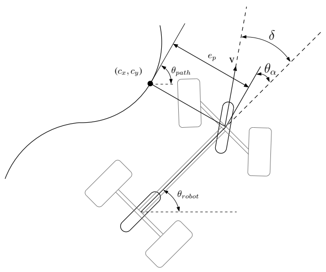

Fig. 4. Path tracking

The first term of the control law simply keeps the wheels aligned with the given path by setting the steering angle δ equal to the heading error.

$$e_\alpha = \theta_{robot} - \theta_{path} \tag{1}$$

where θ_{robot} is the heading of the vehicle and θ_{path} is the heading of the path at (c_x, c_y). In the presence of error e$_p$ a second term adjusts δ such that the intended trajectory intersects the path tangent from (c_x, c_y) at *k v(t)* units from the front axle. The resulting of the steering control law is given as

$$\delta(t) = \theta_e + tan^{-1}\left(\frac{ke_p(t)}{v_x(t)}\right) \tag{2}$$

where *k* is a gain parameter.

Rollover is a important concern in the vehicle locomotion due to the vehicle characteristics and possible rough terrain. According to [15] the preponderant variables for the risk of rollover are sliding parameters (slip angle, tire cornering stiffness) and roll angle. The robot rollover detection is based on roll angle estimation and on velocity and is prevented by limiting the velocity / steering angle relation).

B. Cooperative mission control

In a multi vehicle cooperative setup each vehicle usually provides a high level interface to the team coordination [16]. This can be done in a centralized [16] or distributed [17] ways. Currently there is a strong effort in the research community towards distributed approaches due to advantages in reliability and scalability. In hierarchical coordination approaches [16] higher layers of the vehicle sensing and control infrastructure provide higher level of abstraction information for coordinating purposes. These can be vehicle positions and discrete vehicle states and high level commands.

In TIGRE architecture, multiple vehicle coordination can occur at different levels of hierarchy. Explicit discrete event coordination can occur in a multi-robot setup (such as with other land based robots or with aerial robots) by exchange of coordinating events. These events can be generated at all vehicle control levels and by external entities as other robots. In addition continuous variables (for ex: a particular sensor reading) can be shared between multiple robots leading to tighter multiple vehicle control loops.

Our architecture for multirobot coordination uses an hierarchical approach similar to [16] considering 2 layers, one representing local vehicle control and a multirobot controller. We consider also a coordinating variable $\xi_i = [\xi_c \quad \xi_\sigma]$ and a performance output for each vehicle $z_i = [z_{i_c} \quad z_{i_\sigma}]$ but for these variables we consider not only a continuous part (ξ_c and z_{i_c}) but also an additional discrete part (ξ_σ and z_{i_σ}). In contrast to the two controllers solution proposed by Beard with a discrete event system supervisor and a team formation continuous controller, our architecture has both the multiple robot coordination and the local vehicle controller as hybrid systems allowing discrete and continuous control both locally and for the robot team.

Each robotic vehicle is described in general as a non linear dynamic system S_i with states x_i, inputs u_i and outputs y_i.

The local vehicle controller eq.(3) can be viewed as an hybrid system [12] with a set of continuous time flows for each discrete state q_i and generating the robot control inputs u_i.

$$C_i : \begin{cases} \dot{x}_{q_i} = f_{q_i}\left(x_{q_i}, y_i, \xi_c\right) \\ u_i = b_{q_i}(x_{q_i}, y_i, \xi_c) \\ z_{i_c} = h_{q_i}(x_{q_i}, y_i, \xi_c) \\ q_i = \phi_i({q_i}^{-}, x_{q_i}, y_i, \sigma_i, \xi) \\ z_\sigma = \varphi_i(q_i, x_{q_i}, y_i, \sigma_i) \end{cases} \tag{3}$$

The discrete state update function ϕ_i depends not only on the states, but also on the vehicle output y_i, on local set of events σ_i and on the coordinating variable ξ. The local set of events are produced by guard conditions on discrete state transitions and external events (such as user generated events).

$$M : \left\{ \begin{array}{c} \dot{x}_{q_M} = f_{q_M}\left(x_{q_M}, z_1 ... z_N\right) \\ \xi_c = g_{q_M}(x_{q_M}, \sigma_M, z_1 ... z_N) \\ q_M = \phi_M(q_M^-, x_{q_M}, \sigma_M, z_1 ... z_N) \\ \xi_\sigma = \varphi_M(q_M, x_{q_M}, \sigma_M) \end{array} \right. \quad (4)$$

The global team coordination controller M eq.(4) can also be considered as an hybrid system with both continuous and discrete evolution depending on the performance variables for the N vehicles and generating the coordination variable.

This approach does not impose any particular multi robot coordination topology allowing both centralized approaches (for instance in mixed initiative missions linked human based Command and Control ”C2C” systems) and distributed coordination mechanisms.

From the implementation point of view, the characteristics of the ROS framework [10] allow a large versatility in terms of multi robot cooperation. Without considering the multiple robot scenario ROS limitations referred earlier (currently resolved through the use of additional communication middleware), the software design based on loosely coupled components (such as the ROS computational nodes) allows for information exchange to occur at different vehicle hierarchic levels, and of varying degrees of abstraction (since in the same ROS network, nodes in different robots can publish and subscribe to topics in the same or different robots, whose abstraction level depends on the content and purpose of the publishing nodes).

C. Localization

The primary navigation sensors are GPS and IMU. The vehicle has two GPS receivers being one of them a precision double frequency (RTK capable) receiver. GPS accuracy is augmented with Precise Point Positioning (PPP) when satellite orbit and timing information is available. PPP [18] uses precise satellite orbits and clock data to reduce errors in single GPS receiver. The satellite orbit information (available in the IGS-International GPS Service) can be used in real time when the vehicle operates with internet access (information updated 4 times per day).

Other method for improving the GPS solution precision, is the use of DGPS- RTK (Differential GPS Realtime Kinematic). This method requires the existence of a known base station whose corrections are sent to the vehicle (”rover”). The vehicle is equipped with a dual-frequency (L1/L2) receiver with RTK corrections achieving high positioning accuracy (in order of few cm). GPS logged raw data can also be post-processed with ultra-precise orbit information achieving subcentimeter accuracy for static positioning.

The vehicle localization state (position and attitude) is given (in most situations) by two separate blocks. The position is obtained directly from the GPS since the vehicle dynamics is relatively slow and the GPS receiver provides high precision at 10Hz rate. The attitude is obtained by the IMU mecanization filtering and magnetic compass readings in order to provide an Attitude Heading Reference System (AHRS). This two blocks can when needed (for instance in higher dynamic situations) be combined in a loosely coupled GPS-INS extended kalman filter to obtain full 6DOF vehicle state.

For navigation in areas where GPS is unavailable, a monocular visual odometry SLAM (Simultaneous Localization And Mapping) navigation method is used [19]. In addition, the vehicle can use the navigation sensors described in [19] with a tactical grade INS (iMAR iNAV-FMS-E) replacing the MicroStrain 3DM IMU providing much higher quality IMU data.

V. VISION SYSTEM

The robot vision system is based in a pair of color cameras and an infrared one. The system is used in two types of functions, for situational awareness directly in human supervisioned tasks (such as basic video streaming to a remote operator), or in target detection and scene analysis image processing tasks such as intrusion detection and for navigation purposes.

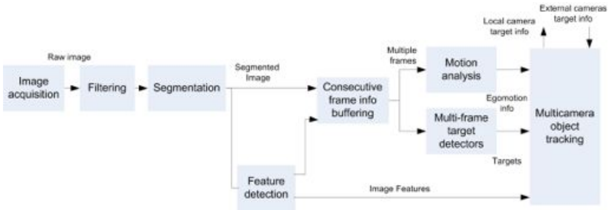

Fig. 5. Vision system

Image acquisition for the color cameras is performed with an external synchronized trigger (in sync with the system clock). This is set at a fixed rate ensuring both the simultaneity of images on the left and right images required for stereo processing and also the frame timestamping needed for other multi-camera processing and for feature based navigation purposes.

The infrared camera 320x240 pixel image is acquired at 30 fps through the USB framegrabber (without external trigger). Currently both color images are captured at maximum resolution (1278x958) and also at 30fps. Each vehicle camera can be used independently or in a multiple camera setup such as conventional stereo arrangement (both color cameras) or in conjunction with the IR one.

Figure 5 presents an overall overview on the vision system architecture. A pipeline structure similar to the architecture presented in [9] is used. For each camera upon acquisition, global filtering can be operated on the image to affect its properties. This type of operation, depending on the particular application can consist in possible multiple options such as sharpening, color adjustments or denoising as examples. In addition for streaming purposes the original raw image can be either logged (this can occur in all pipeline stages) or encoded and transmitted. The possibly processed image is then segmented according to suitable methods (such as color based segmentation, edge detectors or other morphological operators). Region of interest or feature detectors are then applied to the segmented image in other to identify relevant features or targets. Image processing is applied either in a single frame pipeline basis or in a consecutive frame analysis framework. The later type of processing uses multiple consecutive frames to extract information. Examples of this type of processing are visual odometry computations where the motion information is extracted, consecutive frame stereo calculations or multiframe based target detectors. Target 3D positioning can be determined from multi-camera information. In our case, this can be done with images from the stereo color pair or with one from the infrared camera and other from the color (this can be extended to other cameras). Using the standard pinhole camera model, the image points (for an undistorted image) in homogeneous coordinates are given by:

$$\mathbf{x} = \mathbf{K}[\mathbf{R}|\mathbf{t}]\mathbf{X} \tag{5}$$

where $\mathbf{K}$ are the intrinsic parameters, $[\mathbf{R}|\mathbf{t}]$ are the camera extrinsic parameters and $\mathbf{X}$ is the 3D point in the world frame (usually this is the vehicle body fixed frame).

Each camera provides a set of $\mathbf{x_i}$ points. These, when corresponding to the same real world point can be combined in a multi view geometry to provide $\mathbf{X_i}$.

To determine the full 3D position it is necessary to make the correspondence between same target points in the different camera images. In classic dense stereo all the image points are possible candidates and computationally efficient methods like RANSAC [20] are applied to determine point position in the epipolar line. The TIGRE vision system assumes a sparse framework. In this case only the relevant points in the image (for ex: target detected points) are processed (also only for these is the distortion removed). For each corresponding pair of target image points $(\mathbf{x_1}, \mathbf{x_2})$ on different cameras (for the same target) the relative 3D positioning is determined by triangulation or by solving the overdetermined system:

$$\begin{cases} \mathbf{x_1} = \mathbf{K_1}[\mathbf{R_1}|\mathbf{t_1}]\mathbf{X} \\ \mathbf{x_2} = \mathbf{K_2}[\mathbf{R_2}|\mathbf{t_2}]\mathbf{X} \end{cases} \tag{6}$$

The 3D target detection implementation in the ROS framework is depicted in Figure 6. The offline extrinsic camera parameter calibration is also included.

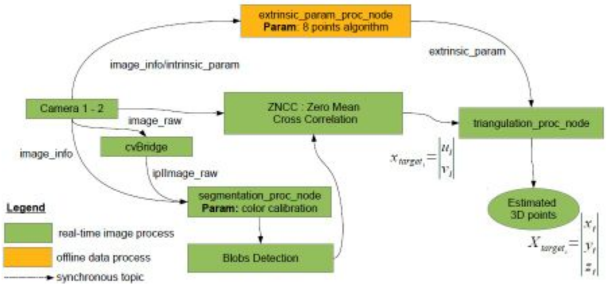

Fig. 6. Vision target detection implementation

VI. RESULTS

Field tests were performed with the robot in the Oporto City Park. These tests were conducted in relatively easy terrain (grass and earth) due to limitations in the allowed area of operation. Human target tracking and following missions were performed using the stereo color camera pair. The target position was logged with a precision GPS receiver (identical to the Septentrio one used on the robot).

In addition, a cooperative mission was performed with an aerial autonomous robot (Ashtec's Pelican UAV) where the UAV detected the target and informed the ground vehicle to follow it (for more information see [21]).

Fig. 7. Two frames from the color cameras with the detected target marked

In Figure 7 two images from the color camera taken during the target tracking maneuver, are presented with the detected target marked. Three images from the IR camera during the same maneuver (but not at the same times) are also shown in figure 8. The human target was wearing a red vest to facilitate detection on the low powered UAV onboard processor. Although this is not a realistic assumption, the detection on the UGV can be performed by more advanced methods and also on the thermographic camera [22], and for the UAV further developments must be pursued in the implementation on the limited resources of realistic vision human target detectors.

Fig. 8. Thermografic camera snapshots

In figure 9 the detected target positions (using the color camera stereo pair), robot trajectory and real target trajectory are indicated for a segment of the tracking maneuver when the robot is approaching the target and stopping afterwards.

VII. CONCLUSIONS

In this work is presented the TIGRE autonomous ground vehicle. The robot was developed for outdoor exploration and to be a versatile robotics research platform. Applications scenarios envision security tasks, precise mapping, cooperative missions with other autonomous robots, and operation in unstructured environments such as underground. The vehicle hardware solutions are described along with the comprehensive set of sensors.

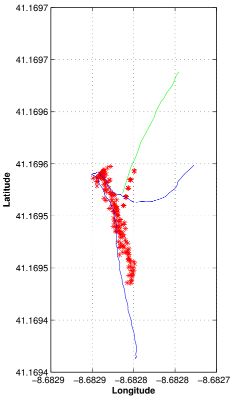

Fig. 9. Target tracked positions (blue-target position, green-robot trajectory, red- vision based target localization

An hybrid systems approach was followed in the vehicle control architecture. Basic motion control is performed by hybrid (discrete and continuous) maneuvers. These are composed in hierarchical finite (for the discrete state) automata providing more complex motion functionalities. The software implementation aspects were also addressed. An overall overview for the vision system is presented along with a particular use for stereo target tracking.

The robot has already performed missions in operational environment both alone and also in cooperation with an unmanned aerial robot. Some demonstrative results from missions performed in outdoor scenario are also presented.

Being the robot for robotics research, an extensive future work is envisioned for further development. Applications of visual odometry and visual features navigation are to be tested in multiple scenarios. The multirobot cooperative framework is to be validated in missions with additional multiple heterogeneous robots, and issues like hybrid systems stability properties in the coodination should be analysed. Integration of new sensors is also to be pursued, namely fast 3D Lidar for obstacle avoidance. These developments are to be considered in a overall goal of achieving long term autonomy in hostile and unstructured outdoor environments. In the immediate future this vehicle will participate in this year Euroathlon trials (an European Union funded initiative for field robotics trials and competitions), were new results under development in navigation and perception are to be validated.

ACKNOWLEDGMENTS

The authors acknowledge the support given by the ISEP-IPP and INESC TEC to this project. This work is financed by the ERDF - European Regional Development Fund through the COMPETE Programme and by National Funds through the FCT - Portuguese Foundation for Science and Technology within project FCOMP-01-0124-FEDER-022701 and under Perception-Driven Coordinated Multi-Robot Motion Control Project with reference PTDC/EEA-CRO/100692/2008.

REFERENCES

[1] K. Berns, K.-D. Kuhnert, and C. Armbrust, “Off-road RoboticsâAn Overview,” *KI - Künstliche Intelligenz*, vol. 25, no. 2, pp. 109–116, Mar. 2011. [Online]. Available: http://www.springerlink.com/index/10.1007/s13218-011-0100-4

[2] S. Ioannou, K. Dalamagkidis, K. P. Valavanis, E. K. Stefanakos, and P. H. Wiley, “On Improving Endurance of Unmanned Ground Vehicles: The ATRV-Jr Case Study,” in *2006 14th Mediterranean Conference on Control and Automation*. IEEE, Jun. 2006, pp. 1–6.

[3] C. Armbrust, M. Proetzsch, and K. Berns, “Behaviour-Based Off-Road Robot Navigation,” *KI - Künstliche Intelligenz*, vol. 25, no. 2, pp. 155–160, Feb. 2011. [Online]. Available: http://www.springerlink.com/index/10.1007/s13218-011-0090-2

[4] L. Bascetta, D. Cucci, G. Magnani, and M. Matteucci, “Towards the implementation of a MPC-based planner on an autonomous All-Terrain Vehicle,” in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2012 -Workshop on Robot Motion Planning: Online, Reactive, and in Real-time*, Vilamoura, Portugal, Oct. 2012.

[5] M. Saptharishi, C. Spence Oliver, C. Diehl, K. Bhat, J. Dolan, A. Trebi-Ollennu, and P. Khosla, “Distributed surveillance and reconnaissance using multiple autonomous ATVs: CyberScout,” *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 826–836, Oct. 2002. [Online]. Available: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=\&arnumber=1068001\&contentType=Journals+\&+Magazines\&pageNumber=3\&searchField=Search_All\&queryText=all+terrain+vehicle

[6] R. R. Murphy, “Findings from NSF-JST-NIST Workshop on Rescue Robotics,” in *2010 IEEE Safety Security and Rescue Robotics*. IEEE, Jul. 2010, pp. 1–4. [Online]. Available: http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=5981570

[7] L. Kuhnert, M. Ax, M. Langer, D. N. Van, and K.-d. Kuhnert, “Absolute high-precision localisation of an unmanned ground vehicle by using real-time aerial video imagery for geo-referenced orthophoto registration,” *Autonome Mobile Systeme 2009, Infromatik Aktuell*, pp. 145–152, 2009.

[8] F. E. Schneider, D. Wildermuth, and I. Processing, “European Land Robot Trial (ELROB) Towards a Realistic Benchmark for Outdoor Robotics,” in *Proceedings of the 1st international conference on Robotics in Education, RiE2010*. FEI STU, Slovakia, 2007, pp. 65–70.

[9] H. Silva, J. M. Almeida, L. Lima, A. Martins, and E. Silva, “A Real Time Vision System for Autonomous Systems: Characterization during a Middle Size Match,” in *RoboCup 2007: Robot Soccer World Cup XI*, ser. Lecture Notes in Computer Science, U. Visser, F. Ribeiro, T. Ohashi, and F. Dellaert, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, vol. 5001, pp. 504–511. [Online]. Available: http://www.springerlink.com/index/10.1007/978-3-540-68847-1

[10] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, “ROS: an open-source Robot Operating System,” in *ICRA Workshop on Open Source Software*, 2009.

[11] A. Huang, E. Olson, and D. Moore, “Lightweight Communications and Marshalling for Low-Latency Interprocess Communication,” MIT, Tech. Rep., 2009.

[12] J. Ding, J. Gillula, H. Huang, M. Vitus, W. Zhang, and C. Tomlin, “Hybrid Systems in Robotics,” *IEEE Robotics & Automation Magazine*, vol. 18, no. 3, pp. 33–43, Sep. 2011. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6016585

[13] R. Alur, C. Courcoubetis, T. A. Henzinger, and P.-H. Ho, “Hybrid Automata: An Algorithmic Approach to the Specification and Verification of Hybrid Systems,” in *Hybrid Systems*, R. L. Grossman, A. Nerode, A. P. Ravn, and H. Rishel, Eds. Springer-Verlag, 1993, vol. 736, pp. 209–229.

[14] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L.-e. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. V. Niekerk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian, and P. Mahoney, “Stanley : The Robot that Won the DARPA Grand Challenge,” *Journal of Field Robotics*, vol. 23, no. April, pp. 661–692, 2006.

[15] M. Richier, R. Lenain, B. Thuilot, and C. Debain, “On-line estimation of a stability metric including grip conditions and slope: Application to rollover prevention for All-Terrain Vehicles,” in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, Sep. 2011, pp. 4569–4574.

[Online]. Available: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=\&arnumber=6094450\&contentType=Conference+Publications\&searchField=Search_All\&queryText=all+terrain+vehicle

[16] R. Beard, J. Lawton, and F. Hadaegh, “A coordination architecture for spacecraft formation control,” *IEEE Transactions on Control Systems Technology*, vol. 9, no. 6, pp. 777–790, 2001. [Online]. Available: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=960341

[17] H. Li, F. Karray, and O. Basir, “A Framework for Coordinated Control of Multiagent Systems and Its Applications,” *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 38, no. 3, pp. 534–548, May 2008. [Online]. Available: http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4490041

[18] K. Chen and Y. Gao, “Real-Time Precise Point Positioning Using Single Frequency Data,” in *Proceedings of the ION GNSS 18th International Technical Meeting of the Satellite Division*, Long Beach, USA, 2005, pp. 13–16.

[19] A. Ferreira, J. M. Almeida, and E. Silva, “Application of Visual-Inertial SLAM for 3D Mapping of Underground Environments,” in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems - Workshop on Planning, Perception and Navigation for Intelligent Vehicles*, Vilamoura, Portugal, 2012.

[20] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge: Cambridge University Press, 2003.

[21] “PCMMC Multi-Robot Coordinated Outdoor Experiments video.” [Online]. Available: http://www.youtube.com/watch?v=vMcDQ6S5Qxc

[22] J. Han and B. Bhanu, “Fusion of color and infrared video for moving human detection,” *Pattern Recognition*, vol. 40, no. 6, pp. 1771–1784, Jun. 2007.