ELROB 2018

24 - 28 September 2018

Mons, Belgium

ELROB 2018 – Technical Paper Preparation Guideline

Submission of a Technical Paper is a mandatory component of the ELROB qualification process, as described in section 3.3 of the official ELROB rules. Each team wishing to participate is required to submit an acceptable Technical Paper in order to remain eligible for ELROB. A scientific advisory board will evaluate all submitted Technical Papers. Teams should adhere to the following guidelines to ensure the acceptability of their submitted Technical Paper.

Format Guidelines

The Technical Paper should be formatted using the layouts and conventions common to published technical papers for a professional, polished appearance. Outlines, checklists, newspaper reports, or magazine articles are not acceptable. Margins of at least 2,5cm should be used (left, right, top, and bottom) on standard ISO A4 size paper. Body text should be 12-point font with 1 line spacing. Times Roman or any other easy-to-read font type may be used for body text, and Arial or other sans serif font for titling. Figures and tables should be incorporated in the text.

The title page must include team and scenario name, and “ELROB” should appear on the title page. Preferably, these information should be repeated in a header or footer and all following pages. Pages must be numbered consecutively. Author names may include title and organizational affiliation, as required. The team’s contact information including email addresses should be included.

Content Guidelines

A short abstract should contain the key system characteristics and novel approaches described in the Technical Paper. A general introductory section may be used to describe the team, its experiences with robot competitions and other important background information. The main part of the document must describe how the team will tackle the challenges of the selected scenarios and should address major challenges common to the ELROB scenarios. As a general guideline, refer to the following section which briefly lists important issues and questions to be considered within the Technical Paper. It is, however, not strictly necessary to answer all those questions in detail. Rather, the list should be taken as a general outline, which topics might be worth addressing in the Technical Paper.

Photographs, tables and diagrams are encouraged. Although not strictly regulated, the overall length of the paper, including title, abstract, captions and figures, should lie between three and four pages.

Return Technical Paper to: elrob@fkie.fraunhofer.de

ELROB 2018

24 - 28 September 2018

Mons, Belgium

ELROB 2018 – Common Scenario Problems and Considerations

Vehicle

Prototype or commercially available platform? Major characteristics? Key capabilities? Operating time and distance?
Why is the vehicle suitable for outdoor/off-road/rough terrain scenarios? What about rain and humidity?
What kind of outdoor/indoor obstacles can the vehicle overcome? How? What about steepness, high inclinations, stairs?
Which standards (DIN, ISO, IEEE etc.) are used?

Processing

What kind of computing systems are used on the vehicle and/or the control station?
Hardware and software complexity issues? Reliability? Common software frameworks or all proprietary? Any special approaches in the realization of the system?
How is the transfer of the scenario input data realized?
Which standards (DIN, ISO, IEEE etc.) are used?

Communication

How is the communication between vehicle and control station realized? Prototype or commercial system? WLAN? UMTS?
What is the expected communication range? What is the influence of weather, especially rain and humidity, and environment, e.g. buildings, hills, dense forest?
How does the system react to temporary communication failures and interruptions? Any novel methods to increase robustness? What bandwidth / which communication means are required to keep the system running?
How is the trial result data transmitted to the control station?
Which standards (DIN, ISO, IEEE etc.) are used?

Localization

What is the indoor/outdoor localization system? GPS, inertial navigation, sensor-based localization or any combination thereof?
If using GPS, how does the system cope with typical problems, e.g. disturbance in/near buildings, tree foliage, temporary blockings?
If external map data is an integral part of the vehicle's navigation system, how is it used? Where does it come from? Online or offline?

Return Technical Paper to: elrob@fkie.fraunhofer.de

ELROB 2018

24 - 28 September 2018

Mons, Belgium

Sensing

What is the location and mounting of the sensors? Sensor type, range and field of view? What are the sensor characteristics and why is the choice of sensors suitable?
Is there any special compensation of outdoor-specific conditions like vibration, bright sun, rapidly changing light, rain, dust or mud?
Sensor fusion strategy? Any means employed to build models of the external environment?
Is there an internal sensing system to sense the vehicle state? Odometry? If odometry, how does it cope with real-world challenges like debris, slippery surfaces, vibrations?
Camera system? Used only by a human operator or is a vision component part of the overall sensing system?

Vehicle Control

How does the vehicle cope with non-standard manoeuvres like abrupt braking, starting on a steep hill or making a sharp?
To what extent is the system working autonomously? Or is it purely tele-operated? If applicable, how are complex operations implemented, e.g. waypoint following, off-road path finding or obstacle avoidance?
How is the vehicle controlled when not in autonomous operation? Is there any support offered by the control station? Semi-autonomous or assistance functions?
Is there any automatic recovery behaviour in case of a communication loss?

System Readiness

How mature is the system? Is it fully functional? Which components are still under development?
How would you rate the Technology Readiness Level (TRL) of hardware and software? *See the appendix for a definition of TRLs.*
Has the team conducted any field tests to ensure readiness for ELROB? If yes, in which environment? What were the results?
Regardless of scenario-specific challenges, what are problems and open issues to be solved before the contest?

ELROB 2018

24 - 28 September 2018

Mons, Belgium

Appendix – Definition of Technology Readiness Levels (TRL) for Hardware/Software

Technology Readiness Level for Hardware	Description
1. Basic principles observed and reported in context of an application	Lowest level of technology readiness. Scientific research begins to be evaluated and translated into applied research and development. Examples might include paper studies of a technology's basic properties.
2. Technology concept and/or application formulated	Invention begins. Once basic principles are observed, practical applications can be postulated. Applications are speculative and there may be no proof or detailed analysis to support the assumptions. Examples are limited to analytic paper studies.
3. Analytical and experimental critical function and/or characteristic proof of concept	Active research and development is initiated. This includes analytical and laboratory studies to physically validate the analytical predictions of separate elements of the technology. Examples include components that are not yet integrated or representative.
4. Technology component and/or breadboard[a] (system / sub-system representation) validation in laboratory environment	Basic technological components are integrated to establish that the pieces will work together. This is relatively low fidelity[b] compared with the eventual system. Examples include integration of "ad hoc" hardware in a laboratory.
5. Technology component and/or breadboard[a] (system / sub-system representation) validation in relevant environment[c]	Fidelity of breadboard technology increases significantly. The basic technological components are integrated with reasonably realistic supporting elements so the technology can be tested in a simulated environment. Examples include high-fidelity[f] laboratory integration of components.
6. Technology system/subsystem model[d] or prototype[e] demonstration in a relevant environment[c]	Representative model or prototype system, which is well beyond the breadboard[a] (representation) tested for TRL 5, is tested in a relevant environment. Represents a major step up in a technology’s demonstrated readiness. Examples include testing a prototype in a high fidelity[f] laboratory environment or in a simulated operational environment[g].
7. Technology system prototype[e] demonstration in an operational environment[h]	Prototype near, or at, planned operational system. Represents a major step up from TRL 6, requiring the demonstration of an actual system prototype in an operational environment (e.g., in an aircraft, in a vehicle, or in space). Information to allow supportability assessments is obtained. Examples include testing the prototype in a test bed aircraft.
8. Actual technology system completed and qualified through test and demonstration	Technology has been proven to work in its final form and under expected conditions. In almost all cases, this TRL represents the end of true system development and demonstration. Examples include developmental test and evaluation of the system in its intended weapon system to determine if it meets design specifications, including those relating to supportability.
9. Actual technology system “mission proven” / “qualified” through successful mission operations	Actual application of the technology in its final form and under mission conditions, such as those encountered in operational test, evaluation, and reliability trials. In almost all cases, this is the end of the last “bug fixing” aspects of true system development. Examples include using the system under operational mission conditions.

Return Technical Paper to: elrob@fkie.fraunhofer.de

ELROB 2018

24 - 28 September 2018

Mons, Belgium

Technology Readiness Level for Software	Description
1. Basic principles observed and reported	Lowest level of software technology readiness. A new software domain is being investigated by the basic research community. This level extends to the development of basic use, basic properties of software architecture, mathematical formulations, and general algorithms.
2. Technology concept and/or application formulated	Once basic principles are observed, practical applications can be invented. Applications are speculative and there may be no proof or detailed analysis to support the assumptions. Examples are limited to analytic studies using synthetic data.
3. Analytical and experimental critical function and/or characteristic proof of concept	Active research and development is initiated. The level at which scientific feasibility is demonstrated through analytical and laboratory studies. This level extends to the development of limited functionality environments to validate critical properties and analytical predictions using non-integrated software components and partially representative data.
4. Module and/or subsystem validation in laboratory environment (i.e. software prototype[e] development environment)	Basic software components are integrated to establish that they will work together. They are relatively primitive with regard to efficiency and robustness compared with the eventual system. Architecture development initiated to include interoperability, reliability, maintainability, extensibility, scalability, and security issues. Emulation with current/legacy elements as appropriate. Prototypes developed to demonstrate different aspects of eventual system.
5. Module and/or subsystem validation in relevant environment[c]	Level at which software technology is ready to start integration with existing systems. The prototype[e] implementations conform to target environment/interfaces. Experiments with realistic problems. Simulated interfaces to existing systems. System software architecture established. Algorithms run on a processor(s) with characteristics expected in the operational environment[h].
6. Module and/or subsystem validation in a relevant end-to-end environment	Level at which the engineering feasibility of a software technology is demonstrated. This level extends to laboratory prototype[e] implementations on full-scale realistic problems in which the software technology is partially integrated with existing hardware/software systems.
7. System prototype[e] demonstration in an operational[h] high-fidelity[f] environment	Level at which the program feasibility of a software technology is demonstrated. This level extends to operational environment prototype implementations where critical technical risk functionality is available for demonstration and a test in which the software technology is well integrated with operational hardware/software systems.
8. Actual system completed and mission qualified through test and demonstration in an operational environment[h].	Level at which a software technology is fully integrated with operational hardware and software systems. Software development documentation is complete. All functionality tested in simulated and operational scenarios.
9. Actual system proven through successful mission- proven operational capabilities	Level at which a software technology is readily repeatable and reusable. The software based on the technology is fully integrated with operational hardware/software systems. All software documentation verified. Successful operational experience. Sustaining software engineering support in place. Actual system.

Return Technical Paper to: elrob@fkie.fraunhofer.de

ELROB 2018

24 - 28 September 2018

Mons, Belgium

Supplementary Definitions:

[a] *Breadboard*: Integrated components that provide a representation of a system/subsystem and that can be used to determine concept feasibility and to develop technical data. Typically configured for laboratory use to demonstrate the technical principles of immediate interest. May resemble final system/subsystem in function only.

[b] *Low fidelity*: A representative of the component or system that has limited ability to provide anything but first order information about the end product. Low-fidelity assessments are used to provide trend analysis.

[c] *Relevant environment*: Testing environment that simulates the key aspects of the operational environment.

[d] *Model*: A functional form of a system generally reduced in scale, near or at operational specification. Models will be sufficiently hardened to allow demonstration of the technical and operational capabilities required of the final system.

[e] *Prototype*: A physical or virtual model used to evaluate the technical or manufacturing feasibility or military utility of a particular technology or process, concept, end item, or system.

[f] *High fidelity*: Addresses form, fit, and function. High-fidelity laboratory environment would involve testing with equipment that can simulate and validate all system specifications within a laboratory setting.

[g] *Simulated operational environment*: Either (a) a real environment that can simulate all of the operational requirements and specifications required of the final system, or (b) a simulated environment that allows for testing of a virtual prototype. Used in either case to determine whether a developmental system meets the operational requirements and specifications of the final system.

[h] *Operational environment*: Environment that addresses all of the operational requirements and specifications required of the final system, including platform/packaging.

Return Technical Paper to: elrob@fkie.fraunhofer.de