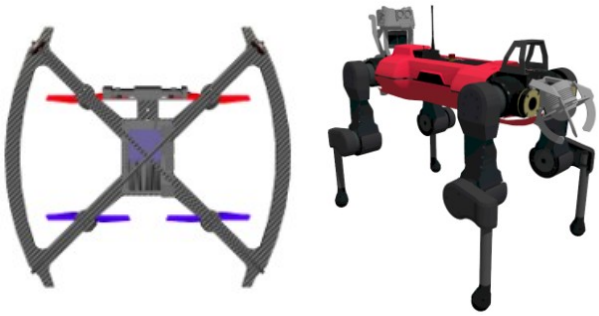
Real-Time Strategy game-like Control of Very Large Robotic Teams



Overview: This thesis deals with the goal of developing the software framework necessary that would allow the commanding of very large fleets [100+] of robotic systems in a manner analogous to that experienced when playing real-time strategy games (RTS). In such video games, the user has access to immediate control of diverse "units" which can be commanded to seamlessly execute a collection of missions including exploration, surveillance and patrolling. The effective result is that of a "mosaic of agents" within which every unit contributes to the final goal but none is absolutely necessary. Thus resilience in the ability to execute the the user's command is achieved by the aggregation of multiple units and the simplicity to operate them. Motivated by this we seek to develop on top of successful simulators in robotics in order to deliver analogous functionality. As modern robotic simulators provide a level of abstraction that allows code developed in them to be API-wise compatible with real robotic systems, this work is a significant prior step towards seamless control of very large fleets of real-life robotic systems in the context of challenging applications. In terms of types of robots, ground and aerial robots such as the legged system and the collision-tolerant Micro Aerial Vehicle presented above are of immediate interest.

Tasks and Sub-objectives

- Select the appropriate simulation environment for robotics.
- Identify the key algorithms that enable the experience provided in RTS games.
- Develop the relevant functionality interfacing the robotics simulator and considering ground and flying robots.
- Develop the necessary UI to deliver an RTS-like experience.
- Evaluate in simulation (i.e., play your RTS-like game moving robots with real dynamics, sensors and mapping capabilities).

Starting Literature

[1] The Gazebo simulator: http://gazebosim.org/

[2] NVIDIA Isaac Simulator: https://developer.nvidia.com/isaac-sim

[3] Robertson, G. and Watson, I., 2014. A review of real-time strategy game AI. Ai Magazine, 35(4), pp.75-104.

[4] Ontanón, S., Synnaeve, G., Uriarte, A., Richoux, F., Churchill, D. and Preuss, M., 2019. RTS AI Problems and Techniques.

[5] Vinyals, O., Babuschkin, I., Chung, J., Mathieu, M., Jaderberg, M., Czarnecki, W.M., Dudzik, A., Huang, A., Georgiev, P., Powell, R. and Ewalds, T., 2019. Alphastar: Mastering the real-time strategy game starcraft ii. DeepMind blog, 2.

[6] Novak, D., Čep, A. and Verber, D., 2018. Classification of modern real-time strategy game worlds. GSTF Journal on Computing, 6(1).

[7] Overmars, M.H., 2005. Path planning for games. In Proc. 3rd Int. Game Design and Technology Workshop (pp. 29-33).

[8] LaValle, S.M., 2006. Planning algorithms. Cambridge university press.

[9] Yannakakis, G.N. and Togelius, J., 2018. Artificial intelligence and games (Vol. 2). New York: Springer.

[10] Game AI Pro: http://www.gameaipro.com/