4.1 Capabilities of autonomous systems

The four sliders of an autonomous system are the planning slider, the action slider, the model slider and the knowledge slider.

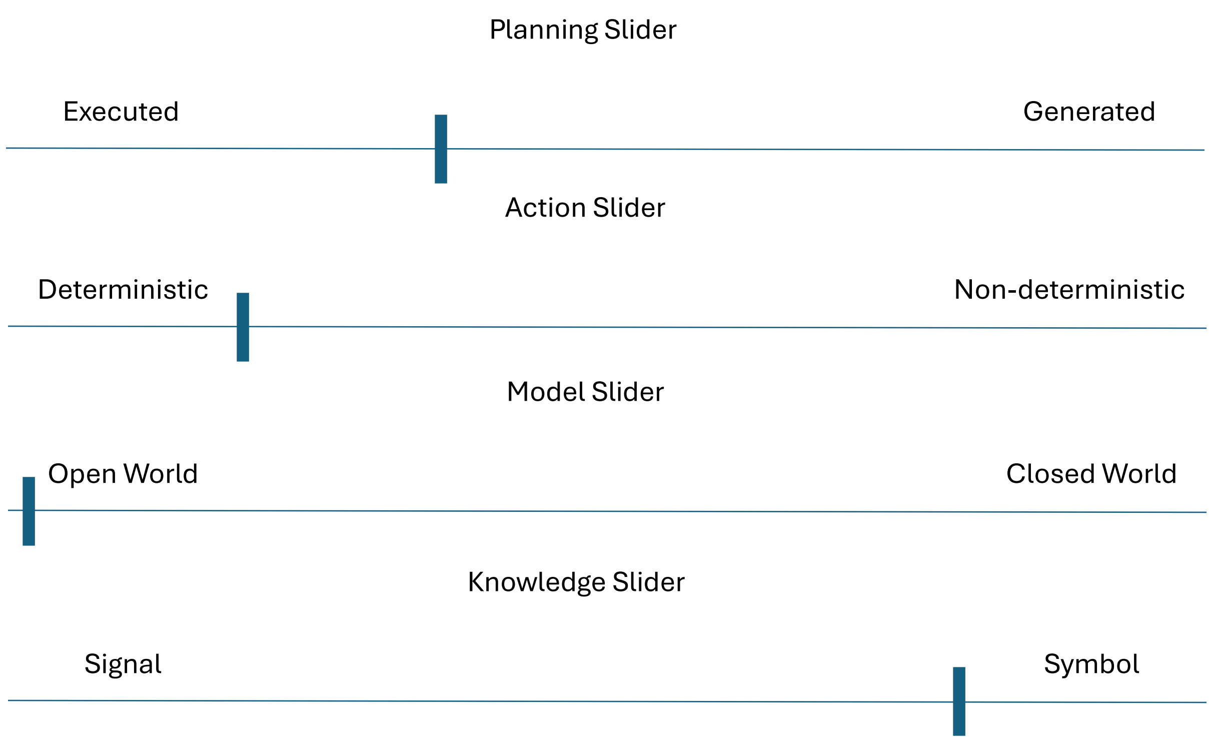
The first one, the planning slider, slides from executed to generated plans and represents how the system decides what its goals are. When designing an autonomous system, the level of preplanned or on-the-fly plans used must be chosen as the preplanned ones must be hardcoded into the system and the generator for on-the-fly planning must be trained.

The second slider, the action slider, represents a choice between deterministic actions and non-deterministic actions. By this it means the level of specific actions taken given a set of input. This influences the complexity of the actions chosen, but also the ways the system can be tested. It could also influence the redundancy of the system's functions if the system has more ways than one of acting.

The third slider is the model slider, and it deals with the mapping of the environment of the system. On one side, you can have an open world solution where nothing is known of the environment beforehand. Then the largest problem would be to map the closest part of the environment to react to this and to find the general direction to move in. If a closed world is used, then the overall map is known, and the problem is to figure out the parts of the map which can move.

Finally, the knowledge slider deals with how knowledge or goals are given to the system. Either the system can receive info concretely, by being told where to go and what to do, or it can be given something which should be done and figure the rest out itself.

For the Autonomous Extraction and Refueling Station, the sliders would look something like this:



4.2 Robotics Framework

For a robotics framework to include tools for intelligent, AI based, autonomous robots it needs to have functionalities for implementing at least the three most important primitives described in chapter 4.3.1 in the AI Robotics book. These primitives are Sense, Plan and Act. However, to reach high levels of intelligence, as described by higher levels of autonomy in 4.4, tools for the Learning primitive is also important. As an example, we can look at two popular frameworks.

ROS contains options for specifically communicating with sensors connected to a robot and a seamless interfacing of tools for processing sensor data. There are advanced tools for mapping, path planning, pose estimation and diagnostics, among other things. Finally, there are multiple robot specific tools for movement and easy integration with other programming languages capable of filling potential gaps in tools. This displays tools and options for all the four primitives and allows for high intelligence development. The only downside is some lack of tools for creation of general machine learning algorithms for decision making, but it allows for this through easy integration with programming languages.

HOP provides options for internal communication and connection between sensors, actuators and higher-level programming languages. The higher-level programming language specifically intended is JavaScript, which allows for sensor processing and machine learning tools to be used. Thus all tools for filling the four primitives are there. However, locking the use to mostly JavaScript could create problems if processing large amounts of data are needed, as JavaScript tends to be slow compared to a lower-level language.

It is clear from these examples that these robotics frameworks allow for development of intelligent systems, but there are a wide range of frameworks, and not all of them provide tools for everything. These can still be used to create low intelligence or even just automatic systems. The frameworks were only tools for sensor interactions and controlling actuators are a good example of this. Without, or with limited tools for developing planning and learning algorithms, intelligence is not possible, but it is also not always needed.

4.3 Industrial Applications

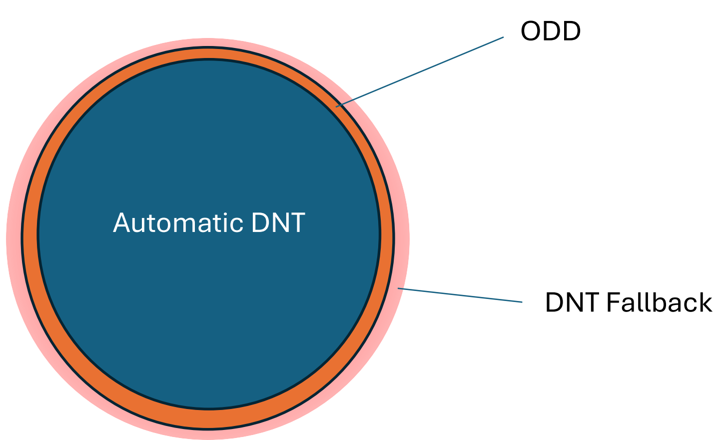
My industry is production of circuit boards and whole electrical products at a medium to high scale. Our biggest problem when it comes to daily work is the uncertainty of how long we will produce for any given customer and when they will order batches of products and when they don’t need any. This has been a large problem in the past as we don’t want a large turnover of workers, and it takes time to hire and train new ones when we suddenly need some. However, autonomous systems capable of adjusting to multiple different products and tasks, could help with this. As of now we have tried to go for simple automatic solutions, either to automate dangerous tasks or remove bottlenecks, but we have found that this only works for large and stable customers. There is also the problem that automatic solutions often take up a lot of space, which takes time to change the use of if they are no longer in use.

With even just low intelligence autonomous solutions, using some system for free flow of products between stations, a multi-product assembly line could be created. This would take up a lot less space and allow for the same tools and space to be used for multiple products. Likely, there should be interaction with humans, as some tasks might be too specific for robots, but the autonomous system can simply treat these as any other stations to.

The problem we are facing with this right now is the large initial investment cost for the factory, but even just a very small and simple solution could be built on and expanded. For future projects, the flow of products between locations could also be baked into the system, and one large autonomous production plant could be created. With multiple locations doing the same tasks, it could even adjust to downtime at some places and shift the workload elsewhere. Adding more intelligence, it might even predict maintenance and downs time or even when to produce or stop production for different customers.

4.4 Operational Boundaries

The ODD defines equipment, operations and environment of the system. For the Autonomous Extraction and Refueling Station, this is an important thing to define. The system will need to perform around the clock in space, but only for at most a week at a time. In this environment, with radiation from the sun and possible objects or debris occasionally on collision course, the system needs to navigate safely between Starships, extract from and refuel tanks in these ships and charge its batteries. To perform these actions, vision sensors, for detecting objects and identifying those of interest must be present. Any dangerous objects must be identified at a distance where evasive maneuvers are still possible. The system also needs a set of propulsion engines for quick movement in all three directions in space. While they need to be quick, they also need to be precise enough for the docking sequence to be possible. Tools for docking, extraction of fuel, detection of full and empty tanks and giving fuel to a rocket are also needed. Finally, equipment for recharging internal batteries and the batteries for storing electricity must be in the system. This equipment needs to function in a low gravity and vacuum environment, while still working quickly and safely. Finally, it needs to learn from experience how to best plan and create its movement paths, as moving in orbit can be both intuitive and complex, to increase the cost efficiency of its operation and allow less use of its precious fuel. As no humans are wanted in direct interaction with the station during operation, all these tasks and problems must be solved automatically. But maintenance, either routine or acute, will be beyond the scope of its autonomous behavior and must be done by astronauts during downtime.



As there are no operators in this system, there is no operator exclusive DNT, and the ODD will only contain automatic DNT. There is still a need for a DNT Fallback in case of dangerous objects or problems with battery or other equipment.

4.5 Overall Capabilities

The Autonomous Extraction and Refueling Station will need sensors for creating a 3D map of the environment. This would ideally be one or multiple optical cameras which can identify objects of interest. These need to take pictures in multiple directions and communicate with some internal data processing software. The system also needs sensors for detecting confirmed docking. This could be a combination of proximity sensors and torque sensors on the station hull and inside the machinery performing the docking. Sensors for detecting fuel level or at least if the tank is full or empty are also required. Sensors detecting possible malfunction in actuators, or to measure the battery level is also required to plan accordingly.

As for actuators, the station needs some propulsion engine, running on the same fuel as it is transporting. The easiest solution would be to use at least three raptor engines, the same ones in use for Starship. Furthermore, actuators for docking with a starship, and then to push or pull fuel to or from the tank is needed. The docking mechanism must hold on to the rocket and seal off any holes to outer space, even while under stress from fuel being pushed or pulled through. The pulling or pushing mechanism would be a mechanical pump either creating vacuum in the station's tank and sucking the fuel in or pushing the fuel out. Finally, depending on how the system should acquire electric power, actuators for folding or deploying solar panels should be present.

Regarding the human interactions of the system, in general it is not wished for humans to interact with the system during operation. However, maintenance must still be performed by humans, and for this there should be some way to anchor oneself to the station for astronauts. There should also be some sort of physical way of accessing the station's machinery and possibly also a socket for accessing the software, to allow updates to be performed and machinery to be maintained.

The robotics framework is required to allow for communication between sensors, actuators and internal software as well as development and running of intelligent autonomous planning and decision-making software. It also needs to allow the storage, updating of and use of sets of data such as maps, movement history, maintenance times, statuses of interesting objects and possible laws and regulations applicable. This framework should also allow for encryption and secure storage of all internal communication and software, as to remove the risk of cyber-attacks from hostile actors. A solution is to use the version of ROS described by B. Dieber et al. in Security for the Robot Operating System, found here:
https://www.sciencedirect.com/science/article/abs/pii/S0921889017302762

All these sensors, actuators, tools and software must be built to withstand the radiation of space, vacuum, low to no gravity and at least microscopic to small meteors hitting the station. Thus, the equipment must be built with extra care to durability and if liquid or gaseous components are included, they need to consider extra sealing measures. For actuators specifically, any mechanical parts prone to erosion due to moveable parts should be avoided, as to keep maintenance cycles as large as possible.