An Intelligent Tutor for Tele-robotics Training

Leo Hartman[1]
Canadian Space Agency, St-Hubert, Québec , J3Y 8Y9

Froduald Kabanza[2] , Khaled Belghith[3]
University of Sherbrooke, Sherbrooke, Québec J1K 2R1

RomanTutor is a tutoring system that uses sophisticated domain knowledge to monitor the progress of students and advise them while they are learning how to operate a space tele-robotic system. It is intended to help train operators of the Space Station Remote Manipulator System (SSRMS) including astronauts, operators involved in ground-based control of SSRMS and technical support staff. Currently there is only a single training facility for SSRMS operations and it is heavily scheduled. The training staff time is in heavy demand for teaching students, developing teaching material and new teaching tools. For example, all SSRMS simulation exercises are developed by hand and this process requires a lot of staff time. Once in orbit ISS astronauts currently have only simple web-based material for skill development and maintenance. For long duration space flight astronauts will require sophisticated simulation tools to maintain skills. RomanTutor addresses these challenges by providing a sophisticated portable training tool. It incorporates a model of the system operations curriculum, a kinematic simulation of the robotics equipment and the ISS and a high performance path planner. For each element of the curriculum that the student is supposed to master, RomanTutor is able to generate example tasks for the student to accomplish within the simulation environment.

I. Introduction

Roman Tutor[1] is a simulation-based tutoring system to support astronauts in learning how to operate the Space Station Remote Manipulator (SSRMS), an articulated robot arm mounted on the international space station (ISS). Figure 1 includes a snapshot of the SSRMS on ISS. Astronauts operate the SSRMS through a workstation located inside one of the ISS compartments. Figure 1 also shows the workstation which has an interface with three monitors, each connected to a camera placed at a strategic location of the ISS. There are a total of 14 cameras on the ISS, but only three of them are seen at a time through the workstation. *Roman Tutor*'s user interface (Figure 2) resembles that of the robotic workstation.

The SSRMS is a key component of the ISS and is used in the assembly, maintenance and repair of the station, and also for moving payloads from visiting shuttles. Operators manipulating the SSRMS on orbit receive support from ground operations. Part of this support consist in visualizing and validating maneuvers before they are actually carried out. Operators have in principle rehearsed the maneuvers many times on the ground prior to the mission, but unexpected changes are frequent during the mission. In such cases, ground operators may have to generate 3D animations for the new maneuvers and upload them to the operator on the station. So far, the generation of these 3D animations are done manually by computer graphic programmers and thus are very time consuming.

RomanTutor integrates robot path-planning and spatial task modeling into an MSS simulator to provide useful feedback to a student operating the SSRMS. To illustrate, when a student is learning to move a payload, RomanTutor invokes a path-planner periodically to check whether there is a path from the current configuration to the target, and provides feedback accordingly. The path planner not only computes collision free paths but is also capable of taking into account the limited direct view of the ISS, the lighting conditions and other safety constraints about operating the SSRMS.

In order to improve the ground support operations on the SSRMS, we have developed as part of RomanTutor an automatic task demonstration generator (ATDG)[2], which generates 3D animations that demonstrate how to perform

[1] Research Scientist, Space Technologies, 6767 Airport Rd, St-Hubert, Quebec J3Y 8Y9.
[2] Professor, Département d'informatique.
[3] Ph.D. Candidate, Département d'informatique.

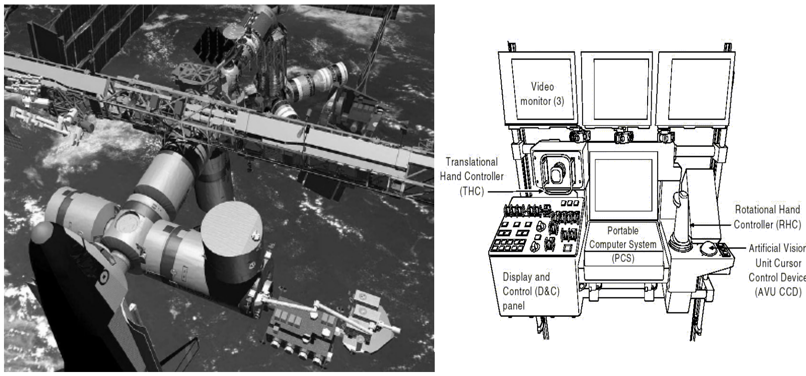

Figure 1. ISS with SSRMS (left) and the Robotic Workstation (right).

a given task with the SSRMS. The ATDG is integrated with a probabilistic roadmap path-planner which is used to compute a collision-free trajectory of the SSRMS from one given configuration to another. The path-planner implements the FADPRM path-planning approach[3] that allows taking into account collisions and visibility constraints. Collisions are treated as hard constraints on trajectories that must be avoided at all cost, whereas visibility constraints are handled as preferences among desirable trajectories. This allows the generation of collision-free trajectories making the robot go into regions visible through cameras and in which the manipulation is safer and easier.

Roman Tutor is still under development. Below the flexible anytime dynamic path planner FADPRM[3] and the automatic task demonstration generator ATDG[2] are described in more detail.

II. FADPRM Path-Planner

In the literature, several approaches dealing with the path-planning problem for robots in constrained environments were found[4-6]. Several implementations were carried out on the basis of these various approaches and much of them are relatively effective and precise. However, the fact is that none of these techniques deals with the problem of restricted sight we are dealing with in our case. That is why we designed and implemented FADPRM[3], a flexible and efficient approach for robot path planning in constrained environments. In more of the obstacles the robot must avoid, our approach holds account of desired and non-desired (or dangerous) zones. This will make it possible to take into account the disposition of cameras on the station. Thus, our planner will try to bring the robot in zones offering the best possible visibility of the progression while trying to avoid zones with reduced visibility.

FADPRM allows us to put in the environment different zones with arbitrary geometrical forms. A degree of desirability *dd*, a real in [0 1] is assigned to each zone. The *dd* of a desired zone is then near 1, and the more it approaches 1, the more the zone is desired; the same for a non-desired zone where the *dd* is in [0 0.5]. On the international Space Station, the number, the form and the placement of zones reflect the disposition of cameras on the station. A zone covering the field of vision of a camera will be assigned a high *dd* (near 1) and will take a shape which resembles that of a cone; whereas a zone that is not visible by any camera from those present on the station will be considered as an non-desired zone with a *dd* near to 0 and will take an arbitrary polygonal shape.

The ISS environment is then preprocessed into a roadmap of collision-free robot motions in regions with highest desirability degree. More precisely, the roadmap is a graph such that every node *n* is labeled with its corresponding robot configuration *n.q* and its degree of desirability *n.dd*, which is the average of *dd*s of zones overlapping with *n.q*. An edge *(n,n')* connecting two nodes is also assigned a *dd* equal to the average of *dd* of configurations in the path-segment *(n.q,n'.q)*. The *dd* of a path (i.e., a sequence of nodes) is an average of *dd* of its edges.

Following probabilistic roadmap methods (PRM)[7], we build the roadmap by picking robot configurations probabilistically, with a probability that is biased by the density of obstacles. A path is then a sequence of collision free edges in the roadmap, connecting the initial and goal configurations.

Following the Anytime Dynamic A* (AD*) approach[8], to get new paths when the conditions defining safe zones have dynamically changed, we can quickly re-plan by exploiting the previous roadmap. On the other hand, paths are computed through incremental improvements so that the planner can be called at anytime to provide a collision-free path and the more time it is given, the better the path optimizes moves through desirable zones. Therefore, our planner is a combination of the traditional PRM approach[7] and AD*[8] and it is flexible in that it takes into account zones with degrees of desirability. This explains why we called it Flexible Anytime Dynamic PRM (FADPRM).

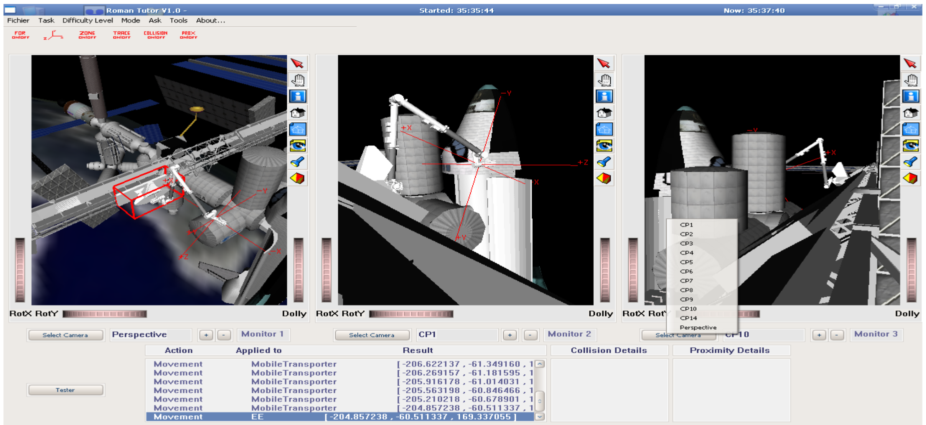

Figure 2. RomanTutor user interface.

We implemented FADPRM as an extension to the Motion Planning Kit (MPK)[7] by changing the definition of PRM to include zones with degrees of desirability and changing the algorithm for searching the PRM with FADPRM. The calculation of a configuration's *dd* and a path's *dd* is a straightforward extension of collision checking for configurations and path segments. For this, we customized the Proximity Query Package (PQP)[9]. In the next section, we show how FADPRM is used as a tutoring resource within *Roman Tutor*.

III. ATDG - Automatic Task Demonstration Generator

Filming the trajectory of the SSRMS fundamentally amounts to selecting the virtual cameras to be used to show the SSRMS at different points of the trajectory and the configurations of these cameras. There is infinity of possible virtual camera positions and possible virtual cameras configurations. If the objective is to give the operator a sense of the task as he will be seeing it from the command and control workstation, then virtual camera positions will be selected from the 14 positions of the cameras on the exterior of the ISS, that is, a finite number of possibilities. But if the objective is to convey some cognitive awareness of the task, then virtual camera can potentially be selected in any position that best help the operator gain a maximal cognitive awareness. For instance, the animation could include a perspective view that overview the entire ISS. Such a view is physically impossible from the inside of the ISS (there is no camera in the space overseeing the entire ISS), but it helps the operator reconstruct mentally the working space by seeing the ISS on a model.

The automatic task demonstration generator (ATDG)[2] takes as input start and goal configurations of the SSRMS. Using those two configurations, the ATDG will generate a movie demonstration of the required manipulations in order to bring the SSRMS from its start configuration to its goal configuration. The top figure in Figure 3 illustrates the internal architecture of the ATDG. The bottom one shows the different steps the data go through in order to transform the two given configurations into a complete movie demonstration.

First, the ATDG uses the FADPRM path planning algorithm, which takes the two given configurations and generates a collision free path between them. This path is then given to the trajectory parser which separates it into categorized segments. This will turn the continuous trajectory into a succession of scenes, where each scene can be filmed by a specific group of idioms. An idiom is a succession of shots that represents a stereotypical way to film a scene category. The parser looks for uniformity in the movements of the SSRMS to detect and recognize the category of scenes.

Once the path is parsed, the camera planner uses TLPlan[10] to find the best shots in order to best convey each scene, while making sure that the whole is pleasing and comprehensive. The use of TLPlan as a camera planner within ATDG provides two advantages. First, Linear Temporal logic (LTL), the language used by TLPlan is more expressive, yet with a simpler defined semantics, than previous camera planning languages such as DCCL[11]. For instance, we can express arbitrary temporal conditions about the order in which objects should be filmed, which objects should remain in the background until some condition become true, and more complex constraints that the LTL language can express. Secondly, TLPlan is more powerful than other camera planners used, for example, in References 11-12, because with TLPlan, LTL shot composition rules convey search pruning capability.

In ATDG, each shot in the idiom is distinguished by three key attributes: shot type, camera placement mode, camera zooming mode.

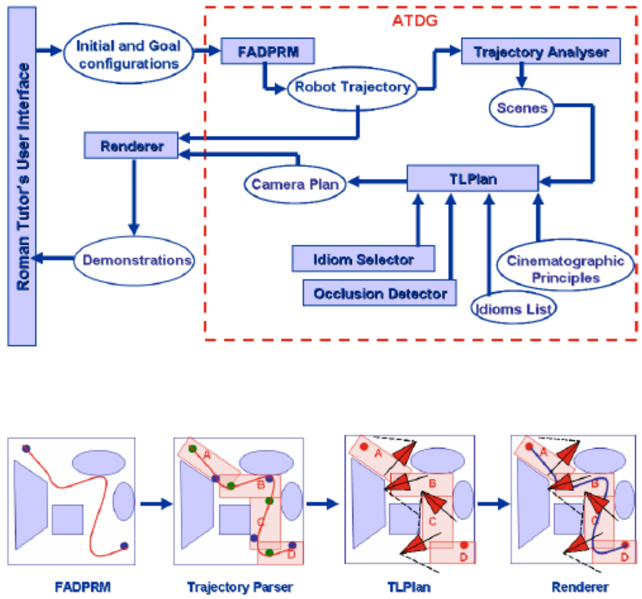

Figure 3. ATDG Internal Architecture.

1) Shot Types: five shot types are currently defined in the ATDG System: Static, GoBy, Pan, Track and Pov. A Static shot for example is done from a static camera when the robot is in a constant position or moving slowly. Whereas in a Track shot, a camera follows the robot and keeps a constant distance from it.
2) Camera Placements: for each shot type, the camera can be placed in five different ways according to some given line of interest: External, Parallel, Internal, Apex and External II. Currently, we take the trajectory of the robot's center of gravity as the line of interest which allows filming of a number of many typical manoeuvres. For larger coverage of manoeuvres, additional lines of interest will be added later.
3) Zoom modes: for each shot type and camera placement, the zoom of the camera can be in five different modes: Extreme Close up, Close up, Medium View, Full View and Long View.

Figure 4 shows an idiom illustrating the anchoring of a new component on the ISS. It starts with a Track shot following the robot while moving on the truss. Then, another Track shot showing the rotation of one joint on the robot to align with the ISS structure. And finally a Static shot focusing on the anchoring operation.

In TLPlan, idioms are specified in the Planning Definition Language (PDDL 3.0). Intuitively, a PDDL operator specifies preferences about shot types in time and in space depending on the robot manoeuver. Parsing the trajectory of the robot with the successive scenes performed, TLPlan will try to find a succession of shots that captures the best possible idioms. TLPlan also takes into account the cinematic principles to ensure consistency of the resulting movie. Idioms and cinematic principles are in fact encoded in the form of temporal logic formulas within the planner.

TLPlan uses also an occlusion detector to make sure the SSRMS is visible all the time. Once TLPlan is done, we are left with a list of shots that is passed to the rendering system to create the animation. The renderer uses both the shots given by TLPlan and the SSRMS trajectory in order to position the cameras in relation with the SSRMS, generating the final task demonstration.

IV. Human Machine Cooperation

One of the most interesting properties of RomanTutor is that it is an expert problem solver in the same perceptual space as the student. This has arisen only rarely in either intelligent systems research or in education research. Normally the computer has access to a very diminished form of the sensory information available to its human user. RomanTutor, because of this property, is in principle on a much more equal footing with its human user and this architecture provides important research opportunities, not only in training and education, but in mixed initiative (i.e., involving people and computers as partners) planning, problem solving and cooperative plan

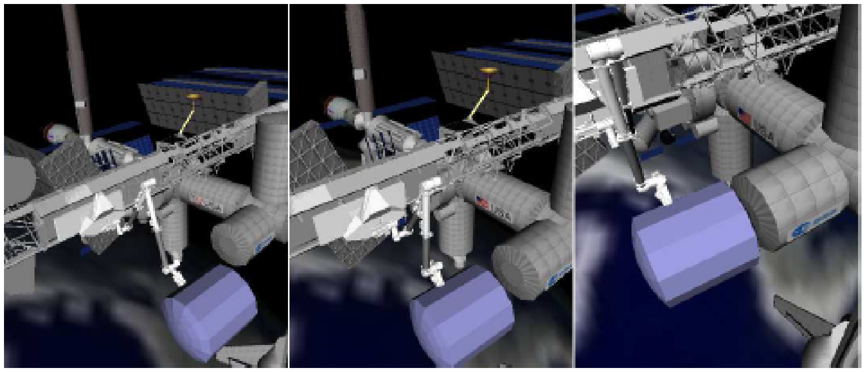

Figure 4. Idiom to film the SSRMS anchoring a new component on the ISS.

execution as well.

Obviously the interaction between the student and RomanTutor is a simple mixed initiative activity. RomanTutor initiates the training task and monitors the student's progress towards accomplishing the task. The student begins the task and can ask RomanTutor for help when they want a recommendation for what to do next. The system monitors the student's activity and regularly evaluates whether the task can be completed from the current configuration of the manipulator and whether it can be completed efficiently. At the point at which RomanTutor discovers that the student would have to backtrack from the current position or that achieving the task takes more than the time allotted for it, RomanTutor will intervene and begin to show the student a more efficient trajectory. Once a better initial trajectory has been demonstrated, the student can take control and resume the task. This error-prompted turn taking repeats until the task is completed. This sort of interaction demonstrates a limited form of mixed initiative activity in which the human and machine address a single system imposed goal.

In general we would like the human and machine participants to be able to address a greater variety of goals and for the machine to recognize its human partner's current goal and contribute to the achievement of that goal in some useful way. Of course, a primary objective is that the machine does not get in the way. Beyond that there is a spectrum of ever increasingly sophisticated ways in which the machine could help its human partner including (1) waiting for commands, (2) carrying out parallel tasks and pausing in order to avoid interfering with the human partner's activities, (3) recognizing that the partner is waiting for it to finish an interfering subtask and proceeding and (4) synchronizing with the human partner and carrying out coordinated actions. A simple example of coordinated activity that could be accomplished in the short term is the selection of camera views while the astronaut operates SSRMS. A system capable of this could be built from the components of RomanTutor with the addition of a sequencing element that tracks cues for the end of usefulness of a current camera view and that tracks cues for when a currently unused camera or camera view becomes useful for the current state of the human partner's primary task. Such cues can be extracted or verified by the motion planning component of RomanTutor operating on the current simulation environment. As cues indicate changes in utility for particular camera views, the sequencer, as it occurs in ATDG, would substitute an increasingly useful camera view for one of the less useful current camera views.

Another example involves a lunar EVA astronaut and a rover acting in a coordinated way to navigate to a site, position a science instrument carried by the rover and carry out some measurements or take samples. We would prefer to let the rover follow the astronaut without burdening the astronaut with driving it in a remote control way. The rover has to follow the astronaut but not so closely that it runs into the astronaut if they stop suddenly. The rover has to recognize obstacles or narrow passages and synchronize with the astronaut about who goes first. Once at the site the astronaut would likely have to direct the rover to the best position for the in situ measurements over the course of several cycles of direct examination of the site and examination of feedback from the rover's science instrument. The more sophisticated the rover's ability to recognize its astronaut partners objectives and to help, the more quickly the astronaut can carry out the task. A large part of this sort of cooperation is based on synchronizing in order to avoid geometric collisions or on achieving a particular geometric relationship between the rover's science instrument and features of the particular science site. Again RomanTutor's motion planning component is able to provide the basic cues that constrain or enable the coordinated activity.

V. Conclusion

RomanTutor's benefits to future training strategies are (1) the simulation of complex tasks at a low cost (e.g., using inexpensive simulation equipment and with no risk of injuries or equipment damage) and (2) the installation anywhere and anytime to provide "just in time" training. Crew members would be able to use it onboard the ISS, for example, to study complex maintenance or repair operations. For very long missions, they would be able to use it to train regularly in order to maintain their skills. In particular RomanTutor is able to generate as many training examples as the student wants. This capacity provides important learning challenges and opportunities that are not possible with the current system based on a fixed set of manually generated examples. RomanTutor with slight modifications would also be a good platform to explore possible solutions for mixed initiative activities.

References

[1] Kabanza, F. Nkambou, R. and Belghith, K. "Path-planning for autonomous training on robot manipulators in space", in International Joint Conference In Artificial Intelligence (IJCAI), pp. 1729–1731, (2005).

[2] Kabanza, F. Belghith, K. Bellefeuille, P. Auder, B. and Hartman, L. "Planning 3D Task Demonstrations of a Teleoperated Space Robot Arm", in International Conference On Automated Planning and Scheduling (ICAPS), pp. 1729–1731, (2008).

[3] Belghith, K., Kabanza, F., Hartman, L., Nkambou, R. "Anytime Dynamic Path-Planning with Flexible Probabilistic Roadmaps". Proc. IEEE Int. Conf. on Robotics and Automation (2006), pp. 2372–2377, (2006).

[4] Latombe, J.C. "Robot motion planning". Kluwer Academic Publishers, Boston, MA, (1991)

[5] Overmars, M.H. "Recent developments in motion planning". Computational Science - ICCS 2002, Part III, Springer-Verlag, LNCS 2331, (2002) 3-13

[6] LaValle, S. M., Branicky, M. S., Lindemann, S.R. "On the relationship between classical grid search and probabilistic roadmaps". Int. Journal of Robotics Research 23: 673–692, (2004)

[7] Sanchez, G., Latombe, J.C. "A single-query bi-directional probabilistic roadmap planner with lazy collision checking". Int. Symposium on Robotics Research (ISRR'01), Lorne, Victoria, Australia, November 2001. Springer Tracts in Advanced Robotics, Springer, 403-417 (2003)

[8] Likhachev, M., Ferguson, D., Stentz, A., Thrun, S. "Anytime Dynamic A*: An Anytime Replanning Algorithme". Proc. of Int. Conf. on Automated Planning and Scheduling, 2005

[9] Larsen, E., Gottschalk, S., Lin, M., Manocha, D. "Fast Distance Queries using Rectangular Swept Sphere Volumes". In Proc. of IEEE Int. Conf. on Robotics and Automation, 4:24-28, 2000

[10] Bacchus, F. and Kabanza, F. "Using temporal logics to express search control knowledge for planning". Artificial Intelligence, 116(1-2), 123–191, (2000).

[11] Christianson, D.B. Anderson, S.E. He, L. Salesin, D.H. Weld, D.S.and Cohen M.F., "Declarative camera control for automatic cinematography". In Association for the Advancement of Artificial Intelligence (AAAI), pp.148–155, (1996).

[12] Bares, W. Zettlemoyer, L. Rodriguez, D. and Lester, J. 1998. Task-sensitive cinematography interfaces for interactive 3d learning environments. In Intelligent User Interfaces, 81–88.