International Conference of Agricultural Engineering
AgEng 2014 Zurich 6–10 July
Ref: C0231

Motion planning algorithm and its real-time implementation in apples harvesting robot

Tien Thanh Nguyen, Erdal Kayacan,

Josse De Baerdemaeker, Wouter Saeys

MeBioS (Mechatronics, Biostatistics and Sensors) Division,

BIOSYST (Biosystems) Department, KU Leuven, Belgium

Tel: +32 163 21444; email: **tienthanh.nguyen@biw.kuleuven.be**

Abstract

A fruit-by-fruit harvesting robot has to reach the target fruits on a tree and perform the detaching movement without any collision with the obstacles (trunks, branches, leaves, etc.) in its workspace. However, the working conditions in a fruit orchard are only partially structured and partially visible. In such cases, a flexible motion planning algorithm, which allows replanning of the ongoing executed motion while receiving additional information from the environment, is required. In this study, an anytime motion planning algorithm for a 9 degree-of-freedom (DoF) manipulator to be used for selective apples harvesting and its real-time implementation are presented. Among the existing motion planning algorithms in literature, the sampling-based approaches are widely applied in robotics due to their capabilities of dealing with many DoFs robots without an explicit representation of the obstacles. In the proposed algorithm, the motion planning task is an ongoing process where the motion is re-planned in every configuration of the trajectory based on the acquired obstacle data. This approach requires a robust and time-efficient motion planner. Therefore, several different sampling-based motion planning algorithms have been evaluated in a specific simulation environment which is similar to the real working conditions to evaluate the most efficient planner. The proposed re-planning approach has been integrated by considering two parallel threads for planning and execution. The planning thread constructs an obstacle free roadmap and finds the trajectory for the user requested query. The execution thread takes the trajectory from the planning, sends the velocity references to the low-level controllers and monitors any change in the user request or new detected obstacles blocking the current executed trajectory. From this monitoring and the pre-defined delay of the trajectory execution, an updated query is generated and sent to the planning thread. This enhancement allows the planner to deal with not only any changing environment including newly observed obstacles or moving obstacles, but also any change in the high-level task which demands a new target location. The effectiveness of the proposed method has been validated in simulation where new obstacles appeared on the original trajectory or the target was changed to another location while executing the predefined path. The proposed method has also been implemented into a real manipulator by using an adaptive delaying buffer in motion command execution. Both in simulation and in the real-time tests the planner handled every situation correctly within an acceptable computation time.

Keywords: motion planning, harvesting, robot, apple

1 Introduction

The research towards selective harvesting robots for delicate fruits like apples started more than 20 years ago and has resulted in many research prototypes, such as those for tomatoes, eggplants, lettuces (Kondo & Ting, 1998), melons (Edan et al., 2000) and apples (Baeten et al., 2007). However, the challenging requirements of practical applications, i.e. combination of high speed and accuracy, ability to deal with the uncertain, complex environment and an acceptable cost, which are important for commercializing the research prototype, have still not yet been fulfilled.

Nguyen et al. (2013) proposed a framework that integrates the task planning in logical space with the motion planning in the geometry space for an apples harvesting robot that can accomplish the harvesting task with assured communication between sensing, planning and execution, and can cope with unstructured obstacles. However, due to the dynamic changes in the environment and the harvesting task, the robot must actively update the information while executing motion, and must be able to re-plan its motion if the current executing motion is not compliant with the updated constraints or if there is a change in the harvesting task.

A common approach is to reactively replan at each time step. This approach has been studied under various nomenclatures i.e. model predictive control, receding horizon control or real-time planning. Also different planners such as numerical optimization, forward search or sampling-based motion planners have been used in this context. Applications of this concept can be found in various field among which artificial intelligence (Karaman S. and Frazzoli E., 2011), control theory (Huynh et al., 2012) and robotics (Sucan et al., 2012)are the best known. Model predictive control (MPC) or receding horizon control is a form of replanning that at each time step formulates an optimal control problem truncated at some horizon. This concept has been successfully applied on agricultural machinery such as autonomous tractors (Kraus et al., 2013; Kayacan et al., 2014) and combine harvesters (Coen et al., 2008). Sampling based motion planners have among others been applied for real-time replanning for a mobile manipulator (Vannoy et al., 2008).

However, the main disadvantage of the replanning approach is that the planning step is computationally expensive which makes the replanning slow. As a consequence, the actions taken after planning can be based on outdated states, leading to major instability and constraint violations. Therefore, the objective of this study was to introduce and test a real-time replanning procedure which can avoid the main problem of the replanning approach.

In section 2 the core of the proposed approach is explained. The implementation of this approach for controlling the manipulator of an apples harvesting robot and the benchmarking of different sampling based motion planners in realistic simulations are described in section 3. The results of these simulations are presented and discussed in section 4. . Finally, conclusions are drawn from this study and suggestions for future research are proposed.

2 Materials and methods

2.1 Motion planning for apple harvesting robot

2.1.1 Sampling based motion planning

In our approach for planning the motion of the manipulator, the probabilistic sampling-based motion planning method is used (Nguyen et al., 2012). The sampling based planner has been designed to build a graph called a roadmap composed of nodes and edges that represent collision-free configurations and local paths. This expresses the network of free local paths in the given environment. Based on this roadmap, a collision-free path that connects the initial and goal configurations is computed using graph search. Different sampling-based motion planner have been proposed, such as rapidly-exploring random trees (RRT), bidirectional RRT (RRTConnect), kinematic planning by interior-exterior cell exploration (KPIECE), bidirectional KPIECE (BKPIECE), lazy bidirectional KPIECE (LBKPIECE), single-query bidirectional lazy collision checking planner (SBL) and expansive space trees (EST) (Elbanhawi, M. and Simic, M., 2014). The sampling based motion planners have the advantage that they are able to find a feasible motion plan relatively quickly. However, the solution from these planners can be unnecessarily long due to the sampling process and searching algorithm. Therefore, sampling based solution has to be executed in combination with an optimization step or trajectory filtering which is often more time consuming than finding the motion solution itself. To overcome this problem, we propose a continuous replanning approach that works simultaneously with the motion execution as will be explained in Section 2.3.

2.1.2 Obstacle representation

One of the advantages of using sampling based motion planners is that they do not attempt to explicitly construct the boundaries of the configuration space obstacles or represent the free space. Instead, they rely on a collision detection procedure. An efficient collision detection procedure will shorten the time needed by the sampling based motion planner to find a solution. On other hand, the collision detection procedure depends on the representation of the obstacles which are detected or captured by the 3D sensors. In our implementation, we use OctoMap (Hornung et al., 2013), a probabilistic 3D mapping framework based on octrees, to represent the collision map. The point clouds, acquired from several sources and registered into the global coordinate system are fused into an OctoMap which represents the collision map. The OctoMap provides an efficient representation with the following characteristics:

- **A probabilistic representation** is used to deal with the uncertainties such as measurement noise and dynamic obstacles, and to integrate data from multiple sensors
- **Modelling of free, occupied and unmapped area** to find the collision-free path only in the area that has been detected to be free. The occupied and unmapped area should be avoided and are therefore represented in separated types.
- **Efficient in access and storage** thanks to the multi-resolution representation of the collision map which reduces the access and collision checking time.

Since the manipulator can move inside the monitoring area of the sensor, it can be detected as an obstacle. This would be considered as a collision. To prevent this problem, a self-filtering procedure has been implemented which removes the point cloud corresponding to the manipulator through combination of the manipulator 3D CAD model and its states.

2.2 Motion execution for apple harvesting robot

2.2.1 Re-planning approach

The real-time replanning approach consists of interleaving threads of replanning, control and sensing. In this approach, the motion controller executes a partial trajectory that is intermittently updated by the replanning thread without interrupting execution. The updated trajectory is calculated from the updated environment model obtained by the sensing. The planner is given a delaying time Δ, during which it generates a new safe trajectory originating from the future state. As the robot moves, planning continues to improve the population of trajectories until the next control cycle, when the robot can switch to a fitter trajectory so that it always follows the probabilistic best trajectory. For that purpose, each trajectory is always updated to start from the following future robot configuration with its velocity when a new control cycle begins. For the trajectory that is being followed, this means that the executed portion of the trajectory is dropped from the trajectory, and the remaining trajectory until the following future state is the trajectory to be executed. To have a non-stop motion, the frequency of the replanning controller must be higher than the frequency of the execution controller and the update frequency of the sensor.

2.2.2 Implementation

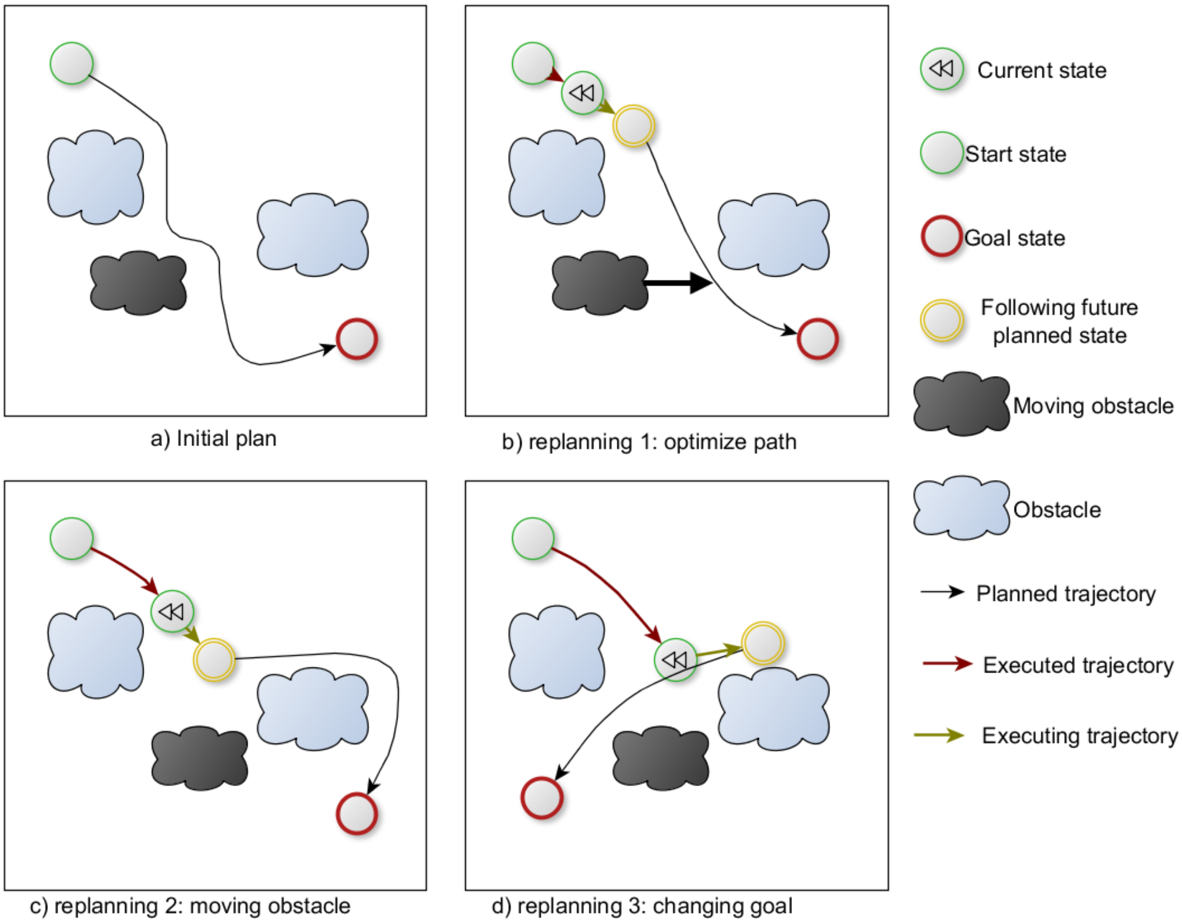


Figure 1: Illustration of the replanning approach

In Figure 1 the implementation of the proposed replanning approach is illustrated for several situations. Firstly, when the request to move from the starting position to the goal position is received, the motion planning will generate an initial collision free path according to recent sensing data. This path could be suboptimal since it is supposed to be improved in the next step. As the robot moves, the motion planning keeps running and plans a new path from the following future state of the robot. With longer time running and more sampling in the configuration space, the subsequent solutions should be closer to the optimum solution than the previous ones. Therefore, this replanning approach allows to obtain a probabilistic optimal solution for the path planning.

A replanning situation is considered in Figure 1(b) and (c). When robot executes the path of the planned trajectory, sensing data is updated and a moving obstacle is detected which blocks the planned path. The motion planning keeps running and plans the next path from the following future state of the robot based on the new updated sensing data. A new path is planned that will avoid the moving obstacle. To have successful obstacle avoidance the robot should still be in a position before the starting point of the next path executing the path from the current position to the next state planned in the previous step when the new path becomes available.

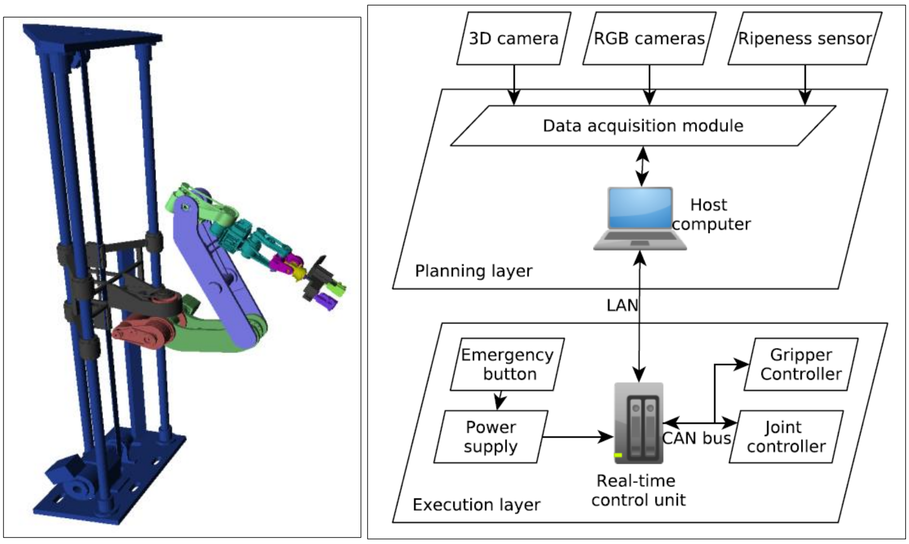
Another replanning situation is considered in Figure 1(d). When the robot executes the path of planned trajectory, a new target position is sent by the higher task planning. So, the robot must change its path to the new goal. The motion planning plans the next path from the following future state of the robot to the new received target. A new path is planned to the new target, but the robot will only start executing the new path when it has reached the starting state from where the new path has been planned.

2.3 The apple harvesting robot

Within the Crops project[1], an apple harvesting robot has been developed. Having a highly configurable, modular design, the apple harvesting robot not only accomplishes the harvesting task for apple, but also is capable of adapting to other agricultural tasks and conditions. The robot consists of a mobile platform equipped with power and compressed air supplies that can easily travel in the apple orchard and a custom built manipulator with the sensors, vision and control system carried by the mobile platform.

2.3.1 The manipulator

In this study, a modular multi-purpose agricultural harvesting manipulator developed by the Institute of Applied Mechanics at the Technical University of Munich (Baur et al., 2012) is used as shown in Figure 2(a). The working area of the robot covers the entire distribution space of ripe fruits on the tree. The manipulator is a redundant manipulator with 9 degrees of freedom (DOF). The redundant DOFs allow the manipulator to reach the fruit from various directions, and to operate at close range as well as at a distant location with a high dexterity which is very important to perform the detaching movement for apples picking. However, with many DOFs, it is impossible for other motion planning methods than sampling based motion planning methods to solve the path planning problem in a reasonable amount of time.



a) b)

Figure 2: 3D model of the apple harvesting manipulator and structure of the control system

2.3.2 The control system

In Figure 2(b), the structure of the control system of the robot is illustrated. It consists of two layers: a planning layer and an execution layer. At the center of the execution layer there is a real-time control unit that calculates the inverse kinematics of the manipulator and communicates with the low-level controllers of the different joints by four CAN bus connections. The main unit of the planning layer is a host computer using ROS as the environment to integrate the control interface and all software modules. The sensor system includes 3D cameras, RGB cameras and ripeness sensors to collect information from the harvesting scene. The real-time control unit connects to the host computer through a LAN interface that provides the status information of the manipulator and receives the desired pose of the Tool Center Point (TCP) for inverse kinematic calculation. However, the host computer can also fully control the manipulator in real-time by sending the desired states of each joint of the manipulator

[1] http://www.crops-robots.eu/

(positions, velocities, accelerations) to the execution layer without using the inverse kinematic calculation module of the real-time control unit.

3 Results and Discussions

3.1 Benchmarking motion planners

The different sampling based motion planners have been benchmarked on the testing situation illustrated in Figure 3. In the benchmarking situation, the obstacle is the model of an apple tree which is 1 meter away from the robot. A set of planning queries, which are simulated for the picking situations and moving to the bin, has been created. For the benchmarking, each motion planner has been tested on 100 planning queries (harvesting 100 apple fruits). All these algorithms were run with the predefined parameters and the limit planning time was set to 5 seconds. Due to the fact that the benchmarking situation is fairly easy with 1 meter of free space in front of the manipulator and the manipulator with 9 DOFs has sufficient flexibility, all planners had a 100% success rate in the limit planning time.

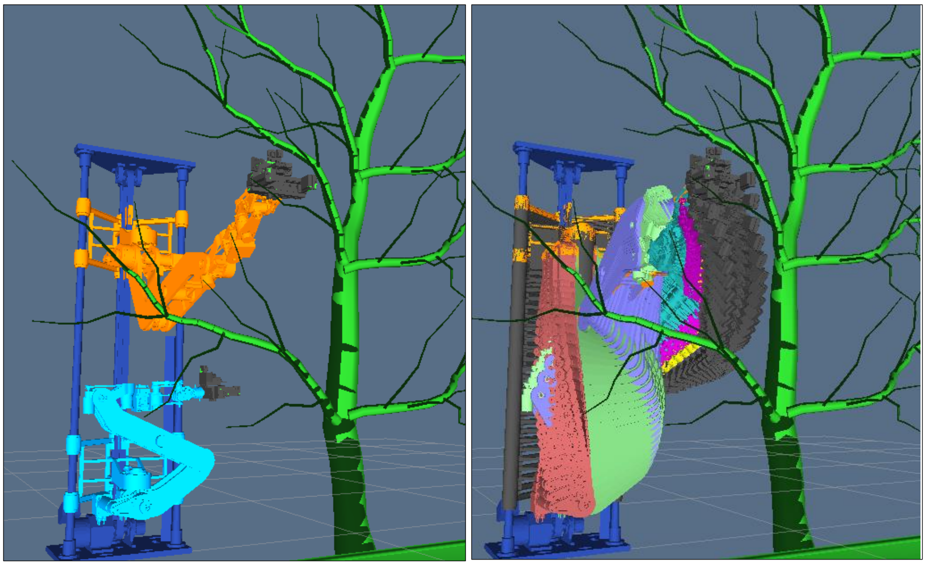


Figure 3: Benchmarking situation for different motion planners

The box plots of the planning times for the different planners are illustrated in Figure 4. It can be clearly seen that in this test the RRTConnect planner was the most efficient algorithm with an average planning time around 250 milliseconds and a maximum planning time below 400 ms. As this would allow replanning at a frequency of 2 Hz, this is very promising for the real-time implementation. Since each motion of the robot from initial position to picking position or from picking position to basket takes approximately 5 seconds, it means that the robot can plan and replan 9 times while executing the motion which allows it can response with the changes in the configuration space on time.

3.2 Implementation results

Once the RRTConnect planner had been chosen based on the simulation results, it has been implemented on the harvesting robot. To choose the most efficient frequencies for motion planning, motion execution and updating sensing, several experiments have been executed with different frequency parameters. The frequency for motion planning is depends on the planning time of the planner for each environment. The frequency for updating sensing is set equal to the frequency of the motion planning since the sensing will provide the input for the motion planning. The frequency of the motion execution must be smaller than the frequency of the motion planning, and need to choose carefully. When the execution time was set too short (less than 0.5 second), the planning time was limited to too few samples in the configuration space, leading to a suboptimal solution. When the execution time was set too long (above 4 seconds) this leads to the problem that the robot finish the motion without replanning the motion. For apple harvesting task, the motion planning and updating sensing frequencies are set to 0.5 seconds, and the motion execution frequency is set to 2 seconds.

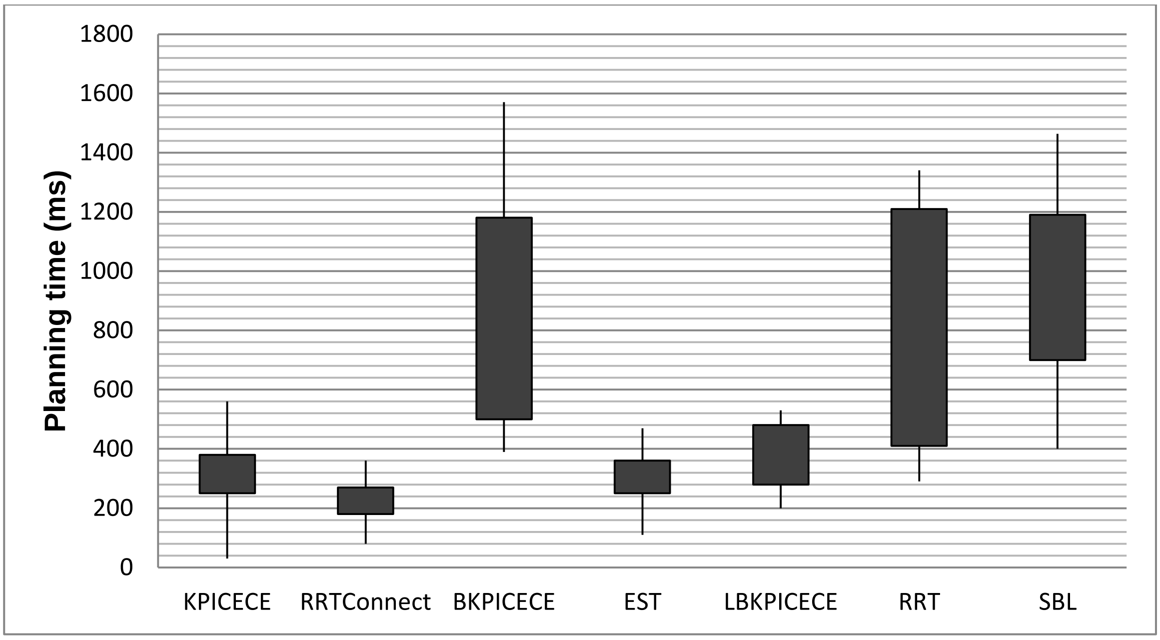


Figure 4: Planning time of different motion planners

4 Conclusions

In this study, we have considered the motion planning problem for 9DOFs manipulator which is used for apple harvesting, and has to deal with dynamic changes in the environment and targets. A real-time replanning approach for motion planning and execution has been elaborated which plans and replans the manipulator trajectory to the target position several times while executing the planned trajectory. The approach was implemented successfully by starting the new replanning procedure from the following future state in the executing trajectory. In this approach, replanning, execution and sensing are performed simultaneously in real-time and feed each other at every time step. OctoMap was selected as an efficient representation for the collision map and the RRTConnect was selected as the most efficient sampling based motion planner to be included in the replanning approach. Although the obtained results are very promising, there is still room for improvement in the proposed approach. The cycle times for sensing, planning and execution were optimized for the case of apple harvesting, but for other tasks and environment the optimal values may be different. A cost function of moving distance should be implemented to choose the best trajectory since different trajectories can be obtained. In future experiments this approach of replanning should be validated in real-time experiments with the apple harvesting robot in an orchard.

5 Acknowledgements

This research has been funded by the European Commission in the 7th Framework Programme (CROPS GA no. 246252). The opinions expressed in this document do by no means reflect the official opinion of the European Union or its representatives. The manipulator used in this paper has been developed by the Institute for Applied Mechanics at the Technical University of Munich – Germany.

6 References

Aoude et al. (2013). Probabilistically Safe Motion Planning to Avoid Dynamic Obstacles with Uncertain Motion Pattern. *Autonomous Robot*, 51-76.

Baeten et al. (2007). Autonomous Fruis Picking Machine: A Robotic Apple Harvester. *6th International Conference on Field and Service Robotics, 42*.

Baur et al. (2012). Design and development of a redundant modular multipurpose agricultural manipulator. *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, (pp. 823-830).

Coen et al. (2008). Cruise control using model predictive control with constraints. *Computers and Electronics in Agriculture*, 227-236.

Edan et al. (2000). Robotic melon harvesting. *IEEE Transactions on Robotics and Automation, 16*(6), 831-835.

Elbanhawi, M. and Simic,M. (2014). Sampling-Based Robot Motion Planning: A Review. *IEEE Access*, 56-77.

Gratal et al. (2012). Visual servoing on unknown objects. *Mechatronics*, 423-435.

Hornung et al. (2013). Octomap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*.

Hsu et al. (2002). Randomized Kinodynamic Motion Planning with Moving Obstacles. *The International Journal of Robotics Research*, 233-255.

Huynh et al. (2012). An Incremental Sampling-based Algorithm for Stochastic Optimal Control. *IEEE International Conference on Robotics and Automation*, (pp. 2865-2872). Minnesota, USA.

Karaman et al. (2011). Anytime Motion Planning using RRT*. *IEEE International Conference on Robotics and Automation (ICRA).* Shanghai, China.

Karaman S. and Frazzoli E. (2011). Incremental Sampling-based Algorithms for a Class of Pursuit-Evasion Games. *Algorithmic Foundations of Robotics IX*, 71-87.

Kayacan et al. (2014). Distributed nonlinear model predictive control of an autonomous tractor–trailer system. *Mechatronics*.

Kondo, N., & Ting, K. (1998). *Robotics for Bioproduction Systems.* American Society of Agricultural Engineers.

Kraus et al. (2013). Moving horizon estimation and nonlinear model predictive control for autonomous agricultural vehicles. *Computers and Electronics in Agriculture*, 25-33.

Luna et al. (2013). Anytime Solution Optimization for Sampling-Based Motion Planning. *IEEE International Conference on Robotics and Automation*, (pp. 5053-5059). Karlsruhe.

Nguyen et al. (2012). Optimum deteching movement for apples-harvesting robot. *International Conference of Agricultural Engineering.*

Sucan et al. (2012). OMPL: The open motion planning library. *IEEE Robotics Automation Magazine, 19*(4), 72-82.

Vannoy et al. (2008). Real-Time Adaptive Motion Planning (RAMP) of Mobile Manipulators in Dynamic Environments With Unforeseen Changes. *IEEE Transactions on Robotics*, *24.*