Available online at www.sciencedirect.com

ScienceDirect

IFAC-PapersOnLine 49-16 (2016) 226–230

Evaluation of a Spray Scheduling System ⋆

Jasmine Cashbaugh * **Stuart Bradley** **

* *Department of Electrical and Computer Engineering, University of Auckland, Auckland, NZ 1010 (e-mail: j.cashbaugh@auckland.ac.nz).*
** *Department of Physics, University of Auckland, Auckland, NZ 1010 (e-mail: s.bradley@auckland.ac.nz)*

Abstract:
There are many applications for outdoor automation in agriculture and horticulture that require liquid to be sprayed variably across a linear boom while a robotic vehicle moves the boom across a field or orchard. This paper examines the modification of an existing scheduling algorithm to take into account real-world effects on spray droplets targeted at overhead flowers. To test the algorithm modifications, a simulation was performed using several different robotic platform velocities to test the effectiveness of the system. These results were then compared to a statistical analysis to ensure their validity.

Keywords: agriculture, algorithms, Monte Carlo simulation, probabilistic models

1. INTRODUCTION

When designing a system to deliver spray to a given target, the behavior of spray droplets is an important consideration as the motion of the spray once it leaves the nozzle affects where and when the spray will land. For example, both the initial magnitude and direction of the spray velocity needs to be taken into consideration when modelling the spray trajectory. Further, the spray itself is subject to effects such as gravity, drag, and wind that must also be taken into consideration when deciding when to command a nozzle to spray. This paper discusses the equations used to model the droplet behavior and the effect of the spray trajectory on system performance.

The application motivating this paper is shown in Fig. 1 and consists of a moving robotic platform actuating spray nozzles at target flowers as they pass. Targets are randomly distributed in the test environment and their locations are not known in advance. Instead, the location of each target is determined by a camera pair in real-time as the platform passes. The goal of this application is to approach the value of 100% of targets hit when the sprayers were on continuously. However, the material to be sprayed should be minimized and only the amount needed by each target should be used; thus, the goal of the scheduling algorithm is amended to hit 95% of targets while using the minimum amount of spray material. This value was chosen as an estimate of the number of flowers that can be seen and reached from a robotic platform travelling below the flowers. There is no limit on the number of sprayers that can fire at once, but only one sprayer should fire at each target. This means that there is a single opportunity to hit each flower so the spray must be aimed precisely.

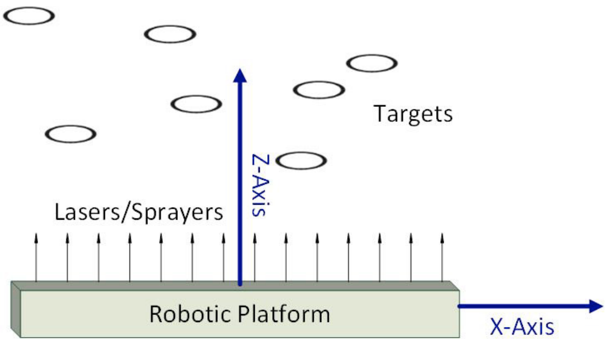

Fig. 1. A diagram of the robotic system application. The y-axis is in the direction of the platform's movement and forms a right-handed coordinate system.

The requirement that this system is capable of at least 95% target coverage leads to the inclusion of a coverage area problem where the goal is to determine how to cover the targets in the most effective manner. Osterman et. al. (2013) examined this problem for an orchard application where three robot arms were used to spray pesticides on trees. A LIDAR system was used to obtain canopy measurements in real-time and a positioning algorithm was used to determine the position and orientation of the three arms that would result in the maximum coverage of the near side of a tree. This work was an improvement upon an earlier model developed by Hočevar et. al. (2010) which used an RGB camera to obtain canopy measurements and made use of threshold values to determine whether to spray with a given nozzle.

In the application presented in this paper, the linear array of spray nozzles is not moveable in segments. Thus, the coverage area problem was explored by using images of the canopy to determine when to turn on each nozzle. This allowed the spray nozzle array to meet the coverage requirements while constrained to a fixed physical shape.

⋆ Funded by MBIE Multipurpose Orchard Robotics contract UOAX1414.

2405-8963 © 2016, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2016.10.042

Each target was assigned to the nearest sprayer based on a geometric analysis. The target's place in the individual spray nozzle queue was found by comparing the current robotic platform position with the calculated target position. This allowed the queuing system to be flexible since it made use of environmental feedback to schedule the spray nozzles.

This problem formulation is similar to a tracking problem with anticipation; the targets are tracked and the spray is fired to intercept the target's future position. This idea is applied in Hirsch et. al. (2011) to a varying number of unmanned aerial vehicles (UAVs) used to track six ground targets. The target motions were not known in advance and the UAVs independently determined their trajectories to minimize the tracking uncertainty over all targets by estimating the future position of the tracked objects. Charlish and Hoffmann (2015) explored this idea as applied to radar performance and assumed that the current state could not be fully known, but only partially observed through noisy measurements. A partially observable Markov decision process was used to anticipate the future position of the target so that the radar could follow a smooth path while keeping the target in range. The authors argued that selecting future actions based on the current state was not enough. Instead, future actions should be selected based on how the system was expected to behave in the future.

Similarly, Spletzer and Taylor (2003) used mobile camera systems to track one or more moving targets. The mobile camera systems had constrained velocities and their positions were limited by a minimum distance from the target. It was assumed that each mobile camera system knew its own position. Dynamic models of the target's motion were obtained using an approximation of the target dynamics. The mobile camera systems moved to minimize the target position estimate error at the next time instant based on the dynamic model.

In the method presented here, the droplet dynamics were modelled as accurately as possible so that the spray would reach the intended target as it passed above the robotic platform. As such, the trajectory of the spray was modelled as a parabola rather than as a simple straight line trajectory. This resulted in a more accurate representation of the real-world spray behavior and better prediction of the spray location at the time of impact with the canopy. Further detail of the scheduler algorithm used in this paper can be found in Section 2 and a description of the spray system can be found in Section 3. Section 4 provides details about the scheduler algorithm modifications while Section 5 provides the simulation results. Finally, the conclusions and future work can be found in Section 6.

2. SCHEDULER ALGORITHM OVERVIEW

The scheduler algorithm presented in this paper made use of a grid system to determine when to schedule spraying of flowers as they passed above the sprayer array. The flower locations were determined by a camera pair which made observations about the flower positions and supplied (x, y, z) locations in the local coordinate frame. The robotic platform velocity information was then used to assess whether each flower was within the designated grid spaces, and, if so, the flower was assigned to the nearest sprayer. This process was repeated for each flower located by the camera pair and a single binary command string representing each sprayer command was sent to the sprayer array. The algorithm was discussed in greater detail in a separate publication by Cashbaugh et. al. (2016).

For the simulations presented here, this algorithm was implemented in MATLAB 2014b on a conventional Pentium-class workstation running Windows 7. The workstation had a 2.70 GHz processor and 8.00 GB of RAM and ran the algorithm in less than 5 ms.

3. SPRAY SYSTEM DESCRIPTION

The spray array for this application consisted of 90 spray nozzles spaced evenly along a linear boom. Each nozzle emitted a spray cone with a small interior angle, consisting of droplets with a known mean diameter and velocity at the nozzle exit. These spray nozzles pollinated the flowers using a mixture of water and pollen with properties very similar to that of water itself.

The targeted performance value of the spray system was to hit 95% of the flowers within reach of the platform. All simulations and physical experiments discussed in this paper were evaluated with this performance criteria in mind and only trials that achieved a 95% hit rate were considered to be successful trials. It is important to note that 95% of the flowers needed to be hit, not pollinated. This is an important distinction since pollination requirements are not immediately obvious and require further investigation. For example, Campbell and Haggerty (2012) claim about 13,000 pollen grains per stigma are required for successful pollination while Hii (2004) claims about 3000 to 4000 grains per stigma are required for the desired export weight of 70 g.

4. SCHEDULER ALGORITHM MODIFICATIONS

Several modifications to the algorithm developed by Cashbaugh et. al. (2016) were necessary in order to use this algorithm for the application discussed in this paper. Previous versions of this scheduling algorithm were designed to work with a laser system, but laser beams are not subject to gravity, wind, or drag and so are easier to schedule and simulate. However, liquid spray is subject to such physical parameters, requiring further scheduler modifications to take these effects into account.

The effect of gravity was taken into account using the following formula for simple projectile motion:

$$h_t = -\frac{1}{2} * g * t_h^2 + v_d * t_h \tag{1}$$

Here, h_t is the height of the target in m, g is the gravitational acceleration of 9.81 m/s^2 in the negative z direction, t_h is the time it takes for the droplet to reach the target height, h_t, in seconds, and v_d is the initial droplet velocity in m/s in the positive z direction. This equation was then solved for time to obtain (2).

$$t_h = \frac{v_d \pm \sqrt{v_d^2 - 2 * g * h_t}}{g} \tag{2}$$