Proceedings Paper:
Davy, J. and Fox, C. (2023) Simultaneous Base and Arm Trajectories for Multi-target Mobile Agri-Robot. In: Towards Autonomous Robotic Systems. 24th Annual Conference, TAROS 2023, 13-15 Sep 2023, Cambridge, UK. Springer, pp. 214-226. ISBN: 9783031433597. ISSN: 0302-9743. EISSN: 1611-3349.

https://doi.org/10.1007/978-3-031-43360-3_18

Simultaneous Base and Arm Trajectories for Multi-Target Mobile Agri-Robot

Joshua Davy[1,2] and Charles Fox[2,3]

[1]STORM Lab, School of Electronic and Electrical Engineering, University of Leeds
[2]School of Computer Science, University of Lincoln; [3]Ibex Automation Ltd

Abstract. Many agricultural robotics tasks require an end effector to hold stationary above individual plants in the field for short periods. Examples include precision harvesting, imaging and spraying. This effector may be mounted on a mobile base such as a large tractor or small robot, driving in the field. We consider how to optimise control of the base and the end actuator together, to minimise total time taken to visit the plants. Our approach is based on low level combination of simple motion primitives, with mid level target clustering, and higher level planning. For the high level, three strategies are compared and evaluated in simulation: baseline stop-and-spray, constant velocity, and variable velocity. The baseline strategy is common in existing systems, and is shown to be outperformed by the new methods. The application considered here is weed spraying, but the methods are applicable to many tasks.

1 Introduction

Precision Agriculture is the differential rather than aggregate treatment of crops. Ideally, it aims for per-plant precision, treating each plant as an individual. This allows time and resources to be utilised more efficiently compared to aggregate treatments. It is well suited to robotics, due to their accuracy and non-reliance on human labour time [1] . Mobile robots used in agriculture usually fall into two classes: automated large tractors, and small human-size robots. Small robots equipped with arms are being applied to precision agriculture processes such as crop harvesting [2], imaging, data collection, [3] [4], and chemical treatments [5] [6]. Such applications require an end effector to be accurately positioned over each plant or region of interest, and held there for a period of time to performs its work, such as spraying or collecting data. Agricultural systems typically have tens of thousands of plants per hectare, so visiting each plant can be prohibitively time-consuming even for autonomous robots. More usually, a subset of plants in need of actuation can be identified in advance, then visited.

In most systems [6] [7] [8] [9], robots with arms drive from one plant of interest to the next, stopping near each one. They then move their arm to accurately position an end effector over the plant, then activate the effector, all with the base stationary. This appears inefficient compared to a system where motion of robot base and arm take place simultaneously, and without the base needing to stop. The present study thus considers generation of simultaneous trajectories for the base and arm in to improve this efficiency.

The duration the effector is required to be held over the plant must be specified for each application. Any arm has a upper limit on its movement speeds [10], which must accounted for along with the speed and maneuverability of the base. The order to visit targets is an important consideration. The optimum order would maximise the number of weeds visited within available time (and roughly equivalently, energy use) subject to the above motion constraints. Computational efficiency is also important for systems to be practically deployed.

We consider only cases where the base is constrained to drive in predefined straight, parallel rows (swathing), but is able to adjust its speed. This is usually required if crops are planted in rows, so that the wheels drive between rather then over crops. Where crops are not in rows, as in grassland, swathing is still often used to ensure coverage of a field. Large tractors with boom style arms are usually most efficient when swathing. Small robots could however drive directly from target to target. Optimising for that case is a larger problem beyond the present study, though the present study might provide a first step towards its solution. Even when driving is constrained along rows, finding an optimal ordering of targets for the arm to visit forms an NP-hard travelling salesman problem [6], so heuristics may be needed.

The only previous work to consider arm motion planning for agri-robots with continuous base movement [11] uses potential field methods to dynamically calculate the motion of a mounted manipulator in real time. However, it assumes approaching crops are linearly spaced and therefore no choice of optimum spray order is needed. An analogous problem exists in industrial manipulation scenarios such as the interception of items on a conveyor belt [12]. Here instead of a moving robot base, objects move towards the manipulator which must intercept the objects as they come into the arm's workspace. As the aim is to maximise the number of objects intercepted by the manipulator, the motion must take place at the upper limits of the arm in order to reduce trajectory time. This upper limit is defined by the dynamic constraints of the system which involve solving complex differential equations numerically in order to find the fastest trajectory [10]. The application considered here is precision weed spraying, but the methods are applicable to many precision agriculture and other tasks. Weed detection from ground platforms [13] and UAV [14] platforms has been demonstrated, so we assume that such detections are available as input. We use a heuristic which considers paths between and within clusters of nearby targets.

2 Methods

Generation of appropriate trajectories for the weed spraying task is here considered at three abstraction levels. The lowest is the formulation of individual weed spraying arm motions where the goal is to generate kinematically feasible trajectories which intercept and match the weed's position and relative velocity with the arm sprayer. This is achieved by a fast, piecewise combination of joint space polynomial and Cartesian space linear segments. The middle level is that of choosing the optimum weed spray ordering in advance in order to simplify the planning process. This is approximated via a K-means based clustering of upcoming weeds in order to group weeds which should be sprayed together in sequence. The highest level chooses the points in the arm workspace at which weeds should be sprayed, together with planning the base velocity to maximise performance.

The focus is on generating arm trajectories and base speeds given known positions of upcoming weeds in the robot's fixed direction of travel. As the robot moves along the field these weed relative positions move closer to the robot. The end effector must be held above the weed as the pesticide is released and therefore must match the Cartesian velocity of the ground relative to the robot.

2.1 Demonstrator Platform

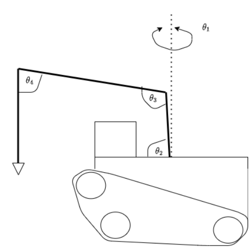

Fig. 1: Demonstrator platform - Kinematic structure and joint parameters.

Our demonstrator robot platform (Figs. 1) consists of a 4DOF arm manipulator mounted to a mobile base. The final joint in the arm is constrained to always point the pesticide sprayer vertically down. This means motion generation is simplified to consider only the first three joints of the system. Inverse kinematic solutions can then be found analytically due to the simplified structure.

Here we solve inverse kinemetics for the particular demonstrator robot, by constraining motion of the arm and considering the its position in cylindrical coordinates. Fig. 1 shows the kinematic structure of the arm with four joint angles $\{\theta_1, \theta_2, \theta_3, \theta_4\}$ and three linkages of lengths $\{l_1 = 0.5\text{m}, l_2 = 1.0\text{m}, l_3 = 0.5\text{m}\}$.The position of the end effector is considered in a 3D cylindrical coordinates system (r, θ, z). Forward kinematics can be solved to determine position of the end effector given the joints,

$$\begin{bmatrix} r \\ \theta \\ z \end{bmatrix} = \begin{bmatrix} l_1 \cos(\theta_2) + l_2 \cos(\theta_2 + \theta_3) + l_3 \cos(\theta_2 + \theta_3 + \theta_4) \\ \theta_1 \\ l_1 \sin(\theta_2) + l_2 \sin(\theta_2 + \theta_3) + l_3 \sin(\theta_2 + \theta_3 + \theta_4) \end{bmatrix} \tag{1}$$

The inverse solution, $\theta_1 = \theta$, is trivial given its direct correspondence with the cylindrical coordinate angle θ. The end linkage is constrained to point the end effector vertically downwards, to direct the spray correctly. θ_4 therefore is set by compensating for the other joints, $\theta_4 = -\theta_2 - \theta_3$. Substituting Eqn 1,

$$\begin{bmatrix} r \\ \theta \\ z \end{bmatrix} = \begin{bmatrix} l_1 \cos(\theta_2) + l_2 \cos(\theta_2 + \theta_3) \\ \theta_1 \\ l_1 \sin(\theta_2) + l_2 \sin(\theta_2 + \theta_3) - l_3 \end{bmatrix}. \tag{2}$$

The inverse solution for θ_2 and θ_3 can be determined by trigonometric identities, as shown in [15], which gives,

$$\theta_3 = -\mathrm{acos}(\frac{r^2 + (z + l_3)^2 - l_1^2 - l_2^2}{2 l_1 l_2}), \tag{3}$$

$$\theta_2 = \mathrm{atan2}(z + l_3, r) + \mathrm{atan2}(l_2 \sin(\theta_3), l_1 + l_2 \cos(\theta_3)). \tag{4}$$

Maximum platform base velocity is 1m/s. Maximum joint velocities are $\dot{\theta}_{1,max} = \dot{\theta}_{2,max} = \dot{\theta}_{3,max} = \dot{\theta}_{4,max} = 3$ rads^{-1}, and maximum accelerations are $\ddot{\theta}_{1,max} = 5$rads^{-2}, $\ddot{\theta}_{2,max} = \ddot{\theta}_{3,max} = \ddot{\theta}_{4,max}\} = 4$rads^{-2}.

Kinematic solutions for other robot configurations may vary but will have a similar structure to our example. The rest of the methods are general to any mobile base with any arm.

2.2 Low Level Motion

The low level motion generates smooth trajectories for the arm joints given the relative velocity of a single weed, a required point for the sprayer and weed to first meet, and a required duration for the sprayer to be held above the weed t_{spray}. This level consists of two phases. The *intercept* phase is getting the effector above the weed and matching its relative velocity. The *tracking* phase is holding it in place during the spraying.

Trajectories for the arm are formed by piecewise combination of two phases. To reduce computational complexity, only kinematic joint limits such as maximum angular velocity and acceleration – as opposed to dynamic limits such as maximum angular momentum and torque – are considered (this heuristic often works in industrial manipulation [10]).

In the *intercept* phase, the trajectory is defined by a 5th degree polynomial generated in joint space. For each of the robot joints, j, its pose q at time t is:

$$q_j(t) = a_j + b_j t + c_j t^2 + d_j t^3 + e_j t^4 + f_j t^5 \tag{5}$$

and the six polynomial parameters $a_j - f_j$ can be found analytically as the solution to six simultaneous equations, formed by imposing a target initial and final constraint on the position, velocity and acceleration of each joint, $(\theta_0, \dot{\theta}_0, \ddot{\theta}_0, \theta_f, \dot{\theta}_f, \ddot{\theta}_f)$ found by inverse kinematics to match the position and velocity of the weed, and the trajectory start and end time t_0, t_f. The six equations are formed by differentiating eqn. 5 twice, for both the start and end times:

$$\begin{bmatrix} \theta_{j,0} \\ \dot{\theta}_{j,0} \\ \ddot{\theta}_{j,0} \\ \theta_{j,f} \\ \dot{\theta}_{j,f} \\ \ddot{\theta}_{j,f} \end{bmatrix} = \begin{bmatrix} 1 & t_0 & t_0^2 & t_0^3 & t_0^4 & t_0^5 \\ 0 & 1 & 2t_0 & 3t_0^2 & 4t_0^3 & 5t_0^4 \\ 0 & 0 & 2 & 6t_0 & 12t_0^2 & 20t_0^3 \\ 1 & t_f & t_f^2 & t_f^3 & t_f^4 & t_f^5 \\ 0 & 1 & 2t_f & 3t_f^2 & 4t_f^3 & 5t_f^4 \\ 0 & 0 & 2 & 6t_f & 12t_f^2 & 20t_f^3 \end{bmatrix} \begin{bmatrix} a_j \\ b_j \\ c_j \\ d_j \\ e_j \\ f_j \end{bmatrix} \tag{6}$$

These can be solved via the Moore-Penrose pseudoinverse. To find the the maximum joint velocities and accelerations – which tells higher levels if the trajectory is feasible – an analytic method can be used. The first and second derivatives of the trajectory form quintic and cubic polynomials respectively. These represent the velocity and acceleration during the trajectory. Differentiating them gain, and solving for their roots, gives their maxima and minima.

For the *tracking* phase, a simple linear trajectory for the end effector is found in Cartesian space, $p(t) = v(t - t_0) + c$, where v is the relative velocity and c is the intercept point. This motion is formed by keeping the effector at a constant velocity, compensating for base motion. To calculate if the kinematic limits are being satisfied (for use by higher levels), the inverse kinematic solution is calculated at discrete points in order to check against the joint limits.

After the tracking phase, the final state of the manipulator is used as the initial state of the next intercept phase as the arm intercepts the next weed. By alternating intercept and tracking motions, multiple weeds can be visited, if the arm can intercept them while maintaining joint kinematics limits, as shown in Fig. 2. After this relatively computationally cheap stage the full sets of joint angles over time that form the trajectory can be calculated by eqns. 5 and 6.

Calculating Validity of Weed Spraying By generating a full trajectory, the validity of a weed to be sprayed by the system can be calculated at a given intercept point. To determine the intercept point for the weed, trajectories could be generated at each time step until the first successful trajectory is found. This however, is computationally inefficient. Most trajectories will be invalid, for not yet entering the arm range, or being beyond the point where the weed can be successfully sprayed. We thus use an initial system to reject weeds by determining a smaller range of intercept points to evaluate with the full trajectory generation system. The reach of the arm is treated as a constant radius around the arm base on the 2D weed plane. As a weed enters this circle, then it is at a valid point of interception. The weed will stay in a valid interception pose until the point at which there will be not enough time to track the weed whilst the pesticide is released. Fig. 4 shows how this valid range is calculated. Consider a weed at a current position (x, y). The first point at which the weed enters a valid intercept point is (x_0, y_0). This point is given as $(x, +\sqrt{r^2 - x^2})$ where r is the radius of valid intercepts. This occurs at time $t_0 = \frac{y - y_0}{v}$ where v is the forward velocity of the robot. The point at which the weed is no longer in a valid state is given by (x_1, y_1). Any point after this, there is no longer time to intercept

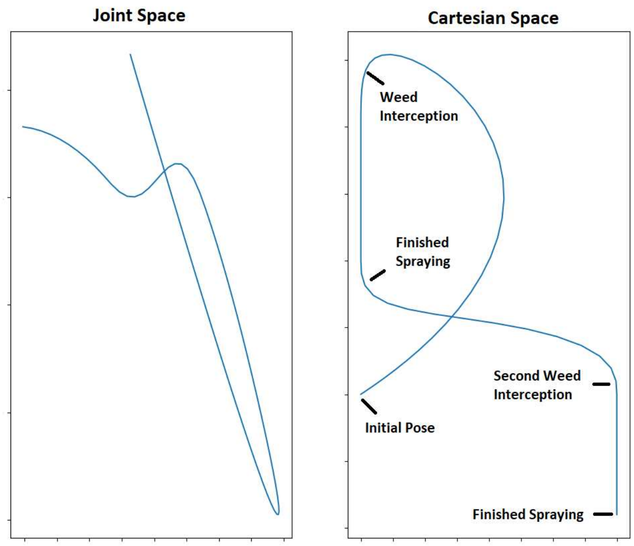

Fig. 2: Example generated trajectories for a two joint arm for spraying two weeds. Consisting of joint space polynomial segments for matching the weed position and velocity followed by Cartesian space linear segments for tracking the weed whilst the sprayer is activated.

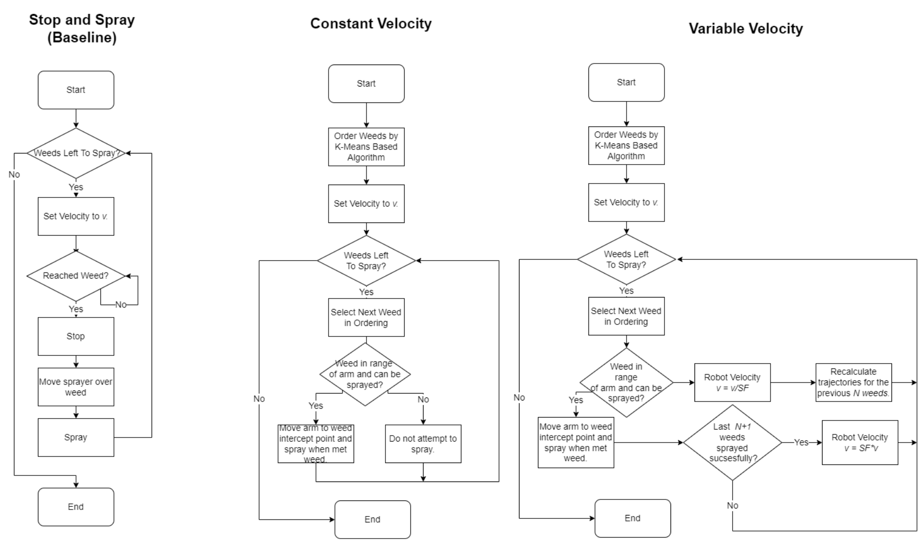

Fig. 3: Flowcharts showing the three proposed spraying strategies.

the weed and spray without going beyond the range of the arm. This point is given as $(x, -\sqrt{r^2 - x^2} + vt_{spray})$ where t_{spray} is the time needed to hover over the weed and release the pesticide. The time of intercept here is given as $t_1 = \frac{y - y_1}{v}$. Between t_0 and t_1 the intercept is valid. This range can then be used with the full trajectory generator to determine kinematic viability. Areas at the horizontal extremes of the circle have no valid intercept points meaning weeds crossing these points cannot be sprayed without a decrease in velocity. Higher velocities increase this area so limit the range of valid weeds.

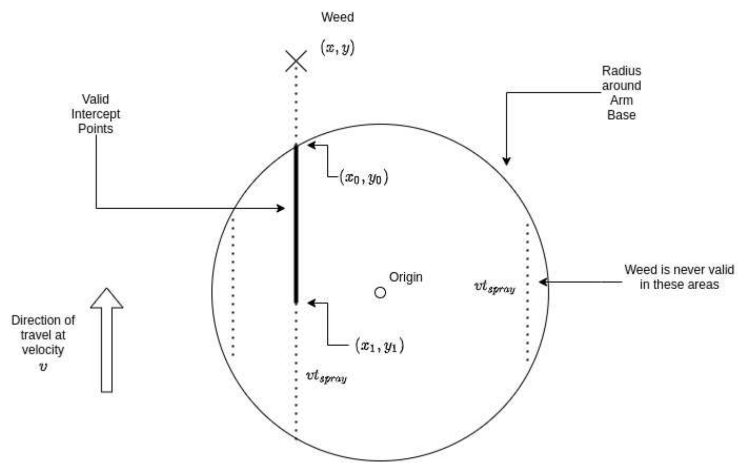

Fig. 4: How the range of valid intercepts is calculated based on a constant radius range. Note the areas at the extremes of the circle form areas where no weed can be sprayed. These areas grow with velocity v.

2.3 Middle Level - Weed Spray Ordering

The middle level takes the locations of all the weeds as input, and creates an ordering in which to visit them as output. This ordering is passed to the low level to generate a trajectory connecting them together. The spray ordering is decided in advance of the high level strategy.

Optimally visiting n weeds with distances between them is a Travelling Salesman Problem which is NP-hard. However, in addition to the distances, we have extra information about their metric positions in space, which simplifies the problem, so a heuristic based ordering algorithm can be utilised. The heuristic is based on the principle that weeds close together should be sprayed together. K-means clustering of the weeds in space is first performed to cluster close weeds. The *distortion* [16] property, α, is then measured as the average Euclidean distance between all weeds in each cluster and its respective centroid. A maximum distortion α_{max} is imposed, and the number of clusters (K) is increased until the maximum distortion requirement is satisfied.

The clusters are ordered by proximity to the robot base. Then within each cluster, the weeds are ordered again by proximity to the robot base. The final ordering is made by concatenating each of the intra-cluster orderings in the order of the inter-cluster ordering as in Algorithm 1.

Algorithm 1: K-Means weed ordering

```
Input weedArray - Array of Weed Poses, α_max - Maximum Distortion
Output weedOrder - Index of Weed Ordering
distortion = +inf;
k := 2;
while α ¿ α_max do
    clusters, α = kMeans(weedArray, k);
    k:=k+1;
end
orderedClusters = orderByX(clusters);
closestCluster = assign(weedArray, orderedClusters);
weedOrder = array();
for cluster in orderedClusters do
    weedsInCluster = closestCluster[cluster];
    weedOrder.append(orderByX(weedsInCluster));
end
return weedOrder;
```

2.4 High Level Spraying Strategies

The task of the high level is to plan the motion of the mobile robot base. It takes as input the mid level's target ordering, and it interacts with the low level's trajectory generation to links base motion plans to arm motion plans.

In general this is a hard problem, with an infinite possible search place of arbitrary, continuous, curved paths available to both the base and the arm. The present study thus considers only a limited subset of solutions which all constrain the base to drive in straight rows, but with the ability to control its velocity. We consider three strategies of increasing velocity control complexity within this directional constraint: stop-and-spray, constant velocity, and variable velocity, shown in Fig. 5.

Stop and Spray (SAS) This simple strategy as used by most existing systems so is considered as the baseline for our more advanced strategies. The base drives forward at a constant velocity until a weed enters the arm workspace. The base then slows to a halt. The arm then performs a point to point movement to position the effector above the weed. The spray is released for the required t_{spray} seconds. Any other weeds in the workspace are also be sprayed from the same stationary base location. Once all weeds in the workspace have been sprayed the

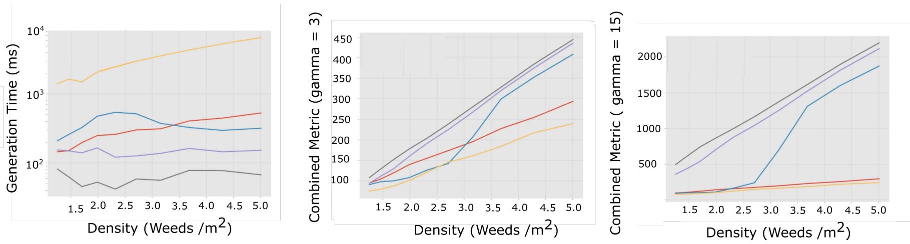

Fig. 5: Graph showing Performance on a 10m by 3m row of uniformally distrubuted weeds at varying densitites.

robot will continue at the given velocity. For the point to point movements, the above low level intercept trajectory is used. For finding the minimum movement time, the trajectory is recalculated at increasing time intervals until the joint limits are satisfied.

Constant Velocity (CV) In this strategy, the base drives forward at fixed speed, without stopping, and the arm moves to match the weed's position and relative velocity. This is more complex due to decisions on spray order and where and when to intercept the weeds as they pass through the arm's range of motion. As discussed in Section 2.3, firstly the spray order is determined. From there, the trajectory generation begins by determining the validity of spraying each weed in the order, skipping those that a viable trajectory cannot be found for. The robot begins to move at a constant velocity v and a trajectory is generated to match the first weed's velocity and acceleration at the earliest point of entering the reach of the arm. If this trajectory is not feasible then later intercept points within the arm range is considered until either a feasible trajectory is found or the weed cannot be sprayed. If a feasible trajectory is found then it is added to the plan; if not the weed is dismissed from the trajectory.

Variable Velocity (VV) In this strategy, the speed – but not the direction – of the mobile base is adapted during interactions with the low level trajectory system, to reduce the number of weeds missed at high densities, and to reduce execution times at low weed densities. The strategy is similar to the constant velocity strategy but with two velocity update rules based on constants N and SF. The first update rule is that to re-plan the previous N weed spray trajectories using the current velocity divided by the scaling factor SF. This is triggered by a weed being unable to be feasibly sprayed. The reason for going back and re-planning for the previous N weeds as well as the weed that was unable to be sprayed is that if a weed was unable to be sprayed then the intercept points will be relatively late in the range of possible intercepts. Re-planning allows these intercept points to be earlier and therefore improve the overall trajectory execution time. The second update rule is the choice to increase the robot velocity. This is triggered by $N + 1$ weeds being successfully sprayed in a row. Here the velocity is then multiplied by SF as long as it is below the maximum velocity v_{max} of the base. This also reduces execution time by allowing the robot base to move faster along the row at low densities of weeds.

3 Evaluation

To evaluate the methods, a 3D simulation was used, with a 10m by 3m row containing weeds placed at random locations drawn from a uniform distribution over the area, at a given parameterised density. The simulation, as shown in Fig. 6 and a Video Demo (https://youtu.be/sKV4pRKN5DA), models the pose of the sprayer arm and the position of upcoming weeds to the robot. For the maximum permitted distortion $a_m ax$ a value of 0.5m was used for the K-means spray ordering. For the Variable Velocity strategy, the two constants N and SF, 3 and 1.2 were used for their respective value after initial experimentation.

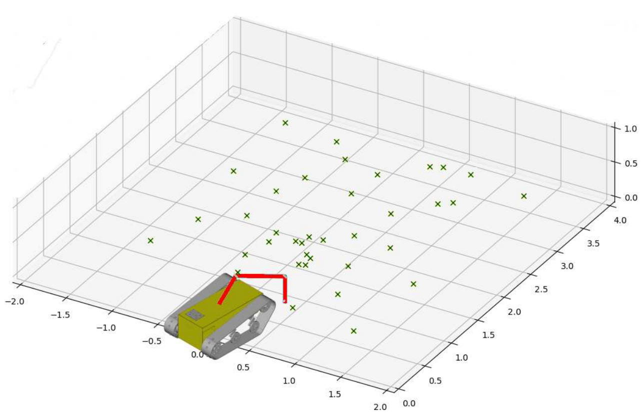

Fig. 6: Simulation setup.

Each strategy was evaluated in the simulation for a a range of densities from 0.1 to 5.0 weeds per m^2. Performance for each of these experiments was averaged over three runs with different random seeds. A Combined Metric is defined which weights the cost of time against the cost of missing weeds, $M(t_r, w_m) = t_r + \gamma w_m$, where t_r is the trajectory runtime and w_m is the amount of weeds missed. γ is a constant representing the cost of missing one weed measured in units of seconds (value of time). For example a value of $\gamma = 10$ would represent that is worth a extra 10 seconds of time in order to spray an extra weed. Lower values of the metric correspond to better strategy performance. Farmers currently pay around 10USD/hr=0.002USD/sec for human sprayers, who walk around 1.8m/s and hit around 95% of weeds. For a typical row density of 1 weed/meter, this costs 0.001USD/weed to spray, equivalent to $\gamma = 0.5$. Robots have a negligible cost of time compared to their human operators. If the human changes role to monitoring 6 robots on a small farm, then $\gamma = 3$; if they monitor 30 robots on a large farm then $\gamma = 15$.

Table 1: Generation time, execution time and weeds sprayed by weed density.

Density (weeds / m^2)	Strategy	Generation Time (ms)	Execution Time (ms)	Weeds Sprayed (%)
0.5	SAS	47.0	49.6	100
	CV v=0.1ms^{-1}	43.3	100.0	100
	CV v=0.2ms^{-1}	48.4	50.0	100
	CV v=0.5ms^{-1}	47.5	20.0	66.6
	VV	323.7	**30.9**	**100**
2	SAS	218.0	138.0	100
	CV v=0.1ms^{-1}	423.1	100.0	98.3
	CV v=0.2ms^{-1}	151.3	50.0	26.6
	CV v=0.5ms^{-1}	59.0	20.0	6.6
	VV	2058.9	**110.0**	**100**
5	SAS	455.8	288.7	100
	CV v=0.1ms^{-1}	290.5	100.0	16.0
	CV v=0.2ms^{-1}	155.1	50.0	4.6
	CV v=0.5ms^{-1}	75.4	20.0	2.6
	VV	6130.3	**241.4**	**100**

4 Discussion

It is possible to construct smooth low-level trajectories from polynomial segments with low computational costs (Fig. 6a), to fit to selected weed orderings from a clustering heuristic, and to combine these with the high level strategies.

Switching from the commonly used high-level stop and spray method to the more advanced constant velocity and variable velocity methods improves time and costs of weed spraying, when weighted with typical agricultural costs (Fig 6b,c) and may deliver similar gains in related tasks.

At low densities (0.5 weeds/m^2), performance between the strategies is similar, but as density increases, performance of both stop and spray and constant velocity strategies begins to drop off due to long execution times and poor spray success rates (Table 1). At higher densities, even at very low base velocities (0.1ms^{-1}), the constant velocity strategy struggles achieving just 16% of weeds sprayed successfully (Table 1). The variable velocity strategy however continues to do well, achieving 100% weed coverage (Table 1), though at the cost of slowing down and taking more time (Table 1). Stop and Spray and Constant Velocity strategies require much lower computational cost then the Variable velocity strategies (Fig. 6a). This is due to the re-planning phase of the variable velocity strategy that leads to a greater number of evaluations of the low level trajectory generator. For the large farm, $\gamma = 15$, the gap in performance between Stop and Spray and Variable Velocity reduces (Fig 6c.) representing that the improved execution times of the Variable Velocity strategy representing less importance in the overall system performance. For the small farm, $\gamma = 3$ (Fig 6b), the improvement is more noticeable.

References

1. T. Duckett, S. Pearson, S. Blackmore, B. Grieve, W.-H. Chen, G. Cielniak, J. Cleaversmith, J. Dai, S. Davis, C. Fox *et al.*, "Agricultural robotics: the future of robotic agriculture," *arXiv preprint arXiv:1806.06762*, 2018.
2. C. W. Bac, J. Hemming, B. Van Tuijl, R. Barth, E. Wais, and E. J. van Henten, "Performance evaluation of a harvesting robot for sweet pepper," *Journal of Field Robotics*, vol. 34, no. 6, pp. 1123–1139, 2017.
3. T. Mueller-Sim, M. Jenkins, J. Abel, and G. Kantor, "The robotanist: A ground-based agricultural robot for high-throughput crop phenotyping," in *IEEE International Conference on Robotics and Automation*, 2017.
4. R. Raja, T. T. Nguyen, D. C. Slaughter, and S. A. Fennimore, "Real-time weed-crop classification and localisation technique for robotic weed control in lettuce," *Biosystems Engineering*, vol. 192, pp. 257–274, 2020.
5. K. Sabanci and C. Aydin, "Smart robotic weed control system for sugar beet," *Journal of Agricultural Science and Technology*, vol. 19, no. 1, pp. 73–83, 2017.
6. J. J. H. Lee, K. Frey, R. Fitch, and S. Sukkarieh, "Fast path planning for precision weeding," in *Australasian Conference on Robotics and Automation, ACRA*, 2014.
7. Y. Xiong, Y. Ge, Y. Liang, and S. Blackmore, "Development of a prototype robot and fast path-planning algorithm for static laser weeding," *Computers and Electronics in Agriculture*, vol. 142, pp. 494–503, nov 2017.
8. R. Raja, T. T. Nguyen, D. C. Slaughter, and S. A. Fennimore, "Real-time robotic weed knife control system for tomato and lettuce based on geometric appearance of plant labels," *Biosystems Engineering*, vol. 194, pp. 152–164, jun 2020.
9. J. Martin, A. Ansuategi, I. Maurtua, A. Gutierrez, D. Obregón, O. Casquero, and M. Marcos, "A generic ros-based control architecture for pest inspection and treatment in greenhouses," *IEEE Access*, vol. 9, pp. 94 981–94 995, 2021.
10. J. Y. Kim, D. H. Kim, and S. R. Kim, "On-line minimum-time trajectory planning for industrial manipulators," *ICCAS 2007 - International Conference on Control, Automation and Systems*, pp. 36–40, 2007.
11. D. Zhao, B. Zhang, Y. Zhao, Q. Sun, C. Li, and C. Wang, "Research on motion planning for an indoor spray arm based on an improved potential field method," *PLOS ONE*, vol. 15, no. 1, p. e0226912, jan 2020.
12. T. Park and B. Lee, "Approach to robot motion analysis and planning for conveyor tracking," *IEEE Systems, Man, Cybernetics*, vol. 22, no. 2, pp. 378–384, 1992.
13. A. Binch and C. Fox, "Controlled comparison of machine vision algorithms for Rumex and Urtica detection in grassland," *Computers and Electronics in Agriculture*, vol. 140, pp. 123–138, 2017.
14. A. Binch, N. Cooke, and C. Fox, "Rumex and Urtica detection in grassland by UAV," in *Proc. International Conference on Precision Agriculture*, 2018.
15. K. Tokarz and S. Kieltyka, "Geometric approach to inverse kinematics for arm manipulator," *Int Conf on Systems*, vol. 1, no. 26, pp. 682–687, 2010.
16. D. T. Pham, S. S. Dimov, and C. D. Nguyen, "Selection of K in K-means clustering," *Proc Inst Mech Eng, C*, vol. 219, no. 1, pp. 103–119, 2005.