Title
Co-robotic harvest-aid platforms: Real-time control of picker lift heights to maximize harvesting efficiency

Authors
Fei, Zhenghao
Vougioukas, Stavros G

Publication Date
2021

DOI
10.1016/j.compag.2020.105894

Peer reviewed

Co-Robotic Harvest-aid Platforms: Real-time Control of Worker Lift Heights to Maximize Harvesting Efficiency

Zhenghao Fei (费铮豪)*, Stavros G. Vougioukas

University of California, Davis, Department of Biological and Agricultural Engineering, One Shields Ave, Davis, CA 95616-5270, USA

Abstract

Harvest-aid platforms are used in modern orchards to improve manual harvesting efficiency, safety, and ergonomics. Typically, workers stand at pre-set heights on a platform's multi-level deck, and each worker harvests fruits inside a canopy zone that is defined by the lowest and highest reach of the worker's arms. However, fruit distributions are non-uniform, and worker picking speeds vary, thus generating a mismatch between labor demand (incoming fruit rates) and labor supply (fruit picking rates) in each zone; this mismatch limits platform-based harvesting efficiencies. To alleviate this problem, we transformed a conventional harvesting platform into a collaborative robot (co-robot) platform. As the co-robotic platform travels forward, it estimates the incoming fruit distribution using a vision system, it measures each worker's picking speed using instrumented picking bags, and controls the heights of hydraulic lifts that move workers up and down. The model-based control algorithm maximizes the machine's harvesting speed by changing the height at which each worker harvests as a response to incoming fruit load because it matches fruit-picking labor supply and demand. Simulation experiments with pre-recorded fruit distribution data validated the approach and provided efficiency gains under various conditions. Apple-harvesting experiments were also performed in a commercial orchard, where 2,307 kg of apples were picked: 1,045kg in variable-height zone

harvesting mode, and 1,262 kg in fixed zone harvesting mode, with workers at fixed heights that were set by the grower. Variable-height zone harvesting mode throughput was 327.6 kg/h vs. 298.8 kg/h for fixed zone harvesting mode at human-controlled platform moving speed, resulting in an improvement of 9.5%.

Keywords:

Co-robotic harvesting, harvest platform, control, human-in-the-loop.

Nomenclature table

A^n	Valid action set for controlling n*th* worker's height
A	Combined actions set for controlling all workers
a_t^n	Height control action for n^{th} worker at timestep t
a_t	Combined height control action for all workers at timestep t.
C	Sparse sampling plan width
E	Sparse sampling extra search depth
f	System's dynamic function
g	Reward function
H	Sparse sampling plan (look-ahead) horizon
K	Sparse sampling extra search sample size
${}^{t}k_w^n$	n*th* worker's picking rate (fruits per second) at time step t. (fruits s^{-1})
${}^{t}\boldsymbol{k}_w$	All workers' picking rate (fruits per second) at time step t
l_x	Worker's reachable windows width (m)
l_y	Worker's reachable windows height (m)
${}^{t}M$	Fruit distribution map at time step t
N	The number of total workers on board
p^n	The n*th* worker
π	Worker height control policy
π^*	Optimal height control policy
$\pi(s)$	Action taken in state s under policy π
$q^{\pi}(s,a)$	Action value, the value of taking action a in state s under policy π

r_t	Reward - the number of picked fruits at time step *t*
γ	Reward discount rate
s_t	System state at time step *t*; it includes [${}^t x_p$, ${}^t v_p$, ${}^t \boldsymbol{y}_w$, ${}^t \boldsymbol{x}_w$, ${}^t \boldsymbol{k}_w$, ${}^t M$]
Δt	Timestep length (s)
$v^{\pi}(s)$	State value of state *s* under policy π
$v^{*}(s)$	Optimal state value of state *s* under optimal policy
$v_H^{*}(s)$	Estimation of the optimal state value using *H*-step expected discounted reward
${}^t v_p$	The platform's moving speed at time step *t* (m s^{-1})
${}^t w^n$	n^{th} worker's current picking window at time step *t*
${}^t x_p$	Platform's horizontal position along the tree row at time step *t* (m)
${}^t x_w^n$	n^{th} worker's horizontal position along the tree row at time step *t*
${}^t \boldsymbol{x}_w$	All workers' horizontal position along the tree row at time step *t*
x_c^n	n^{th} worker's horizontal offset from the platform's horizontal position ${}^t x_p$
${}^t y_w^n$	n^{th} worker's vertical position at time step *t* (m)
${}^t \boldsymbol{y}_w$	All workers' vertical position at time step *t*
Note: the time step *t* is neglected when presenting a symbol in a time invariant context.	

1 Introduction

Harvesting fruits for the fresh-market is a very labor-intensive and costly task (Zhang, 2017), because fruits must be picked carefully, to avoid damage, and selectively, based on marketing criteria. Growers in many countries face a great challenge, because they depend on a large seasonal semi-skilled immigrant workforce, which is becoming less available (Taylor and Charlton, 2018).

During commercial harvesting, workers pick the lower-hanging fruit by walking through the orchard rows, and use tall ladders to reach fruits located at higher parts of the canopies (Figure

1a). Then, they walk to bins that have been pre-positioned in appropriate locations in the orchard, to unload their fruit and resume picking (Figure 1b).

a) b)

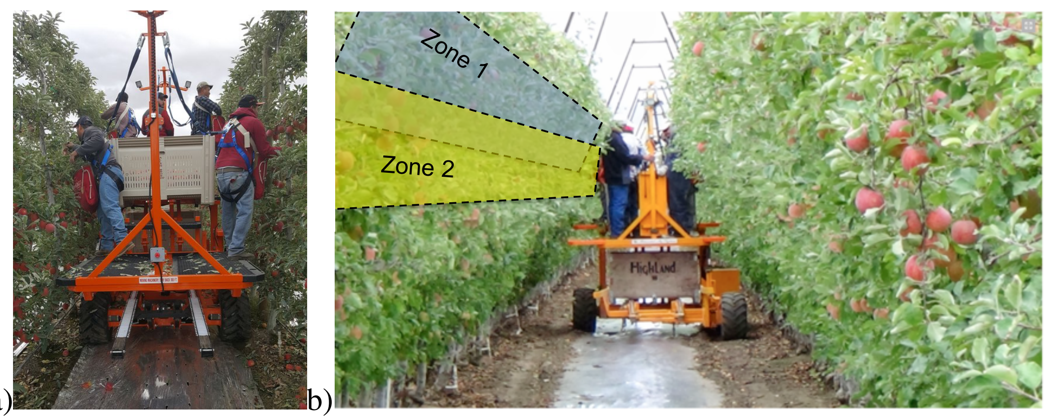
Figure 1 a) Workers use tall ladders to harvest fruit located at higher parts of the canopy. b) After harvesting a full bag, workers walk to bins and unload their fruit.

There are three main approaches to mechanizing the harvesting of fresh-market fruits (Zhang et al., 2016; Zhang et al., 2020). The first is mechanical mass-harvesting. There are three categories of mass-harvesting approaches. One is trunk shaking, where the trunk is shaken and fruits are caught with catching surfaces. This approach causes large numbers of fruits to detach in a very short time interval by using trunk shaking and catching(Peterson et al., 1999, Ortiz et al., 2013, De et al., 2015, He et al., 2017), The second category is the canopy shaking, where a number of rods enter the canopy and shake all together thus hitting the limbs and branches of the tree, and the fruits. (Peterson, 1982, Peterson and Kornecki, 1987, Peterson and Miller, 1989). The third category is the air jet method, where the air is blown to parts of the canopy or the entire canopy. (Thomas, 1964; Berlage, 1973). The mechanical mass-harvesting approach is efficient in fruit harvesting but also causing an unacceptable fruit damage rate from the apple-to-apple collision, apple-to-branch collision, and excessive apple movement (Zhang et al., 2016). In some cases may cause tree damage as well. Due to the high fruit damage rates, the mechanical mass-harvesting is used to harvest fruits for juice and other processed products, but not for fresh-market fruits.

The second approach is selective harvesting using a robotic system. Harvesting robots recognize, locate, and detach fruits individually (Zhang et al., 2016). Some harvesting robots that handle fruits gently, without impacting fruit quality have been demonstrated (e.g., Bac et al., 2014, Silwal et al., 2017, Bogue, 2020), but are still at a pre-commercial stage, mainly because of low cost-effectiveness (high harvest cost), low picking speed compared to the manual harvesting, and inability to harvest a wide range of tree architectures (Vougioukas, 2019).

The third approach - an intermediate, partial-mechanization approach between manual harvesting and fully mechanized harvesting - is *machine-aided harvesting* using harvest-aid platforms. These are self-propelled machines that are deployed in high-density orchards to reduce the amount of labor by increasing workers' harvest efficiencies (Berlage et al., 1972; Peterson, 2005, Lesser, 2008). Many platform variants have been developed and used in practice over the past decades; however, experimental results from their use are not that numerous. Peterson, Miller and Wolford (1996) reported 36 to 44% increase in picker productivity for a two-person platform designed for narrow inclined trellised apple trees. Peterson (2005) tested another harvest-aid platform with two elevated workers and two ground workers, who used two fruit conveyors to transfer the fruit into a collection bin. The harvest productivity increased by up to 22%, but the fruit conveyance system introduced unacceptably high damage rates. Such damage can be partially eliminated by improving the conveyor or the bin filler (Zhang et al., 2018). Besides harvesting, harvest-aid platforms can be used for many other orchard management tasks that involve using ladders, such as pruning and thinning (Sazo et al., 2010). Harvest-aid platforms with vacuum-based conveyance systems have also been introduced and shown to increase workers' efficiencies (Schupp et al., 2011, Zhang et al. 2014). The workers on such platforms insert picked fruits directly in vacuum tubes, and don't spend unproductive time filling and unloading bags.

Workers on modern commercial harvest-aid platform (see for example, Figure 2a) stand on the deck and pick, without having to climb on ladders or walk to bins, thus spending most of their time productively, picking fruit. Typically, the heights of a platform's decks are adjusted prior to harvest at different heights from the ground, thus resulting in each worker picking fruit from a canopy "zone" at a certain height (Figure 2b); the zones may overlap, depending on deck heights, and worker picking styles and reaching abilities.



a) b)

Figure 2 a) Example of harvest-aid platform (Bandit Xpress by Automated Ag, Moses Lake, WA); photo courtesy of Automated Ag. b) Zone-harvesting, from a platform with decks at multiple, pre-configured heights; each worker picks fruit from a zone of reachable fruit.

The performance of multi-worker platforms has been evaluated formally by several researchers. Results were mixed, as the machines' picking efficiencies, were found to be better, similar or worse than that of ladder-based picking, depending on the platform design, tree architecture and fruit load distribution. Berlage et al. (1972) reported that in a two-worker apple harvesting platform without a fruit-conveyance system, one of the workers' picking efficiency decreased 21.9% due to the imbalance of fruit load between the upper and lower region of the canopy. Peterson (2005) also reported that the overall harvest rate of the platform was limited by the slowest worker who caused the faster worker to be idle.

The main reason behind the inefficiency of fixed-height "zone harvesting" is that, as the platform moves forward, the rate of incoming fruit inside a zone may not match the picking rate of the worker(s) picking in that zone. In general, each worker will pick at a different speed, which may vary during the workday; also, fruits are distributed with non-uniform distributions (Figure 3).

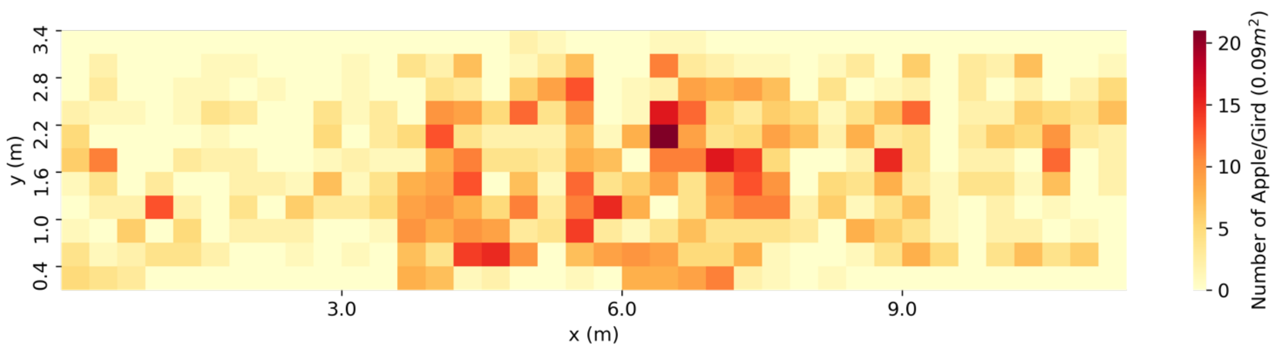


Figure 3 Example of a two-dimensional apple distribution in a high-density V-trellised orchard; data was collected using the method described in Arikapudi et al. (2016). The vertical axis corresponds to height from the ground, and the horizontal axis to distance along the row.

In the worst case, a slow worker harvesting a high-yield zone will act as a bottleneck that restricts the platform from moving forward faster, and will cause other (faster) workers to be idle. In general, fixed-height zone-harvesting will result in various degrees of imbalance between "labor supply" (picking rate) and "labor demand" (incoming fruit rate). The goal of this work is to reduce or eliminate such imbalances that lower the harvest throughput of multi-crew platforms and hinder their wider adoption.

1.1 Related work

Let us consider a number of “jobs” that must be performed by a group of “machines”, and that each job must run continuously on only one machine, and that each machine can perform at most one job at a time. If a subset of the jobs is assigned to a certain machine, the processing time of all the jobs on this machine is known as the *load* of the machine. To maximize the throughput or equivalently minimize the overall processing time of all the jobs requires an appropriate assignment of jobs to machines. Load-balancing algorithms are designed to equally spread the load on machines and maximize their utilization while minimizing the overall job processing time (Zomaya and Teh, 2001). When it is possible to make a priori estimate of work distribution the problem is called *static load balancing*. When jobs arrive continuously, and the amount of job is only known during actual program execution, the computation evolves different machines being responsible for the differing amount of job is called *dynamic load balancing* (Cybenko, 1989). Dynamic load-balancing is essential for the efficient use of highly parallel systems when solving non-uniform problems with unpredictable load estimates (Willebeek-LeMair and Reeves, 1993).

Load-balancing problems arise – and have been studied - in various scenarios. The most studied area of the load-balancing problem is in parallel and distributed computing systems. (Cybenko, 1989, Willebeek-LeMair and Reeves, 1993, Zomaya and Teh, 2001). Also, load-balancing was studied in other areas such as job shop scheduling to improve machine utilization and reduce the makespan (Ramasesh, 1990). In telecommunication networks, load-balancing construct call-routing schemes that distribute the changing load over the system and minimize lost calls. (Schoonderwoerd et al., 1997).

In agricultural robotics, load-balancing has received some attention, in the context of fruit harvesting with multi-arm robots. In particular, the assignment of individual fruits to arms has been studied for a simulated robotic melon harvester with multiple Cartesian arms (Zion et al., 2014, Mann et al., 2016). Fruit harvesting was modeled as a task of coloring an interval graph, and a greedy algorithm was used to compute optimal assignments; a heuristic algorithm was also developed to compute near-optimal solutions, in real-time, as the robot base moves forward. The same problem was addressed by Barnett et al. (2020) for multiple Cartesian robot arms harvesting kiwifruit, while the robot base is stationary. Fruits (jobs) were partitioned in groups with equal numbers of fruits, and each arm was assigned to a partition, for load-balancing purposes; fruits were picked according to the ascending order of their coordinate along the axis of harvester motion. A similar approach was pursued by Xiong et al. (2020) to load-balance the operation of two robot arms on a strawberry-harvesting robot.

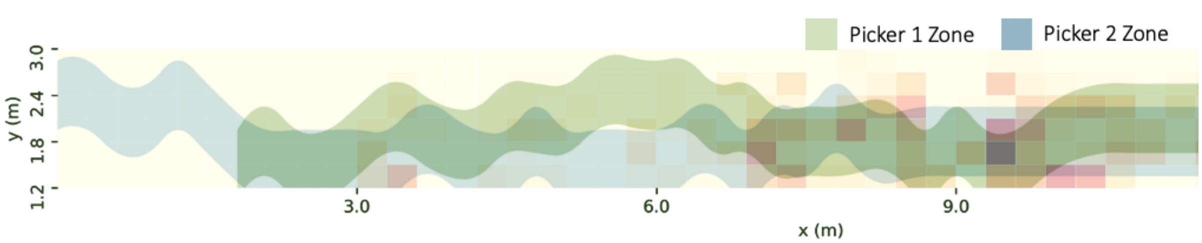
However, load-balancing the work of human workers differs drastically from the applications described above, because we can neither control the workers' picking motions nor assign individual fruits to workers. Hence, the load-balancing scenarios and approaches that were studied previously are not applicable to co-robotic platform-aided harvesting.

1.2 Proposed approach

This paper presents a novel human-robot collaborative approach that addresses the load-imbalance problem of multi-crew harvest-aid platforms by implementing *variable-height zone harvesting*.

Toward our goal, a commercial harvest-aid platform was retrofitted with: a) sensing systems that estimate the position and speed of the platform, the spatial distribution of the incoming fruit, and each worker's picking rate. b) two hydraulically-actuated lifts that control each worker's elevation from the ground (i.e., the height of each worker's picking zone) (Figure 4a). A model-based control system was developed to implement variable-height zone harvesting. The idea is that, as the platform moves forward in a row, instead of optimally assigning fruits to workers (which is impossible), the harvesting zone of each worker changes – in terms of its height from the ground – in a way that load-balances the canopy fruit load with the workers' picking speeds. Hence, in our system, a controller adjusts the lift – and corresponding zone - heights automatically, in real-time, to match each workers' picking rate with the incoming fruit load distribution (Figure 4b).

a) 



b)

Figure 4 a) The co-robotic platform system: an RTK-GNSS provides position and speed; a stereo-camera based vision system estimates a "heat map" of the incoming fruit distribution; instrumented picking bags measure each worker's picking rate; two hydraulic cylinders on one side of the platform move workers up and down. b) The concept of variable-height zone harvesting: A *model-based control system adjusts the lift – and corresponding zone - heights in real-time to match each workers' picking rate with the incoming fruit load distribution, and maximize the machine's picking throughput.*

The stochastic nature of the fruit distributions and worker picking rates led us to model co-robotic platform-aided harvesting as a Markov Decision Process (MDP) (Bellman, 1957), and utilize discrete stochastic optimal control to design a controller that maximizes fruit-picking throughput, a measure of the system's performance over time (Sutton and Barto, 1998). As it will be discussed in Section II, the state space of the MDP is very large, thus prohibiting exact solution approaches such as dynamic programming. Real-time optimization was made possible by computing a near-optimal solution based on a *sparse-sampling control* approach (Kearns et al., 2002), by randomly sampling a look-ahead tree with candidate actions within a time horizon.

The major contributions of this work are:

1) Modeling of the platform-based fruit-picking process, and development of a simulator to enable design, optimization, and evaluation of lift height control policies.

2) Development of a model-based optimal worker/lift height controller that achieves near-maximum picking efficiency.

3) Robotization of a commercial harvest-aid platform that features closed-loop feedback control of worker elevations. (Only two lifts were installed due to budgetary constraints, but the methodology for more lifts would be the same.)

4) Evaluation of the co-robotic platform in simulation, and in commercial harvesting, in an apple orchard.

The rest of the paper is organized as follows. In section II, harvesting with a co-robotic platform is modeled as a Markov Decision Process, and an approach called *sparse sampling* is adopted from the literature - and adapted to our problem instance - to compute in real-time, near-optimal height control actions. Section III presents the hardware and software system of the co-robotic harvest-aid platform. In section IV, we present simulation and real picking experimental results, in an apple orchard. Finally, section V summarizes the work, and discusses conclusions and future work.

2 Methodology

2.1 Problem definition

Our objective is to compute an optimal lift height control policy that maximizes the number of fruits picked by the workers on the machine per unit of time (throughput). To achieve the objective, we needed to develop an optimizing controller that dynamically adjusts the elevation of each worker's lift to match the spatial distribution of incoming fruits (labor demand) with the workers' picking rates (labor supply). The first step is to model the harvesting process, i.e., the process where two workers pick fruit while standing on the decks of two variable-height lifts, on a forward-moving platform. A CAD model of the co-robotic platform system is shown in Figure 5.

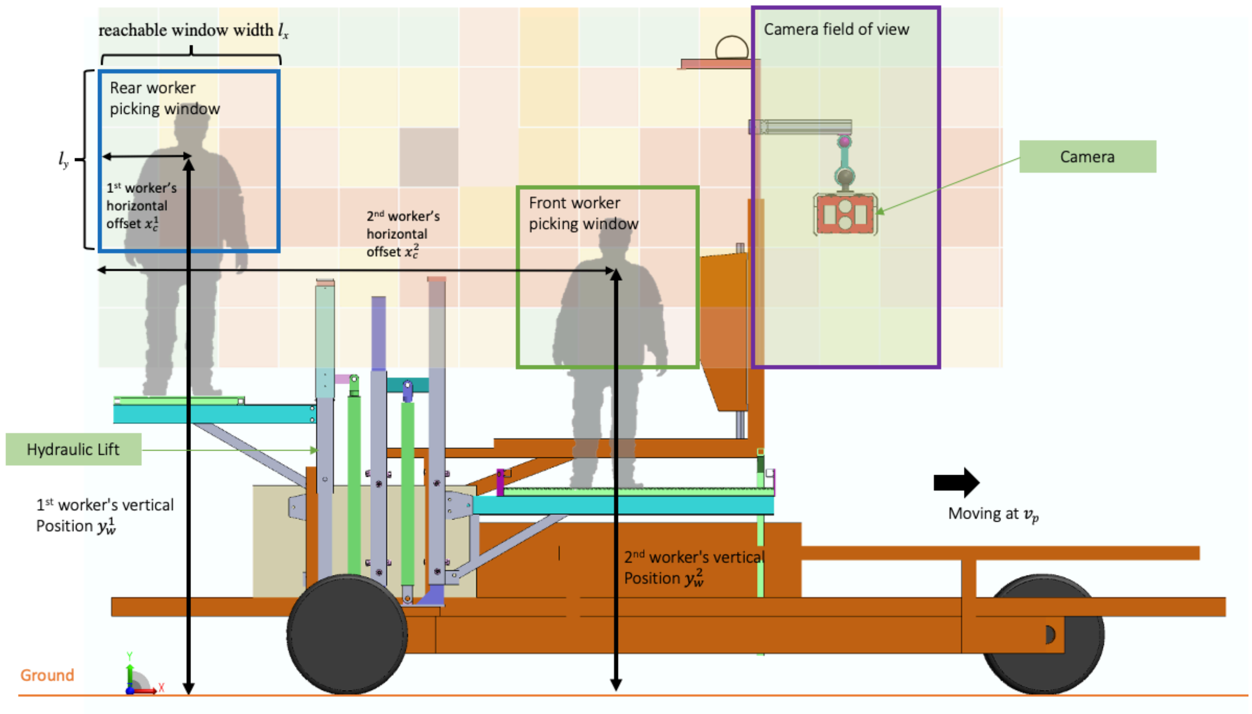


Figure 5 CAD model of the robotized platform system with variables definition; two hydraulic lift can move workers up and down; an RTK-GNSS installed on the top of the platform provides position and speed; a stereo-camera based vision system estimates a fruit distribution map of the incoming fruit distribution (colored heatmap).

The platform moves forward, inside the orchard row, at speed v_p. The camera system in front of the platform detects the incoming fruits – that are inside the camera's field of view - calculates their georeferenced coordinates, and generates a "fruit map", i.e., a heat map of fruit density (number of fruits in the canopy per unit of canopy surface area). The fruit map is represented as a regular two-dimensional grid, M. At any given lift elevation/height, the worker standing on the lift deck can reach fruits inside a rectangular picking window that has width l_x and height l_y, and is centered at the worker's current position (x_w, y_w). The fruits within a worker's picking window are picked at the worker's picking rate k_w. If there is no fruit inside the picking window, the worker is idle.

2.2 System modeling - Markov Decision Process Model

Markov Decision Processes (MDPs) offer a popular framework for modeling decision-making problems (Bellman 1957). In an MDP, at every time step *t*, the next state can be computed given the current system state s_t and the current action. A controller observes s_t and selects an action $a_t \in A(s_t)$, where $A(s_t)$ is the set of valid actions in s_t. Then, the system transitions to a new state s_{t+1} according to a state transition function (system's dynamic function) $f : s_t \times a_t \rightarrow s_{t+1}$ and receives a reward r_t based on a reward function $g: s_t \times a_t \rightarrow r_t$. An optimal control policy is one that results in the maximum expected cumulative reward.

2.2.1 Co-robotic platform system states

The state of the co-robotic platform is $s_t = \left[{}^t x_p, {}^t v_p, {}^t \boldsymbol{y}_w, {}^t \boldsymbol{x}_w, {}^t \boldsymbol{k}_w, {}^t M \right]$, as defined in the nomenclature table. The fruit distribution map *M* is an important and relatively complex part of the state. Each cell of the map represents a physical area of 0.3 m × 0.3 m on the surface of the fruiting wall; the value of each grid represents the number of fruits in its area. The values of the grid cells inside the camera's field of view are updated by the vision system. As the platform moves forward, a cell's value decreases, if a worker harvests/removes fruit from the canopy area corresponding to that cell. A state transition function will be presented to model the update of the cell values. An example *M* is shown in Figure 6.

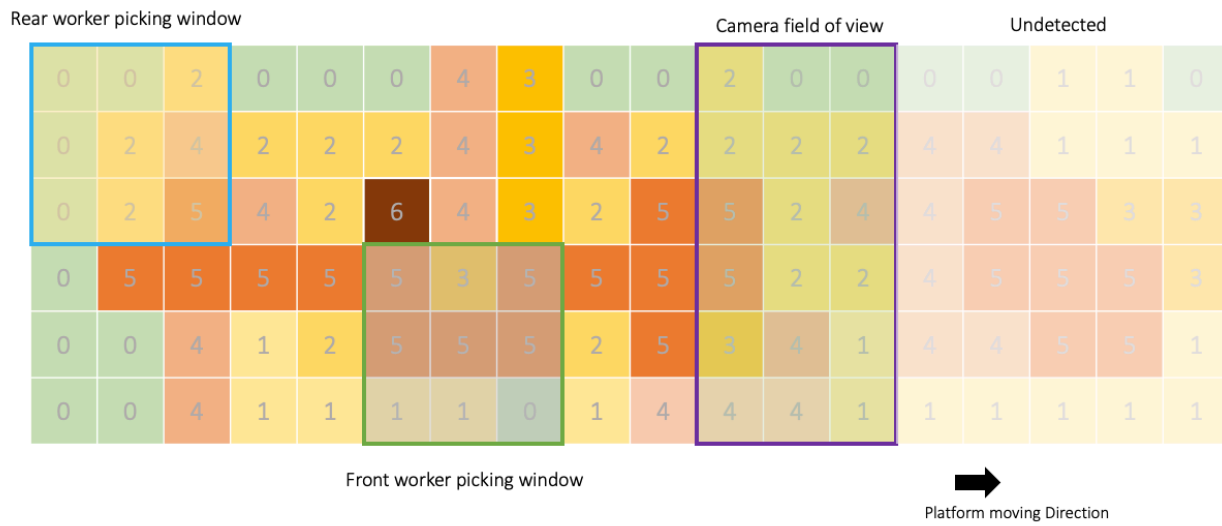


Figure 6 An example of a fruit map M. The width of the grid map is from the end of platform (the rear worker's reaching limit) to the camera's sensing range in front of the platform. The height of the grid map is the valid vertical picking range (from the lowest worker reaching limit when the lift is at lowest height to the highest worker reaching limit when the lift is at highest height). The purple box is the camera field of view. Green and blue boxes are workers' picking windows.

2.2.2 Model parameters

There are some MDP model parameters that are assumed to be constants. A worker's current horizontal position x_w^n is at a constant offset x_c^n (values for the offsets are listed in Table 1) relative to the platform's current position x_p. The assumption implies that each worker stands on their respective lift at a fixed position:

$$x_w^n = x_p + x_c^n \quad (1)$$

In reality, workers can reach the fruits left and right, each worker's picking window w^n is centered at the worker's current position (x_w^n, y_w^n) and its size is assumed to be fixed.

$$w^n = \left[x_w^n - \frac{l_x}{2} : x_w^n + \frac{l_x}{2}, y_w^n - \frac{l_y}{2} : y_w^n + \frac{l_y}{2}\right] \quad (2)$$

The ascending (v_{up}) and descending (v_{down}) speeds of the hydraulic cylinders are also constants (values are listed in Table 1).

2.2.3 Actions

For the n[th] worker, the valid action set A^n is $\{0, +1, -1\}$, where "0" encodes the "*keep_still*" action, "+1" the "*move*_up" action, and "-1" the "*move_down* action for all s_t, except when that worker is already at the top or bottom. If the worker is already at the top or bottom, the effect of the *move_up* or *move_down* action, respectively, will be the same as that of the *keep_still* action. The controller's full action set includes all the combinations of actions for two workers $A = A^1 \times A^2$.

2.2.4 State transition function - system dynamics

The system's state transition function $f\ : s_t \times a_t \rightarrow\ s_{t+1}$ defines how the state change after a time step, given an action a. The fruit map constitutes part of the state, and human picking patterns are unknown and unpredictable. Therefore, it is impossible to represent the state transition function with a closed-form equation. Next, we describe the transition/update equations for each component of the state. If each time step is Δt, the platform's horizontal position x_p is updated as:

$$ {}^{t+1}x_p = {}^{t}x_p + {}^{t}v_p \times \Delta t \tag{3} $$

The n^{th} worker's vertical position y_w^n follows

$$ {}^{t+1}y_w^n = {}^{t}y_w^n + v_{vert}^n \times \Delta t \tag{4} $$

$$ v_{vert}^n = v_{up} \text{ , if } {}^{t}a^n = +1 $$

$$ v_{vert}^n = v_{down} \text{ , if } {}^{t}a^n = \text{-}1 $$

$$ v_{vert}^n = 0 \text{ , if } {}^{t}a^n = 0 $$

The platform's moving speed v_p and each worker's picking rate k_w^n are measured by sensors. Based on our observations, these values do not change during the time period used in our work (Δt = 5 s), so, they are assumed to be constant during each step of the state transition function. The fruit distribution map *M* updated as follows. For each worker p^n, we refer to the maximum number of fruits the worker can pick during a time step Δt as the picking capacity ${}^t o^n$:

$$ {}^t o^n = {}^t k_w^n \times \Delta t \tag{5} $$

Given that a worker's fruit-picking sequence/strategy is not known, the assumption is made that the fruits in the nth worker's current picking window ${}^t w^n$ will be picked *randomly* within the timestep Δt. If there are enough (more than ${}^t o^n$) apples in this window, we subtract ${}^t o^n$ fruits randomly from *M's* corresponding picking window area. If the number of apples in the corresponding picking window area is less than ${}^t o^n$, we remove all the fruits.

$$ {}^{t+1}M[{}^t w^n] = \textbf{randomPick}({}^t M[{}^t w^n], {}^t o^n) \tag{6} $$

Function **randomPick** will try to remove up to ${}^t o^n$ fruits from ${}^t M$'s corresponding picking window area randomly.

2.2.5 Reward function

The reward function $g: s_t \times a_t \rightarrow r_t$ defines the reward signal r_t when action a_t is executed at state s_t. The way the reward is computed is explained next. When there are fewer than o^n fruits inside a worker's picking area, the worker will pick all the fruits and remain idle, until the platform takes her/him to a position where there is more fruit. Therefore, the number of picked fruits r_t^n by worker *n*, is always less or equal to the fruit picking capacity ($r_t^n \leq {}^t o^n$).

$$r_t^n = {}^t o^n \text{ , if } \mathbf{sum}({}^t M[{}^t w^n]) \geq {}^t o^n$$
$$r_t^n = \mathbf{sum}({}^t M[{}^t w^n]) \text{ , if } \mathbf{sum}({}^t M[{}^t w^n]) < {}^t o^n \tag{7}$$

The reward signal r_t, is the sum of all workers' picked fruits at time step t.

$$r_t = \Sigma_{n=0}^{n=N} r_t^n \tag{8}$$

2.2.6 Optimization objective

Our overall objective is to find an optimal height control policy π^* that maximizes the total number of picked fruits by all workers, from the time the platform enters the row ($t = t_0$) to the time it exits the row ($t = t_e$).

$$\pi^* = \underset{\pi}{\text{argmax}} \ \Sigma_{t=t_0}^{t=t_e} r_t \tag{9}$$

2.3 Optimization Algorithm

Following the definition in Sutton & Barto, (1998), we defined a policy $\pi(s)$ as a mapping from the current state, s, to an action a. A state-value $v^{\pi}(s)$ estimates "how good" it is to be in state s and follow policy π, in terms of future expected returns with a discount factor γ:

$$v^{\pi}(s) = \mathbf{E}\left[\Sigma_{i=0}^{\infty} \gamma^i r_i \middle| s, \pi\right] \tag{10}$$

Similarly, we defined the state-value for a state-action pair (s, a) under the policy π as an action-value, denoted $q^{\pi}(s, a)$; the action value estimates the future expected returns starting from the state, s, taking action a, and following the policy π thereafter.

$$q^{\pi}(s, a) = \mathbf{E}[r_t + \gamma v^{\pi}(s_{t+1}) | s, a, \pi] \tag{11}$$

In an MDP, there is always at least one policy that is better or equal to all other policies. Its expected return is greater or equal to all other policies' at all states ($\pi(s) \geq \pi'(s)$ if and only if

$v^{\pi}(s) \geq v^{\pi'}(s)$ for all $s \in S$). This policy is called the optimal policy π^*, and the state-value of π^* is called the optimal state-value $v^*(s)$.

$$v^*(s) = \max_{\pi} v_{\pi}(s) \text{ for all } s \in S \tag{12}$$

The action-value of π^* is called optimal action-value $q^*(s, a)$.

$$q^*(s, a) = \max_{\pi} q_{\pi}(s, a) \text{ for all } s \in S, a \in A(S) \tag{13}$$

We can also rewrite the optimal action-value as the expected return for taking action a in state s and follow the optimal policy π^* afterwards.

$$q^*(s, a) = \mathbf{E}[r_t + \gamma v^*(s_{t+1})|s, a] \tag{14}$$

In a reverse way, once we have an optimal action-value $q^*(s, a)$ we can obtain an optimal policy:

$$\pi^* = \arg\max_{a} q^*(s, a) \tag{15}$$

The MDP that models co-robotic platform-based harvesting has a huge state space. Although the state variables are discrete, the number of possible states is huge because of the large number of cells in the fruit map and the randomness in the fruit-picking sequence. Hence, the "curse of dimensionality" renders exact optimization methods, such as dynamic programming, impractical. One approach that deals with MDPs with a finite discrete action space and a huge state space is the *sparse sampling* algorithm proposed by Kearns et al. (2002). We used this algorithm to solve the proposed MDP. This approach is a model-based online planning approach that uses an existing generative model, i.e., a model that returns a randomly-sampled next state s_{t+1} and corresponding reward r_t, given as input any state-action pair (s_t, a_t). Our system's state-transition function f and reward function g can serve as a generative model. Given any state, instead of computing all the possible next states and rewards, this algorithm draws samples using a generative model for state-action pairs, step by step. Thus the running time required to compute a near-optimal action is not related to the size of the state space, and depends only on the size of the action space $A(S)$, the planning horizon H, and the width C. Specifically, the planning horizon H is the look-ahead depth of the state expansion tree, and the width C is the number of next-state samples that are generated for each state-action pair, as shown in Figure 7.

The main idea of the sparse sampling algorithm is to estimate the optimal q-values of the current state s_t for all the possible actions and determine a policy using $\pi^* = \arg\max_{a_t} q^*(s_t, a_t)$.

Because of the duality between the state-value and action-value, estimating the action-value can be done through estimating the state-value. Furthermore, $v^*(s)$ can be approximated by the H-step value $v_H^*(s)$:

$$v_H^*(s_t) = \boldsymbol{E}\left[\sum_{i=0}^{H} \gamma^i r_i \middle| s_t, \pi^*\right] = \boldsymbol{E}[r(s_t, a^*) + \gamma v_{H-1}^*(s_{t+1})] \tag{16}$$

Kearns et al. (2002) proved that the error of this approximation can be made controllably small by choosing H sufficiently large. The optimal value of the current state for H horizons can be approximated as:

$$\hat{v}_H^*(s_t) \approx \max_{a_t}(\boldsymbol{E}[r(s_t, a_t) + \gamma \hat{v}_{H-1}^*(s_{t+1})]) \tag{17}$$

The expectation in Eq. 17 will be approximated by a sample mean using C samples. Now, we can recursively obtain an estimate of $\hat{v}_{\mathrm{H}}^*(s_t)$ by using the estimate of $\hat{v}_{H-1}^*(s_{t+1})$. For mathematical simplicity, the original paper uses $\hat{v}_0^*(s) = 0$ for all s; however, doing so, ignores the policy's effect beyond H steps, and effectively limits the planning horizon. Instead, in this work, we modify the algorithm by using a random policy to rollout E extra steps from state s, and record the cumulative discounted return. We repeat this random E-step rollout K times, and use the average cumulative discounted return as an approximation of $\hat{v}_0^*(s)$.

$$\hat{v}_0^*(s_t) = \frac{1}{K}\sum_{j=1}^{K}\left(\sum_{i=1}^{E}\left(\gamma^i r_i \middle| s_t, \pi^{random}\right)\right) \tag{18}$$

In practice, the planning horizon is chosen according to the available computational resources. We will analyze the effect of different planning horizons in later experiments. An example of a planning tree of this algorithm is shown in Figure 7, and the pseudo-code is given in Algorithm 1-4.

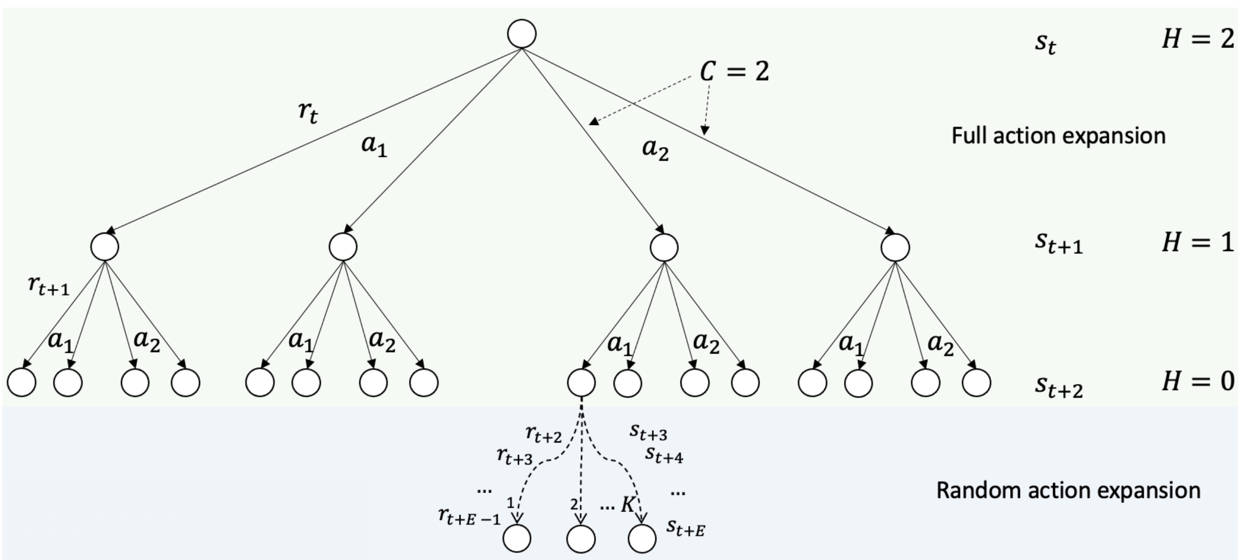


Figure 7 Planning tree with action number A(s)=2, sample size C=2 and planning horizon H=2

Algorithm 1 EstimateQ

Input: $s_t, C, H, E, K, \gamma, f, g$
Output: estimation of optimal action-value for all actions at s_t: $\hat{q}_H^*(s_t, a_i)$
for each $a_i \in A(s_t)$ **do**
  Use state transition function f to generate C next state samples and get their corresponding rewards using reward function g. Let S_{t+1} be the set of the generated next states and R_t be the set of corresponding rewards .

$$\hat{q}_H^*(s_t, a_i) = \frac{1}{C}\left(\sum_{r_t \in R_t} r_t + \gamma \sum_{s_{t+1} \in S_{t+1}} \textbf{EstimateV}(s_{t+1}, C, H-1, E, K, \gamma, f)\right)$$

end for
return $[\hat{q}_H^*(s_t, a_i)\ for\ a_i \in A(s_t)\ ]$

Algorithm 2 EstimateV

Input: $s_t, C, H, E, K, \gamma, f, g$
Output: estimation of optimal H-step value function at s_t: $\hat{v}_H^*(s_t)$
if $H = 0$ and $E > 0$ **then**
  return RandomPolicyValue(s_t, E, K, γ, f, g)
else if $H = 0$ and $E = 0$ then
  return 0
else
$\hat{q}_H^*(s_t, a_i)$ for $a_i \in A(s_t) =$ **EstimateQ**($s_t, C, H, E, K, \gamma, f, g$)

return $max_{a_i \in A(s_t)} \hat{q}_H^*(s_t, a_i)$

Algorithm 3 RandomPolicyValue

Input: s_t, E, K, γ, f, g
Output: $\hat{v}_0^*(s_t)$
Use random policy π^{random} to control the system from s_t for E steps. Repeat this procedure K times.
$\boldsymbol{v} = 0$
for $k = 0$ to K-1 **do**
$\boldsymbol{s} = \boldsymbol{s_t}$
$v_k = 0$
  for $m = 0$ to E -1 **do**
    $\boldsymbol{a} = \pi_h{}^{random}(s)$
    $\boldsymbol{r}, \boldsymbol{s} = \boldsymbol{g}(\boldsymbol{s}, \boldsymbol{a}), \boldsymbol{f}(\boldsymbol{s}, \boldsymbol{a})$
    $\boldsymbol{v_k} += \gamma^m * \boldsymbol{r}$
  end for
$\boldsymbol{v} += \boldsymbol{v_k}$
end for
$\hat{v}_0^*(s_t) = \frac{v}{K}$
return $\hat{v}_0^*(s_t)$

Algorithm 4 SelectAction

Input: $s_t, C, H, E, K, \gamma, f, g$
Output: optimal action a

$\hat{q}_H^*(s_t, a_i)$ for $a_i \in A(s_t) =$ **EstimateQ**($s_t, C, H, E, K, \gamma, f, g$)

return $argmax_{a_i \in A(s_t)} \hat{q}_H^*(s_t, a_i)$

3 Materials

3.1 Robotized co-robotic platform

A commercial harvest-aid platform (Bandit Xpress by Automated Ag Systems, Moses Lake, WA) was "robotized" to work in variable-height zone harvesting mode. Hydraulic cylinders were installed to move workers up and down, based on commands from the central control computer. Guard rails and hooks for safety harnesses were installed to ensure worker safety. Workers were equipped with instrumented picking bags (Fei et al., 2017), which measure the worker's picking rate in real-time and transmit the data to the central computer wirelessly, via a Digi Xbee® RF module. The picking bags measure the picking rate in kg s^{-1}. The picking rate numbers were converted into fruits s^{-1} using the average fruit weight, which was estimated prior to the experiments by measuring the weight of one hundred random fruits from a bin. An RTK-GNSS (Real Time Kinematic - Global Navigation Satellite System) with cm-level localization accuracy was installed on the top of the platform to estimate the platform moving speed and the relative position between workers and incoming fruits. A stereo camera system was mounted in the front of the platform, and vision software was used to detect and localize the incoming fruits (Pothen and Nuske, 2016). The robotized platform is shown in Figure 8. Parameter values of the system are listed in Table 1.

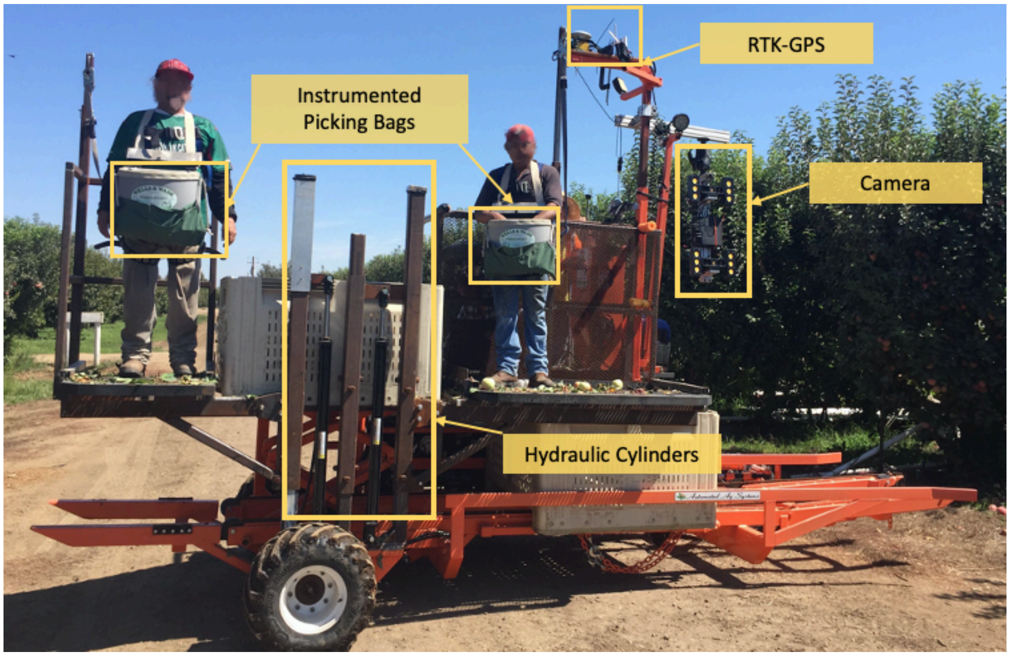


Figure 8 Sensors (Instrumented picking bag, RTK-GNSS, Camera) and actuators (Hydraulic cylinders) on the robotized platform with workers.

System parameters:

Number of workers: N = 2

Lowest lift deck height from the ground: 1.1 m

Highest lift deck height from the ground: 2.0 m

Worker's vertical position to standing deck offset: 0.95 m

Worker's lowest reaching height, at lowest lift height: 1.6 m

Worker's highest reaching height, at highest lift height: 3.4 m

Hydraulic lift ascending speed: $v_{up} = 0.037\ m/s$

Hydraulic lift descending speed: $v_{down} = 0.074\ m/s$

Worker horizontal offset from the rear end of platform: $x_c^1 = 0.45m, x_c^2 = 2.55m$

Camera horizontal offset from the rear end of platform: 4.05 m

Worker's reachable windows size: $l_x = 0.9\ m, l_y = 0.9\ m$

Camera horizontal field of view: 0.9 m
Time step: $\Delta t = 5s$
Reward discount rate: $\gamma = 0.99$

Table 1 Robotized platform system parameters

3.2 Software system overview

The software is implemented using the Robot Operating System (ROS) (Quigley et al., 2009). Each functional module constitutes a node in the ROS network. The nodes are listed as follows:

a) Lift-control node: This node executes on an Arduino microcontroller, which interfaces to the hydraulic cylinders of the platform that controls the worker heights. The microcontroller is connected to a central computer through a USB serial.

b) Instrumented picking bag nodes: Each instrumented bag has an Arduino microcontroller, and a node executes on it. The node calculates picking rates and sends the data to the central computer via Digi Xbee®. When the worker fills a bag and wants to unload, an unloading button on the bag is pressed, and a command is sent to the lift control node to lower the worker's lift to the elevation of the bin's lift height.

c) Camera node: The node executes on a laptop computer connected to the stereo camera. The node estimates the incoming fruit density from the image stream in real-time using an algorithm developed by Pothen and Nuske (2016), and sends the incoming fruit density map to the central computer through an Ethernet connection. The performance metrics of the vision system's fruit detection in the specific orchard were: precision 0.923, and recall 0.91.

d) Worker height optimization node: This node implements the model and optimizing control algorithm (sections 2.2 and 2,3) and runs on the central computer. It uses all the sensor information to compute optimal heights for the worker in real-time – using the sparse sampling algorithm - and sends the commands to the lift-control node.

e) RTK-GNSS node: This node publishes the current location and speed of the platform. The overall co-robotic platform software structure is shown in Figure 9.

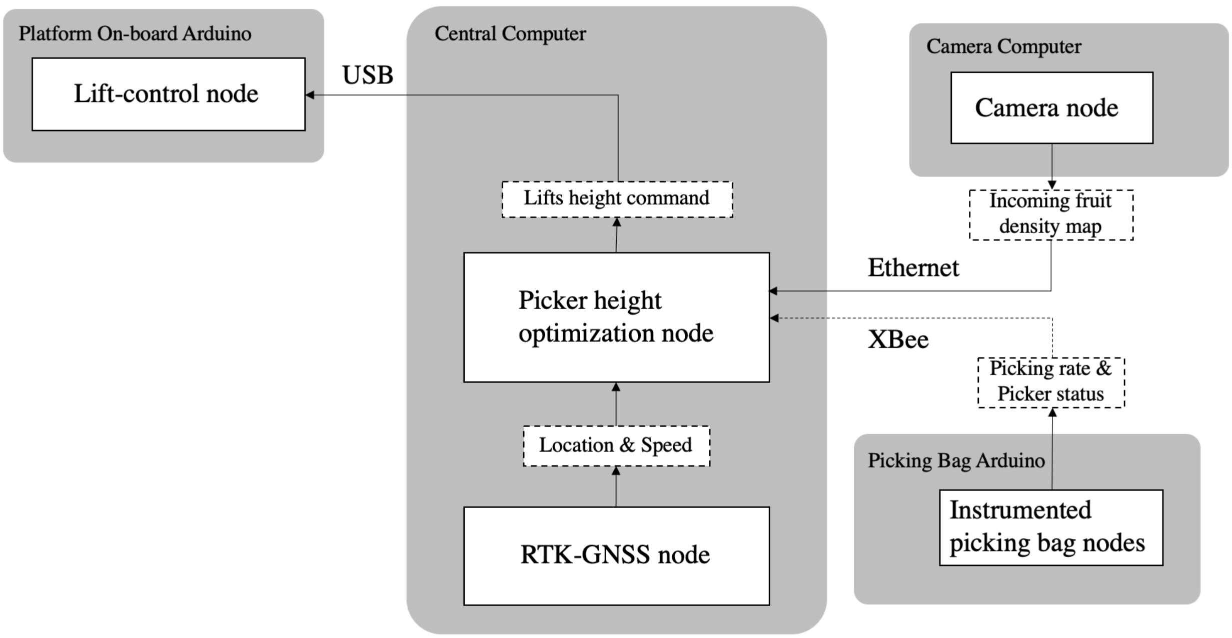


Figure 9 Computational system architecture for the co-robotic harvest-aid platform

4 Experimental design

4.1 Simulation experiments

Simulation experiments were performed using digitized fruit distribution data. In these experiments, we analyzed the efficiency gains of the optimized height control policy under different platform moving speeds v_p, picking rates k_p, and fruit distributions.

4.1.1 Evaluation metric for simulation experiments

We use *PP* (Picking Percentage) as our evaluation metric to evaluate the performance of a policy. The metric is defined as:

$$PP = \frac{Number\ of\ picked\ apples}{Total\ number\ of\ apples} \tag{19}$$

We compare the *PP* of different policies on the same fruit distribution, so the total number of apples is constant for a specific fruit distribution. Hence, maximizing *PP* is equivalent to maximizing the number of picked apples, which is the same as the optimizing objective of the sparse sampling algorithm.

In all simulations, the experiment starts when the front worker reaches the evaluation area and ends when the rear worker leaves the evaluation area. The *PP* is calculated using the number of apples inside the evaluation area before and after picking. For example, in Figure 10, this area has *x*-coordinate between [3m, 14m].

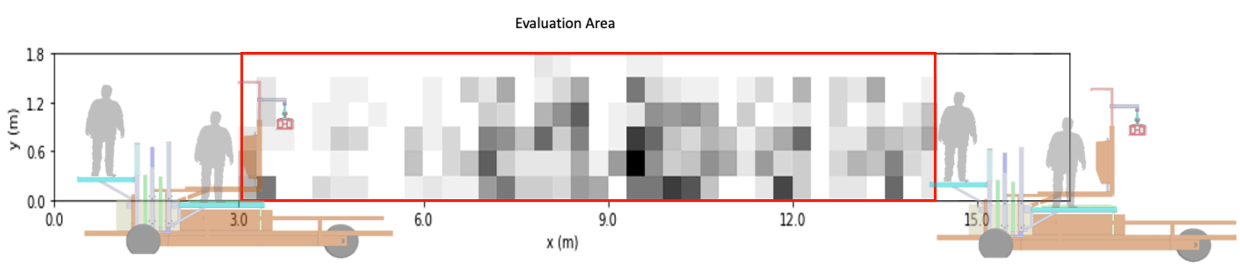


*Figure 10 An example evaluation grid map, each grid represents a 0.3m*0.3m physical area. The darkness indicates fruit density. The x-coordinate between [3m, 14m] is the evaluation area which PP is calculated.*

4.1.2 Baseline height control policies

4.1.2.1 Fixed height policy – fixed-height zone harvesting

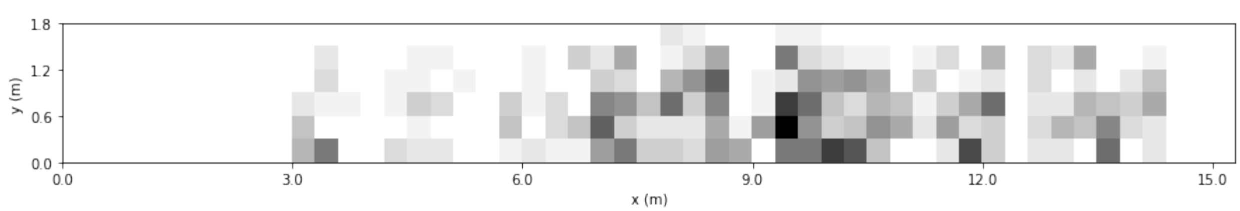
In fixed-height zone-harvesting, worker elevations are pre-configured and fixed. One worker is assigned to pick the bottom half of the fruit, and the other worker is assigned to pick the top half. We call this policy the *fixed height policy* and use it as a baseline in our experiments.

4.1.2.2 Random Policy

The second baseline policy we use for comparison is random policy. A random policy selects an action from the valid action set {0, -1, +1} randomly, with a uniform distribution, for each worker in every time step. This baseline is used to compare the performance of a random, uninformed height policy, against the proposed near-optimal policy.

4.1.3 Digitized – real fruit distributions for simulation experiments

In order to evaluate the improvement of the sparse sampling policy over the other policies, we collected real apple distribution data in a commercial apple orchard by logging every fruit's 3D location. The apple coordinates in two rows were measured manually using the method proposed by Arikapudi et al. (2016); the first row is 11.43 m long, and the second row is 12.61 m long; the fruit coordinates were digitized into a grid map using the parameters described in Table 1. The digitized fruit distributions are shown in Figure 11.



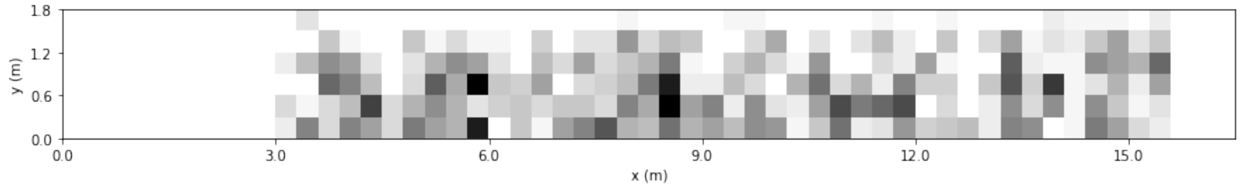


Figure 11 The top fruit grid map is real fruit distributions 1, the density of apples along the x-axis is 58.5 fruits per meter. The bottom grid map is real fruit distributions 2, the density of fruit along the x-axis is 109.2 apples per meter.

4.1.4 Experiments with varying platform moving speed

The optimization algorithm does not control the platform's moving speed; it tries to achieve maximum *PP* under any platform moving speed. Increasing the platform speed can decrease the worker's idle time and shorten the time needed to traverse a row, but it is expected to result in more unpicked fruits (lower *PP*). Also, the potential efficiency gain of using a good height control policy over a bad one varies with the changing of the platform moving speed (if the platform moved extremely slowly, workers could easily pick all the fruits on the tree and achieve 100% *PP*; if the platform moved extremely fast, all workers would pick at full capacity and leave no room for optimization).

To study the potential efficiency gain of using the sparse sampling policy over the baseline policies, and the tradeoff between the *PP* and the platform moving speed, we varied the platform moving speed from 0.003 m s^{-1} to 0.03m s^{-1} (by 0.003m s^{-1} increments) and compared the performance of all the height control policies on the digitized fruit distributions. We tested three picking speed combinations: 1) equal picking speed for each worker, at 0.5 fruits s^{-1}; 2) the top worker picking at 0.4 fruits s^{-1} and the bottom worker at 0.6 fruits s^{-1}.; 3) the top worker picking at 0.6 fruits s^{-1} and the bottom worker at 0.4 fruits s^{-1}. All picking speed combinations had the same total picking rate of 1.0 fruit s^{-1}.

4.1.5 Planning horizon study

The sparse sampling algorithm is designed to provide a near-optimal policy and is only guaranteed to converge to optimal when the planning (look-ahead) horizon H and width C are large enough (Kearns et al. 2002). However, in practice, the computational resources are limited, and we need to compute an action within one timestep (5 s); hence, H and C cannot be arbitrarily large. The algorithm searches over the entire action space up to horizon H, for C samples at each action, so the computational complexity of the sparse sampling method is $O((LC)^H)$ which is exponential to the planning horizon (L is the number of possible actions, and is equal to 9, in our case). A larger H means the algorithm plans further in the future, and a large C can approximate a better expectation value with the sample mean. The effect of using an extra search depth E is that after expanding the full state-action for H steps, the algorithm will continue to search for an extra E depth for K times, using a random policy. So, E can be considered as a computationally low-cost way to enlarge the planning horizon; We refer to $H+E$ as the *effective planning horizon*. In this planning horizon study, we varied H from 1 to 5 and varied extra E from 0 to 5 to study the effects of the H and E to the optimization results (PP) to help us understand how to select them. The planning horizon study experiments were done in the digitized fruit distributions row1. In all these experiments, simulated workers had the same picking rate of 0.6 fruits s^{-1}, and the simulated platform moved at a constant speed of 0.01 m s^{-1}. The planning width C and the extra search sample size K were set equal to 5.

4.2 Field experiments

The performance of the algorithm, and of the overall co-robotic harvesting system was evaluated in apple harvesting experiments that were conducted in a commercial Fuji apple orchard with V-trellised trees in Lodi, CA, in September 2018. A total of 2307 kg of apples were picked during the experiments. Among them, 1045kg were picked in variable-height zone harvesting mode (using sparse sampling height control policy), and 1262 kg were picked in fixed-height zone harvesting mode. Two experienced workers worked on the platform during all experiments (Figure 12). The optimizer parameters used in the field experiment were *H=2, E=5, K=5*, and *C*=2.

Figure 12 Harvesting experiments were conducted in a commercial Fuji apple orchard, with V-trellised trees in Lodi, CA, in September 2018.

Unlike simulation experiments, harvesting the exact same fruit row multiple times with different parameters is not possible in real orchards. Thus, one cannot generate -and compare - platform speed vs. *PP* plots for the different height control policies. Also, the limited time window for harvesting limited the amount of data we could collect. We compared the variable-height zone harvesting mode against the fixed-height zone harvesting mode, using two different platform speed control modes: fixed speed, and variable/adaptive speed, controlled by an operator. The details of the speed control policies are explained in section 4.2.3.

4.2.1 Evaluation metric

Unlike the simulation experiments, accurate estimates of the numbers of apples on the trees before and after picking were not available. So, PP could not be used as a metric in real apple picking experiments. Instead, we used throughput (kg s^{-1}) as the evaluation metric.

$$throughput = \frac{sum\ of\ picked\ apple\ weight\ (kg)}{sum\ of\ effective\ picking\ time\ (s)} \tag{20}$$

The platform's moving speed affects the throughput, so only the throughputs achieved under the same platform speed control mode can be compared against each other. The optimizer's objective, which maximizes the total picked fruit, is equivalent to maximizing the throughput at constant moving speed. The total weight of picked apple was calculated by segmenting each bag's signal (to identify individual bag-fill cycles) and adding the weights of each individual bag. Outliers in the data (caused by bag or other system temporary failures) were discarded, i.e., only the weights of "valid bag loads" were used. The effective picking time was the sum of the time used to fill the valid bag loads; the time spent on unloading apples into the collection bin was not included.

4.2.2 Height control modes

4.2.2.1 Fixed-height zone harvesting

The baseline fixed height mode used in the field experiment was same as the fixed height policy described in section 4.1.2.1. Worker lift elevations were pre-set by the grower, based on his experience.

4.2.2.2 Variable-height zone harvesting

The variable-height zone harvesting in the field experiment was implemented by using the sparse sampling policy proposed in this work.

4.2.3 Speed control modes for the platform

In commercial harvesting, the platform's travel speed is controlled by a worker in the front. The goal is to move the platform as fast as possible, while not leaving any – or many - apples unpicked. We refer to this speed-control mode as *adaptive speed* mode and used it to compare the picking performances of variable-height zone harvesting mode and fixed-height zone harvesting mode. In addition to the adaptive speed control, we used two *fixed moving speeds* ($v_p = 0.015\ m\ s^{-1}$, and $v_p = 0.03\ m\ s^{-1}$), to evaluate the improvement of the optimized height control policy. These two speeds were selected based on the simulation results and the actual platform moving speeds observed in the orchard.

4.2.4 Experiment settings

We compared the sparse sampling optimized policy against the fixed height policy under fixed and adaptive platform speed policies. Table 2 shows all possible combinations for speed policy and height control policy. The entry of the adaptive speed and fixed height control policy (set #1) corresponds to the conventional picking process during commercial harvesting.

Parameter set	**0**	**1**	**2**	**3**	**4**	**5**
Speed policy	Adaptive	Adaptive	Fixed 0.03 m s^{-1}	Fixed 0.03 m s^{-1}	Fixed 0.015 m s^{-1}	Fixed 0.015 m s^{-1}
Height control policy	Fixed-height zone harvesting	Variable-height zone harvesting	Fixed-height zone harvesting	Variable-height zone harvesting	Fixed-height zone harvesting	Variable-height zone harvesting

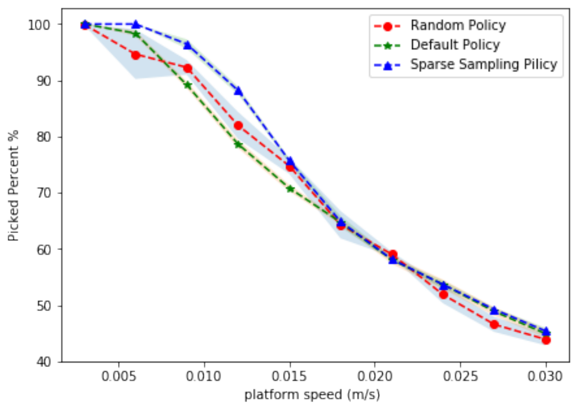
Table 2 Apple harvesting experiment settings: combinations of speed policy and height control policy.

5 Experimental results

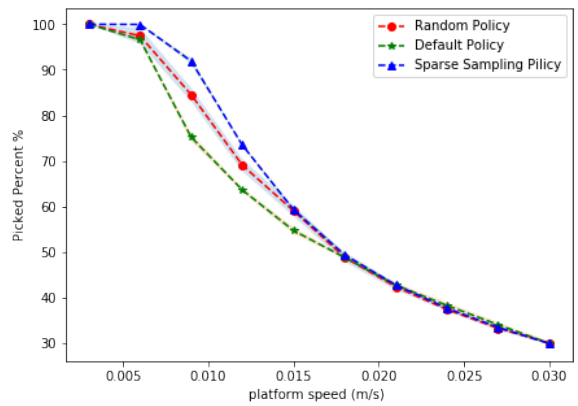
5.1.1 Simulation results

5.1.1.1 Experiments with varying platform moving speed

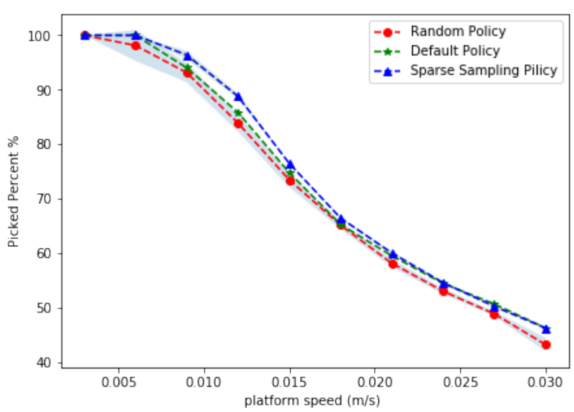
The experiment results for three picking-rate settings on two digitized real fruit distributions with varying platform moving speeds are shown in Figure 13.



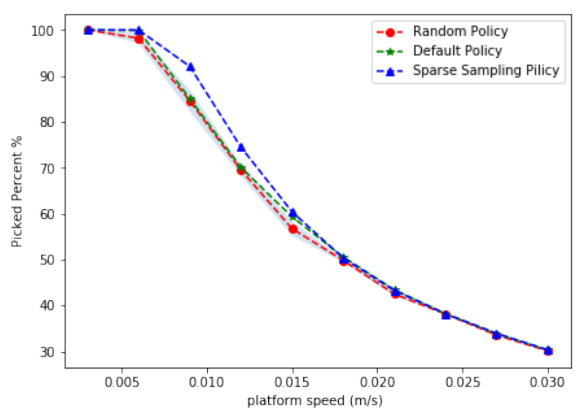
a



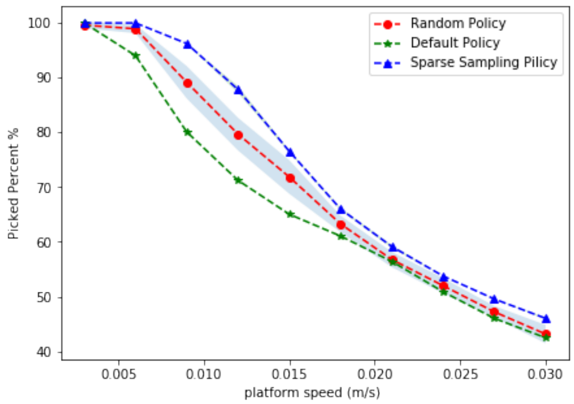
b



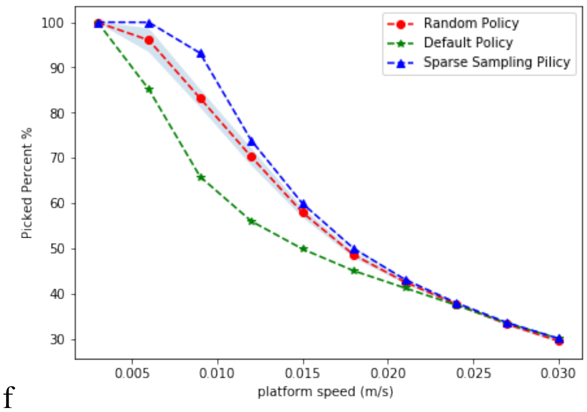
c



d



e



f

Figure 13 The picking percentage with respect to the platform moving speed for three height control policies. a, c, e show the results of experiments on the real fruit distributions 1 and b, d, f show the results of experiments on the real fruit distributions 2; a, b show the experiment results under *picking rate setting 1);* c, d show the experiment results under *picking rate setting 2);* e, f show the experiment results under *picking rate setting 3).*

The results show that in the low-speed range, all policies achieve almost 100% *PP*; in the high-speed range, all policies were limited by the workers' picking speeds. In the middle-speed range, where there is room for optimization, the sparse sampling method was always better than the other policies. If the initial distribution of the picking rate does not match the distribution of fruits well, the sparse sampling policy can be very beneficial as the cases in Figure 13 e, f under picking rate setting #3. In Figure 13f, when the platform is moving at 0.01m s^{-1} using the sparse sampling policy, workers can pick 95% of the fruits; the same workers can only pick 62% of the fruits, under the fixed height policy. The *PP* is improved by more than 30% in this specific case using the sparse sampling policy. In other words, if a grower decides to accept 95% *PP* to pick row 2, using the sparse sampling policy, the platform can move at 0.01m s^{-1}, whereas the platform can only move at 0.005 m s^{-1} using the fixed height policy; equivalently, 50% of the working time can be saved using the sparse sampling policy. However, the improvement was not always so large. In particular, when the vertical distribution of picking rate matched the vertical distribution of fruits better, such as in Figure 13 c, d under picking rate setting #2, the difference between the sparse sampling policy and the fixed height policy became small. In practice, if the fruit distribution were known before picking and did not change along rows (not realistic), one could simply use the fixed height policy and assign the slower worker to the zone with less fruit and the fast worker to the zone with more fruit. However, if the order were reversed, the consequence of using the fixed height policy would be cause the *PP* to decrease significantly, as in Figure 13 e, f. With the sparse sampling policy, the initial position of the two workers makes no big difference. The sparse sampling policy can always assign the workers to the optimal position and achieve optimal picking efficiency.

5.1.1.2 Planning horizon study

Figure 14 shows the effect of the planning horizon H and extra search depth E on harvesting the digitized fruit row #1. When the effective planning horizon ($H + E$) was equal to 1, the workers picked only 88% percent of the fruits; when the effective planning horizon increased to more than 2, the same two workers could pick more than 96% of the fruits. For the same effective planning horizon value, using a higher H can be slightly better than using a higher E, because the planning horizon contributed by H considers all possible actions, whereas E corresponds to Monte Carlo (non-exhaustive) estimation. However, in practice, setting $H > 3$ takes too much time to get a solution in real-time. So, it is preferable to use both H and E to extend the effective planning horizon. To conclude, we need to use both H and E to extend the effective planning horizon, and choose an effective planning horizon larger than 3, to allow the sparse sampling policy achieve near-optimal results in real-time.

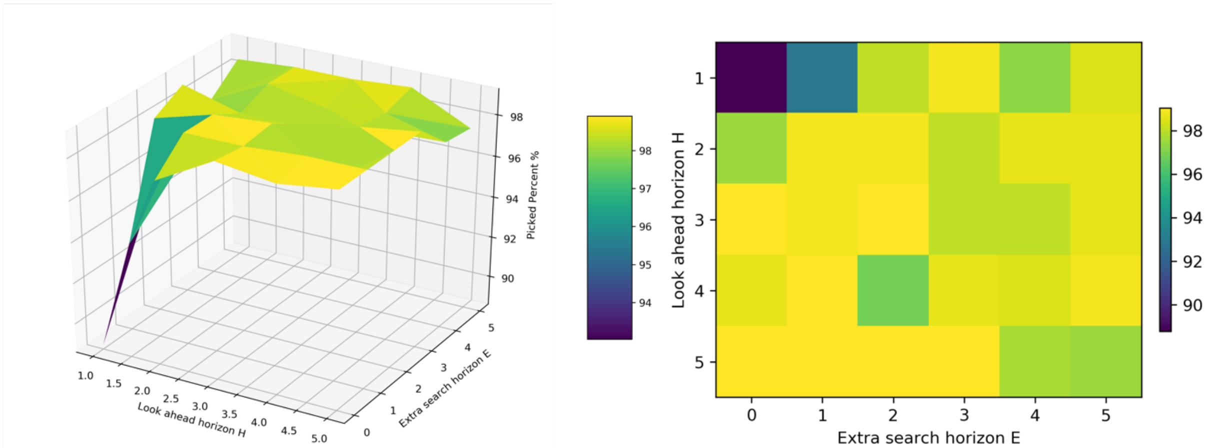


Figure 14 The 3D plot (left) and its 2D projection (right) show the dependance of PP on the look-ahead horizon H, and the extra search horizon E; the digitized fruit distribution from row #1 was used.

5.1.2 Field experiment results

The GNSS traces of the platform's location as it moved inside rows during the orchard experiment are shown in Figure 15. The trace is superimposed on a satellite image of that part of the orchard.

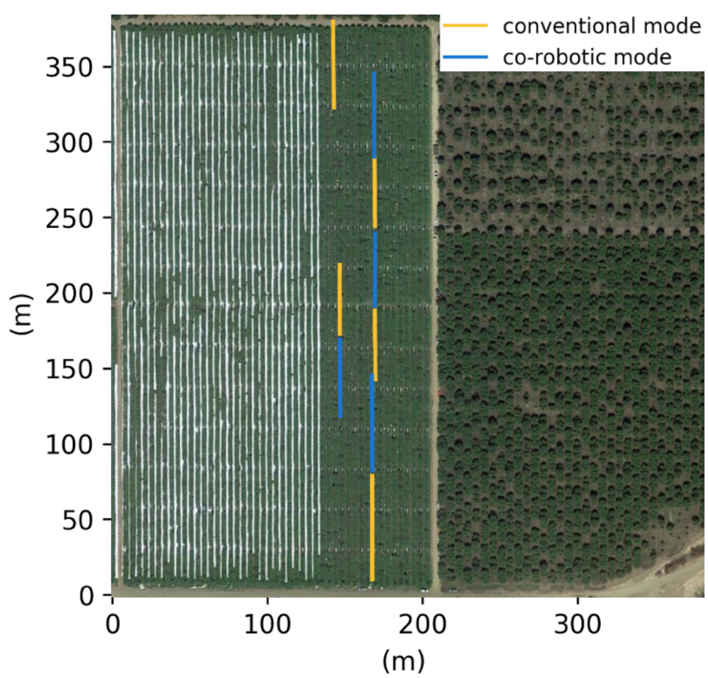


Figure 15 GNSS trace of the platform's location as it moved inside rows during the orchard experiment. The trace is superimposed on a satellite image of part of the orchard.

Because the adaptive speed mode is used in commercial harvesting, more data was collected using this mode than with the fixed-speed mode. Table 3 shows the weights of the picked apples, the corresponding effective picking times, the throughputs under all the settings, and the improvements of the variable-height zone harvesting vs. the fixed-height zone harvesting.

Speed policy	**Adaptive**	**Adaptive**	**Fixed 0.03 m s^{-1}**	**Fixed 0.03 m s^{-1}**	**Fixed 0.015 m s^{-1}**	**Fixed 0.015 m s^{-1}**
Height Control Mode	fixed-height zone harvesting	variable-height zone harvesting	fixed-height zone harvesting	variable-height zone harvesting	fixed-height zone harvesting	variable-height zone harvesting
Total Weight (kg)	788.28	648.04	251.32	161.60	222.36	235.52
Total Time (h)	2.64	1.98	0.94	0.57	0.91	0.88
Throughput (kg h^{-1})	298.80	327.60	266.40	280.80	244.80	266.40
Variable-height zone Harvesting Improvement	9.50%		5.47%		9.52%	

Table 3 Harvesting throughput results from apple-harvesting field experiments. Harvesting was done using fixed-height and variable-height zone harvesting under adaptive and fixed speed control.

The results show that for both the platform speed control modes, the optimized policy achieved higher throughput than the (conventional) fixed height policy. The improvement was 9.50% when adaptive speed control was applied, and 9.52% when the platform travel speed was constant at 0.015m/s. The throughput improvement decreased as the platform's moving speed increased: the improvement in throughput was smaller at 0.015 m s^{-1} compared to 0.03 m s^{-1}; this trend was the same as in the simulation results.

6 Summary, conclusions and future work

In this study, we developed an optimized variable lift height control policy, to implement variable-height zone harvesting for multi-crew harvesting platforms. The policy uses a sparse sampling algorithm to solve the "labor supply and demand" mismatch problem that limits platforms' harvesting efficiencies. To evaluate our approach, we converted a commercial picking platform into a "co-robotic" system. Simulations and commercial apple harvesting experiments showed that the proposed lift height control policy improved the overall picking throughput by up to 9.52%. The potential gain in picking efficiency depended on the platform's speed, the workers' picking rates, and the fruit distribution. The efficiency gain was significant when the platform speed was neither too fast nor too slow, and when the workers' picking rates were unequal.

Our method can be directly applied to harvest-aid platforms that use vacuum tubes to transport picked fruits from each worker into the bin, without using picking bags. The only difference is that each worker's picking rate would be measured by counting the fruits going through the worker's tube.

Extensions of our work are discussed next. The co-robotic platform did not control its own travel speed. Future work will extend our approach to include an adaptive speed controller that matches the platform speed with the fruit load and workers' picking speeds. Speed control is expected to result in higher picking rate for the front worker (who has to adjust speed) and increased throughput. One limitation of our method is that the optimization model assumes that fruit detection is perfect. In reality, fruit detection is not perfect, and the detection errors presumably result in non-optimal lift height assignments. Future research can investigate optimization algorithms that take into account the error statistics (e.g. by using a fruit distribution probability map instead of a fruit-count map) to compute the optimal policy. Another limitation is that the picking rate estimated using the worker's picking bag is affected by the number of fruits the worker has in front of them (local fruit density), so it does not directly measure the worker's "intrinsic" picking capacity. One way to estimate this capacity is to use the camera-sensed fruit distribution and the platform travel speed to remove the effect of variable fruit density. Finally, fruit quality was not assessed in our experiments. In our work, the workers on the platform picked in the same way when the platform operated in conventional mode, and when it operated in co-robotic mode; hence, no changes in fruit damage rates were expected. However, it is conceivable that when a worker picks fruits while her/his lift is being raised or lowered, the vertical motion could affect the fruit-detachment action, and potentially, fruit quality. Further harvesting experiments followed by post-harvest comparison studies are needed to evaluate this aspect of the system.

Acknowledgement

This work was funded by NIFA grants 2016-67021-24535 and W3009 Hatch Multi-State 1016380. We appreciate research engineer Dennis Sadowski for retrofitted the harvest-aid platform. We thank Zania Pothen and Abhisehi Silwal from Carnegie Mellon University providing support on the fruit detection system and their help during the field experiment. We thank grower Jeff Colombini (Lodi Farming) for providing the experiment field during harvest season and thanks workers helped in the experiments.

References

Arikapudi, R., Vougioukas, S. G., Jiménez-Jiménez, F., Anjom, F. K. (2016). Estimation of fruit locations in orchard tree canopies using radio signal ranging and trilateration. Computers and Electronics in Agriculture, 125, 160-172.

Bac, C. W., van Henten, E. J., Hemming, J., Edan, Y. (2014). Harvesting robots for high-value crops: State-of-the-art review and challenges ahead. Journal of Field Robotics, 31(6), 888-911.

Barnett, J., Duke, M., Au, C. K., Lim, S. H. (2020). Work distribution of multiple Cartesian robot arms for kiwifruit harvesting. Computers and Electronics in Agriculture, 169, 105202.

Bellman, R. (1957). A Markovian decision process. Journal of mathematics and mechanics, 679-684.

Berlage, A. G. (1973). Apple harvesting trials with oscillating air jets. Transactions of the ASAE, 16(3), 460-0461.

Berlage, A. G., Yost, G. E., Langmo, R. D. (1972). Limitations of single-and multi-man platform harvesting aids.

Bogue, R. (2020), "Fruit picking robots: has their time come?", Industrial Robot, Vol. 47 No. 2, pp. 141-145.

Cybenko, G. (1989). Dynamic load balancing for distributed memory multiprocessors. Journal of parallel and distributed computing, 7(2), 279-301.

De Kleine, M. E., Karkee, M. (2015). A semi-automated harvesting prototype for shaking fruit tree limbs. Transactions of the ASABE, 58(6), 1461-1470

Fei, Z., Shepard, J., Vougioukas, S. (2017). Instrumented picking bag for measuring fruit weight during harvesting. In 2017 ASABE Annual International Meeting (p. 1). American Society of Agricultural and Biological Engineers.

He, L., Fu, H., Sun, D., Karkee, M., Zhang, Q. (2017). Shake-and-catch harvesting for fresh market apples in trellis-trained trees. Transactions of the ASABE, 60(2), 353-360.

Kearns, M., Mansour, Y., Ng, A. Y. (2002). A sparse sampling algorithm for near-optimal planning in large Markov decision processes. Machine learning, 49(2-3), 193-208.

Lesser, K., Harsh, R. M., Seavert, C., Lewis, K., Baugher, T., Schupp, J., Auvil, T. (2008, January). Mobile platforms increase orchard management efficiency and profitability. In International Symposium on Application of Precision Agriculture for Fruits and Vegetables 824 (pp. 361-364).

Mann, M. P., Zion, B., Shmulevich, I., Rubinstein, D., Linker, R. (2016). Combinatorial Optimization and Performance Analysis of a Multi-arm Cartesian Robotic Fruit Harvester—Extensions of Graph Coloring. Journal of Intelligent & Robotic Systems, 82(3-4), 399-411.

Ortiz, C., Torregrosa, A. (2013). Determining adequate vibration frequency, amplitude, and time for mechanical harvesting of fresh mandarins. Transactions of the ASABE, 56(1), 15-22.

Peterson, D. L. (1982). Rod press fruit removal mechanism. Transactions of the ASAE, 25(5), 1185-1188.

Peterson, D. L. (2005). Development of a harvest aid for narrow-inclined-trellised tree-fruit canopies. Applied engineering in agriculture, 21(5), 803-806.

Peterson, D. L., & Kornecki, T. S. (1987). Mechanical apple harvester for T-trellis canopies. Transactions of the ASAE, 30(3), 597-0600.

Peterson, D. L., & Miller, S. S. (1989). Advances in mechanical harvesting of fresh market quality apples. Journal of Agricultural Engineering Research, 42(1), 43-50.

Peterson, D. L., Miller, S. S., Woldorf, S.D. (1996). Apple harvesting concepts for inclined trellised canopies. Applied engineering in agriculture, 12(3), 267-2.

Peterson, D. L., Bennedsen, B. S., Anger, W. C., Wolford, S. D. (1999). A systems approach to robotic bulk harvesting of apples. Transactions of the ASAE, 42(4), 871.

Pothen, Z., Nuske, S. (2016). Texture-based fruit detection via images using the smooth patterns on the fruit. In 2016 IEEE International Conference on Robotics and Automation (ICRA) (pp. 5171-5176). IEEE.

Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Ng, A. Y. (2009, May). ROS: an open-source Robot Operating System. IEEE International Conference on Robotics & Automation (ICRA): workshop on open source software (Vol. 3, No. 3.2, p. 5).

Ramasesh, R. (1990). Dynamic job shop scheduling: a survey of simulation research. Omega, 18(1), 43-57.

Sazo, M. M., Marree, A. D., & Robinson, T. (2010). The platform factor-labor positioning machines producing good results for NY apple industry. New York Fruit Quarterly, 18(2), 5-10.

Schoonderwoerd, R., Holland, O. E., Bruten, J. L., Rothkrantz, L. J. (1997). Ant-based load balancing in telecommunications networks. Adaptive behavior, 5(2), 169-207.

Schupp, J., Baugher, T., Winzeler, E., Schupp, M., & Messner, W. (2011). Preliminary results with a vacuum assisted harvest system for apples. Fruit Notes, 76(4), 1-5.

Silwal, A., Davidson, J. R., Karkee, M., Mo, C., Zhang, Q., Lewis, K. (2017). Design, integration, and field evaluation of a robotic apple harvester. Journal of Field Robotics, 34(6), 1140-1159.

Sutton, R. S., & Barto, A. G. (1998). Reinforcement learning: An introduction (Vol. 1, No. 1). Cambridge: MIT press.

Taylor, J. E., & Charlton, D. (2018). The Farm Labor Problem: A Global Perspective. Academic Press.

Thomas, R. L. (1964). An engineering investigation in the use of a pulsating air stream to mechanically detach apples from the tree. Unpublished MS thesis, Rutgers—The State University, New Brunswick, New Jersey.

Vougioukas, S. G. (2019). Agricultural robotics. Annual Review of Control, Robotics, and Autonomous Systems, 2, 365-392.

Willebeek-LeMair, M. H., Reeves, A. P. (1993). Strategies for dynamic load balancing on highly parallel computers. IEEE Transactions on parallel and distributed systems, 4(9), 979-993.

Xiong, Y., Ge, Y., Grimstad, L., From, P. J. (2020). An autonomous strawberry-harvesting robot: Design, development, integration, and field evaluation. Journal of Field Robotics, 37(2), 202-224.

Zhang, Q. (editor), 2017. Automation in Tree Fruit Production: Principles and Practice. ISBN: 9781780648507. CABI.

Zhang, Z., Heinemann, P. H., Liu, J., Baugher, T. A., & Schupp, J. R. (2016). The development of mechanical apple harvesting technology: A review. Transactions of the ASABE, 59(5), 1165-1180.

Zhang, Z., Heinemann, P., Liu, J., Schupp, J., Baugher, T. (2014). Design, fabrication, and testing of a low-cost apple harvest-assist device. In 2014 Montreal, Quebec Canada July 13–July 16, 2014 (p. 1). American Society of Agricultural and Biological Engineers.

Zhang, Z., Igathinathane, C., Li, J., Cen, H., Lu, Y., & Flores, P. (2020). Technology progress in mechanical harvest of fresh market apples. Computers and Electronics in Agriculture, 175, 105606.

Zhang, Z., Pothula, A. K., Lu, R. (2018). Improvements and evaluation of an infield bin filler for apple bruising and distributions. In 2018 ASABE Annual International Meeting (p. 1). American Society of Agricultural and Biological Engineers.

Zion, B., Mann, M., Levin, D., Shilo, A., Rubinstein, D., Shmulevich, I. (2014). Harvest-order planning for a multiarm robotic harvester. Computers and Electronics in Agriculture, 103, 75-81.

Zomaya, A. Y., Teh, Y. H. (2001). Observations on using genetic algorithms for dynamic load-balancing. IEEE transactions on parallel and distributed systems, 12(9), 899-911.