Proceedings of the 14th International Conference on Flexible Automation and Intelligent Manufacturing • Toronto, Ontario • July 12–14, 2004

# Sensing-System Reconfiguration: A Comparison of On-Line Methods

Michael D. Naish[a,*], Elizabeth A. Croft[b] and Beno Benhabib[c]

[a] Dept. of Mechanical and Materials Engineering, University of Western Ontario, London, Ont., Canada N6A 5B9
[b] Dept. of Mechanical Engineering, University of British Columbia, Vancouver, B.C., Canada V6T 1Z4
[c] Dept. of Mechanical and Industrial Engineering, University of Toronto, Toronto, Ontario, Canada M5S 3G8

***ABSTRACT***

*This paper investigates the performance of two dispatching approaches applied to the real-time coordination of multiple, mobile sensors. The sensing system is targeted towards the surveillance of objects in the context of autonomous manufacturing systems. Sensors are assigned and manoeuvred to collect data at specific points on the object trajectory. A technique based on reinforcement learning (RL) is compared to a heuristic dispatching method and a system that does not use dispatching at all. Through a number of simulation examples, it is shown that, on average, the RL-based dispatcher achieves very similar, if not slightly better, performance than the heuristic dispatcher. Both approaches appear to provide a benefit over non-dispatching systems, thereby validating the efficacy of the dispatching approach, despite very different underlying implementations.*

## 1. Introduction

The use of industrial robots in an automated manufacturing environment typically requires presentation of the objects to be manipulated in specific fixed poses. These positions and orientations are achieved by structuring the environment using jigs and fixtures; the rate at which objects are presented is controlled by various types of indexing systems. The migration towards flexible autonomous manufacturing systems necessitates the ability to reliably detect, recognize and continuously track objects within the workspace of a robot. The availability of such real-time sensory information, combined with recent advances in robot control, will enable a robot or other automated system to operate autonomously in a semi-structured or unstructured environment. The ultimate aim is to enable a robot system to operate with the same degree of flexibility as possessed by a human worker.

The majority of work in the area of vision-based surveillance has employed single or multiple static cameras (e.g., [1],[2]). The coordination and control of multiple sensors provides an opportunity to significantly improve the quality and robustness of the acquired data as compared to single-sensor systems (e.g., [3]-[5]). This paper investigates the use of a reconfigurable surveillance system that may comprise multiple static and dynamic sensors. The underlying approach is based on the concept of dispatching, as used for the coordination of service vehicles, such as ambulances and taxicabs [6],[7]. The premise is that the quality and robustness of the data acquired by a set of sensors can be improved through appropriate selection and positioning. Additionally, sensor fusion may be used to combine information from these multiple, coordinated sensors into a single representation, reducing the uncertainty of the measured data [8].

## 2. Sensor Dispatching

In the context of object surveillance, sensor dispatching attempts to maximize the effectiveness of a sensing-system used to provide estimates of object parameters at predetermined times or positions along the object trajectory. Herein, it is assumed that the times at which the information is desired are fixed. These predetermined times are referred to as demand instants, $t_j$. The position of the object at a particular demand instant is a demand point, $D_j$; demand points are estimated using observations of the object motion. At any one time, a finite set, or rolling horizon, of (future) demand points is considered by the system. The period between two demand instants is the service interval—the amount of time available for planning and positioning of the sensors before data acquisition must occur.

[*] Corresponding author: Tel.: (519) 661-2111 x. 88294; Fax: (519) 661-3020; E-mail: naish@eng.uwo.ca

If the sensing-system comprises multiple redundant sensors, a subset of these may be sufficient to satisfy the sensing requirements of a demand. In other words, a sensor fusion process does not need to combine the information from all of the sensors in the system. Instead, a subset of sensors ($k \leq n$, where $k$ is the subset size and $n$ is the total number of sensors) may be selected to survey the object at a particular demand instant.

Thus, we consider the following sensor dispatching problem in the context of moving object surveillance: Given a set of sensors and the predicted motion of a manoeuvring object (represented as a set of time-varying demand points), select a subset of sensors to acquire data at each demand point and specify the pose of each sensor, thereby, maximizing the quality of the acquired data and ensuring that the remaining sensors are adequately distributed throughout the workspace for possible future use.

There are a number of different methods by which this sensor dispatching problem may be approached. These include optimization techniques, heuristics and machine learning approaches. While the computational complexity of optimization approaches render them unsuitable for real-time implementation, specific implementations of the last two of these approaches are detailed in the following sections. Both approaches tackle the dispatching problem using two complementary strategies. A coordination strategy is used to specify which sensors will be used at each demand instant. This requires the fitness of each sensor to be evaluated with respect to the sensing demand under consideration. Subsequently, the pose of each sensor is specified through a positioning strategy. These strategies consider the demands sequentially (starting with the first), limited to the rolling horizon.

### *2.1. Assessment of Sensor Data Quality*

Integral to dispatching is an estimate of the quality of data that each sensor can provide for the demand point at hand, given its pose in the workspace and its motion capabilities. This estimate is used to select sensors for inclusion in a fusion subset (i.e., assignment to the sensing task) and assess the desired pose of each sensor during surveillance. Herein, a visibility measure, that is inversely proportional to the measurement uncertainty, is used to assess the fitness of a single sensor or a group of fused sensors. Assuming that the demand point is visible to the sensor, the single-sensor visibility measure is considered to be [9]:

$$v_{\mathrm{s}} = \frac{1}{\|R\|} \tag{1}$$

where $\|R\|$ is the Euclidean norm of the covariance matrix associated with the sensor measurement. The visibility measure for a fusion subset comprising $k$ sensors, whose measurements are combined using sensor fusion, is defined as:

$$v_{\mathrm{f}} = \frac{1}{\|P\|}, \tag{2}$$

where $P$ represents the fused covariance matrix,

$$P = \left[ \sum_{i=1}^{k} R_i'^{-1} \right]^{-1}.$$

Typically, the measurement uncertainty (covariance) changes as the pose of the sensor(s) with respect to the demand point is varied. Considering a single sensor, the goal of the dispatcher is to manoeuvre the sensor such that the visibility measure $v_{\mathrm{s}}$ is maximized. The sensor pose at this maximum reflects the best-possible visibility, $v_{\mathrm{p}}$. The best-possible visibility is constrained by the capabilities of the sensor, the task, and the workspace. The best-achievable visibility, $v_{\mathrm{a}}$, is the visibility that can be expected, given a finite amount of time for manoeuvring. Thus, $v_{\mathrm{a}}$ is constrained additionally by the dynamic characteristics of the sensor (maximum velocity, acceleration, etc.) and time. Similarly, $v_{\mathrm{p}}$ and $v_{\mathrm{a}}$ may be computed for a group of sensors by maximizing $v_{\mathrm{f}}$. The only difference for multiple sensors is that $v_{\mathrm{p}}$ must also consider the optimal fusion subset. Together, $v_{\mathrm{a}}$ and $v_{\mathrm{p}}$ may be used to compute the normalized best-achievable visibility:

$$v_{\mathrm{n}} = \frac{v_{\mathrm{a}}}{v_{\mathrm{p}}} \tag{3}$$

The normalized visibility measure allows the performance of difference sensing systems to be compared to one another, even when different object trajectories are observed.

### *2.2. A Heuristic Approach to Sensor Dispatching*

A heuristic approach to the dispatching of multiple sensors has been proposed in [9]. This approach divides the problem among three different submodules:

1. *Assignment and positioning*: Assignment is triggered by an object entering the workspace or the completion of a previous service interval. Only the first demand point on the rolling horizon is considered. The coordination strategy is implemented by performing a search that selects a subset of sensors (of size $k$) to service the demand instant from the set of all sensors (total of $n$). Once assigned, the subset of sensors cannot be altered until the demand is serviced (i.e., data is acquired), completing the service interval. In addition, the search method specifies a desired pose for each assigned sensor at the time of data acquisition, thereby implementing the positioning strategy. Note that, the desired sensor poses may be altered in real-time (see Replanning below); however, the fusion subset does not change until the next demand instant is considered.
2. *Preassignment and prepositioning*: The goal of preassignment and prepositioning is to position sensors in anticipation of future sensing requirements. Those sensors that have not been assigned to the most immediate demand instant (the first demand on the rolling horizon) may be preassigned to subsequent demands. The approach is very similar to that used for assignment and positioning; however, there are some differences. First, while the suitability of each sensor is considered for each demand, only those that have not been previously (pre)assigned may be preassigned to the demand under consideration. Second, the preassignment algorithm loops to consider additional demand instants until either all sensors have been preassigned or the end of the rolling horizon has been reached. This approach aims to service each demand with the sensors that can provide the highest quality data.
3. *Replanning*: The uncertainty of the predicted demand point locations at the time of assignment necessitates reevaluation over the service interval to ensure that the sensors are in the best positions. As improved estimates become available, the desired poses of assigned sensors may be adjusted to reflect the change in demand position (provided that the demand point location has changed by an amount that exceeds the uncertainty of the original prediction).

### *2.3. A Learning-Based Approach to Sensor Dispatching*

Rather than relying on a system designer to generate appropriate dispatching rules, machine learning offers the possibility of developing a suitable dispatcher through repeated exposure to surveillance episodes. Three main types of machine learning approaches exist [10]: (1) supervised learning, which uses the "correct" strategy for training purposes; (2) unsupervised learning, which classifies input values; and (3) reinforcement learning, which uses measures of goodness in place of specific targets. Since the best dispatching strategy is likely not known and the sensors must be controlled, reinforcement learning (RL) is, perhaps, the most suitable approach.

Modelled after human learning, RL provides a mechanism by which a system can learn an appropriate control strategy without knowing what the "best" strategy is. Instead, rewards are used as a form of instructive feedback, guiding the system towards actions that maximize reward. This parallels the human tendency to repeat actions that produce favourable results and suppress those actions that cause unfavourable results. From the rewards received after each action, RL attempts to learn the long-term value of the actions that may be taken from any given state. These values are referred to as $Q$-values. Learning proceeds by estimating the expected reward, taking action $a$ from state $s$, for all state-actions pairs, $Q(s,a)$. Once training has determined $Q$-values that accurately reflect the value of each state-action pair, control simply selects those actions with the highest $Q$-value, given the current state.

There are many different implementations of RL [11]; herein, SARSA($\lambda$), a form of temporal difference (TD) learning, is adopted. During training, the action-value function $Q_t(s,a)$ is updated by SARSA($\lambda$) as follows [12]:

$$Q_{t+1}(s,a) = Q_t(s,a) + \eta \delta_t e_t(s,a) \quad \forall s,a\,, \tag{4}$$

where

$$\delta_t = \underbrace{r_{t+1} + \gamma Q_t(s_{t+1}, a_{t+1}) - Q_t(s_t, a_t)}_{TD-error}; \tag{5}$$

$$e_t(s,a) = \begin{cases} \gamma\lambda e_{t-1}(s,a) + 1 & \text{if } s = s_t \text{ and } a = a_t, \\ \gamma\lambda e_{t-1}(s,a) & \text{otherwise} \end{cases} \quad \forall s \in S, a \in A,; \tag{6}$$

and $\eta$ controls the learning rate. $\delta_t$ represents the temporal difference of the estimated action-value functions, $r$ is the received reward, $\gamma$ $(0 \leq \gamma \leq 1)$ represents the discount rate (which balances between immediate and future rewards); $e_t(s,a)$ is an eligibility trace that uses current errors to correct previous $Q$-value estimates, $\lambda$ is referred to as the trace-decay parameter and is responsible for temporal credit assignment: $\lambda = 0$ results in no feedback beyond the current time step; $\lambda = 1$ propagates the error without decay, arbitrarily far in time.

When applied to a continuous-valued state-action space, it becomes impossible to represent all possible states, let alone repeatedly visiting each one to ensure adequate learning. One approach that may be used to generalize the state space is neural networks. Neural networks can approximate complex functions and scale well with the input space [11]. A general algorithm for training a feed-forward neural network is outlined in [13]. The approach uses eligibility traces to adjust the nodal weights during back propagation.

The application of RL to sensor dispatching requires the problem to be framed in an appropriate manner. One approach, introduced in [14], elects to handle the coordination strategy at the system level using a supervisory controller. Fusion subsets are constructed by assessing the capabilities of each sensor heuristically. The positioning strategy is implemented at the sensor level using RL. In effect, each sensor is treated as an autonomous agent, determining its pose in the workspace independently. The goal of each sensor is to choose an appropriate action from a finite set. The number of possible actions is determined by the constraints imposed on the sensor motion. The simplest case involves sensors capable of translation along a single axis, e.g., $A$ = {right, left, stop}.

The selection of an appropriate action requires that the state space be encoded in a form that can be input to the networks. The action-value function $Q(s,a)$ is represented using a separate neural network for each action. All sensors share the same egocentric representation. The following salient features of the state space are encoded (all other potential features are ignored): whether or not the sensor belongs to the fusion subset, the normalized range from the sensor to the first demand point on the rolling horizon (ahead or behind, relative to the direction of the target motion), the normalized position of the first demand point on the rolling horizon and the normalized position of the sensor in the workspace. After encoding the state space in this manner, the networks may be used to produce an estimate of the value of each action. During training the received rewards are used to adjust the network weights (which reflect the overall strategy). Once trained, the action with highest estimated value is always selected. Additional details regarding network training for sensor dispatching may be found in [14].

The positioning strategy of the RL-approach is quite different from the heuristic approach. Unlike the heuristic dispatcher that sets a desired pose at the time of assignment, the RL-based dispatcher simply chooses a path (direction) for the sensor to follow. If the sensor position comes within a predefined tolerance of the demand point, it stops moving.

### *2.4. Sensing-system Configuration*

It must also be noted that the quality of information provided through sensor dispatching is dependant on the initial pose of each sensor within the workspace (i.e., the pose of each sensor prior to object entry). The effect of the initial sensor placement becomes more pronounced as the speed of the sensors decreases relative to the speed of the object. In general, as sensor speed decreases, adequate performance requires the sensors to become more widely distributed in the workspace. If the object trajectory is known a priori, the sensing system can be reconfigured in an optimal manner. While beyond the scope of this paper, one approach for determining initial sensing-system configurations is outlined in [15].

## 3. Results and Discussion

The performance of different sensing systems is investigated and compared in this section. All systems utilize the same basic sensing system that is composed of four sensors. The differences between each system (identified as A, B, C, and D) are limited to the maximum achievable translational and rotational velocities. Velocities range from

$\dot{x} = 0.2$ m/s and $\dot{\alpha} = \pi/2$ rad/s, for System A, to 0 for the static System D. (Note that, while the static sensors cannot move, the fusion subsets are still adjusted in real-time.) Each system is coupled with both a heuristic dispatcher and a learning-based dispatcher so that the performance of each dispatcher may be compared. As a final comparison, the sensing system is used without dispatching of any kind (System E). Instead, all four (static) sensors are used to acquire data for each demand point.

The workspace considered herein is illustrated in Figure 1. The sensors are constrained to rails at the edge of the workspace. These rails allow translation along the rail ($x$) and rotation about the $z$-axis (by an amount $\alpha$). For each episode, the object moves at approximately 0.1 m/s. The object trajectory itself is corrupted by Gaussian white noise ($\mu$ = 0, $\sigma$ = 0.05 m). Fusion subsets of size $k$ = 3 are assigned at each demand instant (except for the non-dispatching system, where $k$ = 4). Prior to evaluation (and training of the RL-based dispatcher), an optimal initial sensing-system configuration was determined for each system. The systems were configured using the approach outlined in [15], assuming that the object would follow the straight-line trajectory of Figure 2(a).

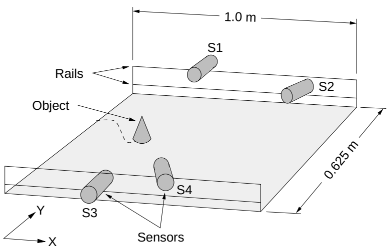
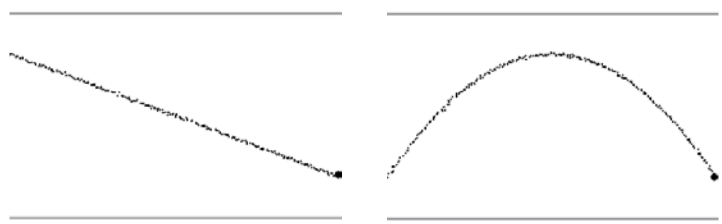



Figure 1: Example workspace.



(a) Straight-line

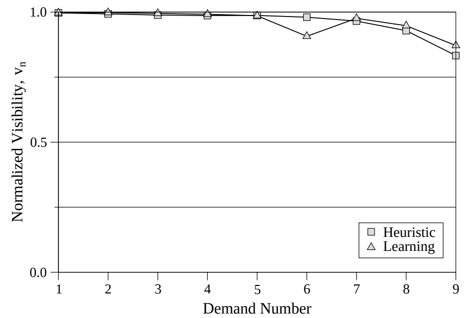
(b) Parabolic

Figure 2: Observed object trajectories.

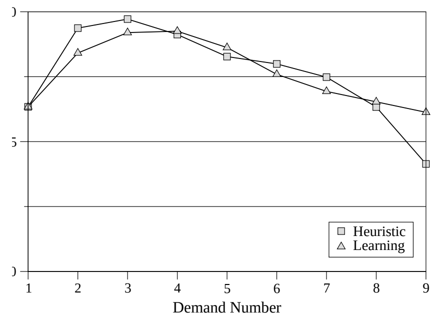
The heuristic dispatcher applies the same strategy, independent of the sensing system type. In contrast, a different RL-based dispatcher is used for each system type. Each RL-based dispatcher uses 3 action networks (for right, left and stop) composed of 23 input nodes, 10 hidden nodes, and 1 output node. The networks were trained using the same methodology and parameters as [14].

Comparing the overall performance of the RL-based dispatchers with their heuristic counterparts, both appear to be quite similar. As may be observed in Figure 3, the RL-based dispatchers improve the worst-case visibility by reducing the visibility of other demand points. This trade-off reflects the limitations of the sensing systems. Without reserve motion capability (with respect to the object), adjusting the sensor poses to better observe one part of the object trajectory necessarily impacts other parts. The result is that improvements in the mean visibility, while possible, are limited in magnitude. This is particularly evident for the fast surveillance system (System A), where the visibility of $D_6$ is reduced to increase the visibility of the final three demand points. In this case, the sensors are "hedged" towards the end of the object trajectory.

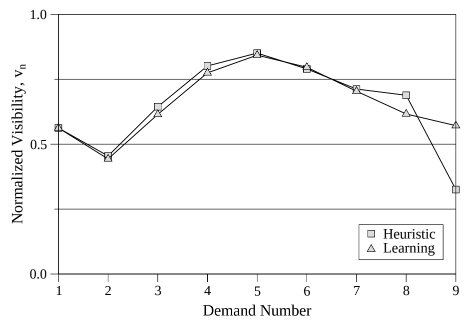
Table 1 presents the differences between the heuristic dispatcher and learning-based dispatchers, based on 500 samples. In each case, the RL-based dispatcher realizes a small improvement in the mean normalized visibility for the object trajectory. While small, these differences are statistically significant. For dispatchers using the fast sensing system (System A), a two-sample $t$-test rejects the two-tailed null hypothesis ($H_0 : (\mu_{\bar{v}_n})_{\text{learning}} = (\mu_{\bar{v}_n})_{\text{heuristic}}$) with nearly 100% confidence ($P$ = 3.986 × $10^{-9}$). The more conservative, non-parametric Mann-Whitney test [16] rejects the null hypothesis with 98.88% confidence. Both tests reject the null hypothesis for Systems B and C. The performance of the dispatchers for the static system (D) is identical. Clearly, the inability to adapt the sensing system to the object motion impacts its overall effectiveness. Comparing the performance of System D to the non-dispatching System E, the benefit of dispatching is not clear. While the dispatching systems do show better average and worst-case performance, the non-dispatching system provides higher visibility for the majority of the demand points. One may conclude from this that the benefits of dispatching are only realized by sensing systems that are capable of at least limited motion capability.



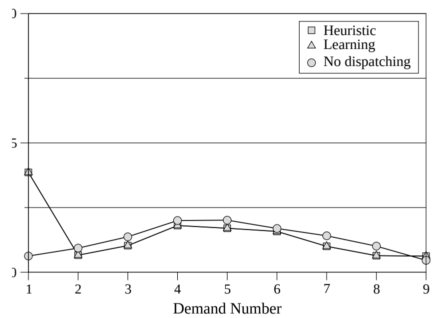

(a) System: A




(b) System: B




(c) System: C




(d) Systems: D and E

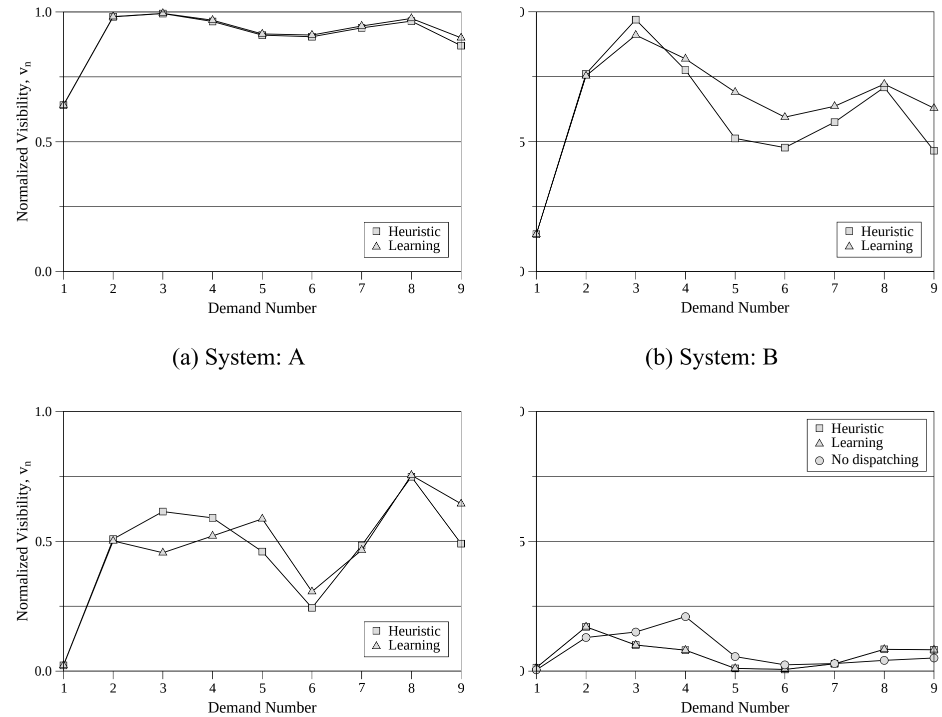
Figure 3: System performance for dispatchers observing a straight-line object trajectory (see Table 1 for system descriptions).

Table 1: Comparison of RL-based dispatchers with heuristic dispatchers observing a straight-line trajectory.

| System | Sensor characteristics | | $\bar{v}_n$ | | | $(v_n)_{min}$ | | |
|---|---|---|---|---|---|---|---|---|
| | $\dot{x}$ [m/s] | $\dot{\alpha}$ [rad/s] | Heuristic | Learning | % chg | Heuristic | Learning | % chg |
| A | 0.2 | π/2 | 0.96230 | 0.96337 | 0.11 | 0.83297 | 0.87152 | 4.53 |
| B | 0.1 | π/3 | 0.74617 | 0.76675 | 0.34 | 0.41408 | 0.61289 | 48.01 |
| C | 0.025 | π/4 | 0.64760 | 0.65838 | 1.66 | 0.32521 | 0.44325 | 36.30 |
| D | 0.0 | 0.0 | 0.14333 | 0.14333 | 0.0 | 0.06234 | 0.06234 | 0.0 |
| E | No dispatching | | 0.12790 | | — | 0.04596 | | — |

A much larger improvement is observed for the worst-case visibility, $(v_n)_{min}$, for those sensing systems that have dynamic capability (Systems A, B and C). Thus, for situations where a minimum level of performance is required over the entire object trajectory, a learning-based implementation of the dispatcher may be desirable.

The robustness of the dispatcher to unanticipated object trajectories is an important characteristic. Specifically, the dispatcher should be able to adapt the surveillance system to accommodate an object trajectory that is different from expectation. The different trajectory is not anticipated by the sensing system; however, the dispatcher adjusts the sensing strategy is real-time. To illustrate this point, consider the same sensing systems that have been used above (trained and optimized to observe a straight-line trajectory, Figure 2(a)) but presented with a parabolic object trajectory, Figure 2(b), instead. The performance of the learning-based systems observing the unexpected trajectory is compared with heuristic systems in Figure 4.




(a) System: A

(b) System: B

(c) System: C

(d) Systems: D and E

Figure 4: System performance for dispatchers expecting a straight-line object trajectory and observing a parabolic trajectory.

Table 2 illustrates the differences between the two types of dispatcher for this scenario. Note that, the visibility of the first demand point cannot be altered by the dispatching strategy as it is acquired before the sensors have an opportunity to adjust their pose. Since the first demand point is the true minimum, no change can be affected, i.e., the worst-case visibility is constant. To provide some measure of the impact of each dispatching approach, the demand point with the second-lowest visibility is used as a measure of the worst-case performance, i.e., $(v_{\mathrm{n}})^*_{\min} = \min_{j=2}^{M} (v_{\mathrm{n}})_j$. Considering this, learning shows a statistically significant improvement in both the mean normalized visibility and the minimum visibility for all of the systems with dynamic capability (Systems A, B and C). Again, for System D (with static sensors) the type of dispatcher does not influence overall performance.

Table 2: Comparison of RL-based dispatchers with heuristic dispatchers expecting a straight-line trajectory and observing a parabolic trajectory. Note that, $(v_n)^*_{\min}$ refers to the minimum visibility of all demands, excluding the first.

| **System** | **Sensor characteristics** | | $\bar{v}_{\mathrm{n}}$ | | | $(v_{\mathrm{n}})^*_{\min}$ | | |
|---|---|---|---|---|---|---|---|---|
| | $\dot{x}$ **[m/s]** | $\dot{\alpha}$ **[rad/s]** | **Heuristic** | **Learning** | **% chg** | **Heuristic** | **Learning** | **% chg** |
| A | 0.2 | π/2 | 0.90755 | 0.91477 | 0.80 | 0.87015 | 0.90036 | 3.47 |
| B | 0.1 | π/3 | 0.59868 | 0.65519 | 9.44 | 0.46511 | 0.59408 | 27.73 |
| C | 0.025 | π/4 | 0.46249 | 0.47305 | 2.28 | 0.24385 | 0.30622 | 25.58 |
| D | 0.0 | 0.0 | 0.06414 | 0.06415 | 0.01 | 0.00658 | 0.00658 | 0.0 |
| E | No dispatching | | 0.07740 | | — | 0.02424 | | — |

Note that, sensing-system performance is directly proportional to the speed of the sensors. As the speed of the sensors increases, so too does the object visibility and robustness to trajectory variation. For static sensing systems, dispatching does appear to offer any significant benefit over non-dispatching systems. In such cases, the only advantage that dispatching may offer is the ability to reduce post-processing loads by selecting a subset of sensor data for processing.

## 4. Conclusion

A learning-based method for sensor dispatching has been compared with a heuristic method in this paper. The RL approach utilizes heuristics for the coordination strategy, selecting subsets of sensors to service the first demand point on the rolling horizon. Sensors are positioned via a RL-based positioning strategy. A single set of action networks is shared by all sensors. The commanded motion of each sensor is dependent on the encoded system state: the pose of the demand point, the sensor pose, and whether the sensor is part of the fusion subset.

Once trained, the RL-based approach has been shown to achieve similar, if not slightly better, performance than the heuristic approach introduced in [9]. Modest improvements in the mean normalized visibility are realized along with significant increases to the minimum visibility for a variety of sensing-systems, observing different object trajectories. The major disadvantage of learning is its requirement for training. While heuristics can provide a consistent level of performance upon start up (uninitialized) and can react to change instantly, learning requires a number of episodes before performance becomes acceptable. Thus, for highly dynamic environments, the reactive nature of a heuristic approach may be advantageous.


## Acknowledgements

The financial support of the Natural Sciences and Engineering Research Council of Canada (NSERC) is gratefully acknowledged.


## References


*[1] B. Rao and H. Durrant-Whyte, "A Decentralized Bayesian Algorithm for Identification of Tracked Targets," IEEE Transactions on Systems, Man and Cybernetics, vol. 23, no. 6, pp. 1683–1698, 1993.*

*[2] T. Ellis, "Multi-Camera Video Surveillance," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Fort Collins, Texas, pp. 253–259, 1999.*

*[3] S. Abrams, P. K. Allen and K. Tarabanis, "Computing Camera Viewpoints in an Active Robot Work Cell," International Journal of Robotics Research, vol. 18, no. 3, 1999.*

*[4] R. Collins, O. Amidi and T. Kanade, "An Active Camera System for Acquiring Multi-View Video," in Proceedings of the IEEE International Conference on Image Processing, Rochester, New York, pp. 517–520, 2002.*

*[5] T. Matsuyama, T. Wada, and S. Tokai, "Active image capturing and dynamic scene visualization by cooperative distributed vision," in Advanced Multimedia Content Processing, S. Nishio and F. Kishino, Eds., pp. 252–288, Springer-Verlag, 1999.*

*[6] B. L. Golden and A. A. Assad, Eds, Vehicle Routing: Methods and Studies, North-Holland, 1988.*

*[7] M. Shrivastava, P. K. Chande and A. S. Monga, "Taxi Dispatch: A Fuzzy Rule Approach," in Proceedings of the 1997 IEEE Conference on Intelligent Transportation Systems, pp. 978–982, Nov. 9-12, 1997.*

*[8] R. C. Luo and M. G. Kay, Eds., Multisensor Integration and Fusion for Intelligent Machines and Systems, Albex, 1995.*

*[9] M. D. Naish, E. A. Croft and B. Benhabib, "Coordinated Dispatching of Proximity Sensors for the Surveillance of Manoeuvring Targets," Journal of Robotics and Computer Integrated Manufacturing, vol. 19, no. 3, pp. 283–299, 2003.*

*[10] S. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, Prentice-Hall, 1995.*

*[11] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction, The MIT Press, 1998.*

*[12] R. S. Sutton, "Learning to Predict Through the Methods of Temporal Differences," Machine Learning, vol. 3, pp. 9–44, 1988.*

*[13] R. S. Sutton, "Implementation Details of the TD(λ) Procedure for the Case of Vector Predictions and Backpropagation," GTE Laboratories, Tech. Rep. TN87-509.1, Aug. 1989.*

*[14] M. D. Naish, E. A. Croft and B. Benhabib, "Object Surveillance Using Reinforcement Learning Based Sensor Dispatching," Proceedings of the 2004 IEEE International Conference on Robotics and Automation, New Orleans, Louisiana, Apr. 26-May 1, 2004.*

*[15] M. D. Naish, E. A. Croft and B. Benhabib, "Sensing-System Configuration for Dynamic Dispatching," in Proceedings of the 2001 IEEE International Conference on Systems, Man and Cybernetics, vol. 5, pp. 2964–2969, Tuscon, Arizona, 2001.*

*[16] J. H. Zar, Biostatistical Analysis, $4^{th}$ ed., Prentice-Hall, 1999.*