# Sequential $Q$-Learning With Kalman Filtering for Multirobot Cooperative Transportation

Ying Wang, *Member, IEEE*, and Clarence W. de Silva, *Fellow, IEEE*

***Abstract*—This paper presents a modified, distributed $Q$-learning algorithm, termed as sequential $Q$-learning with Kalman filtering (SQKF), for decision making associated with multirobot cooperation. The SQKF algorithm developed here has the following characteristics. 1) The learning process is arranged in a sequential manner (i.e., the robots will not make decisions simultaneously, but in a predefined sequence) so as to promote cooperation among robots and reduce their $Q$-learning spaces. 2) A robot will not update its $Q$-values with observed global rewards. Instead, it will employ a specific Kalman filter to extract its real local reward from the global reward, thereby updating its $Q$-table with this local reward. The new SQKF algorithm is intended to solve two problems in multirobot $Q$-learning: credit assignment and behavior conflicts. The detailed procedure of the SQKF algorithm is presented, and its application is illustrated using a prototype multirobot experimental system. The experimental results show that the algorithm has better performance than the conventional single-agent $Q$-learning algorithm or the team $Q$-learning algorithm in the multirobot domain.**

***Index Terms*—Decision making, multirobot systems, $Q$-learning.**

## NOMENCLATURE

| | |
|---|---|
| $R_i(i = 1, \ldots, n)$ | The $i$th robot. |
| $\Lambda_i(i = 1, 2, \ldots, n)$ | The action set of the $i$th robot. |
| $a^i_j(j = 1, 2, \ldots, m_i)$ | The $j$th action of the $i$th robot. |
| $Q_i(i = 1, \ldots, n)$ | The $Q$-table of the $i$th robot. |
| $n$ | The number of robots. |
| $s$ | World states. |
| $a$ | Actions. |
| $\tau$ | "Temperature" factor. |
| $\Psi$ | A set including all actions selected by the robots thus far. |
| $\phi$ | The empty set. |
| $\Omega$ | A set including all actions that can be taken by more than one robot at the same time. |
| $\Delta_i$ | A set including the actions that can be selected by the $i$th robot up till now. |
| $P(a^i_j)$ | The probability for the $i$th robot to select its $j$th action. |
| $r_i$ | The local reward received by the $i$th robot. |
| $\varepsilon$ | Learning rate. |
| $\mu$ (Section II) | Discount rate. |
| $g_t$ | The global reward received by the robots at time $t$. |
| $b_t$ | The noise process. |
| $\mu$ (Section III) | Mean. |
| $\sigma^2_w$ | Variance. |
| $\|s\|$ | The total number of world states. |

Manuscript received March 14, 2008; revised November 30, 2008. First published July 7, 2009; current version published March 31, 2010. Recommended by Technical Editor P. R. Pagilla. This work was supported in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada, in part by the Canada Research Chair, in part by the Canada Foundation for Innovation (CFI), in part by the British Columbia Knowledge Development Fund (BCKDF), and in part by the National Natural Science Foundation of China under Grant 60772127.

Y. Wang is with the Industrial Automation Laboratory, Department of Mechanical Engineering, The University of British Columbia, Vancouver, BC V6T 1Z4, Canada, and also with the Faculty of Maritime, Ningbo University, Ningbo 315211, China (e-mail: ywang@ mech.ubc.ca).

C. W. de Silva is with the Industrial Automation Laboratory, Department of Mechanical Engineering, The University of British Columbia, Vancouver, BC V6T 1Z4, Canada (e-mail: desilva@mech.ubc.ca).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMECH.2009.2024681

## I. INTRODUCTION

MULTIROBOT cooperation has rapidly become an important research area in the field of robotics. Most multirobot systems are considered to work in dynamic and unstructured environments such as planet surfaces, where the terrain features and the obstacle distribution change with time [1]. Furthermore, even if the external physical environment is stationary, the overall system structure is still dynamic from the viewpoint of a single robot because other robots may take actions thereby changing the environment of that particular robot continuously. The environmental dynamics makes multirobot decision making quite challenging, where the traditional task planning approach [2] may become useless because a planned optimal policy can become inappropriate a few minutes later due to changes in the environment. In addition, although the behavior-based approach is quite successful in simple environments [3], it is shown to be weak in a complex multirobot environment where there are thousands or ten thousands of world states, and some states may be unobservable [4].

By observing human capabilities of dealing with a dynamic environment, it is easy to draw a conclusion that a human employs not only planning or reactive (behavior-based) techniques but also learning techniques to successfully complete a task in such an environment. Through learning, a human learns new world states, finds optimal actions under these states from past experiences, and improves his planning and reactive techniques continuously. Learning enables him to deal with unexpected world states and uncertainties in the environment, and makes his decision-making capability more robust in a dynamic environment.

Therefore, machine learning has become popular in the multirobot field where the robotic environment is usually dynamic and partially observable [5]. Among the existing machine learning

approaches, reinforcement learning [6], especially $Q$-learning, is used most commonly in multirobot systems because of its simplicity and good real-time performance. In particular, the single-agent $Q$-learning and the team $Q$-learning algorithms [7]–[9] are two commonly used $Q$-learning algorithms in multirobot cooperative control. Our previous work [7] addressed how to employ two $Q$-learning algorithms, to find optimal cooperative strategies for a multirobot box-pushing task. While they can help robots to find good cooperation policies, it is observed that there are some serious disadvantages in both $Q$-learning algorithms. On one hand, directly extending the single-agent $Q$-learning algorithm to the multirobot domain violates its assumption of static environment, and makes the $Q$-values not converge to the correct values [10]. Although robots still can find some good policies, the performance of the entire team is degraded. On the other hand, although the team $Q$-learning algorithm models a dynamic environment inherently and seems to be more suitable for multirobot systems, it requires an extensive learning space (state/action space) that increases rapidly when the number of robots becomes bigger. It is very difficult for a robot to manipulate and manage such a vast learning space in a real-time manner [10].

In view of these observations, it is advantageous to develop a new $Q$-learning algorithm that is suitable for multirobot systems. In this paper, a modified $Q$-learning algorithm, termed as the sequential $Q$-learning algorithm with Kalman filtering (SQKF), is developed to meet the challenges of $Q$-learning in the multirobot domain. There are two important contributions in the new SQKF algorithm: sequential learning and Kalman-filtering-based reward estimation. Both contributions will help promote the cooperation among the robots and speed up the $Q$-learning algorithm. In addition, the algorithm itself can be generalized, i.e., it is scalable and can be applied to many existing multirobot tasks (e.g., multirobot foraging, multirobot assembly [11], [12], multirobot cooperative observation, and teleoperation [13]) through some minor modifications of the definitions of states and actions. The details of the developed SQKF algorithm are presented in Sections II and III, and the algorithm is validated using experimental results in Section IV.

## II. Sequential $Q$-Learning

The SQKF is especially designed to cope with various challenges such as the extensive learning space and the dynamic environment, which the single-agent $Q$-learning and team $Q$-learning algorithms are unable to handle in a multirobot environment. The SQKF algorithm has two parts: Sequential $Q$-learning and Kalman-filtering-based reward estimation.

The basic idea of the sequential $Q$-learning algorithm [14] comes from a strategy that is typically employed by humans when they cooperatively push a large and heavy object to a goal location. In the transportation of the object, the group members usually do not select their pushing locations and forces concurrently. Instead, one of them will select his pushing location and apply a force first. Then, by observing the first person's action, the second person will select his pushing action (i.e., a cooperative strategy) accordingly. Next, the third person will determine his action by observing the actions of the first two people. In this manner, once all the group members have determined their actions, they will execute the overall pushing operation through an approach of synchronization like simple communication or force/acceleration sensing. This cooperative strategy is known to work well in manual tasks. The same strategy is used here to develop the sequential $Q$-learning algorithm for multirobot transportation tasks.

The sequential $Q$-learning algorithm may be summarized as follows.

1) *Assume that there are n robots, $R_1, R_2, \ldots, R_n$, which are arranged in a special sequence. The subscripts represent their positions in this sequence.*
2) *$\Lambda_1, \Lambda_2, \ldots, \Lambda_n$ are the corresponding action sets available for the robots. In particular, the robot $R_i$ has $m_i$ actions available for execution as given by $R_i$: $\Lambda_i = (a_1^i, a_2^i, \ldots, a_{m_i}^i)$.*
3) *$Q_1(s,a), Q_2(s,a), \ldots, Q_n(s,a)$ are the corresponding Q-tables for the $n$ robots, where $s$ represents environmental states and $a$ represents actions of the robots. All entries in the Q-tables are initialized to zero.*
4) *Initialize the "Boltzmann temperature" factor $\tau$ to 0.99.*
5) *$\Psi$ is a set including all actions selected by the robots thus far, and $\phi$ represents the empty set. In addition, we assume a set $\Omega$ of all actions that can be taken by two or more robots simultaneously.*
6) *Observe the current world state s*
7) *Do repeatedly the following*:
   i) *Initialize $\Psi = \phi$*
   ii) *For ($i = 1$ to n)*
      a) *Generate the currently available action set $\Delta_i = (\Lambda_i - (\Lambda_i \cap (\Psi - \Psi \cap \Omega)))$*
      b) *The robot $R_i$ selects the action $a_j^i \in \Delta_i$ with probability*

$$P(a_j^i) = \frac{e^{Q_i(s,a_j^1,a_j^2,\ldots,a_j^i)}}{\sum_{c=1}^{k} e^{Q_i(s,a_j^1,a_j^2,\ldots,a_c^i)}} \qquad (1)$$

      *where $a_c^i \in \Delta_i (c = 1, 2, \ldots, k)$ and k is the size of the set $\Delta_i$*
      c) *Add action $a_j^i$ to the set $\Psi$*
      d) *$i \leftarrow i + 1$*
   iii) *End For*
   iv) *For each robot $R_i (i = 1, 2, \ldots, n)$, execute the corresponding selected action $a_j^i$*
   v) *Receive an immediate global reward g*
   vi) *Extract its local reward $r_i$ from the global reward g by using the Kalman filtering algorithm presented in Section III.*
   vii) *Observe the new state $s'$*
   viii) *For each robot $R_i (i = 1, 2, \ldots, n)$, update its table entry for $Q_i(s, a_j^1, a_j^2, \ldots, a_j^i)$ as follows*:

$$Q_i(s, a_j^1, a_j^2, \ldots, a_j^i) = (1-\varepsilon) Q_i(s, a_j^1, a_j^2, \ldots, a_j^i) + \varepsilon\Big(r_i + \mu \max_{a^1, a^2, \ldots, a^i} Q_i[s', a^1, a^2, \ldots, a^i]\Big) \qquad (2)$$

*where* $0 < \varepsilon < 1$ *is the learning rate and* $0 < \mu < 1$ *is the discount rate*.

ix) $s \leftarrow s', \tau \leftarrow \tau * 0.999$

In the sequential $Q$-learning algorithm, in each step of decision making, the robots do not select their actions at the same time. Instead, they select their actions one by one according to a predefined sequence that may be fixed or adaptive. Because the robots in this paper have the same sensing and pushing capabilities, their positions in the sequence will be determined randomly before the system begins to run, and will remain unchanged in the ensuing learning process. However, it is also possible for the sequential $Q$-learning algorithm to employ an adaptive sequence where the positions of the robots can be adjusted online based on their performance when a cooperative task is carried out. In addition, it is clear as well that the sequential $Q$-learning algorithm allows for heterogeneous robots with different capabilities because the algorithm assumes different action sets for different robots.

In the sequential $Q$-learning algorithm, every time a robot wants to select its action, it will observe which actions have been selected by the robots that precede it in the sequence. By not selecting the same actions as those of the preceding robots, this robot is able to successfully solve the behavior conflict problem and promotes effective cooperation with its teammates.

The benefits of the sequential $Q$-learning algorithm are obvious. Because each robot observes the actions of the robots preceding it in the sequence before it makes its own decision, the sequential $Q$-learning algorithm is likely to possess more effective cooperation than a single-agent $Q$-learning algorithm, thereby expediting the convergence in multirobot cooperative tasks. Furthermore, because a robot only observes the actions of a subset of its teammates, the sequential $Q$-learning algorithm results in a significantly smaller learning space (or $Q$-table) than that for the team $Q$-learning algorithm. In addition, in a multirobot box-pushing system, it is usually forbidden for any two robots to select the same action (pushing location) due to the space limitation. By selecting their actions according to a particular predefined sequence, the robots avoid selecting the same action that would result in behavior conflicts and cooperation failure in multirobot cooperative transportation tasks.

## III. Kalman-Filtering-Based Reward Estimation

When the conventional $Q$-learning algorithms (single-agent $Q$-learning or team $Q$-learning) are employed in a multirobot environment, there are several factors that can confuse the learning agents or even cause failure. First, as argued before, a multirobot environment is essentially dynamic. In a dynamic environment, the learning agent will find that it is difficult to assess the rewards received and update its learning process effectively [7], [10]. The second challenge results from the partially observed environment that is common in multirobot systems. The feature of partial observation makes it difficult for the learning agent to assess an action under a specific state. In particular, the agent may be confused when an action receives different rewards under the "same" world state. The third challenge in multirobot learning is the credit assignment problem. How to estimate the real reward of each robot from the global reward signal has become a key topic in multirobot $Q$-learning [10].

The issue of Kalman-filtering-based reward estimation was first studied by Chang *et al.* [15]. While their simulation results validated this approach in a simple grid world, they had assumed some crucial assumptions to simplify their model, which degraded its value to some degree.

This paper investigates cooperative and intelligent control of autonomous multirobot cooperative transportation systems in a dynamic, unstructured, and unknown environment. This problem is more challenging than the grid world game addressed in [15]. In particular, compared to the approach in [15], there are two important improvements in this paper. First, a more general noise process is assumed and a method to estimate and update the parameters of the noise process is incorporated into the algorithm to improve its performance. Both of them make the new learning algorithm generalizable and applicable to many multirobot systems. Second, the approach of Kalman-filtering-based reward estimation is integrated into the sequential $Q$-learning algorithm presented in Section II, which is an original proposition from us, which will promote cooperation among robots and speed up the convergence process.

### A. System Model

In the approach of Kalman-filtering-based reward estimation, the global reward received by a learning agent is thought to be the sum of its real local reward and a random noise signal caused by changes in the environment and unobservable world states. In particular, if the agent is in the world state $i$ at time $t$ and it receives a global reward $g_t$, then it can be expressed as

$$g_t = r(i)_t + b_t \tag{3}$$

where $r(i)_t$ is the real reward the agent receives in state $i$ at time $t$, and $b_t$ is an additive noise process that models the effect of the unobservable states on the global reward and evolves according to the following equation:

$$b_{t+1} = b_t + z_t, \qquad z_t \sim N(\mu, \sigma_w^2). \tag{4}$$

Here, $z_t$ is a Gaussian random variable with mean $\mu$ and variance $\sigma_w^2$. Based on these assumptions, the system model may be presented as

$$\begin{cases} x_t = Ax_{t-1} + w_t, & w_t \sim N(\Delta, \Sigma_1) \\ g_t = Cx_t + v_t, & v_t \sim N(0, \Sigma_2) \end{cases} \tag{5}$$

where

$$x_t = \begin{pmatrix} r(1)_t \\ r(2)_t \\ \vdots \\ r(|s|)_t \\ b_t \end{pmatrix}_{(|s|+1)\times 1}$$

is the state vector, $|s|$ is the total number of world states, $w_t$ is the system noise having a Guassian distribution with mean

$$\Delta = \begin{Bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \mu \end{Bmatrix}_{(|s|+1)\times 1}$$

and covariance matrix

$$\Sigma_1 = \begin{pmatrix} 0 & \cdots & 0 & 0 \\ \vdots & \ddots & & \vdots \\ 0 & & 0 & 0 \\ 0 & \cdots & 0 & \sigma_w^2 \end{pmatrix}_{(|s|+1)\times(|s|+1)},$$

and $v_t$ is the observation error, which is a zero-mean Gaussian white noise.

In addition, $\Sigma_2$ is set to zero because no observation error is assumed. The system matrix is $A = I$, and the observation matrix is $C = (\,0 \quad \cdots \quad 0 \quad 1_i \quad 0 \quad \cdots \quad 0 \quad 1\,)_{1\times(|s|+1)}$, where the $1_i$ occurs at the $i$th position when state $i$ is observed. In particular, if the current state is $s_i$, then the $i$th element and the last element in $C$ are equal to 1, and the remaining elements in $C$ are all zero.

### B. Kalman Filtering Algorithm

Kalman filters [16] are powerful tools to estimate states of a linear system with Gaussian noise. Here, the Kalman filtering algorithm is employed to dynamically estimate the real rewards and the noise through observing global rewards received by the learning agent. Then, the estimated real reward $r(i)_t$ in state $i$ at time $t$ instead of the global reward $g_t$ will be used to update the $Q$-table in the learning algorithm presented in Section II. The standard Kalman filtering algorithm is based on a system model with zero-mean Gaussian white noise. In order to employ the standard Kalman filtering algorithm, the system model in (5) is transformed into (6) as follows:

$$\begin{cases} x_t = Ax_{t-1} + Bu + \varepsilon_t, & \varepsilon_t \sim N(0, \Sigma_1) \\ g_t = Cx_t + v_t, & v_t \sim N(0, \Sigma_2) \end{cases} \tag{6}$$

where

$$B = \begin{pmatrix} 0 & & & \\ & 0 & & \\ & & \ddots & \\ & & & 0 \\ & & & & 1 \end{pmatrix}_{(|s|+1)\times(|s|+1)}, \quad u = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \mu \end{pmatrix}_{(|s|+1)\times 1}$$

$\varepsilon_t$ is a zero-mean Gaussian white noise signal with covariance matrix $\Sigma_1$, and $\Sigma_2 = 0$. A formal proof of justification for this system transformation is provided in the Appendix.

Based on the model given by (6), the Kalman filtering algorithm is presented next.

1) Initialize $x_0 = (0, \ldots, 0)^T$ the covariance matrix $P_0 = I$, $u = (\,0 \quad \cdots \quad 0 \quad 0\,)^T$, $\sigma_w^2 = 0.1$, and $t = 1$.
2) While (true)
   i) From current state $i$, select an action $a$ with $Q$-learning and execute it, observe a new state $k$, and receive a global reward $g_t$.
   ii) Update the estimate $\hat{x}_t$ and its covariance matrix $\hat{P}_t$ according to

$$\hat{x}_t = Ax_{t-1} + Bu$$
$$\hat{P}_t = AP_{t-1}A^T + \Sigma_1.$$

   iii) These *a priori* estimates are updated using the current observation $g_t$

$$C_t = (0, \ldots, 1_i, 0, \ldots, 0, 1),$$

   whose $i$th element is 1.

$$K_t = \hat{P}_t C_t^T (C_t \hat{P}_t C_t^T)^{-1}$$
$$x_t = \hat{x}_t + K_t(g_t - C_t\hat{x}_t)$$
$$P_t = (I - K_t C_t)\hat{P}_t.$$

   iv) Replace the local reward $r$ in (2) with $x_t(i)$ to update the $Q$-table.
   v) Reestimate the mean $\mu$ and variance $\sigma_w^2$ of the noise process with the history of $b_t$ (i.e., $x_{t-\mathrm{width}}(|s|+1), \ldots, x_t(|s|+1)$), and update $u$ and $\Sigma_1$ (*width* is the length of the history).
   vi) $t \leftarrow t+1, \qquad i \leftarrow k$
   vii) End While

### C. Online Parameter Estimation of the Noise Process

In the original approach proposed by Chang *et al.* [15], a value of $\sigma_w^2$, the covariance of the noise, has to be guessed before the Kalman filtering algorithm is run. However, in a real multirobot project, it is not practical, and usually very difficult, to guess this covariance value. In this section, an online estimation method is developed to estimate the covariance $\sigma_w^2$ with the history of $b_t$ (i.e., $x_t(|s|+1)$). Moreover, the mean $\mu$ is also estimated online because it is not assumed to be zero in the system model presented in (6). The estimation method is given next.

1) Initialize $\mu_0 = 0$, $\sigma_{w0}^2 = 0.1$, and $t = 0$.
2) Run the Kalman filtering algorithm for $n$ ($n > 200$) iterations with constant $\mu_0$ and $\sigma_{w0}^2$, and record the history values of $x_{t+1}(|s|+1), x_{t+2}(|s|+1), \ldots, x_{t+n}(|s|+1)$.
3) While (true)
   i) Estimate the mean and covariance as follows:

$$\mu_t = \frac{1}{n-1}\sum_{i=2}^{n}\left(x_{t+i}(|s|+1) - x_{t+i-1}(|s|+1)\right)$$

$$\sigma_{wt}^2 = \frac{1}{n-1}\sum_{i=2}^{n}\left(x_{t+i}(|s|+1) - x_{t+i-1}(|s|+1) - \mu_t\right)^2.$$

   ii) Run the Kalman filtering algorithm with $\mu_t$ and $\sigma_{wt}^2$, and record the value of $x_{t+n+1}(|s|+1)$.
   iii) $t \leftarrow t+1$
4) End While

## IV. Experimentation

A physical multirobot transportation system has been developed in the Industrial Automation Laboratory at the University of British Columbia. It is a distributed system consisting of several state-of-the art robots with local sensing capability. The mobile robots are manufactured by MobileRobots, Inc. (formerly ActivMedia Robotics Company), which is a main player in the robot market. In this project, one four-wheel-driven Pioneer 3-AT robot and two two-wheel-driven Pioneer 3-DX robots are used. They are agile, versatile intelligent mobile robotic platforms built on a core client–server model. The robots contain an embedded Pentium III computer, opening the way for onboard vision processing, Ethernet-based communications, laser, sonar, and other autonomous functions. An experiment is carried out to validate the SQKF in a real environment with sensor noise (i.e., the robot localization error due to the wheel slip, measurement noise of the laser/sonar distance finder, and so on).

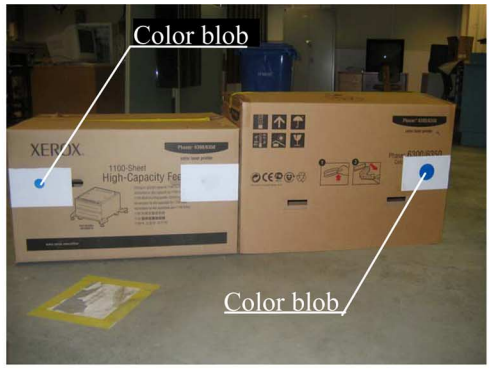
In the experiment presented here, an environment with one obstacle is incorporated to test the cooperative transportation capability of the developed system. In this system, three mobile robots equipped with the SQKF algorithm are used to transport a big box from one location to another within the environment.

In order to focus on the SQKF algorithm developed in this paper, a simple global coordinate system is employed. When the multirobot system begins to operate, each robot is given its initial position and orientation in this global coordinate system. Then, the robots will estimate and update their latest positions and orientations by recording and analyzing the data from the encoders mounted in their wheels and their compass sensors, while moving in the environment. Each robot can also inquire the current positions and orientations of its peer robots via the wireless Ethernet network. In addition, the robots know the global coordinate of the goal location in advance.

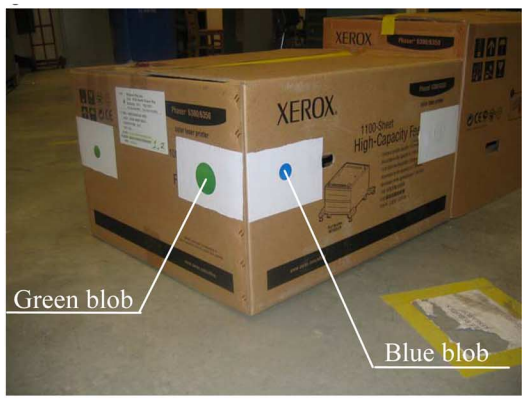
The transported object (box) and the obstacles are placed on the ground randomly. As a result, the robots have to search and estimate the poses of the box and the obstacles in the environment by fusing the sensory data from their sonar units, laser distance finders, and charge-coupled device (CCD) cameras. An approach has been developed by us to complete this task [17]. Furthermore, the sensors are assumed to only detect objects within a radius of 1.5 m although a robot can exchange its sensory information with its peers via wireless communication to establish a bigger local world state. In essence, this is a typical local sensing system, and the robots only know a local segment of the whole environment.

There are different color blobs on the four lateral sides of the box so that a robot can estimate the orientation and position of the box by identifying these color blobs with its own CCD cameras and fusing this information with the data from its sonar unit and laser distance finder. If an object without any color blobs is detected, it will be regarded as an obstacle in the environment. The box with the color blobs is shown in Fig. 1.

In the developed system, although each robot makes decisions independently, before it makes a decision, it needs to exchange information with its peers so as to form a cooperation strategy, as presented in Sections II and III. There is a computer server in the developed system, which is used to synchronize actions




(a)




(b)

Fig. 1. Color blobs used to identify the orientation of the box. (a) Big blue blob and a small blue blob on the same surface of the box. (b) Blue blob on one side and two green blobs on another side.

of the robots when they are ready to push the box. However, it does not serve as a centralized decision maker because the system is fully distributed, where each robot makes decisions independently. The use of the synchronization server here enables the designers to develop the program easily. This server may be replaced with other communication approaches such as broadcast communication.

Before the experiment is carried out, a simulation system is developed and employed to train the robots so that their $Q$-tables (knowledge bases) are improved. After 10,000 rounds of simulated box-pushing, the $Q$-tables of three simulated robots are exported to the three real robots to complete the physical cooperative box-pushing task.

A different policy from the simulation system is employed in the developed experimental system. In the simulation system, after each step of box-pushing, the robots will identify the new world state and select the corresponding actions with the SQKF algorithm. However, it is time-consuming and not practical for the real robots to frequently change their actions (pushing locations) in a physical environment. Instead, a new policy is employed here, where the robots will resume their actions selected in the previous step unless the reward earned in the previous step is lower than a predefined threshold value, which is selected by the designers through simulation. This policy enables the robots to avoid changing their actions frequently because changing the actions is so expensive and time-consuming for a physical multirobot box-pushing system.

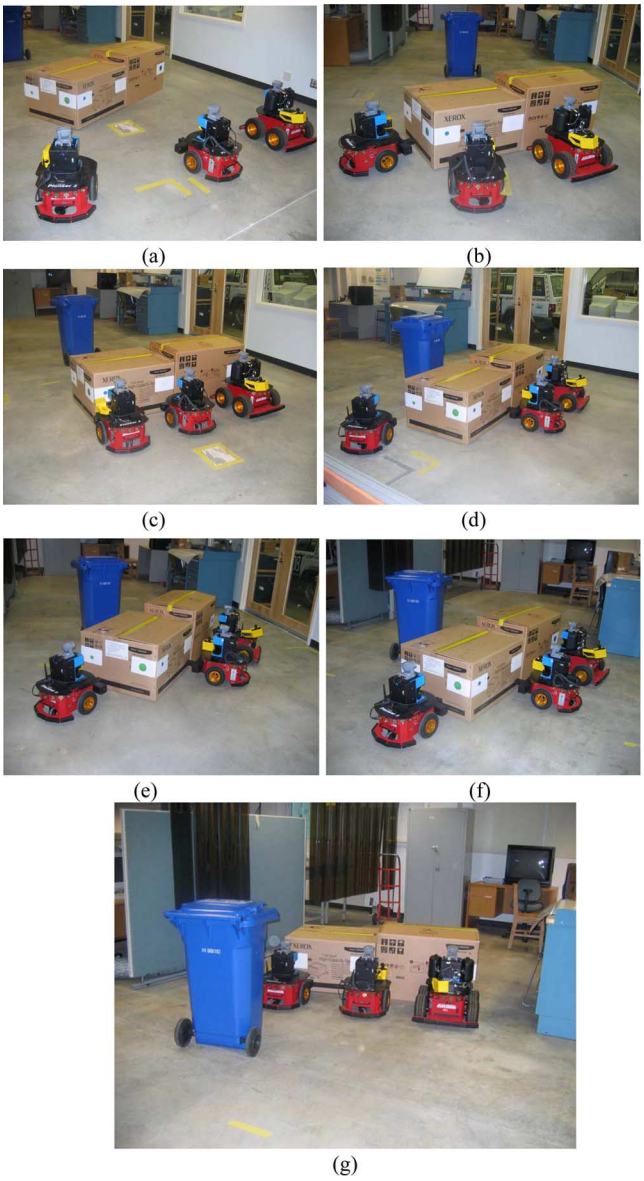



Fig. 2. Multirobot cooperative transportation in a real environment.

The experimental results of cooperatively transporting a box in a real environment are presented in Fig. 2.

In Fig. 2(a), a big box is placed on the ground, which is within the detection radius of the sensors of the mobile robots. The three robots are instructed about their initial positions and orientations in the global coordinate system before they begin to work. When the system starts to run, each robot uses its CCD camera to search and identify the color blobs on the box surface so that the orientation of the box relative to the current pose of the robot can be estimated. By fusing this relative orientation of the box with the depth data from its laser distance finder, the robot estimates the position and orientation of the box in the global coordinate system. If one robot cannot detect the box with its laser and vision sensors, it will communicate with other robots in the environment to request the position and orientation information of the box from them. If no robot finds the box, they will wander in the environment until one of them detects the box.

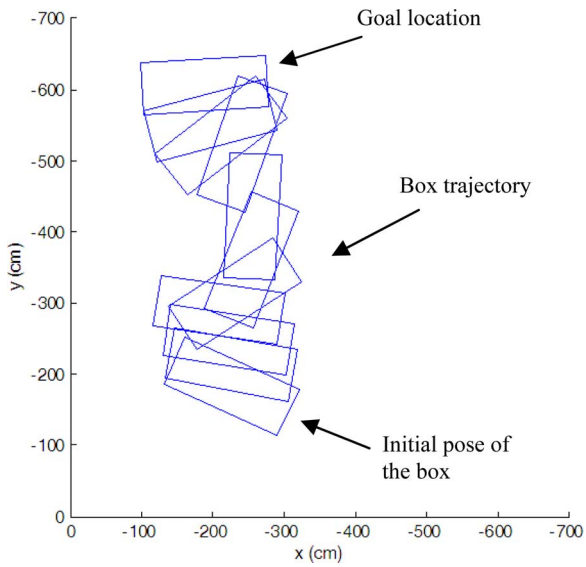



Fig. 3. Representative trajectory of the box in the experiment.

At the same time, each robot scans its local environment with the laser distance finder and identifies its peers that are close to it by requesting their positions through communication. If one object is detected, which is not the box or one of the peer robots, the object will be regarded as an obstacle.

By fusing the information of the box pose with the information of local obstacle distribution, a local world state is temporarily established by the robot, and the optimal action under this state is selected with the SQKF algorithm.

Fig. 2(b) shows how the robots push the box with the selected actions and Fig. 2(c) shows that the robots have changed to another formation so that the box is pushed with a bigger net force.

In Fig. 2(c), the robots detect an obstacle (the blue garbage bin) in the path, which had not been detected earlier by their sensors due to the limited detection radius. In order to determine the position and area of the obstacle, Fig. 2(d) shows how the first robot in the predefined sequence moves closer to the obstacle while measuring the distance between the obstacle and itself. The obstacle position estimated by this robot is sent to its two peers so that they can recalculate their local world states and select the corresponding actions so as to adapt to the new local world.

Fig. 2(e) and (f) shows how the robots have changed their formation to adapt to the new world state. Here, they attempt to change the orientation of the box so that the obstacle is avoided.

Fig. 2(g) shows that the robots have avoided the obstacle successfully and restored to the formation that generates the largest net pushing force.

From Fig. 2(a)–(g), it is observed that the SQKF-learning-based multirobot system has been successful in completing a cooperative transportation task in an environment with unknown obstacles. The learned $Q$-tables in the training stage help the robots select good cooperation strategies in a robust and effective manner. A representative trajectory of the box in the experiment is shown in Fig. 3.

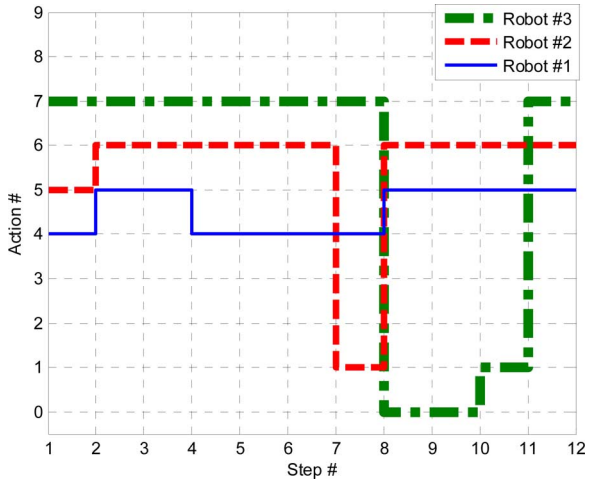



Fig. 4. Action switching history of the three robots in a round of box-pushing.

In Fig. 3, after each step of box-pushing, based on the feedback data from their laser distance finders and CCD cameras, the robots estimate and record the new pose of the box using a technique developed in our previous work [17]. Once the robots push the box to the goal location, the pose data collected in each step are put together to reconstruct the trajectory of the box, as shown in Fig. 3.

The action switching history of the three robots in the experiment is shown in Fig. 4.

A total of 15 trials of the experiment were completed. The robots could not complete the task in three of these trials. These failures were caused by incorrect path planning of a robot that did not avoid other robots or the box, in reaching its desired pushing position. In the successful trials, the robots usually spent 345–480 s in completing the task. Box-pushing experiments with multiple obstacles in the environment have not been attempted due to difficulties with path planning of multiple mobile robots and obstacle avoidance in a complex environment.

## V. Discussion and Future Work

The new SQKF algorithm is an extension of the single-agent $Q$-learning algorithm in the multirobot domain. However, it cannot guarantee to converge to the optimal policies because each robot does not observe the actions of all the other robots in the environment. Instead, before a robot makes a decision, it only observes actions of the robots preceding it in the sequence to promote its cooperation with them. Therefore, the environment is still dynamic in the eyes of this robot. As stated before, a $Q$-learning algorithm (including the SQKF algorithm) usually cannot guarantee to converge to optimal policies if the environment is dynamic [10].

The advantage of the SQKF algorithm is that it takes into account the existence of the other robots in the environment, which is not the case in the single-agent $Q$-learning algorithm when it is used in a multirobot domain, and attempts to estimate the effects of their actions on the rewards and state transition using a Kalman filter. Another advantage of the SQKF algorithm is that the sequential learning helps improve cooperation with other robots by observing the actions of the robots preceding it in the sequence, while maintaining a small learning space. Finally, although the SQKF algorithm cannot guarantee convergence to optimal policies in a dynamic environment, the simulation and experimental results show that it still enables the robots to find a good policy to complete the cooperative task.

If the disturbance caused by other robots in the environment can be estimated accurately by the Kalman filter, the SQKF algorithm may be able to converge to optimal policies even in a multirobot dynamic environment. However, a formal proof is not available to date. In order to estimate the disturbance of other robots, a nonlinear model and an extended Kalman filtering approach may be needed.

## VI. Conclusion

In this paper, a modified $Q$-learning algorithm suitable for decision making in multirobot cooperative tasks was developed. By arranging the robots to learn in a sequential manner and employing Kalman filtering to estimate the disturbances caused by other robots in the same environment, the developed learning algorithm displayed better performance than the conventional single-agent $Q$-learning and team $Q$-learning algorithms in a multirobot cooperative transportation project. In addition, from the algorithm formulation, it was evident that this new algorithm was generalizable and could be applied to many existing multirobot projects. The new $Q$-learning algorithm was implemented and validated using a prototype multirobot cooperative transportation system in laboratory. The experimental results showed that the developed system was able to successfully complete the desired task in a real environment with unknown obstacle distribution.

## Appendix

This appendix provides a formal proof for transforming the system in (5) to the one in (6) given in Section III. First, consider a more general case as follows. Let $X$ be a random variable that ranges over an open interval $(-\infty, +\infty)$ and has a probability density function $f_X(x)$ which is a continuous and positive function on this interval. Let $F_X(x)$ denote the probability distribution function of $X$. Thus, $F_X(x)$ is continuously differentiable in $x$, and $dF_X(x)/dx = f_X(x)$. Set $Y = aX + b$, where $a$ and $b \in \mathbb{R}$, and $a > 0$. Then, it is required to determine the density function $f_Y(y)$ of $Y$. It is determined as follows.

Assume that $F_Y(y)$ is the probability distribution function of $Y$. Then, one has

$$\begin{aligned} F_Y(y) &= P(Y \le y) \\ &= P((aX + b) \le y) \\ &= P\left(X \le \frac{y-b}{a}\right) = F_X\left(\frac{y-b}{a}\right). \end{aligned} \tag{7}$$

Thus,

$$\begin{aligned} f_Y(y) &= \frac{dF_Y(y)}{dy} = \frac{d}{dy}\left(F_X\left(\frac{y-b}{a}\right)\right) \\ &= \frac{1}{a} f_X\left(\frac{y-b}{a}\right). \end{aligned} \tag{8}$$

Return to (5) and (6), where $a = 1$, $b = \mu$, $Y = X + \mu$, and $X$ is a zero-mean normal distribution variable with a density function of

$$f_X(x) = \frac{1}{\sigma_w \sqrt{2\pi}} e^{-x^2/2(\sigma_w^2)}.$$

From (8), one obtains

$$f_Y(y) = \frac{1}{\sigma_w \sqrt{2\pi}} e^{-(y-\mu)^2/2\sigma_w^2}. \quad (9)$$

Equation (9) shows that $Y$ is now a normal distribution variable with mean $\mu$ and variance $\sigma_w^2$. Therefore, it can be concluded that a normal distribution variable with mean $\mu$ and variance $\sigma_w^2$ has the same distribution as the sum of the deterministic real number $\mu$ and a normal distribution variable with mean 0 and variance $\sigma_w^2$.

## REFERENCES


[1] T. Arai, E. Pagello, and L. E. Parker, "Guest editorial: Advances in multi-robot systems," *IEEE Trans. Robot. Autom.*, vol. 18, no. 5, pp. 655–661, Oct. 2002.

[2] N. Miyata, J. Ota, T. Arai, and H. Asama, "Cooperative transport by multiple mobile robots in unknown static environments associated with real-time task assignment," *IEEE Trans. Robot. Autom.*, vol. 18, no. 5, pp. 769–780, Oct. 2002.

[3] T. Huntsberger, P. Pirjanian, A. Trebi-Ollennu *et al.*, "Campout: A control architecture for tightly coupled coordination of multi-robot systems for planetary surface exploration," *IEEE Trans. Syst., Man, Cybern.*, vol. 33, no. 5, pp. 550–559, Sep. 2003.

[4] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 2002.

[5] E. Martison and R. C. Arkin, "Learning to role-switch in multi-robot systems," in *Proc. 2003 IEEE Int. Conf. Robot. Autom*, Taipei, Taiwan, pp. 2727–2734.

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

[7] Y. Wang and C. W. de Silva, "Multi-robot box-pushing: Single-agent $Q$-learning vs. team $Q$-learning," in *Proc. 2006 IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Beijing, China, pp. 3694–3699.

[8] M. L. Littman, "Value-function reinforcement learning in Markov games," *J. Cogn. Syst. Res.*, vol. 2, no. 1, pp. 55–66, 2001.

[9] M. L. Littman, "Markov games as a framework for multi-agent reinforcement learning," in *Proc. 11th Int. Conf. Mach. Learn. (ML 1994)*, New Brunswick, NJ, pp. 157–163.

[10] E. Yang and D. Gu. (2004). "Multiagent reinforcement learning for multi-robot systems: A survey," Tech. Rep., Univ. Essex, Colchester, U.K. [Online]. Available: http://robotics.usc.edu/∼maja/teaching/cs584/papers/yang04multiagent.pdf

[11] X. Yuan and S. X. Yang, "Multirobot-based nanoassembly planning with automated path generation," *IEEE/ASME Trans. Mechatronics*, vol. 12, no. 3, pp. 352–356, Jun. 2007.

[12] D. Sun and J. K. Mills, "Manipulating rigid payloads with multiple robots using complaint grippers," *IEEE/ASME Trans. Mechatronics*, vol. 7, no. 1, pp. 23–34, Mar. 2002.

[13] W.-T. Lo, Y. Liu, I. H. Eihajj, N. Xi, Y. Wang, and T. Fukuda, "Cooperative teleoperation of a multirobot system with force reflection via Internet," *IEEE/ASME Trans. Mechatronics*, vol. 9, no. 4, pp. 661–670, Dec. 2004.

[14] Y. Wang and C. W. de Silva, "A machine learning approach to multi-robot coordination," *Eng. Appl. Artif. Intell.*, vol. 21, no. 3, pp. 470–484, 2008.

[15] Y.-H. Chang, T. Ho, and L. P. Kaelbling, "All learning is local: Multi-agent learning in global reward games," presented at the Neural Inf. Process. Syst. (NIPS), Whistler, BC, Canada, 2003.

[16] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Trans. ASME, J. Basic Eng.*, vol. 82, ser. D, pp. 35–45, 1960.

[17] H. Lang, Y. Wang, and C. W. de Silva, "Mobile robot localization and object pose estimation using optical encoder, vision and laser sensors," in *Proc. IEEE Int. Conf. Autom. Logistics*, Qingdao, China, Sep. 2008, pp. 617–622.


**Ying Wang** (S'03–M'08) received the Bachelor's and Master's degrees from Shanghai Jiao Tong University, Shanghai, China, in 1991 and 1999, respectively, and the Ph.D. degree in robotics and mechatronics from The University of British Columbia, Vancouver, BC, Canada, in 2008.

Since 1999, he has been a Faculty Member at Ningbo University, Ningbo, China. He is currently a Postdoctoral Fellow in the Industrial Automation Laboratory, Department of Mechanical Engineering, The University of British Columbia. His current research interests include robotics, controls, mechatronics, and automation.

**Clarence W. de Silva** (S'75–M'78–SM'85–F'98) received Ph.D. degrees from the Massachusetts Institute of Technology, Cambridge, in 1978, and from the University of Cambridge, Cambridge, U.K., in 1998, and the Honorary D.Eng. degree from the University of Waterloo, Waterloo, ON, Canada, in 2008.

He has held the Mobil Endowed Professorship in the Department of Electrical and Computer Engineering at the National University of Singapore and Honorary Chair Professorship at the National Taiwan University of Science and Technology. He was a Senior Fulbright Fellow at the University of Cambridge, an Erskine Fellow at the University of Canterbury, Christchurch, New Zealand, a Lilly Fellow at Carnegie Mellon University, Pittsburgh, PA, a National Aeronautics and Space Administration (NASA)/American Society for Engineering Education (ASEE) Fellow, an Advanced Systems Institute of BC Fellow, and a Killam Fellow. He is currently a Professor of mechanical engineering at The University of British Columbia, Vancouver, BC, Canada, where he has held the Natural Sciences and Engineering Research Council of Canada (NSERC)-British Columbia (BC) Packers Chair in Industrial Automation since 1988. He also holds the Tier 1 Canada Research Chair in Mechatronics and Industrial Automation. He is the author or coauthor of 181 journal papers, 19 books, 18 edited volumes, 43 book chapters, and 207 conference papers, including the most recent books *Modeling and Control of Engineering Systems* (Taylor & Francis/CRC, 2009), *Sensors and Actuators—Control System Instrumentation* (Taylor & Francis/CRC, 2007), *Vibration—Fundamentals and Practice* (Taylor & Francis/CRC, 2007), *Mechatronics—An Integrated Approach* (Taylor & Francis/CRC, 2005), and *Soft Computing and Intelligent Systems Design—Theory, Tools, and Applications* (Addison-Wesley, 2004).

Prof. de Silva is a Fellow of the Royal Society of Canada, the American Society of Mechanical Engineers (ASME), and the Canadian Academy of Engineering. He is a Registered Professional Engineer in the Province of British Columbia, Canada. He has served on the editorial boards of 12 international journals including IEEE TRANSACTIONS and *ASME Transactions*, and as Editor-in-Chief of the *International Journal of Control and Intelligent Systems*, Editor-in-Chief of *the International Journal of Knowledge-Based Intelligent Engineering Systems*, Regional Editor, North America, of the IFAC *International Journal—Engineering Applications of Artificial Intelligence*, and Senior Technical Editor of *Measurements and Control*. He has been the recipient of several awards including the Killam Research Prize, the Paynter Outstanding Investigator Award, the Takahshi Education Award of the ASME Dynamic Systems and Control Division, the Outstanding Engineering Educator Award of the IEEE Canada, the Outstanding Contribution Award of the IEEE Systems, Man, and Cybernetics (SMC) Society, and the Meritorious Achievement Award of the Association of Professional Engineers of British Columbia.