Intelligent Cooperative Control for Urban Tracking with Unmanned Air Vehicles

Kevin Cook†, Everett Bryan*, Huilil Yu‡, He Bai‡, Kevin Seppi†, and Randal Beard*
*Electrical and Computer Engineering, Brigham Young University, Provo, Utah 84602
eabryan@gmail.com, beard@byu.edu
†Computer Science, Brigham Young University, Provo, Utah 84602
kevincook13@gmail.com, kseppi@byu.edu
‡UtopiaCompression Corporation, Los Angeles, California 90064
{huili,he}@utopiacompression.com

***Abstract*—We introduce an intelligent cooperative control system for ground target tracking in a cluttered urban environment with a team of Unmanned Air Vehicles (UAVs). We extend the work of Yu et. al. [1] to add a machine learning component that uses observations of target position to learn a model of target motion. Our learner is the Sequence Memoizer [2], a Bayesian model for discrete sequence data, which we use to predict future target location identifiers, given a context of previous location identifiers. Simulated cooperative control of a team of 3 UAVs in a 100-block city filled with various sizes of buildings verifies that learning a model of target motion can improve target tracking performance.**

I. Introduction

We introduce an intelligent cooperative control solution to the problem of persistently monitoring the road network of a city and tracking a ground target of interest with a team of Unmanned Air Vehicles (UAVs). The city is cluttered with buildings, trees, bridges, and tunnels, which may occlude the target from the sensor footprint of each vehicle. Each UAV is equipped with a high definition pan-tilt-zoom camera.

We use the term *intelligent cooperative control* as it is used in iCCA [3], to refer to a cooperative control system which learns from observations. Our system learns a model of target motion, from observations. A conventional use of observations of current target position is to update a Bayes filter for prediction of future target position, given an assumed model of target motion. A simple model of target motion may assign high probability to continuing in the current direction at the current speed, and low probability to a change in velocity. Rather than only use observations to update estimates of position assuming a model of motion, we use observations to learn a model of motion (see Figure 1).

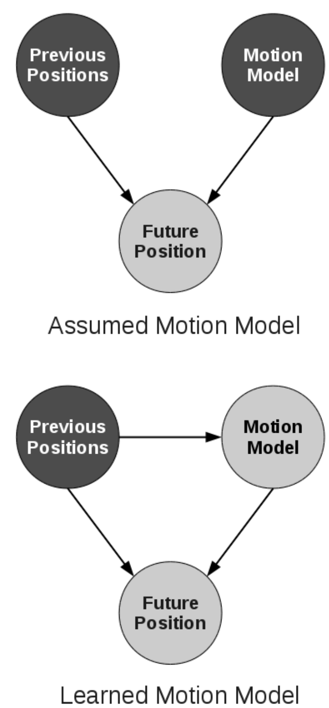

Fig. 1. Bayes nets illustrating contrast between a conventional approach to estimating future target position, using an assumed model of target motion (on the left), and our approach to estimating future target position, using a learned model of target motion. (Shaded circles represent random variables with given values.)

To facilitate such learning, we partition an area of interest into discrete locations. We then map each incoming observation to a location identifier, resulting in a sequence of identifiers which is used to train a model of target motion.

We refer to our model of target motion as a sequence model, because it predicts motion as a sequence of locations. Our sequence model leverages a state-of-the-art machine learning algorithm successfully implemented in statistical natural language processing and in text compression, the *Sequence Memoizer* [2], to learn a model of target motion. In Section II-F, we explain why we select the Sequence Memoizer and how we use it to estimate the probability that a target of interest will visit a specific location at a specific future point in time.

In Section II, we describe how we combine the learned sequence model with the cooperative probabilistic path planning system of Yu et. al. [1], to implement the intelligent cooperative control system.

In Section III, we simulate a team of three UAVs tracking a target in a 100-block city containing buildings of various sizes. We compare performance of our system to that of several baseline systems and verify that learning a model of target motion can improve target tracking performance.

II. Intelligent Cooperative Control System

Yu et. al. [1] introduce a probabilistic path planning system for cooperative target tracking using unmanned air and ground vehicles. We extend their work with the ability to learn patterns of target motion, to improve target tracking performance. Although the path planning system is designed for decentralized control, we do not explore the details of communication between vehicles as is done in related work [4].

A block diagram of the intelligent cooperative control system is shown in Figure 2. An auction algorithm provides efficient means for planning multiple vehicle paths to maximize the probability that at least one vehicle detects the target of interest. At the beginning of each time step t, each vehicle considers various paths to follow for the duration of a look-ahead window of length n time steps, and calculates a reward associated with each potential path. Calculation of reward is decentralized, in that each vehicle calculates its own rewards given information available from other vehicles. Each vehicle takes its highest reward to an auction. The vehicle with the highest reward wins the auction and follows the path associated with that winning reward. Remaining vehicles return to another auction, repeating the process until all vehicles determine a path.

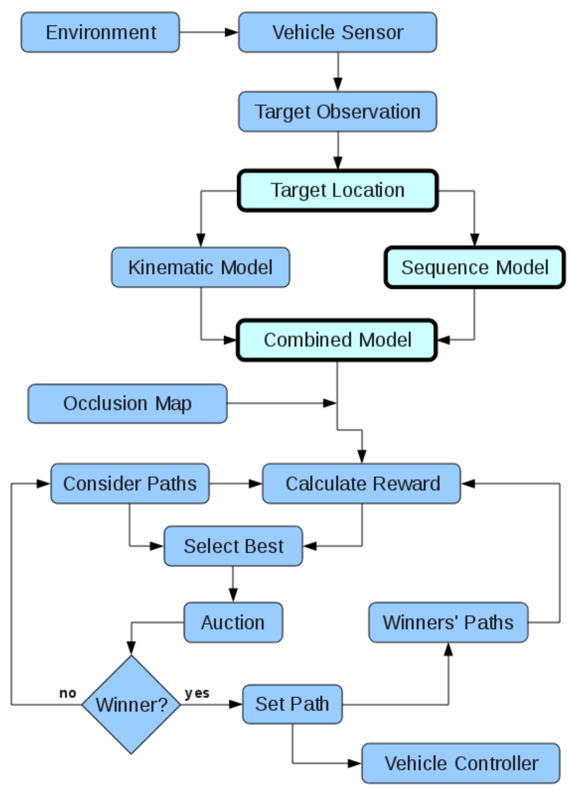

Fig. 2. Intelligent Cooperative Control System for Target Tracking (Our extension of Yu et al. [1] shown as highlighted blocks.)

We describe each component of the system and how it relates to the whole. Sections II-A - II-J describe various inputs required to calculate reward, and Section II-K describes calculation of reward. Remaining sections describe the auction process and vehicle control.

A. Environment

The environment is a cluttered urban area of interest $\mathcal{A}$ which we partition into some number of discrete locations, each with an associated location identifier l. A ground target of interest moves within the environment.

B. Vehicle Sensor

Each vehicle observes the environment with a sensor, attempting to detect the target.

C. Target Observation

We assume some process exists, either manual or automated, by which a target of interest is detected, given that the target is observable by the vehicle sensor. We also assume that when the target is detected, its position in the area of interest is observed.

D. Target Location

Target observations, and the lack thereof, are represented by a sequence of identifiers. At each time step, an identifier is appended to the sequence, either identifying the discrete location containing the observed position of the target, or a special identifier denoting that the target was not observed. The period of a time step is fixed and exactly one identifier is appended to the training sequence at each time step, regardless of whether the target moves from its current location, and regardless of whether the target is observed, resulting in a sequence which includes information about the speed at which the target moves.

E. Kinematic Model

Target observations are used by a kinematic model to estimate future target location, given an assumed model of target motion. Recent observations are used to determine the last known target location and direction. The motion model assumes with high probability that the target will continue in that same direction.

The kinematic model estimates what we refer to as a *location model* (essentially a dynamic occupancy grid [1]), a probability distribution over discrete locations for each time step j in a look-ahead window of length n. At each time step t, the probability $P(l|t,j)$ that the target is at location l at time step j in the look-ahead window, is estimated for all locations $l \in \mathcal{A}$ and for all time steps in the look-ahead window.

The estimated location model, or kinematic model $\hat{P}_k$, is combined with a sequence location model $\hat{P}_s$, to provide a combined location model $\hat{P}_c$.

F. Sequence Model

Target observations are used by a sequence model to learn a model of target motion, to estimate future target location.

1) Learning a Motion Model: The sequence of identifiers representing observations are used to train a learner, a machine learning algorithm which learns patterns in the training sequence. Given a context sequence of identifiers, the learner predicts subsequent identifiers. The learner we select is the Sequence Memoizer, an unbounded-depth, hierarchical, Bayesian nonparametric model for discrete sequence data [5]. We choose the Sequence Memoizer for the following reasons:

- It performs well in other sequence prediction tasks, such as language modeling and text compression [2], [5], [6].
- It can be trained on-line, meaning that the learner can be extended with additional training data at linear cost in both processing time and memory usage.
- It does not limit the length of context considered, which is important when prediction of future location depends upon more observations than just those at the previous few time steps.

2) Estimating a Location Model: At each time step t, the learner is used to estimate an intermediate location model $\hat{P}_{s'}$ as follows:

1) The learner is extended with an incoming observation, a location identifier or the unobserved identifier.
2) A fixed number n_r of probable target trajectories $\mathbf{q}_t^{(r)}$ $r \in \{1, 2, \cdots, n_r\}$, sequences of future locations, are generated by the learner, given a context of previous observations. The length of each trajectory is equal to the length n of the look-ahead window:

$$|\mathbf{q}_t^{(r)}| = n \tag{1}$$

3) The probability assigned to a location at a future time step $\hat{P}_{s'}(l|t,j)$ is equal to the proportion of generated tragectories, in which that location appears at that future time step:

$$\hat{P}_{s'}(l|t,j) = \frac{|\{r|\mathbf{q}_t^{(r)}[j] = l\}|}{n_r} \tag{2}$$

When a target is not observed in recent time steps, it becomes difficult for the learner to predict future target locations. To help mitigate this difficulty, a current sequence model $\hat{P}_s$ is estimated by combining the intermediate location model $\hat{P}_{s'}$ with recent sequence models, as follows: The probability assigned by the current sequence model to a location at a future time step is equal to the mean of the probabilities assigned to that location, at that absolute point in time, by recent sequence models and by the intermediate location model:

$$\hat{P}_s(l|t,j) = \left(\frac{1}{1+n-j}\right) \left(\hat{P}_{s'}(l|t,j) + \sum_{j'=1}^{n-j} \hat{P}_s(l|t-j', j+j')\right) \tag{3}$$

G. Combined Model

Both the kinematic model and the sequence model are used to estimate a combined location model $\hat{P}_c$. When only few observations have been acquired, the sequence model has difficulty learning a model of target motion, and is likely to be less effective than the kinematic model. Therefore, we use the kinematic model in addition to the sequence model. The combined model is the mean of the kinematic model and the sequence model:

$$\hat{P}_c(l|t,j) = \frac{1}{2}\left(\hat{P}_k(l|t,j) + \hat{P}_s(l|t,j)\right) \tag{4}$$

The combined location model is one of several inputs used to calculate reward associated with a considered vehicle path (see Figure 3).

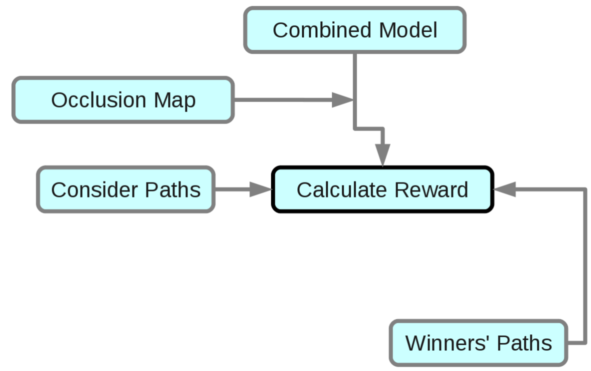

Fig. 3. Inputs required to calculate reward.

H. Occlusion Map

An occlusion map $\mathcal{M}$ provides information about objects which may occlude the target from view of a vehicle, and which are not expected to change often, such as buildings. The occlusion map is also used to calculate reward.

I. Consider Paths

Each vehicle i, which is not already an auction winner, considers a small number of potential paths for the duration of the look-ahead window, by calculating the reward $J(\Theta)$ associated with each path Θ. The path with the highest reward competes in an auction with other vehicles, which also have yet to win an auction at the current time step.

J. Winners' Paths

Auction winners K have paths Θ_K which are already determined, for the current time step. Before the end of the current time step, all vehicles have become auction winners, having determined paths. At the start of the next time step, there are no auction winners ($K = \emptyset$).

The paths of auction winners are used to calculate reward by each vehicle which is not yet an auction winner.

K. Calculate Reward

Reward is a function of the established paths of auction winners Θ_K, the combined location model $\hat{P}_c$, the occlusion map $\mathcal{M}$, and a considered path Θ_i

of a vehicle i which is not yet an auction winner $(i \notin K)$. We refer to auction winners K and the vehicle i considering a path, collectively as a set K' of proposed winners. Reward $J(\Theta_{K'})$ is the sum, over all time steps in the look-ahead window, of the reward associated with each time step:

$$J(\Theta_{K'}) = \sum_{j=1}^{n} P(D_{K'}|t, j, \Theta_{K'}, \mathcal{M}) \tag{5}$$

The reward associated with each time step is the probability $P(D_{K'}|t, j, \Theta_{K'}, \mathcal{M})$ that the target will be detected at that time step by any proposed winner, which probability is calculated by marginalizing a joint probability over all locations. The joint probability is factored into two parts: One factor is the probability $P(D_{K'}|\mathbf{x}_{K'}, l, \mathcal{M})$ that a target at location l is detected by any proposed winner, given positions $\mathbf{x}_{K'}(t, j, \Theta_{K'})$ of proposed winners at the future time step and the occlusion map $\mathcal{M}$. The other factor is the probability that the target will be at that location at the future time step $P(l|t, j)$, estimated by the combined location model $\hat{P}_c(l|t, j)$:

$$P(D_{K'}|t, j, \Theta_{K'}, \mathcal{M}) = \sum_{l \in \mathcal{A}} P(D_{K'}|\mathbf{x}_{K'}(t, j, \Theta_{K'}), l, \mathcal{M}) P(l|t, j) \tag{6}$$

The probability $P(D_{K'}|\mathbf{x}_{K'}, l, \mathcal{M})$ that a target at location l is detected by any proposed winner is calculated as one minus the probability that no proposed winner detects the target:

$$P(D_{K'}|\mathbf{x}_{K'}, l, \mathcal{M}) = 1 - \prod_{i \in K'} (1 - P(D_i|x_i, l, \mathcal{M})) \tag{7}$$

The term $P(D_i|x_i, l, \mathcal{M})$ is the probability that a target at location l is detected by vehicle i, and is discussed further in Section III-D6.

1) Adjustments: We adjust calculation of reward as follows, to accomplish the following objectives:

- Ensure that vehicles do not collide with each other. We set reward to a large negative value if any proposed winners' paths $\Theta_{K'}$ do not keep vehicles within safe distance from each other.
- Ensure that vehicles stay close to the area of interest. If a path leads a vehicle away from the area of interest, then we set reward to draw the vehicle towards the center of that area.
- Encourage vehicles to cooperatively search the area of interest. We add a small amount of probability mass to each location l of the location model $P(l|t, j)$, ensuring that all locations have at least some probability. Doing so ensures that even in the absence of any target observations, vehicles will cooperatively attempt to observe as many locations as possible at each time step.

L. Select Best

The path considered by a vehicle, which results in the highest reward, is the best path. At the auction, the vehicle bids the value of that highest reward.

M. Auction

Vehicles which have not yet determined paths for the current time step, compete in an auction. Each vehicle bids the value of the reward associated with its best path.

N. Winner

The highest bidder wins the auction, and sets its path accordingly. All remaining vehicles reconsider paths for the next auction.

O. Set Path

The auction winner sets its path to its best path, the auction winning path. This path is added to the list of winners' paths to be used by remaining vehicles to calculate reward for subsequent auctions.

P. Vehicle Controller

The low-level vehicle controller follows the determined winning path.

III. Experiments

We perform experiments to verify that learning a model of target motion can improve target tracking performance. We simulate a city, one ground target selected from five target types, three UAVs, our intelligent cooperative control system, and various baseline systems.

A. City

The simulated city is 10 by 10 blocks, where each block is 100 meters square. Each block contains at most one building, centered in the middle of the block. Building height is distributed exponentially, from 6 to 140 meters. Building length equals building width, which is distributed uniformly from 25 to 30 meters. A few blocks are empty.

The city is partitioned into 121 discrete locations, where each location is centered at an intersection.

B. Ground Target

The ground target travels in the center of streets and at speeds between 75 and 100 percent of a speed limit, which is sometimes 5 meters/sec and at other times 20 meters/sec.

The target is one of five target types, one deterministic, three probabilistic, and one Markov. We expect that the three probabilistic targets are somewhat representative of the behavior of targets in the real world. By contrast, we do not expect that the deterministic or the Markov target are similarly representative, however we do believe that they serve as interesting test cases. Each target type is described below:

- The deterministic target always follows the same path: counter-clockwise along the border of the city.
- The three probabilistic targets each repeatedly visit a list of waypoint intersections, with some uncertainty regarding the exact path to take between waypoints. Each target has its own list of waypoints and uncertainty parameter.
- The Markov target follows a Markov motion model, wandering aimlessly throughout the city. This target follows a model of target motion which is very similar to the model of target motion assumed by the kinematic model. Thus, the kinematic model has an inherent advantage tracking this type of target.

C. UAVs

Three UAVs fly over the city at an altitude of 150 meters and at a constant speed of 17 meters/sec. Each UAV contains one camera, gimbaled to point downwards, with a viewing angle of 75 degrees.

D. Intelligent Cooperative Control System

We implement the intelligent cooperative control system described in Section II, using the parameters described below:

1) Communication between UAVs: We assume perfect communication between UAVs. If the target is observed by one UAV, it is observed by all UAVs. Only one learner is shared among all UAVs.

2) Time Steps: Each time step t is 5 seconds. The length n of the look-ahead window is 3 time steps.

3) Kinematic Motion Model: The kinematic model assumes a model of target motion, as follows: At each time step, the probability of moving forward one intersection is 0.9. The remaining probability of 0.1 is uniformly distributed over three other possibilities, moving left one intersection, moving right one intersection, and staying at the current intersection.

4) Sequence Location Model: The sequence model learner estimates an intermediate location model $\hat{P}_{s'}$ by generating $n_r = 100$ probable trajectories using the previous 20 target observations as context.

5) Considered Paths: The set of paths considered by a UAV are those that result from choosing one of nine considered roll angles, and maintaining that chosen roll angle throughout the look-ahead window. The set of considered roll angles is $\{$-0.4, -0.3, -0.2, -0.1, 0, 0.1, 0.2, 0.3, 0.4$\}$ degrees.

6) Probability of Detection: To estimate a probability of detection for a single UAV, the probability $P(D_i|x_i, l, \mathcal{M})$ that a target at location l is detected by UAV i, we define a grid of sky points in the UAV plane of flight. The distance between adjacent sky points is 10 meters. UAV position x_i is estimated as the nearest sky point. Target position is estimated as the center of location l. We estimate the probability of detection as one, if the line of sight between estimated UAV position and estimated target position is not occluded by a building and if estimated target position is within the field of view of a UAV camera at the estimated UAV position. Otherwise, we estimate the probability of detection as zero.

7) Safe Distance: We define the safe distance between UAVs to be at least 50 meters.

E. Baseline Systems

We compare performance of our system to that of several baseline systems. Each system is described below:

- KM+SM This is our system, combining both the kinematic model and the sequence model.
- KM Only the kinematic model is used. It is same as our system except that the sequence model is disabled.
- SM Only the sequence model is used. It is same as our system except that the kinematic model is disabled.
- ML Only a maximum likelihood model is used. It is same as our system except that the kinematic model is disabled and the sequence model is altered. The smoothing mechanism of the learner is disabled, resulting in a maximum likelihood learner. All discount parameters of the Sequence Memoizer are forced to zero. We do this to explore the effect of smoothing, which sets aside probability for unseen events.
- None This system is same as our system except that both the sequence model and the kinematic model are disabled. UAVs randomly search the city attempting to collectively view as many intersections as possible.
- Fixed The team of UAVs follows a fixed pattern of flight. We manually design five different flight patterns, in an attempt to cooperatively search the city without any information about target position. UAVs follow circular paths of various diameters and positions. For each experiment, only the performance of the best fixed pattern is reported.

F. Results

We measure performance as the proportion of time the target is detected by any UAV. A target is detected if the line of sight between a UAV and the target is not occluded by a building and if the target is within the field of view of the UAV camera.

Figures 4-8 show the performance comparison of our intelligent cooperative control system (KM+SM) and each of the baseline systems, for the five target types. Performance is averaged over five Monte Carlo simulations. The proportion of time that the target is detected by any UAV, is always measured from the beginning of the simulation up to the current time step. For example, Figure 4 shows results for the deterministic target. At 500 time steps into the simulation, our system (KM+SM) detects the target approximately 22% of the time, of all time steps from the beginning up to that point in time. At the end of the simulation, 5000 time steps, our system detects the target approximately 40% of the time, of all time steps in the simulation.

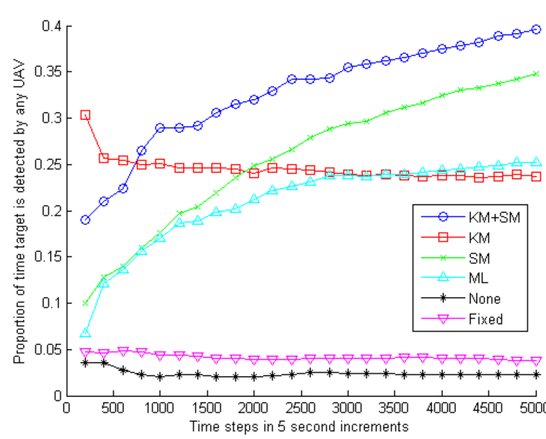

Fig. 4. Deterministic Target

Simulations verify that learning a model of target motion can improve the target tracking performance.

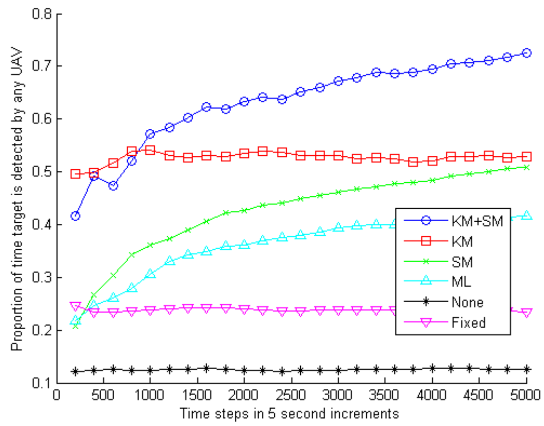

Fig. 5. Probabilistic1 Target

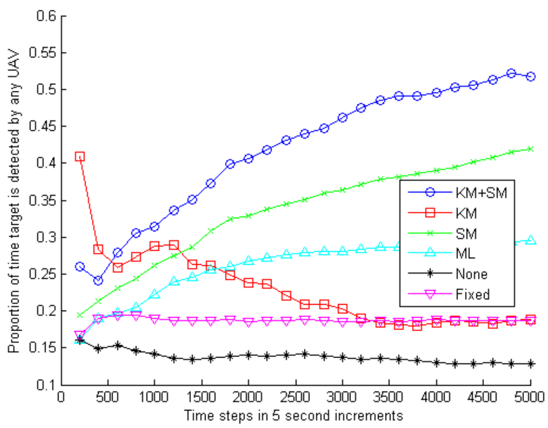

Fig. 6. Probabilistic2 Target

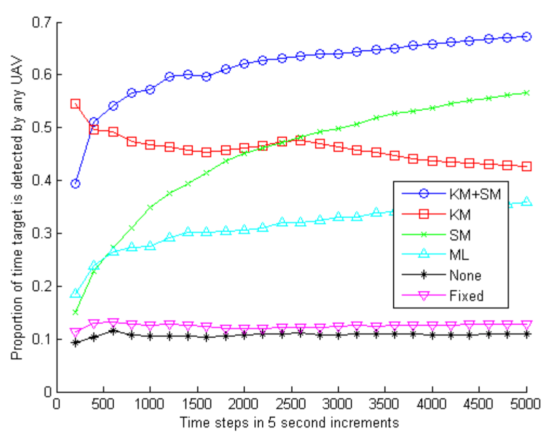

Fig. 7. Probabilistic3 Target

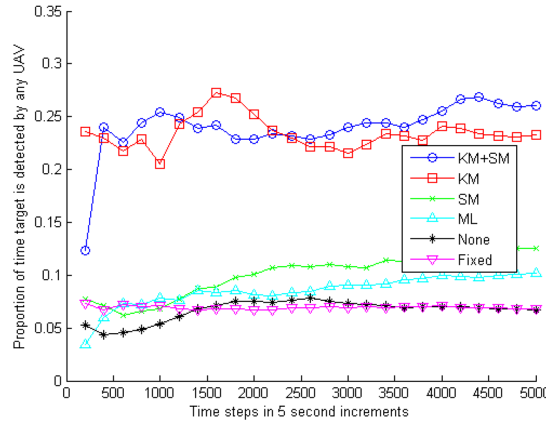

Fig. 8. Markov Target

Performance of all the three systems, KM+SM, SM, and ML, which incorporate learning, improves over time, as the number of observations increases. Also, our system performs better than any of the baseline systems we tested, after some initial learning period.

The effect of learning is least pronounced with the Markov target, due to the fact that its true motion model is to wander aimlessly about the city. Unlike the other targets, there is nothing to learn about how the target moves in relation to any specific location in the city.

IV. Conclusion

We introduce an intelligent cooperative control system for target tracking, which learns a model of target motion from observations. We simulate performance of the system using a team of three UAVs to track a ground target in a cluttered urban environment, and demonstrate that cooperative target tracking performance can improve as a result of the learned motion model. Performance improves over time, as more observations are obtained. Also, performance improves over any of the baseline models we evaluated, after an initial learning period.

Future work may explore alternate methods to combine the benefits of the kinematic and sequence models, such as using only the kinematic model until a sufficient number of observations have been acquired, and afterwards only using the sequence model. Future work may also include: simulating imperfect communication between UAVs, tracking multiple targets simultaneously, and experimenting with various methods for estimating probability of detection for a single UAV.

ACKNOWLEDGMENT

This project is funded by the Air Force under the Small Business Innovation Research contract no. FA8650-12-M-3217.

The views expressed are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

REFERENCES

[1] H. Yu, R. Beard, M. Argyle, and C. Chamberlain, "Probabilistic path planning for cooperative target tracking using aerial and ground vehicles," in *2011 American Control Conference*, July 2011, pp. 4673 –4678.

[2] F. Wood, J. Gasthaus, C. Archambeau, L. James, and Y. Teh, "The sequence memoizer," vol. 54, no. 2. ACM, 2011, pp. 91–98.

[3] J. Redding, A. Geramifard, A. Undurti, H. Choi, and J. How, "An intelligent cooperative control architecture," in *American Control Conference (ACC)*, Baltimore, MD, July 2010, pp. 57–62. [Online]. Available: http://people.csail.mit.edu/agf/Files/10ACC-iCCA.pdf

[4] M. Hirsch, H. Ortiz-Pena, and M. Sudit, "Decentralized cooperative urban tracking of multiple ground targets by a team of autonomous UAVs," in *Proceedings of the 14th International Conference on Information Fusion*, July 2011, pp. 1 –7.

[5] F. Wood, C. Archambeau, J. Gasthaus, L. James, and Y. W. Teh, "A stochastic memoizer for sequence data," in *Proceedings of the 26th Annual International Conference on Machine Learning*, ser. ICML '09. New York, NY, USA: ACM, 2009, pp. 1129–1136. [Online]. Available: http://doi.acm.org/10.1145/1553374.1553518

[6] J. Gasthaus, F. Wood, and Y. Teh, "Lossless compression based on the Sequence Memoizer," in *2010 Data Compression Conference*. IEEE, 2010, pp. 337–345.