RSSI-based Physical Layout Classification and Target Tethering in Mobile Ad-hoc Networks

Prashant P. Reddy and Manuela M. Veloso
Carnegie Mellon University, Pittsburgh, USA
{ppr|mmv}@cs.cmu.edu

***Abstract*— We investigate mobile ad-hoc indoor networks consisting of simple inexpensive robots, LANdroids, with limited wireless communication range and without any range or location sensors. We focus on the problem of using the mobile LANdroids to take responsibility for maintaining connectivity between a static Gateway and mobile Targets that move beyond the communication range of an established network. We refer to such a tracking task as *Target Tethering*. This type of network commonly uses IEEE 802.11 wireless protocols for communication, with Received Signal Strength Indicator (RSSI) as a measure of radio signal strength. RSSI data is noisy and poorly relates to distance in indoor environments, leading to a challenging Target Tethering task. Some algorithms use the trace of single-source RSSI data to infer distance between two nodes and use it to compute a Target Tethering policy. However, such distance estimates are poor. We instead aim at inferring physical network layout from RSSI data among multiple nodes. We introduce a novel approach based on *Cluster Geometries*, classes of network nodes corresponding to rotation-invariant physical layouts of LANdroids and a mobile Target, with the conjecture that multi-robot RSSI data can distinguish the Cluster Geometries and therefore the physical layouts. We proceed with extensive experiments and support our conjecture by showing successful classification of the designed Cluster Geometries given the multi-robot RSSI-based data. We then combine the estimated Geometries with motion patterns of the moving Targets to show that suitable multi-robot Target Tethering policies for unknown indoor environments can be learned using multi-agent reinforcement-learning. Specifically, we use an interesting variation of Q-learning where we first learn offline base policies in general open environments and later *specialize* the policies seamlessly during online execution to account for obstacles in the indoor environment.**

I. Introduction

LANdroids [1] are simple inexpensive robots designed to function as nodes in mobile ad-hoc networks that enhance the capabilities of human-robot teams in urban environments. Specifically, such networks extend the range over which the team can communicate with static Gateways that link the team to external communication channels. This scenario is relevant to important real-world situations. In urban conflicts, security forces often need to enter indoor environments where electronic communication is preferred over voice communication. Firefighters and other emergency personnel often need to enter large buildings whose communication infrastructure has been destroyed, so the team must deploy ad-hoc communication infrastructure.

Let $\mathcal{N} = \{N : N \in \mathcal{G} \cup \mathcal{T} \cup \mathcal{L}\}$ define a *LANdroids network* where:

- *Gateways*, $\mathcal{G} = \{G_i : i = 1..I, I \geq 1\}$, provide IP-connectivity to some communication channel;
- *Targets*, $\mathcal{T} = \{T_j : j = 1..J, J \geq 1\}$, typically represent humans carrying wireless radios;
- *LANdroids*, $\mathcal{L} = \{L_k : k = 1..K, K \geq 2\}$, relay IP-based communication between Gateways and Targets.

An example network, with a single Gateway, two Targets and eight LANdroids is shown in Figure 1. The Gateway is in a fixed physical location. The Targets are characterized by their unpredictable, although not random, movement about the environment as they attempt to complete their tasks.

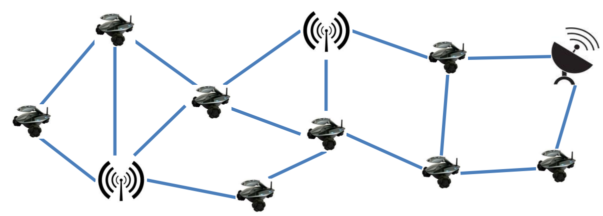

Fig. 1. An example of a LANdroids network showing one static Gateway, two mobile Targets and eight LANdroids.

When LANdroids are deployed, they self-organize by physically moving about until they form a network such that all Targets are directly or indirectly connected to one or more Gateways. The collective goal of the LANdroids in the network is to keep all the Targets connected to a Gateway. Let $c(N_1, N_2)$ be 1 if two network nodes, N_1 and N_2, have IP-connectivity between them; 0 otherwise. Then, in a fully connected graph over all $N \in \mathcal{N}$, for each T_j, there exists at least one path from that T_j to some G_i such that the product of $c(N_1, N_2)$ over all edges along that path is equal to 1.

We study the subsequent scenario where a Target, T_j, moves away from the established network thus risking loss of connectivity. We focus on the problem of moving one or more LANdroids such that the network can be stretched to maintain connectivity with the moving Target, T_j. We refer to this tracking task as *Target Tethering*.

LANdroids robots are envisioned as being inexpensive enough that they need not be retrieved after deployment. Because of this constraint, LANdroids do not have many sensors; the only sensor available for communication is a wireless radio which operates over a limited physical range on the IEEE 802.11 (Wi-Fi) protocols. *Received Signal*

Strength Indicator (RSSI) is a measure of radio signal strength in such wireless networks. RSSI data is noisy and poorly relates to distance in indoor environments, leading to a challenging Target Tethering task.

In this paper we present an approach for inferring physical layout of sections of the LANdroids network using RSSI data from multiple network nodes and using the inferred layout as a basis for Target Tethering. We introduce *Cluster Geometries*, classes of network nodes corresponding to rotation-invariant physical layouts of LANdroids and a mobile Target, with the conjecture that multi-robot RSSI data can distinguish the Cluster Geometries and therefore the physical layouts. We also classify *Target motion patterns* relative to Cluster Geometries using the RSSI data. We use the learned Cluster Geometries and Target motion patterns to develop a Target Tethering algorithm based on Markov Decision Processes (MDPs) and multi-agent reinforcement learning. We proceed with extensive experiments and support our conjecture by showing the successful classification of designed Cluster Geometries and Target motion patterns using only RSSI-based data. We also demonstrate, using a realistic simulation environment, that a multi-agent MDP policy, learned offline using Q-learning in an open environment, can be successfully specialized online with further learning for a specific indoor environment to accomplish the Target Tethering task.

II. Related Work

Kotz, *et al.*, [2] demonstrate that it is difficult to find a simple function that models the relationship between RSSI values and physical distance. Zickler and Veloso [3] show that it is possible for a LANdroid to probabilistically infer distance using a trace of RSSI values combined with motion odometry and to then use the inferred probability distribution to localize a stationary Target. They then use an auxiliary compass to synchronize the geographic orientation of the Target and the LANdroid. When the Target is in motion it communicates its odometry readings to the LANdroid and the LANdroid is able to follow the Target and keep it connected. While this approach is shown to work well in open environments, it is not as successful in environments with walls. We proceed with the conviction that while RSSI data is noisy and it relates poorly to physical distance, this relationship is not random and can be revealed as a useful pattern using RSSI data from multiple robots. Further, our approach to solving the Target Tethering task does not rely on auxiliary sensors, like compasses, and therefore solves for the LANdroids domain more accurately.

Ahmadi and Stone [4] provide a model for a *biconnected* LANdroids network that optimizes the physical layout of a network such that every Target in the network is connected to the same or different Gateways via two paths such that no link in the network is shared by both paths. Further, only one of the two paths is active at any given time and any network link that is part of an active path is called a *priority link*–a concept that we utilize in our approach.

III. Physical Layout Classification

When a LANdroids network is initially setup, the LANdroids are deployed in an unknown environment randomly such that they have no common geographic orientation frame and no knowledge of each other's relative physical location. The static Gateways and the mobile Targets are identifiable but their relative locations are also unknown. The lack of an environment map, the lack of a common geographic orientation frame, and the lack of initial relative location information all combine to make Target Tethering an extremely challenging task.

The simplest approach to Target Tethering is to follow the moving Target using the LANdroid closest to it. However, this poses two fundamental problems. Since we do not know the relative physical locations of the network nodes, we cannot identify the *closest* LANdroid. Since we do not have a common orientation frame, we do not know in which direction the LANdroid should move such that it follows the Target. Consequently, prior work in this domain has focused on the problems of (i) deducing physical distance from RSSI data, and (ii) acquiring a common orientation frame.

The wireless radios on the LANdroids are the only sensors available to estimate the relative layout of the network nodes in physical space. Specifically, LANdroids do not have physical distance sensors (e.g., RF range sensors, LIDAR) nor do they have geographic location or orientation sensors (e.g., GPS, compass) that could be used to estimate the layout. Therefore, our approach, which classifies the physical network layout in the neighborhood of the mobile Target using RSSI data alone is highly valuable.

Deducing distance from RSSI data is difficult due to several factors such as the presence of walls and other large obstacles between nodes. For example, two network nodes that are 20 meters apart and are separated by two concrete walls usually have a significantly lower RSSI value than two nodes at the same distance that are separated by wood plank walls. Since, a LANdroids deployment occurs in unknown environments, we do not know a priori how to scale observed RSSI values to reflect true physical distance.

We offer a radically different approach to this problem whereby it is unnecessary to map RSSI values to physical distance estimates. Instead we introduce the concept of *Cluster Geometries* which allow us to infer relative physical layout of network nodes without relying on the poor distance metric; we then use these Cluster Geometries to define a multi-agent MDP policy for the LANdroids in the Cluster to follow when the Target in the Cluster starts moving.

A. Clusters

A *Cluster*, C, in a LANdroids network, $\mathcal{N}$, consists of:

1) One Target, $T_j \in \mathcal{T}$;
2) Two or more LANdroid Cluster members, $L_k^C \in \mathcal{L}$, k=1,2,... from the *neighborhood* of the Target, T_j;
3) A *head node* role assigned to one Cluster member, L_H^C, which identifies that LANdroid as being responsible for coordinating the behavior of that Cluster.

In a given LANdroids network, at any specific time, only one LANdroid has the *priority link* to a specific Target, T_j [4]. That LANdroid claims for itself the role of the head node, L_H^C, for the Cluster associated with T_j. L_H^C then traverses the network to designate a small number of neighboring LANdroids as Cluster members. Figure 2 shows a sample Cluster.

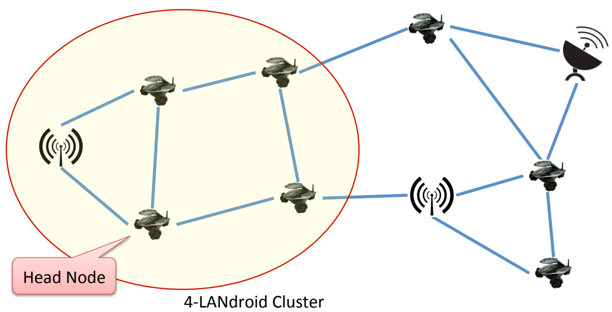

Fig. 2. The shaded area of the network shows a 4-LANdroid Cluster (κ=4) with the head LANdroid node connected to the Target.

Additional Cluster members are chosen by the head node on the basis of whether or not they are connected to the Target, T_j, connected to the Cluster head node, L_H^C, or connected to an already chosen Cluster member until κ Cluster members have been chosen. κ, usually in the range of 2 to 5, is set according to the expected density of LANdroids in the specific deployment. To illustrate, a 3-LANdroid Cluster (κ=3) and a 4-LANdroid Cluster are shown in Figure 3. In general, the larger the value of κ, the more useful RSSI data we can gather, but this needs to be balanced by the increased communication overhead amongst the Cluster members.

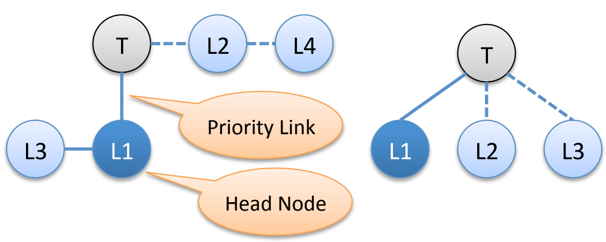

Fig. 3. A Cluster with 4 LANdroids (κ=4) and another with 3. The node marked *T* denotes the Target and the nodes market *L#* denote LANdroids.

B. Cluster Geometries

We next introduce the concept of *Cluster Geometry*. Without loss of generality, consider the three Clusters in Figure 4. Note that each Cluster has the same number of LANdroid Cluster members, κ=4. Priority links are denoted by solid lines and secondary links are denoted by broken lines. Further note that the three Clusters are indistinguishable from an IP network topology perspective. However, if we take as given that the Clusters actually represent the approximate physical layout of the Target and the LANdroids, we can then say that the three Clusters are indistinguishable in *Cluster configuration* but clearly distinguishable in *Cluster Geometry*. Therefore, Cluster Geometries distinguish the physical layout of Clusters that have identical network topology.

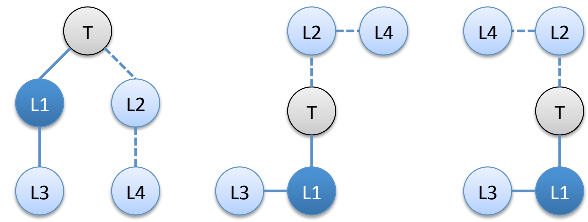

Fig. 4. Three κ=4 Clusters with identical configurations and IP network topology but distinct Cluster Geometries.

Cluster Geometries offer the following key benefits:

1) We can infer an approximate physical layout of the Target and the LANdroids in its neighborhood without relying on the highly unreliable pairwise RSSI-to-distance mapping. Note that inference of physical layout relies only on relative distances and not an estimating actual distances based on RSSI values.
2) The Cluster Geometry is rotation-invariant in physical space if it is used such that any actions by Cluster members are described relative only to the Geometry. For example, if L_2 is asked to move *Left* relative to the center Geometry in Figure 4, it can infer that to mean that it should move away from L_1 and L_4 whether or not that direction is the true geographic West.
3) Given a Cluster Geometry, Cluster members can synchronize their orientation frames by executing coordinated movement patterns and observing how the RSSI values to all the other Cluster members change.

We hypothesize that we can distinguish Cluster Geometries for a given Cluster configuration using only the RSSI data between the nodes in the Cluster. We demonstrate this to be true in Section V. The ability to classify Cluster Geometries is inversely related to the number of distinct Geometries that we try to distinguish amongst for the same Cluster configuration; *i.e.*, if a large number of only mildly different Geometries are introduced, then classification accuracy is reduced. As a domain-specific heuristic, if different Geometries result in the same MDP policy for the Target Tethering task, those Geometries can be collapsed into a single Geometry. In many situations, even just two Geometries–one Geometry with the Target outside the convex hull of the LANdroids (*e.g.* the first Geometry in Figure 4) and the second Geometry with the Target within the convex hull (*e.g.* the other two Geometries in Figure 4)–can offer a meaningful reduction in uncertainty about the physical layout of the LANdroids network in the vicinity of a Target.

Using multi-robot RSSI data from all the Cluster members allows us to reduce uncertainty about the Cluster Geometry even if we cannot identify the Geometry definitively. Since this approach relies only on very basic sensor data in the form of RSSI values, any reduction in uncertainty is valuable. Note further that this reduction of uncertainty is usefully applicable, outside of the LANdroids domain, to any set of network nodes with RSSI-based connectivity where it is helpful to distinguish whether the network nodes are, for example, stretched out in a line versus clustered in a polygon. Furthermore, other approaches such as LIDAR and RF range sensors, when available, can be used to further reduce uncertainty.

IV. Learning Target Tethering Policies

A. Target Motion Patterns

We assume that a Target's motion is always linear and categorize the motion pattern as being approximately:

1) away from the head node;
2) across from the head node; or
3) towards the head node.

Each of these motion patterns is illustrated in Figure 5. We hypothesize further that given a Cluster Geometry, we can determine the best match motion pattern based on how the RSSI values between the Target and the Cluster members evolve as the Target moves in physical space over time. Since the motion pattern is a time series signal, it is in general easier to classify than a static Cluster Geometry.

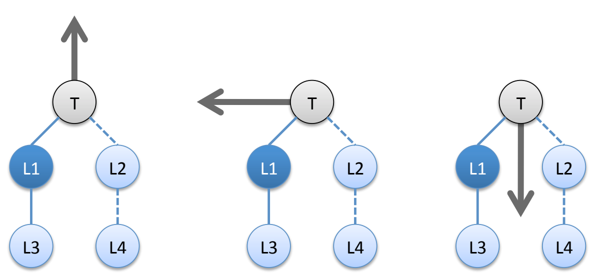

Fig. 5. Arrows show three distinct Target motion patterns: (1) away from head node, (2) across from head node, and (3) towards head node.

B. Multi-agent Q-learning

We aim to define an action policy for a given Cluster as:

$$\pi_C : G_C \times M_C \rightarrow A_C$$

where:

- G_C is the estimated Geometry of the given Cluster;
- M_C is the estimated motion pattern of the Target relative to the Cluster Geometry, G_C;
- A_C is the chosen compound action for all the LANdroids in the Cluster.

We use reinforcement learning to derive the action policy, π_C. To review briefly, the MDP for a single Cluster, C, is defined as:

$$M_C = \langle \mathcal{S}, \mathcal{A}, \delta, r \rangle$$

where:

- $\mathcal{S} = \{s_i : i = 1..I\}$ is a set of states;
- $\mathcal{A} = \{a_j : j = 1..J\}$ is a set of actions;
- $\delta(s, a) \rightarrow s'$ is a transition function;
- $r(s, a)$ is a reward function.

We define the states, $\mathcal{S}$, as the set of possible combinations of Cluster Geometries and Target motion patterns:

$$\mathcal{S} = \bigcup_{C \in \mathcal{C}} \mathcal{G}_C \times \mathcal{M}_C$$

where $\mathcal{C}$ is the set of all Cluster configurations. We define the actions, $\mathcal{A}$, as the set of compound action assignments where each LANdroid in the Cluster is assigned one of five actions for moving up, left, down or right or to stay in place:

$$\begin{aligned} &a_{L,C} \in \{Up, Left, Down, Right, Stay\} \\ &A_C = \{a_{L,C};\ \forall L \in C\} \\ &\mathcal{A} = \{A_C;\ \forall C \in \mathcal{C}\} \end{aligned}$$

The transition function, δ, is defined by numerous stochastic interactions within the simulation environment. The reward function, r, unknown to the MDP, is calculated by the environment as:

- +1, if the *Stay* action for all LANdroids in the Cluster for some time period, T, maintains connectivity to the moving Target with strong signal strength;
- +7, if an action increases signal strength to the Target;
- -5, if an action decreases signal strength to the Target;
- 0, otherwise.

State is shared amongst all the LANdroids in the Cluster, so the learning algorithm needs to be executed only on the head node. We use the Watkins-Dayan [5] Q-learning update rule which can be summarized as:

$$\hat{Q}_t(s, a) \leftarrow (1-\alpha_t)\hat{Q}_{t-1}(s, a) + \alpha_t[r_t + \gamma \max_{a'} \hat{Q}_{t-1}(s', a')]$$

where α_t is the learning rate: $1/numVisits(s, a)$.

The presence of walls in indoor environments makes the challenge of applying reinforcement learning to this problem particularly difficult. Since we do not know ahead of time the specific configuration of walls that the LANdroids will encounter at execution time, we cannot learn an MDP policy for that specific wall configuration. Instead, we learn a *base policy* for a given Cluster Geometry and Target motion pattern combination in an open environment. Then, during execution, we continue the learning process to *specialize* the policy for the specific wall configuration encountered.

For example, consider the scenario in Figure 6 which depicts the learned policy for a 4-LANdroid cluster. The arrows show the possible actions, for each LANdroid in the cluster, that could be suitable given the Target's motion pattern, shown by the thinner arrow. The relative lengths of the arrows indicate the relative magnitude of the Q-values for each of those actions. We see that the Up action of the bottom-right LANdroid has the highest Q-value. However, if the Cluster tries to execute this action on the bottom-right LANdroid, it may discover that there is a wall or some other obstacle that prevents that LANdroid from moving Up. We solve this problem using the following approach.

The Cluster categorizes the possible actions, $a_{L,C}$, of each LANdroid into *good* and *bad* actions based on whether the Q-values are above a state-specific threshold, $Q^{\tau}_{G_C,M_C}$. The

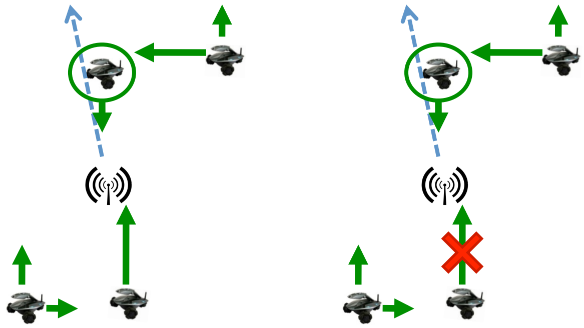

Fig. 6. Solid arrows show the learned policy for a 4-LANdroid Cluster, given a Target motion pattern shown by the dashed upward arrow, where the length of each solid arrow represents the magnitude of Q-values for that action, with (left) and without (right) the maximum value action available.

head node then identifies the Cluster member with the best combination of possible actions, by summing the Q-values for that LANdroid that are above the threshold:

$$Q^S_{L_C} = \sum_{a \in A_C : Q(a) > Q^{\tau}_{G_C, M_C}} Q(a)$$

We make a design choice to have only one Cluster member follow the Target at a given time although this is not a strict requirement. The head node assigns responsibility for following the Target to the Cluster member–including possibly itself–that has high Q^S. It does not simply choose the Cluster member with the highest value of Q^S, but instead chooses amongst the Cluster members according to a probability that is distributed proportional to the Q^S value of each Cluster member. The assigned Cluster member LANdroid then chooses an action from amongst its set of good actions again using a probability distribution proportional to the Q-values of each action. If the assigned LANdroid is successful in following the Target by maintaining good connectivity with the Target throughout the motion pattern, then the tethering task for the whole Cluster is considered successful for that motion pattern and all Cluster members continue with the $Stay$ action until the next time the Target moves.

On the other hand, if the assigned LANdroid is not successful in following the Target, *i.e.*, its RSSI value to the Target drops significantly or it is not able to increase its RSSI value sufficiently within a fixed time period, it reports that development to the head node, which then assigns the task of following the Target to the next most promising Cluster member. An action that encounters a wall or otherwise deteriorates signal strength to the Target accumulates negative reward, so the Q-value for that action decreases during its unsuccessful attempt to follow the Target. Therefore, even if the head node chooses that same LANdroid again for the next attempt, that LANdroid is more likely to choose a different action because the unsuccessful action's likelihood in the probability distribution is now lower. This approach effectively adapts, or *specializes*, the previously learned base policy to the current environment.

Our approach to Target Tethering can be applied recursively to achieve powerful results. For example, the Target in the Cluster is the base case or T_0. The LANdroid that ends up following the Target can then be viewed as T_1 and the other LANdroids in the network can be tasked with also tethering to T_1. This logic recursively leads to T_2, T_3, and so on to yield a chain of moving LANdroids where the sequence of the chain is determined at runtime in accordance with environmental constraints. Such behavior causes the whole LANdroids network to stretch with the moving Targets.

V. Experimental Results

We conducted four sets of experiments to test our hypotheses that Cluster Geometries and Target motion patterns can be classified using only multi-robot RSSI data and that effective Target Tethering policies can be learned using a combination of offline and online Q-learning. For the simulation experiments, we use a realistic simulator developed specifically for the LANdroids domain [6].

A. Geometry Classification in Simulation

We setup the three Cluster Geometries shown in Figure 4 in a representative indoor environment with walls. We collected multi-robot RSSI data for each Geometry by moving the Target and Cluster members to various locations within that indoor environment while still approximately maintaining the chosen Geometry. We choose sufficiently different locations for each observation such that the number and angles of walls between various nodes in the Geometry varies for each observation.

We extract features for classification from the RSSI data. We denote the RSSI value between the head node and the Target as $rssi_{HN}$, the RSSI value between Cluster member L_2 and the Target as $rssi_{L2}$ and so on.

$$R_i = \frac{rssi_{Li}}{rssi_{HN}}, \quad i = 2..|\{L \in C\}|$$

We use the R_i values as features for classification using a 10-fold cross-validated Support Vector Machine (SVM). We obtain test set classification accuracy of $94\% \pm 3\%$.

Normalizing the RSSI values for classification is a critical element of our procedure; it contributes a degree of *environment-invariance* to the trained SVM. If the SVM is trained using R_i values from an environment with wood walls but the online classification is done in an environment with concrete walls, the absolute RSSI values in the two environment will differ significantly but the normalized R_i values will be relatively consistent. Similarly, Geometries whose physical layout is smaller or larger than the one used for training will still have equivalent R_i values.

B. Geometry Classification using Robots

We also executed the simulation experiment described above using real robots. We collected multi-robot RSSI data over six floors of two different buildings. Constructing features as in simulation, we obtained test set accuracy of about 60%. We then added additional features, R_{ij}, capturing the relative RSSI values amongst the LANdroid Cluster members themselves. Figure 7 compares the normalized RSSI values for the first two 4-LANdroid clusters in Figure 4.

The plots represent 175 observations of 3 R_i values and 6 R_{ij} values, each R_i and R_{ij} computed relative to the RSSI value between the head node and the Target. We obtained classification accuracy of $84\% \pm 3\%$ with the added features.

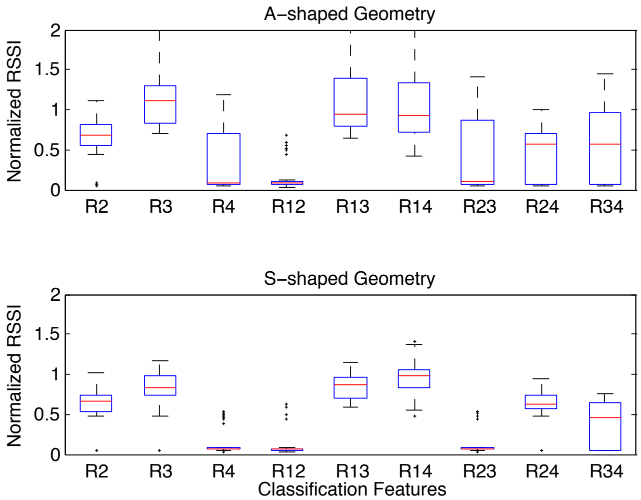

Fig. 7. Comparison of normalized RSSI values for 9 classification features from the A- and S-shaped Geometries shown in Figure 4.

C. Target Motion Pattern Classification

We setup the three Geometries of Figure 4 in the simulator and executed three Target motion patterns as in Figure 5. Figure 8 traces, for the first of the three Geometries, a sample time series of normalized RSSI values observed by each Cluster member. The normalization is done relative to the initial RSSI value between the head node and the Target. We find that we can distinguish different Target motion patterns for the same Geometry and also that the same Target motion pattern generates distinct traces for different Geometries.

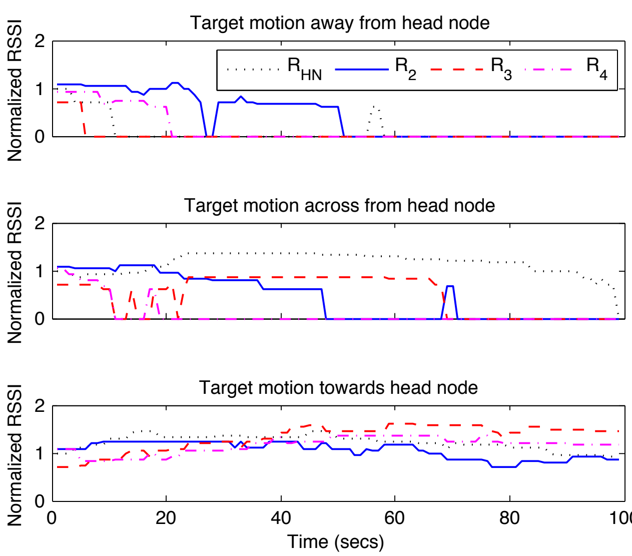

Fig. 8. Each subplot shows the distinct trace pattern of normalized RSSI values observed by the four LANdroid Cluster members for the three Target motion patterns represented in Figure 5.

D. Learning a Target Tethering Policy

We implemented the reinforcement learning approach described in Section IV in simulation. We setup the Geometry shown in Figure 5 and executed 50,000 training episodes of those Target motion patterns in an environment with no walls to obtain the base policy. The exploration-exploitation tradeoff is biased towards exploration during offline training and strongly biased towards exploitation for online execution. At the start of online execution the base policy is loaded and the MDP state is initialized to the estimated Geometry and Target motion pattern as classified by the Cluster. We find that the head node assigns responsibility for following the Target to different Cluster members until the Tethering task is successfully completed. The video attachment shows three scenarios where different Cluster members successfully execute Tethering actions when the Target moves away from the head node in a given 4-LANdroid S-shaped Geometry.

VI. Conclusion

In this paper, we contribute a solution to the Target Tethering problem in the LANdroids domain. We introduced Cluster Geometries, which characterize the physical layout of LANdroids in the vicinity of a Target and provide a solid foundation upon which Tethering policies relative to the Geometries can be defined. This approach avoids the difficult task of reliably mapping RSSI values to physical distance. We have shown through our experimental results, in simulation and using real robots, that such Geometries can be classified with a high degree of accuracy using only the multi-robot RSSI data from the Cluster. We further note that Cluster Geometries can be used to reduce uncertainty about physical layout of any Wi-Fi connected network nodes, outside of the LANdroids domain. We also contribute an interesting approach to Q-learning that first learn a base policy in a general environment and later specializes the policy for the encountered environment. Our experiments demonstrate that this approach produces successful Tethering policies. Future work in this area is expected to focus on methods to increase classification accuracy using features from additional sensors such as video cameras and LIDAR.

Acknowledgements

This research was partly sponsored by Intelligent Automation, Inc. (IAI) under subcontract no. 654-1, and by US Dept of the Interior under Grant No. NBCH-1040007. The views contained in this document are those of the authors only.

References

[1] M. McClure, D. Corbett, and D. Gage, “The DARPA LANdroids program,” in *SPIE*, vol. 7332, 2009.
[2] D. Kotz, C. Newport, and C. Elliott, “The mistaken axioms of wireless-network research,” Dartmouth College TR2003-467, Tech. Rep., 2003.
[3] S. Zickler and M. Veloso, “RSS-Based Localization and Tethering for Moving Robots in Unknown Environments,” in *ICRA*, 2010.
[4] M. Ahmadi and P. Stone, “Keeping in Touch: Maintaining Biconnected Structure by Homogeneous Robots,” in *AAAI*, 2006.
[5] C. Watkins and P. Dayan, “Q-learning,” *Machine Learning*, 1992.
[6] V. Manikonda, P. Ranjan, and Z. Kulis, “A Mixed Initiative Controller and Testbed for Human Robot Teams in Tactical Operations,” in *AAAI Fall Symposium*, 2007.