Using Semantic Fields to Model Dynamic Spatial Relations in a Robot Architecture for Natural Language Instruction of Service Robots*

Juan Fasola and Maja J Matarić, *Fellow, IEEE*

Abstract— **We present a methodology for enabling service robots to follow natural language commands from non-expert users, with and without user-specified constraints, with a particular focus on spatial language understanding. As part of our approach, we propose a novel extension to the semantic field model of spatial prepositions that enables the representation of dynamic spatial relations involving paths. The design, system modules, and implementation details of our robot software architecture are presented and the relevance of the proposed methodology to interactive instruction and task modification through the addition of constraints is discussed. The paper concludes with an evaluation of our robot software architecture implemented on a simulated mobile robot operating in both a 2D home environment and in real world environment maps to demonstrate the generalizability and usefulness of our approach in real world applications.**

I. Introduction

For autonomous service robots to provide effective assistance in real-world environments, they will need to be capable of interacting with and learning from non-expert users in a manner that is both natural and practical for the users. In particular, these robots will need to be capable of understanding natural language instructions for the purposes of user task instruction, teaching, modification, and feedback. This capability is especially important in assistive domains, where robots are interacting with people with disabilities, as the users may not be able to teach new tasks and/or provide feedback to the robot by demonstration.

Spatial language plays an important role in instruction-based natural language communication. For example, consider the following instruction given to a household service robot:

(1) Go to the kitchen

If the user says (1), the robot should understand, in principle, what that means. That is, it should understand which task among those within its task/action repertoire the user is referring to. In this example, the robot may not know where the kitchen is located in the user's specific home environment, but it should be able to understand that (1) expresses a command to physically move to a desired goal location that fits the description "the kitchen".

* Research supported by National Science Foundation grants IIS-0713697, CNS-0709296, and IIS-1117279.

J. Fasola is with the University of Southern California, Los Angeles, CA 90089 USA (e-mail: fasola@usc.edu).

M. J. Matarić is with the University of Southern California, Los Angeles, CA 90089 USA (e-mail: mataric@usc.edu).

The path relation in (1) was expressed with the use of the preposition "to". Spatial relations expressed by language are often expressed by prepositions [1]. Therefore, the ability for robots to understand and differentiate between spatial prepositions in spoken language is critical for the interpretation of user-guided instructions to be successful.

Spatial language understanding is also especially relevant for interactive robot task learning and task modification. Continuing with the household robot example, the user might teach the robot the complex task "Clean up the room", through natural language, by specifying the subgoals of that task individually, each represented by its own spatial language instruction (e.g., "Put the clothes in the laundry basket", "Stack the books on top of the desk in the right-hand corner", "Put all toys under the bed", etc.). In addition, user modification of known robot tasks can also readily be accomplished with spatial language. For example, the user might modify the task defined by (1) by providing spatial constraints, or rules, for the robot to obey during task execution, such as "Don't go through the hallway," or "Move along the wall." These user-defined constraints do not change the meaning of the underlying task, but allow the user to interactively modify the manner in which the robot executes the task in the specific instance.

Finally, spatial language can be used to provide teacher feedback during task execution, to further correct or guide robot behavior. In the context of our example, as the robot is moving along the wall en route to the kitchen, the user may provide additional feedback by saying "Move a little further away from the wall," or "Move close to the wall but stay on the paneled floor". These examples illustrate the importance of spatial language in the instruction and teaching of in-home service robots by non-expert users. In this paper, we present an approach for enabling autonomous service robots to follow natural language commands from non-expert users, including under user specified constraints such as those mentioned, with a particular focus on spatial language understanding.

II. Related Work

The use and representation of spatial prepositions, and spatial language in general, in human-agent interaction scenarios has been investigated by previous work. Skubic et al. [6] developed a mobile robot capable of understanding and relaying static spatial relations (e.g., "to the right", "in front of", etc.) in natural language instruction and production tasks. The use of computational field models of static relations has also been explored in the context of human-

robot cooperation tasks [5], and for visually situated dialogue systems [11]. These works all incorporated pre-defined models of spatial relations, however, researchers have also examined learning these types of static spatial relations both online during interaction and offline from a corpus of training data (e.g., [8, 12, 13]). Our approach extends upon this related work by modeling not only static spatial relations for natural language instruction understanding, but also dynamic spatial relations involving paths, as presented in the following section.

Recent work has, however, explored the use of dynamic spatial relations in the context of natural language robot instruction. Tellex et al. [3] constructed a probabilistic graphical model to infer spatial task/actions commanded through natural language for execution by a forklift robot. Kollar et al. [4] presented a Bayesian approach for the interpretation of route directions on a mobile robot, using learned models of dynamic spatial relations (e.g., "past", "through") from a set of positive and negative schematic training examples. In both of these works the representations of the spatial relations used, static or otherwise, were not pre-specified and instead were derived from labeled training data. However, these approaches typically require the system designer to provide an extensive corpus of labeled natural language input for each new application context, without taking advantage of the domain-independent nature of spatial prepositions. In contrast, our approach develops novel, pre-defined templates for spatial relations, static and dynamic, that facilitate use and understanding across domains, and whose computational representations enable guided robot execution planning.

Methods for mapping natural language instructions onto a formal robot control language have also been developed by researchers using a variety of types of parsers, including those that were manually constructed [9, 10], learned from training data [17], and learned iteratively through interaction [18]. Among these examples, the work of Rybski et al. [9] and Matuszek et al. [17] relied on pre-defined robot behaviors as primitives, as opposed to spatial relations, which limits, if not restricts, the user's ability to introduce feedback modifications and/or constraints on robot execution of a specific primitive behavior. The work of Kress-Gazit et al. [10] and Cantrell et al. [18] leave the definition of primitives up to the system designer, however, the parsers utilized in their systems mapped words to meanings based on dictionary-based rules. Our methodology employs domain-generalizable spatial relations as primitives, and probabilistic reasoning for the grounding and semantic interpretation of phrases, thereby enabling context-based instruction understanding and user-feedback modifiable robot execution paths.

III. Approach and Methodology

In this section we present our methodology for autonomous service robots to receive and interpret natural language instructions involving spatial relations from non-expert users. Our approach is motivated by related research in linguistics, cognitive science, neuroscience, and computer science, and proposes the encoding of spatial language within the robot *a priori* as primitives, with particular focus on the representation of prepositions. Specifically, our approach extends the semantic field model of spatial prepositions, proposed by O'Keefe [2], to include dynamic spatial relations and provides a computational framework for human-robot interaction which integrates the proposed model.

A. Semantic Fields

The *semantic field* of a spatial preposition is analogous to a probability density function (pdf), parameterized by schematic figure and reference objects, that assigns weight values to points in the environment depending on how accurately they capture the meaning of the preposition (e.g., points closer to an object have higher weight for the preposition 'near'). This field representation for the semantics of spatial prepositions, while based on insights gathered from neuroscience research in rats, was shown by O'Keefe [2] to closely resemble the form and continuous nature of spatial preposition representations demonstrated by humans [20]. These types of continuous spatial field functions have also been shown to transfer seamlessly into higher dimensions [16], thus enabling similar relational comparisons in 2D and 3D space. Example semantic fields are shown in Fig. 1 for the static prepositions "near", "away from", and "between" for illustration purposes. The semantic field for *near* was produced by calculating the weights ($\mathbb{R}[0,1]$) for each point in the environment using the following equation:

$$f_{near}(dist) = \exp[-(dist^2)/2\sigma^2] \quad (2)$$

Where *dist* is the minimum distance to the reference object; σ is the width of the field (dropoff parameter) which is context-dependent. The equation in (2) utilizes a Gaussian for the computation of the field, however other exponential or linear functions could instead be applied depending on the domain requirements. For further information regarding static field computation, we refer the reader to [2].

B. Modeling Dynamic Spatial Relations

While appropriate for static relations, the semantic field model, by itself, is not sufficient for representing dynamic spatial relations that involve paths. *Paths* are comprised of a set of points connected by direction vectors that define sequence ordering. Path prepositions include, among others: to, from, along, across, through, toward, past, into, onto, out of, and via. To account for paths in the spatial representation of prepositions, our approach employs multiple methods. The primary method modifies the traditional semantic field model with the addition of a weighted vector field at each point in the environment. As an example, the preposition "along" denotes not only proximity, but also a path parallel to the border of a reference object. Thus, in our proposed model, the semantic field for *along* contains not only weights for each point in the environment to encapsulate proximity, but also weighted direction vectors at each point to encapsulate optimal path direction. Among these direction vectors, those that coincide with the meaning of the relation are favored (in this example, those more parallel to the reference object have higher weight). By multiplying the

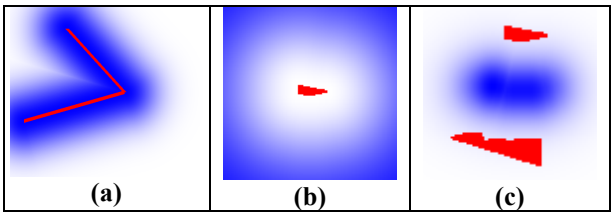

Figure 1. Semantic fields for static prepositions (a) near; (b) away from; (c) between.

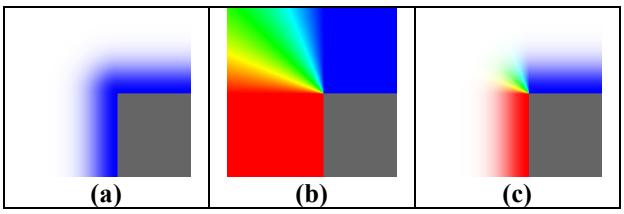

Figure 2. Semantic field for "along" (a) near subfield; (b) direction subfield (90°= red, 0°= blue); (c) combined field.

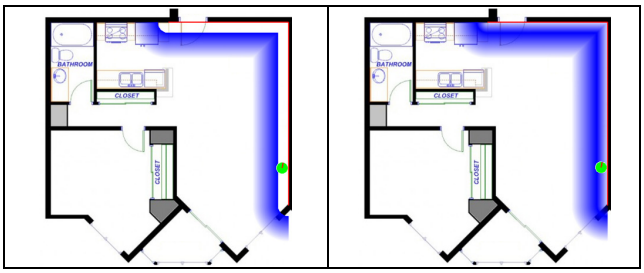

Figure 3. Two example alterations to semantic field due to user feedback statement "Move a little away from the wall"

weights of these two subfields together (proximity and path direction) at each point in the environment, we are able to produce the semantic field for the dynamic spatial relation *along* (see Fig. 2).

The advantage of modeling spatial relations as pdfs, as opposed to using classification-based methods (e.g., [4]), is that generating robot action plans for instruction following is as simple as sampling the pdf, which can be used to find solution paths incrementally (one path segment at a time). In other words, there is no need to search the action space (randomly or exhaustively) to find appropriate solutions by classifying candidate paths as a whole, which may be prohibitive in time-complexity. Furthermore, user teaching, feedback, and refinement of the robot task execution plan can easily be incorporated as an alteration of the pdf. For example, the feedback statement "Move a little away from the wall" could alter the semantic field of the task by attributing higher weight to points further from the wall from the robot's current location; for example, by shifting the entire field over, or by simply shifting the mean of the field. Fig. 3 illustrates these two forms of field alterations for the task "Walk along the wall".

While it is true that some dynamic spatial relations can be modeled by specialized semantic fields that capture optimal path direction at a local level (e.g., along, toward, up, down, etc.), many path prepositions require the existence of certain characteristics achieved at a global level in order to satisfy their meaning. To represent these more complex prepositions, our approach identifies four classical AI conditions that each path preposition may subscribe to, they are: 1) *pre-condition*, 2) *post-condition*, 3) *continuing-condition*, and 4) *intermediate-condition*. A unique characteristic of our model is that each condition is represented by either a semantic field, or by another path preposition (which is in turn represented by semantic fields). In addition, each path preposition may contain none, one, or many of each of the four conditions, but must have at least one identifiable condition in its representation. For example, "to" has a single post-condition containing the semantic field for *at*, signifying that the path denoted by *to* terminates *at* the region in question; here the reference object for the field is passed in as a parameter to the preposition. "From" is the reverse of "to", with the *at* field as a pre-condition. The paths "into" and "onto" are both special cases of "to", wherein the *at* field post-condition is replaced by the fields for *in* and *on*, respectively. One example of a path with intermediate conditions is "through", which contains *into*, *along*, and *out of*, as ordered conditions. "Across" is the same as "through", but with movement along the minor axis of the reference object as opposed to the major axis.

These condition-based representations were incorporated into our model of dynamic spatial relations in light of findings from linguistics and cognitive science research into the meanings of path prepositions, which suggest the existence of such constraints [1, 19]. For more information regarding our approach to modeling dynamic spatial relations with global properties, we refer the reader to [7].

IV. Robot System Modules and Architecture

Our robot software architecture contains five system modules that enable the interpretation of natural language instructions, from speech or text-based input, and translation into agent execution. They include: the syntactic parser, noun phrase (NP) grounding, semantic interpretation, planning, and action modules. The following sections discuss the primary modules in detail.

A. Syntactic Parser

Natural language instructions are received by the syntactic parser as textual input. The text string may be provided by a speech recognizer (e.g., [21]) or keyboard-based input. While both methods have been implemented with our system, we focus this discussion on well-formed English sentences provided via keyboard input.

The first step of the syntactic parser is to extract the part-of-speech (POS) tags from the natural language text string; these tags identify words in the input as nouns ('N'), verbs ('V'), adjectives ('A'), determiners ('Det'), etc. Our system uses the Stanford NLP Parser [15] for extracting the POS tags for all words, except for the prepositions ('P'), which are instead identified using a lexicon for single and multi-word prepositions (e.g., "to", "away from", "in line with").

Our system does not attempt to provide a solution for natural language processing in the general case, but instead focuses on directives, and more specifically, on natural language English instructions involving spatial language.

To parse these instructions, a phrase structure grammar is utilized. Following are the constituency rules:

S → V P* NP NP → N' P+ NP

N' → (Det) A* N+ NP → NP and NP

NP → N'

Here, S defines a valid sentence, NP a noun phrase, and N' a terminal noun phrase. It is important to note that the grammar presented, although limited, is capable of parsing spatial language sentences that do not contain prepositions (e.g., "Enter the room"), those with multiple prepositions (e.g., "Come up on over here"), as well as partial parses of well-formed English sentences (e.g., "PR2, can you please *wait at the counter by the entryway*, thanks").

B. *Grounding Noun Phrases*

After the noun phrases in the natural language input are identified by the syntactic parser, the cognitive system attempts to ground the NPs in its representation of the world. Due to the hierarchical nature of NPs, the grounding process is recursive; it first attempts to ground any child NPs before expanding to ground root NPs. To perform this grounding procedure, the nouns in the NP are first checked against the system's knowledge base of labels for grounds (e.g., objects, rooms, etc.) in the world. These labels are domain-dependent and can either be learned online or, as in our system, loaded from a file along with additional world details, including: a map of the environment, object properties, grounding types, and the locations of known objects in the map.

If the knowledge base is unable to find a matching label, the grounding process fails, at which point the system may prompt the user for additional information and/or clarification. If a single match is found, the NP is successfully grounded. Lastly, if multiple matches are found, the system relies on higher-level NPs (for a child NP), or the user (for a root NP), for disambiguation.

In our methodology, disambiguation of multiple matches for a child NP is accomplished in two steps: 1) the semantic field for the prepositional phrase of the child NP's root NP is computed, and 2) each of the candidate grounds are evaluated against the computed semantic field to find the optimal match for the NP.

To illustrate this probabilistic, semantic field-based grounding procedure, consider the instruction "Go to the table by the kitchen". First, the syntactic parse of the input is obtained (see Fig. 4(a)), yielding a single root NP with two children NPs ("the table" and "the kitchen"). In our example world, there is a single ground match for "the kitchen", but there are five possible groundings for "the table". To disambiguate among the five candidate groundings, the semantic field for *near* (determined by the use of the preposition "by" in the root NP) is computed for the reference object (i.e., the ground match for the NP "the kitchen"). The field values at each of the candidate ground locations are then evaluated, and the candidate with the highest value is returned as the optimal (most likely) ground match for the NP "the table". Fig. 4(b) shows the semantic field for *near the kitchen* in the example world along with

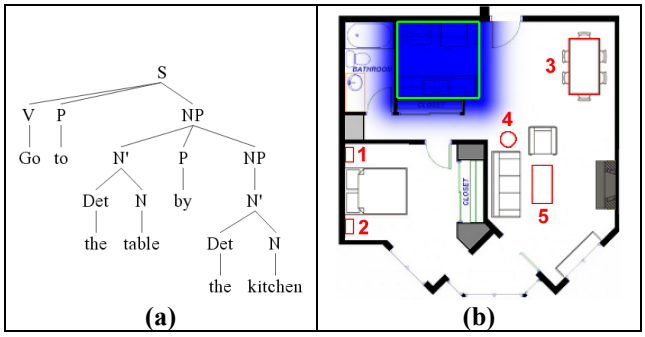

Figure 4. (a) Parse tree for "Go to the table by the kitchen"; (b) Semantic field for 'near kitchen' with candidate tables.

TABLE I. SEMANTIC FIELD VALUES OF CANDIDATE GROUNDINGS FOR NP "THE TABLE"

Candidate Ground	log(Semantic Field Value)
1	-13.60
2	-46.77
3	-27.67
4	**-5.92**
5	-28.51

Note. Log semantic field values are reported. Optimal grounding highlighted in bold.

the candidate groundings for "the table"; Table I lists the field values computed for all of the candidates, for reference.

After all of the NPs in the natural language input have been successfully grounded to known items in the world, the system proceeds to interpret the semantics of the instruction for appropriate robot command execution.

C. *Semantic Interpreter*

Our methodology employs a probabilistic approach to interpreting the semantics of the natural language input. Specifically, the problem statement for the semantic interpretation module is to infer the most likely command type, path type, and static spatial relation, given the observations. The system considers five observations in total, determined by the syntactic parser and grounding modules, including: the verb, the number of NP parameters, the figure type, the reference object type, and the preposition used, if any, in the sentence root.

The command types are domain-dependent, and may include, for example, robot movement, object manipulation, speech production, learned tasks, etc. In evaluating the feasibility of our methodology, our system focuses on two command types: robot movement (translation), and robot orientation. In instructing these types of movement commands, users often utilize spatial relations as opposed to precise quantitative descriptions [14]. Therefore, inference of these underlying dynamic and static spatial relations is necessary for correct interpretation of the command. This is especially evident in instructions where path prepositions are not specified (e.g., "Enter the room" vs. "Go into the room"). Static relations are inferred as part of the path specification. For example, the path for *to*, as described earlier, relies on a

static spatial relation to determine the termination condition (e.g., *at* for "to", *in* for "into", *out* for "out of").

The Bayesian inference method utilized by our system is Naïve Bayes, however our methodology allows the use of any probabilistic inference method, leaving the choice up to the system designer. Following is the formula used to perform command inference in our system:

$$\arg\max_{C} P(C|o_1, o_2, \ldots, o_n) = \frac{1}{Z} P(C) \prod_{i}^{N} P(o_i|C)$$

Where the N=5 observations are the same as those previously listed. The likelihood and prior probabilities are calculated from a database of labeled training data, which could be provided to the system *a priori* or gathered incrementally through interaction. The inference of path type and static relations is achieved similarly, with the addition of the inferred command and path type (for static inference) as observations. An example labeled input for the instruction "Stand by the kitchen" is provided below:

Observations	Labeled Semantics
{ Verb: "Stand" Number of NP Parameters: 1 Figure Type: None Ref. Object Type: Room Preposition: "by" }	{ Command: Robot Movement Path: *to* Static Relation: *near* }

D. Planning

Once the semantic interpreter has inferred the instruction parameters (i.e. the command type, path type, and static spatial relation), the planning module attempts to find a solution for the robot given these command specifications, as well as any other constraints indicated by the user. Constraints are specified to the system the same as instructions, through natural language, and thus their grounding and semantic interpretation are also equivalent.

The A* path planning algorithm is used in our system to find the minimum cost solution for robot action given the command and constraint specifications, which is then passed on to the action module for task execution. In the simplest case, constraints are handled by the planner through modification of the A* cost function. For example, the constraint "Stay away from the TV set" would apply the semantic field of the inferred static relation *away from* (attached to the reference object) to every point in the environment, and thus points with lower field values would subsequently have higher cost during A* search. More complex constraints would require the planner to segment the search into multiple steps to achieve intermediate goals [7]; this procedure is left beyond the scope of this paper.

V. Evaluation

To evaluate the ability of our robot system to follow natural language directives, we first analyzed the effectiveness of the semantic interpretation module to infer the correct command specifications given the natural language input. Our testing domain consisted of a simulated mobile robot operating within a 2D home environment map.

A dataset of 128 labeled training examples (each containing a list of observations with correct command specifications), were used in the evaluation of the semantic interpretation module. This dataset included the use of 8 different dynamic spatial relations (path types), 10 separate static spatial relations, 2 commands, and 22 different verbs, each appearing multiple times (and in novel combinations) among the examples. In order to create a training set and a test set for evaluation, the dataset was split into two equal parts, using randomized selection of the examples. A two-fold cross validation was performed on the dataset: the semantic interpreter first utilized the training set to gather probability statistics for the inference process, and was consequently evaluated against the test set. Subsequently the test set and training set were swapped and the inference performance was again evaluated. The results of both evaluations were then averaged to obtain the final inference accuracy results.

The results of the testing show that the semantic interpreter was able to achieve an inference accuracy of 99.2% for commands, 87.8% for paths, and 80.7% for static spatial relations. Table II contains a summary of these results. Given the relatively small size of the data set, the performance of the semantic interpreter is encouraging. Future work will include performing additional tests to confirm whether or not enhancing the sample size, and/or utilizing a more complex probabilistic model (e.g., Bayesian Network), would result in an increase in inference accuracy.

Next, to validate the potential of our methodology towards enabling natural language directive following in service robots, with and without user-specified constraints, we present four example test runs of our system. These examples illustrate the ability of the system to parse natural language input, ground noun phrases, infer command semantics, plan, and execute an appropriate solution while obeying natural language directive constraints.

In the first test run, the command given to the robot was "Go to the room by the entryway", without constraints. According to the map, the referenced room corresponded to the kitchen, which was correctly grounded by the system using the semantic field for '*near the entryway*'. The robot successfully planned and executed the optimal path to the kitchen (see Fig. 5(a)). In run #2, the same command was given but with the added constraint "Walk along the wall", which the robot was also able to account for by utilizing the semantic field values for the dynamic spatial relation *along*

TABLE II. Inference Accuracy of Semantic Interpretation Module

Inference Variable	Inference Accuracy
Command	99.2%
Path	87.8%
Static Relation	80.7%

Note. Results of two-fold cross validation of entire semantic dataset with 128 entries

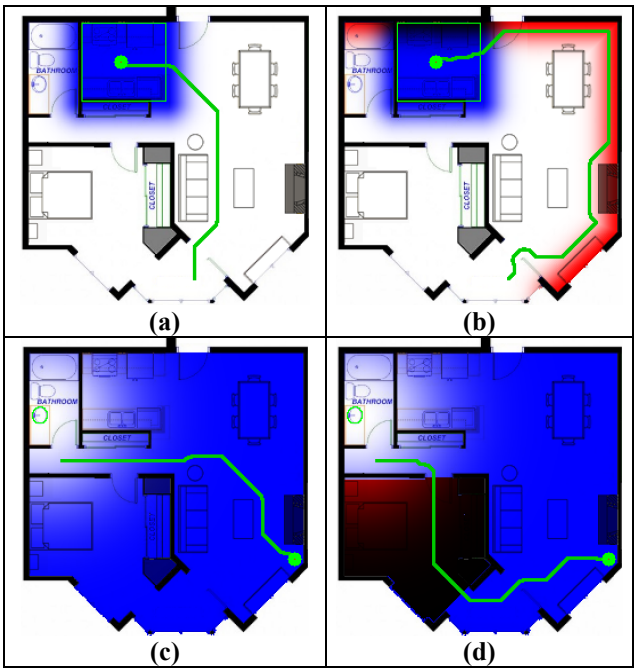

Figure 5. Executed paths and semantic fields (command = blue, constraint = red) for test runs (a) run 1; (b) run 2; (c) run 3; (d) run 4.

TABLE III. SEMANTIC INFERENCE RESULTS FOR INSTRUCTIONS AND CONSTRAINTS OF TEST RUNS

Inference Variable	Run 1 instruction	Run 2 constraint	Run 3 instruction	Run 4 constraint
Command	RM	RM	RM	RM
Path	*to*	*along*	*to*	*to*
Static Relation	*at*	-	*away*	*in*

Note. RM = robot movement command

in the cost function during the planning process (Fig. 5(b)). In runs #3 and #4, the command to the robot was "Stand away from the sink in the bathroom" (differentiating from the kitchen sink), with the addition in run #4 of the constraint "Enter my room" (see Fig. 5(c) and (d)).

To illustrate the usefulness of the semantic field model towards representing static and dynamic spatial relation primitives for use in path generation and classification, Fig. 6 shows the progression of the *at*, *along*, *away from*, and *in* semantic field values along the execution paths generated for test runs #1-4, respectively. As demonstrated by the results, the values returned by the semantic fields are highly correlated with the progress made during path execution towards accomplishing the goals of the dynamic spatial relation inferred from the natural language instructions.

As evidenced by inference results shown in Table III, and all four robot execution paths displayed in Fig. 5, the system was able to demonstrate its potential by successfully following the natural language directives, with and without constraints, during each of the test runs performed for the purposes of system evaluation.

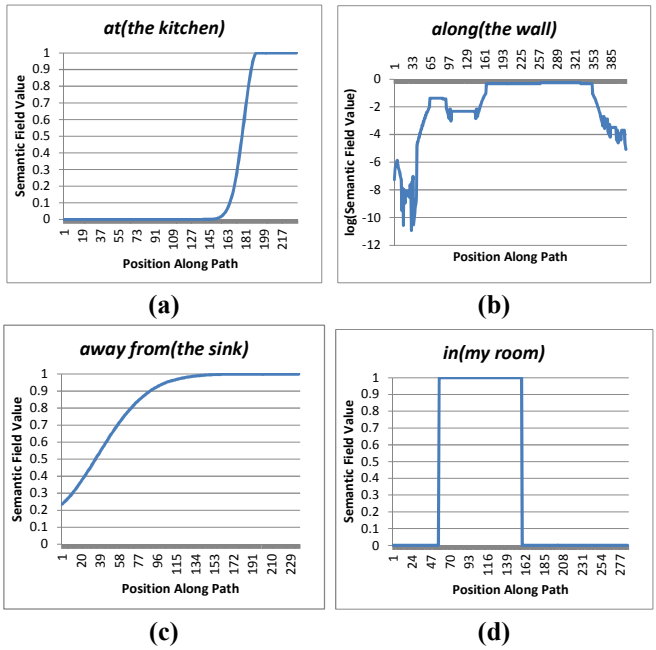

Figure 6. Semantic field values along execution paths in test runs (a) run 1; (b) run 2; (c) run 3; (d) run 4.

To demonstrate the generalizability of our approach and its usefulness in practice with real robots in real environments, next we present evaluation results of our robot software architecture using maps of real environments that were generated by physical robots implementing SLAM with onboard laser sensors. We provide the results of spatial language instructions given to a simulated mobile robot within these environments, with and without user-specified natural language constraints, to showcase the ability of our methodology to generate semantic fields, both dynamic and static, to accomplish spatial language tasks in real world task scenarios. The two maps that were used for this additional evaluation were collected from the Radish data set [23], and consist of a map of a building at the University of Freiburg (FR079), and a map of the interior of the Intel Research Lab in Seattle (intel_lab). The maps were manually annotated to specify landmark locations (e.g., rooms, walls, objects) and were given to the robot *a priori*. In practice, annotation for the robot-generated maps would be accomplished by the user and/or by a qualified technician during installation prior to first use.

As previously mentioned, the syntactic parser module can accept text input from either a speech recognizer or keyboard input. To illustrate the feasibility of our approach to operate with human users in real world environments, we provide the results of our implemented speech recognition module on the 128 natural language instruction training examples in our test database. Each of the entries was spoken exactly once for analysis, using a headset microphone placed approximately 1 inch from the speaker's mouth, and with minimal background noise. Table IV presents the accuracy results of the speech recognition module. The speech recognizer used

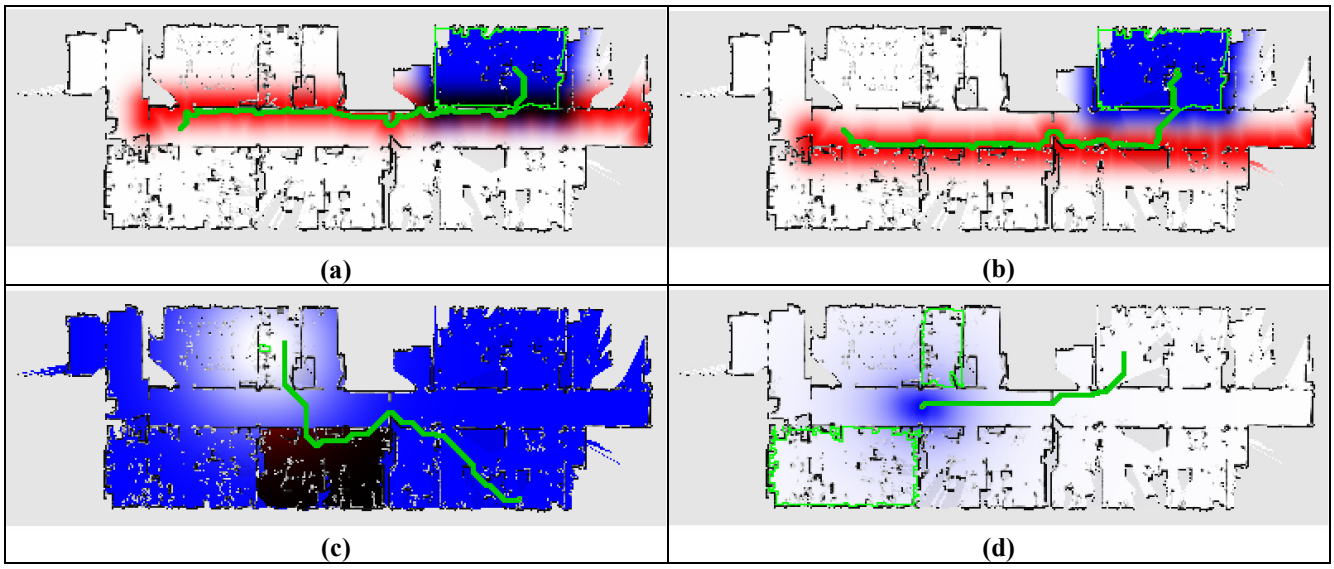

Figure 7. Executed paths and semantic fields (command = blue, constraint = red) for test runs (a) run 5; (b) run 6; (c) run 7; (d) run 8.

TABLE IV. SPEECH RECOGNITION MODULE ACCURACY

Sentence Error Rate (total correct / total sentences)	
Sentence Error Rate	9/128 = 7.03%
Sentence Semantic Error Rate	6/128 = 4.68%
Word Error Rate (substitutions + deletions + insertions / total words)	
Word Error Rate	(8 + 5 + 2)/686 = 15/686 = 2.18%
Word Semantic Error Rate	(4 + 5 + 2)/686 = 11/686 = 1.6%

Note. Semantic error rates exclude errors resulting in semantically equivalent sentences/words

in the module was Nuance's Dragon NaturallySpeaking [21].

The low error rate of the speech recognition module observed under our test conditions (low ambient noise and using a user mounted headset microphone), combined with the availability of algorithms to interpret spoken language under various forms of disfluency and repetition (e.g., [25]), demonstrate the feasibility of obtaining grammatically correct text input from spoken language in real world scenarios for use in our software architecture for service robots.

Eight additional test runs of our robot architecture were conducted using real world maps generated by robots using onboard laser sensors, as noted above. The natural language instructions, with their associated constraints, given to the robot in test runs #5-12 are provided in Table V. The robot execution paths for each of the test runs, along with associated semantic fields displayed for reference purposes, are provided in Fig. 7-8.

While these results were obtained from simulations in 2D, it is very common for robots operating in real-world environments (such as a home or office) to utilize a 2D map representation of the environment for localization and spatial task planning. The SLAM maps presented in this section are identical to the maps that would be used by a real robot

TABLE V. INSTRUCTIONS GIVEN IN TEST RUNS 5-12

Type[Run #]	**Natural Language Instruction**
Instruction[5]:	Go to the cafeteria
Constraint:	*Walk along the north hallway wall*
Instruction[6]:	Go to the cafeteria
Constraint:	*Walk along the south hallway wall*
Instruction[7]:	Stand away from the desk in my office
Constraint:	*Enter the meeting room*
Instruction[8]:	Stand between my office and the lab
Instruction[9]:	Relocate to the lounge area next to the lab
Instruction[10]:	Relocate to the lounge area next to the lab
Constraint:	*Roll along the central wall*
Instruction[11]:	Get to the kitchen
Instruction[12]:	Get to the kitchen
Constraint:	*Travel inside the central area*

operating in the actual 3D environments, and the methods employed for spatial language instruction understanding and task following, as presented in this paper, would also be identical. As a last step towards implementing our methodology on a real robot, translation of the discretized plan returned by the planner to continuous robot motor commands (e.g., wheel velocities) is necessary. This translation can be accomplished using a local planner which utilizes the returned A* path to fill a cost map covering the robot's local environment to determine the wheel velocities that would result in robot movement best following the generated path. This local planner would also be able to respond to dynamic obstacles not represented in the map (e.g., people, objects, etc.) and can be implemented using the dynamic window approach proposed by Fox et al. [24], for which there is an available ROS package which facilitates its usage in practice [22].

The evaluation results presented above, regarding speech recognition accuracy and spatial navigation task performance using SLAM maps, demonstrate the feasibility of our approach for use in practical applications with real robots. The evaluation of the robot software architecture in multiple

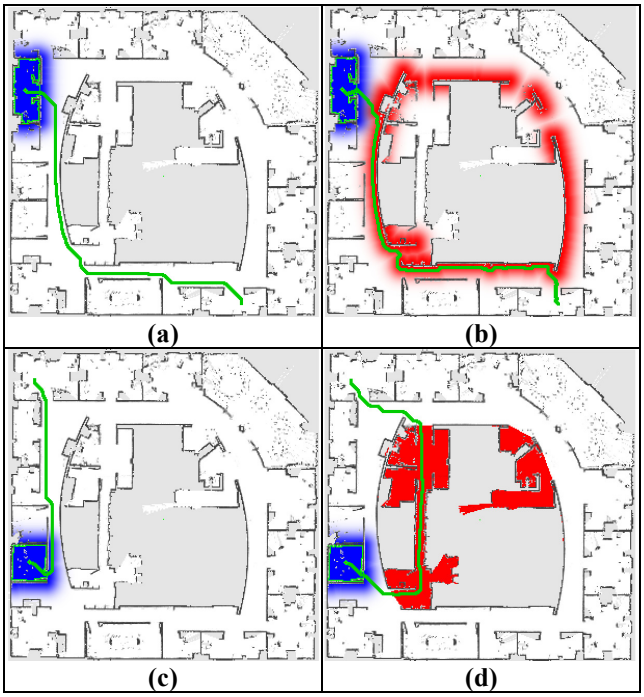

Figure 8. Executed paths and semantic fields (command = blue, constraint = red) for test runs (a) run 9; (b) run 10; (c) run 11; (d) run 12.

environments, using both manually-created and robot generated maps, demonstrates the generalizability of the approach and its effectiveness in accomplishing spatial language instruction tasks, with and without user specified constraints, across domains and in novel real world environments.

VI. Conclusion

We have described the need for enabling autonomous service robots with spatial language understanding to facilitate natural communication with non-expert users for task instruction, task modification, and user feedback on robot task execution, and have presented a general approach we have developed toward addressing this research challenge.

The results obtained from our evaluation testing demonstrate the potential of our methodology for representing dynamic spatial relations, grounding and interpreting the semantics of natural language instructions probabilistically, and generating appropriate robot execution plans under user-specified natural language constraints.

References

[1] B. Landau and R. Jackendoff. What and where in spatial language and spatial cognition. *Behavioral and Brain Sciences*, 16(2):217–265, 1993.

[2] J. O'Keefe. Vector grammar, places, and the functional role of the spatial prepositions in English, E. van der Zee and J. Slack, Eds. Oxford: Oxford University Press, 2003.

[3] S. Tellex, T. Kollar, S. Dickerson, M. R. Walter, A. G. Banerjee, S. Teller, and N. Roy. Approaching the Symbol Grounding Problem with Probabilistic Graphical Models. *AI Magazine*. 32(4): 64-76, 2011.

[4] T. Kollar, S. Tellex, D. Roy, and N. Roy. Toward Understanding Natural Language Directions. In Proc. ACM/IEEE Int'l Conf. on Human-Robot Interaction (HRI), 259–266, 2010.

[5] Y. Sandamirskaya, J. Lipinski, I. Iossifidis, and G. Schöner. Natural human-robot interaction through spatial language: a dynamic neural field approach. 19th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), pp. 600-607, 2010.

[6] M. Skubic, D. Perzanowski, S. Blisard, A. Schultz, W. Adams, M. Bugajska, and D. Brock. Spatial Language for Human-Robot Dialogs. *IEEE Transactions on SMC Part C*, Special Issue on Human-Robot Interaction, 34(2):154-167, 2004.

[7] J. Fasola and M. J. Matarić. Modeling Dynamic Spatial Relations with Global Properties for Natural Language-Based Human-Robot Interaction. In Proc. IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), Gyeongju, Korea, Aug. 26-29, 2013.

[8] D.K. Roy. Learning visually grounded words and syntax for a scene description task. In *Computer Speech & Language*. 16(3-4): 353-385, 2002.

[9] P.E. Rybski, J. Stolarz, K. Yoon, and M. Veloso. Using dialog and human observations to dictate tasks to a learning robot assistant. *Journal of Intelligent Service Robots*, 1:159-167, 2008.

[10] H. Kress-Gazit, G.E. Fainekos, and G.J. Pappas. Translating structured English to robot controllers. *Advanced Robotics*, 22, 1343–1359, 2008.

[11] J.D. Kelleher, F.J. Costello. Applying Computational Models of Spatial Prepositions to Visually Situated Dialog. *Computational Linguistics*, 35(2):271-306, 2009.

[12] S. Mohan, A. Mininger, J. Kirk, and J.E. Laird. Learning Grounded Language through Situated Interactive Instruction. In AAAI Fall Symposium on Robots Learning Interactively from Human Teachers (RLIHT), 2012.

[13] N. Hawes, M. Klenk, K. Lockwood, G.S. Horn, J.D. Kelleher. Towards a cognitive system that can recognize spatial regions based on context. In Proceedings of the Twenty-Sixth Conference on Artificial Intelligence (AAAI), 2012.

[14] L.A. Carlson, and P.L. Hill. Formulating spatial descriptions across various dialogue contexts. In K. Coventry, T. Tenbrink, & J. Bateman (Eds.), Spatial Language and Dialogue, 88-103. New York, NY: Oxford University Press Inc, 2009.

[15] D. Klein, and C.D. Manning. Accurate Unlexicalized Parsing.In Proceedings of the 41st Meeting of the Association for Computational Linguistics, pp. 423-430, 2003.

[16] K.P. Gapp. Basic meanings of spatial relations: Computation and evaluation in 3d space. In Proceedings of the 12th National Conference on Artificial Intelligence (AAAI'94), 1393–1398. AAAI Press, 1994.

[17] C. Matuszek, E. Herbst, L. Zettlemoyer, and D. Fox. Learning to Parse Natural Language Commands to a Robot Control System. In Proc. of the International Symposium on Experimental Robotics (ISER), Québec City, Canada, 2012.

[18] R. Cantrell, P. Schermerhorn, M. Scheutz. Learning actions from human-robot dialogues. In Proc. IEEE RO-MAN, pp.125-130, 2011.

[19] J. Bohnemeyer. The Unique Vector Constraint: The impact of direction changes on the linguistic segmentation of motion events, E. van der Zee and J. Slack, Eds. Oxford: Oxford University Press, 2003.

[20] G.D. Logan, and D.D. Sadler. A computational analysis of the apprehension of spatial relations. In P. Bloom, M.A. Peterson, L. Nadel,& M. Garrett (Eds.), *Language and space*, pp. 493-529, Cambridge, MA: MIT Press, 1996.

[21] Nuance. Dragon NaturallySpeaking, 2013. www.nuance.com

[22] E. Marder-Eppstein, and E. Perko. ROS package: base local planner, 2012. www.ros.org/wiki/base_local_planner

[23] A. Howard and N. Roy. The Robotics Data Set Repository (Radish), 2003. http://radish.sourceforge.net

[24] D. Fox, W. Burgard, and S. Thrun. The dynamic window approach to collision avoidance. *IEEE Robotics and Automation*, 4(1), 1997.

[25] M. Scheutz, R. Cantrell and P. Schermerhorn. Toward Humanlike Task-Based Dialogue Processing for Human Robot Interaction. *AI Magazine*, 32(4):77-84, 2011.