Trained Probabilistic Models for the NAO Robot in a Labyrinth

Samantha Eaton, Kleanthis Zisis Tegos, Sagar Vilas Jagtap, Akram Alghanmi, Godwyll Aikins, Neda Keivandarian, Ran Bi, Jesse Torres, Kholud Alghamdi, Marius Silaghi

Florida Institute of Technology

ABSTRACT

A framework based on a Probabilistic Model for the moving behavior of the NAO humanoid robot in the environment given by $20in \times 20in$ vinyl maze cells is being trained and made available for future applications. NAO is one of the most advanced humanoid robots, having advanced speech, vision, and behavior based on artificial intelligence already implemented on it, and being a precursory of the larger Pepper robot famous for being used as host at certain hotels. Pepper uses wheels, probably since the leg-based movement of Nao proved hard to harness with precision and robustness. Indeed, most of the intelligence currently present in NAO is speech and gesture-related, while its autonomous walking capabilities are only little exploited in existing available software, and only with reflexes without high-level utility-driven intelligence.

We test that it is possible to exploit a public NAO sensor database made recently available, to build a sample probabilistic model for walking and turning in a controlled vinyl maze. The probabilistic model is a new and powerful representation of related phenomena based on random variables and with conditional probability tables for the NAO sensors computed from experimental measurements given relevant environment states. The model allows for complex planners and reasoners, that are based on rich POMDP models, to be built on top of it.

Such a high-level AI framework allows for easily giving NAO new tasks by just specifying the corresponding utilities. The proposed model is tested with a simple particle filter localizer on a predefined trajectory, and improvements and data missing in the Nao database are being identified for future work.
Keywords: Humanoid, framework, model, AI, Nao.

1. Introduction

The humanoid NAO robot, together with its wheeled larger version Pepper, are among the most advanced robots available to the general public. These robots are known to gracefully dance, walk, talk and entertain the users with intelligent dialog. However, the walking intelligence of these robots is rather low, and no common application enables NAO to robustly walk with high-level, (i.e., utility-driven) intelligence; not even in restricted environments.

Nao is being used for a robotic soccer contest called Robocup where teams of robots play without central controllers. However, the robots in such games are endowed only with reactive behavior for their role in the game and are not utility driven with high-level intelligence. Futher, the Robocup robots are based on a secret operating system not available to the rest of the public owning Nao robots, like us. Also, the games are organized on hard terrain that is optimal for Nao movements, but walking on other surfaces, like carpets, is much harder for it.

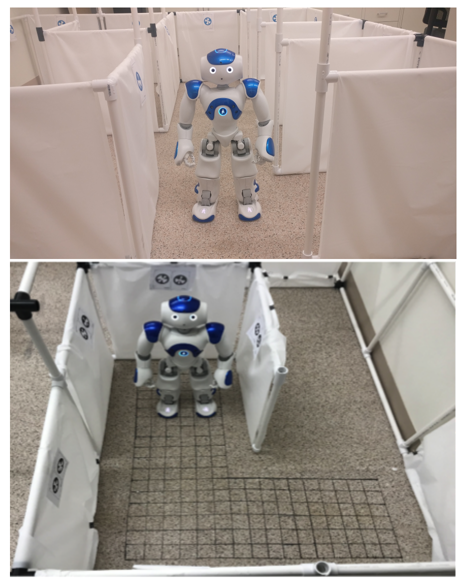

Figure 1. (a) Restricted Environment; (b) Markers for Standardizing Data Measurement

In previous work, groups of researchers in our lab started to address this problem by standardizing a restricted maze environment based on $20in \times 20in$ white vynil cells [1], Figure 1.a,b, and started organizing a public database publishing sets of contributed sensor data and movement measurements [2], usable for characterizing and training models of the interaction between Nao and such restricted environments. However, the test of the Nao sensor database was very preliminary and in this work we undertake the extension of the data gathering and framework formalization, as well as testing, for the Nao robot.

We now propose a general probabilistic model of the movement capabilities of Nao in a controlled environment, defined by the labyrinth of 20 inch square rooms with vinyl white walls.

Motivation. Our work aims to build a probabilistic model of the NAO robot's sensors and actions non-determinism, enabling the application of high-level intelligent algorithms for tasks in a vynil labyrinth. The tasks context include school maze problem solving contests for educating students in advanced approaches to artificial intelligence concepts like uncertainty and non-determinism.

The foreseen contest application is the navigation of the maze while making educated movement decisions based on the information available. Tasks can consist of guiding users in the labyrinth, patrolling, searching, transporting. With high-level intelligence, new tasks should be achievable simply by having users communicate to NAO utilities in terms of rewards for desired states. Decision components can include operations such as localization, mapping, and planning.

Contribution. In this particular work we identify and propose a particular Partially Observable Markov Decision Process (POMDP) model that is elegant, almost textbook level, and still represents sufficiently well the NAO robots' interactions with a vynil labyrinth to accomplish many tasks.

Further, we present a Dynamic Decision Network (DDN) representation of this POMDP, which allows us to incrementally and educationally show the construction of our NAO POMDP from components. This part is intended as a crash tutorial on POMDP/DDNs to the FCRAR community. As such the DDN is :

- firstly showed constructed as a Bayesian Network (BN) for static inferences of location based on the NAO's available sensor inputs (sonars and landmark detection),
- further enhanced to a Dynamic Bayesian Network (DBN) by including movement and transition models, and
- finally promoted to a POMDP/DDN by enabling the addition of rewards that specify tasks to do, in a utility-driven paradigm.

Lastly, we report on the experimental evaluations that prove and quantify the non-determinism and uncertainty of the NAO architecture, not only confirming the needs for frameworks like ours, but also actually providing the numerical Conditional Probability Tables (CPTs) components of the POMDP models which can be used for actual inference with the proposed framework. The CPTs are trained on a combination of new measurements and existing data in the NAO database [2].

Moreover, movement of NAO in the labyrith with our probabilistic model-based localization is executed based on the CPTs we construct according to the described mechanisms, illustrating applications and evaluating the accuracy achievable using the BN/DBN parts of the proposed POMDP model.

2. Background and Related Work

Let us introduce the main background on the Nao robot, on dynamic Bayesian networks and on Partially Observable Markov Decision Problems (POMDPs), which are among the most principled general frameworks in artificial intelligence for addressing robotic problems.

2.1 The Nao Humanoid

Nao is a humanoid [3], that has advanced capabilities being able to talk and walk. His sensors consist of two sonars placed symmetrically on the right and left of his torso, as well as two cameras placed one above the other between the LED eyes. The cameras come with software to detect a special type of landmarks, called Naomarks.

Among the closest related humanoid robots that are available we mention SoftBank's Pepper, that has wheels locomotion instead of legs, and Honda's Asimo [4] who also has a camera on his face but two asymmetrical sonars on its torso, of which one is specialized for detecting the ground. Asimo has walking that is more advanced, being able to climb stairs, but it is not available for sale.

2.2 Relevant Nao Sensors and Actuators

The only relevant sensors in our approach are the two Sonars and the Landmark detector. The Landmark detector is a high-level sensor implemented in software on top of the Nao cameras. While this detector works relatively well in static scenarios, the version in the standard edition we have fails frequently at detection when the robot is executing walking tasks. These failures have to be modeled and accounted for by the high level intelligent architecture and this sensor non-determinism is also quantified in this work.

Similarly the only actuator exploited is the `walk` command, whose analysis and probabilistic model is the subject of the second part of Section 4.1. While Nao has multiple walk commands, for now we restrict ourselves to three instances:

- step 2 inches forward
- turn 10 degrees right
- turn 10 degrees left

Rotations of 5 degrees are also studied but their non-determinism was observed to be significantly higher in preliminary evaluations, as frequently the robot just does not achieve any displacement on receiving the command.

2.3 Bayesian Networks

A Bayesian network is a graphical probabilistic model representing conditional independence information with graphs consisting of nodes and edges. A node represents a random variable, and an edge represents a direct conditional dependency between the random variables (nodes) it links. Edges are directed with parents preferably being selected as causes of descendants, which simplifies estimation of conditional probabilities. Nodes are conditionally independent of indirect ancestors in the graph given certain nodes in-between (a Markov blanket). The domains of the random variables could be either discrete or continuous. The random variables in the Bayesian networks can be classified in three types: observable (evidence), non-observable (hidden), and controllable (also evidence, graph source nodes). Generally Bayesian network graphs are constructed to be acyclic [5]. Every node is associated with a conditional probability table specifying its conditional probability given all possible value combinations for its parent variables in the graph, or some compressed representation thereof.

A Bayesian Network is a compressed representation of the full joint probability distribution of its variables. The main application of using Bayesian networks is the enabling of probabilistic inferences in environments where state or phenomena are uncertain or non-deterministic.

A Dynamic Bayesian Network (DBN) is a Bayesian network of a special type, where sets of nodes are repetitions of a pattern in a sequence that can potentially be infinite. It is frequently used to represent sequential processes, each repetition corresponding to a time step, where the time is discretized. It thus represents a probabilistic temporal model. Dynamic Bayesian networks are generative models. The graph in DBN is composed of a sequence of time steps or slices, namely $(0,...,t-1,t,t+1,...)$ [6].

As an example, Particle Filtering, one of the main reasoning techniques for filtering in DBNs, works as follows. After

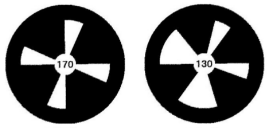

Figure 2. Standardized wall markers, in pairs

(i) an initial population at time t with N samples from the hidden state variables set x_t is created from the distribution $P(X_t)$,
(ii) firstly, for the subsequent time step $t+1$, its next state value x_{t+1} sample is generated based on the transition model $P(X_{t+1}|x_t)$
(iii) and then weighted by the likelihood that it assigns to the corresponding new evidence e_{t+1}, $P(e_{t+1}|x_{t+1})$.
(iv) Lastly, this weighted population is interpreted as a probability distribution $P(X_{t+1})$, that in a subsequent round is resampled per the aforementioned procedure to generate the next set of samples, thus repeating these inference operations at each remaining time step.

2.4 POMDPs

The Partially Observable Markov Decision Process (POMDP) is a framework for modeling dynamic uncertain environments, adding rewards associated to states and/or transitions in Dynamic Bayesian Networks. A POMDP is defined by a set of states, a set of possible observations, a set of actions, a transition probability function between state repetitions, observation conditional probabilities, an initial belief, a discount factor, and rewards associated to states.

Through observation of evidence, the robot will update the probability distribution of its current state (aka. belief) in order to optimize its choice of the next action maximizing future rewards. Actions at a time step contribute in the next state transition.

A reward represents the benefit of a state (and potentially, action). The typical goal of reasoning in a POMDP is to generate an optimal policy (map belief-action) which maximizes the amount of future rewards [7]. A belief is a probability distribution over states. The robot will optimize the next action, making decisions, based on its current belief [8].

One way to depict POMDPs graphically is based on extensions of DBNs called Dynamic Decision Networks (DDNs). In DDNs, each time step is associated with the set of actions that can be performed in it, treated as additional evidence factors, as well as with a reward that depends of state and actions.

2.5 The Nao database structure

We use the database of experimental measurements constructed by the previous work in [2]. This database contains sensor and transition measurement data, as well as its documentation and relevant code snippets. The sensor data contains the measurements obtained from the various sensors on the NAO robot; the left and right sonar sensors for detecting objects and the visual sensor for detecting the standardized landmarks on the walls, see Figure 2. The transition data is measured as the center of mass displacement when the robot is given the command to move between two adjacent squares, each $2in \times 2in$, in the labyrinth.

The transition data is organized into 4 column datasheets. Each datasheet describes experiments with a given action, and its columns are: the given position, given orientation, recorded position, and recorded orientation. As the name of the columns suggests, each datasheet records actual responses measured for the command associated with it.

The sensor data consists of 12 column datasheets (see Figure 3).

- Two columns were dedicated for the position of the robot in the cell of the labyrinth, namely the "Row" and "Column" in the maze.
- "Angle" is the degree of the rotation of robot with respect to the reference vector.
- "Head Pitch" and "Head Yaw" (the orientation of the robot's head) are represented in the sensor datasheets but were controlled at the still position and not used in the localization of the robot.
- The "Left" and "Right" sonar sensor readings of the robot were the most robust Sensor data readings.
- In the Nao sensors database data, two landmarks are placed on each maze wall center, and data pertaining to their detection was recorded in the sensor data: "Landmark detection" (whether the landmark was detected), "Landmark ID" (Number ID on the Nao Mark), "Alpha" and "Beta" (which are the line of sight angle of the landmark's position from the robot's head axis), and "Width" and "Height" of the detected landmark.

2.6 Related Work

In [9], a Nao robot is addressing the problem of simultaneous localization and mapping (SLAM) in a room which has multiple Nao marks with random locations. The contribution proposed to enhance SLAM addressing the real-time incorporation of new landmarks in exploration.

An autonomous algorithm to obtain the parameters of the POMDP in a navigation system for indoor environment based on WiFi and ultrasound observations is implemented in [10]. This algorithm can learn the observation and transition matrix in autonomous mode which is coming from wifi simultaneous localization and mapping so it could obtain the WiFi and Ultrasound environment map with minimum effort. The localization algorithm converges faster by adding a global navigation system sensor.

A multimodal interaction system between a human and a robot was suggested in [11] by expanding a POMDP framework in a humanoid robot called "Pepper" to observe various multimodal information streams with its own sensors. The interaction system is made of two strata in the framework of POMDP – set the first stratum so as it decides to start interaction based on the physical distance between Pepper and a user, and set the second stratum to control multimodal interaction. The effort led to stratified interaction to reduce the increase of the user's belief states, making the problem more manageable. It shows that POMDPs were previously used for emotive behaviors, even if not for walking, with such robots.

A variable resolution technique is proposed in [12] for reducing the complexity/state space of the POMDP. This is accomplished by automatically adjusting the number of the states in the grid based on features of the environment map while maintaining the level of detail required for planning at a given region to find a robust and efficient policy. This is validated in a POMDP-based simulation of a navigation environment, a realistic robot simulator, and an autonomous robot.

col, row	Represent the location variable in DBN; has discrete values [(0,0),,,, (3,7)]
angle	correspond to orientation of robot, represent angle variable in DBN, has discrete values [0, 10, 20, ...,320]
head pitch, head yaw	Represent the orientation of the head
sonar left, sonar right	Sensor reading for distance to the wall
Landmark detected	Represent Boolean values; True, False
Landmark id	Represent the type of id, there are three types.
Alpha, beta, width, height	Properties of a landmark, direction of view and dimension

Col	Row	angle	Head pitch	Head yaw	Sonar right	Sonar left	Landmark detection	Landmark id	Alpha	Beta	Width	height
0	1	10	NA	NA	.45	.47	True	68	-0.00892	-0.0649	0.09798	0.09798
3	5	230	NA	NA	.48	.26	True	85	0.219224	-0.1276	0.18434	0.18434

Figure 3. Sensor Inputs, as organized in the Nao Database

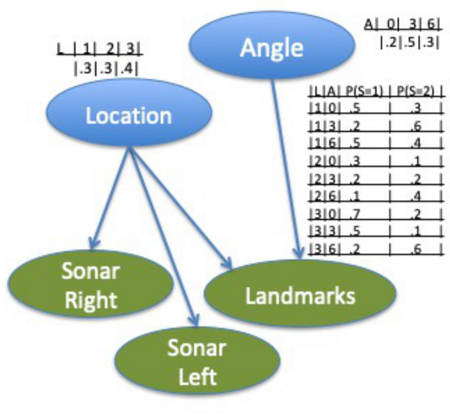

Figure 4. Conceptual Belief Network for Observing the Situation

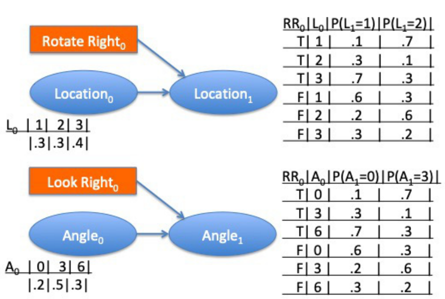

Figure 5. Conceptual Belief Network for Transitions

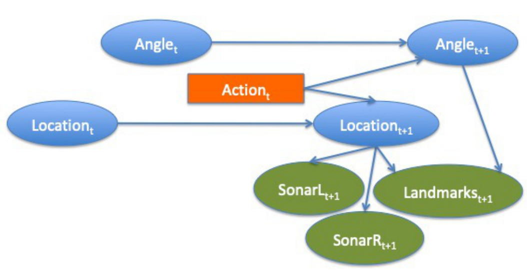

Figure 6. Dynamic Belief Network utilized

3. Concepts and Data Structures

To model the phenomena taking place when a Nao robot is observing or walking in the maze, belief networks are designed to generate the CPTs from sample data, as conceptually shown in Figures 4 and 5.

The Bayesian Network in Figure 4 abstracts the relations between: the relative position or "Location" of the center of the robot (the center of the space between its feet) with respect to walls, the torso heading angle, the angle between its head and torso, the measurements of its sonars, and the landmark detection (the actual sets of values of the variables are more complex, and described later).

The Bayesian Network in Figure 5 describes abstractly the impact of actions on location and angles, for rotations. Similarly, the actual sets of values of the variables are more complex, and described later. With most types of walking, both the previous angle and previous location have an impact on the resulting location.

The Location or position node holds one of the M unique $x-y$ coordinate points within the labyrinth, and possible heading angles of the torso. If the maze would have m rows and n columns, then $M = m \times n$. The Angle node represents the angle between the head and torso, as controlled by the robot.

The Left Sonar node represents the measured value of the left sonar based on the $x-y$ coordinate point within the labyrinth and the overall angle the robot body is facing. Its conditional probability table (CPT) is a table of size $M \times N \times O$, where M represents the total number of unique $x-y$ coordinate points within the labyrinth, N is the number of unique torso heading angles, and O is the number of possible sonar readings. The Right Sonar node is defined similarly, but represents the measured right sonar values.

The landmark node represents the event of detecting some of the landmarks in the environment. It is related to the $x-y$ coordinate position within the labyrinth and the overall position angle of the head of the robot. Its conditional probability table (CPT) is compressed based on an independent success assumption into a table of size $M \times N \times L$, where M represents total number of unique $x-y$ coordinate points within the labyrinth, N is the number of possible unique sums between torso heading angle and the torso-head angle, and L is the number of possible landmarks.

A possible value for the Landmark random variable is any bag of landmarks with their possible sizes and positions in Nao's visual field. The representation can be compact by using a single Boolean per landmark to represent it's visibility, while the distribution of the rest of the landmark location and size values can be integrated as deterministic nodes with Gaussian noise.

This representation is more general and flexible than the earlier version [2] where there are three landmark nodes per time frame,

Table 1. Sonar Left Node

X	Y	Angle	P(Sonar=0.46)	P(Sonar=0.47)	...
0	0	0	0.25	0.000	...
0	0	11.25	0.000	0.000	...
...	...	...	...	...	...
m	n	N	...	...	...

Table 2. Landmark Node

X	Y	Angle	Landmark 1	Landmark 2	...
0	0	0	0.8	0	...
0	0	11.25	0.8	0	...
...	...	...	...	...	...
m	n	N	...	...	...

that can recognize up to maximum three landmarks, an assumption that is not always true with complex mazes. The bag representation also simplifies the handling of matching different landmark IDs in different locations. The case of landmarks with repeating IDs is described later.

While the shown figures display conditional probability tables for illustration, the actual conditional probability tables of these Bayesian networks are estimated by counting from the measurements in the Nao sensor database, as described further in the article.

4. Models and Algorithms

The database extension we provide offers algorithms for estimating the belief (i.e., conditional probability distribution), given current sensor measurements, as well as for filtering the new beliefs along sequences of actions and measurements of sensor readings .

4.1 Training Conditional Probability Tables

Building on top of the sensor data made publicly available by a previous work [2], libraries for estimating the conditional probability tables for our Belief Network (refer to Figure 4) were developed.

The proposed belief network is **discrete**. While location is a continuous quantity, the state space was divided into discrete cells. The same approach was used for the sensor measurements, which are also continuous, but were categorized into discrete groups.

The Sonar nodes each relate data from the respective sonar sensors with the location and angle information. Extracting the sonar data for each sonar results in a $(m \times n) \times N \times O$ matrix, where N is the number of torso heading angles recorded at each position and O the number of sonar readings handled. As some measurement readings never occurred in the training set leading to zeroes as shown below, the probability mass is later redistributed to compensate for possible noise by assigning a small probability to each possible value.

For the Landmark node, we separately represent the Boolean landmark detection event and its actual parameters in terms of position. When a landmark is detected its position is pretty accurate and therefore can be approximated with a deterministic function with small Gaussian noise. For the Landmark detection event, the CPT can be represented compactly by exploiting an *assumption of cause independence between individual landmarks' detection success*. As such one only needs to store separately the probability of detecting each individual landmark, and the probability of each bag of landmarks can be inferred from these assumption of independence with the product rule. The resulting table is a $(m \times n) \times N \times L$ matrix, where L is the number of landmarks in the environment. At training the landmark data is True if the landmark was encountered, and False otherwise.

For the case where landmarks with the same ID are reused in the environment, the compact representation needs to include one column for each count of detected markers with identical IDs, and

Table 3. Translational Transition Probability for Location Node

$State_{cmd}$, $Orientation_{cmd}$	0, -90	0, 90	...	8, 0
0, -90	0.8	0	...	0
0, 90	0.1	0	...	0
...	...	...	...	...
8, 0	...	...	...	...

the aforementioned deterministic probability node linking the position to the parameters of the detection are normalized over all such repeated occurrences.

Transition probabilities for the NAO humanoid robot were calculated using the transition data that came with the dataset. Recorded readings and each action command sent to the robot were used to compute the probabilities by counting.

$$P(r|0,S) = \frac{\text{Number of times state r = 0, and orientation S}}{\text{Total times the command was executed}} \quad (1)$$

From the Nao database, the following translational conditional transition probability table of size $(l \times o) \times (r \times u)$ was produced, where l is the relative state the robot was asked to move to, o is the relative orientation the robot was asked to be in, r is the possible state the robot may be in, and u is the possible orientation the robot may be in after execution of a command. Both $l, r \in S\{0,1,2,3,4,5,6,7,8\}$, in which each number represents either the robot's current or neighboring position.

Specificity to the NAO robots. What makes this implementation specific to NAO is the structure and the parameters that each node represents. The sonar sensors are placed symmetrically right and left on the NAO's chest and therefore, the sonar data will only depend on the location and orientation of the robot's torso. The landmark node utilizes visual data from the cameras placed one on top of the other on NAO's head. Because the parameters that the network makes use of are highly dependent on their sensor's physical placement on the robot's body, the model and its implementation that we present to describe the robot behavior is unique to the NAO robot.

If this network was to be generalized for use, for instance, on another robot with cameras that the landmark node draws data from, fixed to the robot's torso; it would completely invalidate the network as the landmark node would not depend anymore on the head angles. It is however possible to use this network on another robot that shares the same physical features and sensors as NAO or with appropriate adaptation for robots that share similar features.

Specificity to the used labyrinth. All the landmark measurements are based on the positions of NaoMarks in the labyrinth so, inherently, the conditional probability tables (CPTs) of the computed model are only relevant in the context of the given labyrinth.

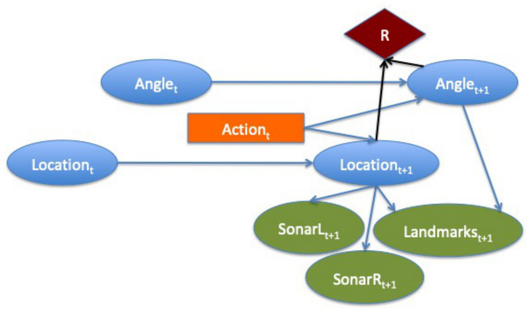

Figure 7. Dynamic Decision Network

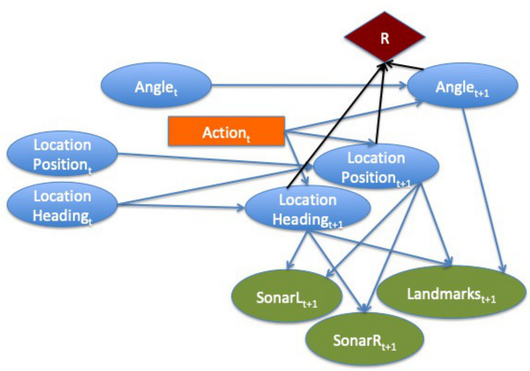

Figure 8. Dynamic Decision Network optimized

The readings of the sonar sensors also depend on the labyrinth wall material. Different materials reflect the sonar waves differently. Therefore its CPTs are also relevant only within this particular setting.

The labyrinth also provides the testing environment for the sample models test application described in experiments. Measurements of the sensors were extracted at various locations as described in the previous team's article [2], measurements which were used to train the models described in this paper. The labyrinth was used to measure the accuracy of the obtained models, in terms of number of times the network guessed the current location correctly, given its sensor measurements.

4.2 The POMDP Model Design

From the Bayesian Network concept describing observation and transition phenomena in Figures 4 and 5, we build a Dynamic Belief Network shown in Figure 6.

Performing exact inference for Dynamic Bayesian Networks (DBN) is hard for complex networks, given that the cost of the update procedure is O($d^{|x|+2}$) for domain sizes d with $|x|$ variables in a densely connected graph, as stressed in [5]. For stochastic methods sample generation is possible with various approaches. In the first approach, the distribution used in sampling comes from a pooled multivariate Gaussian mixture with weights given by the previous round samples. Practically, the distribution used for re-sampling is interpolated using a radial-basis function with Gaussian kernel. In the competing approach, the distribution used for sampling is obtained with multi-variate interpolation (Shepard method or linear) [13].

The Dynamic Decision Network. Integrating rewards conveying the utility of goal states, a dynamic decision network is obtained, as in Figure 7. The Dynamic Decision Network (DDN) is the graphical representation of POMDPs.

Since for many actions of the humanoid it can be assumed that the transition is independent for location heading and location position, the CPT storage space is improved with the finer grain dynamic decision network in Figure 8, where the "Location" random variable in the previous models is replaced by two random variables: "Location Position" and "Location Heading", for the $x-y$ coordinate and for the torso heading, respectively.

Rewards Specification. The rewards needs not be calculated by us, as it is supposed to be the way for end-users of our library to specify tasks. This is similar to how queries are not included in a first order logic knowledge database, but are posed by the end users. So, while we here provide an AI library of probabilistic models, the end-users of such models sets the rewards for achieving specific tasks.

For example, in some designs the reward for the POMDP could be proportional to the distance of a cell to any of the surrounding walls, as well as the expected distance to the end position. What this means is that the robot will try to choose the move that will bring the robot closer to the end goal, taking the walls into consideration.

5. Experiments

Preliminary experiments that enabled extensions to the Nao sensor database with additional landmark measurements were achieved.

The experiment consisted in measuring and evaluating sensor inputs for more situations, and analyzing the data into CPTs as reported in the article for better estimation of the scale of non-determinism. The CPTs, and related software for access, are too large for reproduction in the article but they are made available in the database.

5.1 Setup

As shown in Figure 1b, the humanoid was placed at the starting point with three immediate landmark pairs around it (left, right, and back). Markers were also drawn on the floor beneath the humanoid to standardize the observation of the change in position results.

A series of trials was executed using the humanoid to find the left and right sonar readings and the landmark visual readings based on the various $x-y$ coordinate points within the labyrinth and the torso and head positions. The generated data was saved in CSV format, added to the database, and used to generate the belief network.

5.2 Obtained Models

The belief network with sample node data is shown in Figure 4. The main inputs to the overall system is the $x-y$ position coordinate point and the angle of the head.

In our experiments, M is 32 for the unique $x-y$ coordinate points within the labyrinth and N is 32 for the unique degree angles (0 degrees to 360 degrees in steps of 11.25 degrees). Data in the database was available for $L=3$ distinct landmarks.

The Figures 9, 11, and 10 quantify the measured non-determinism in actions and sensors for Nao.

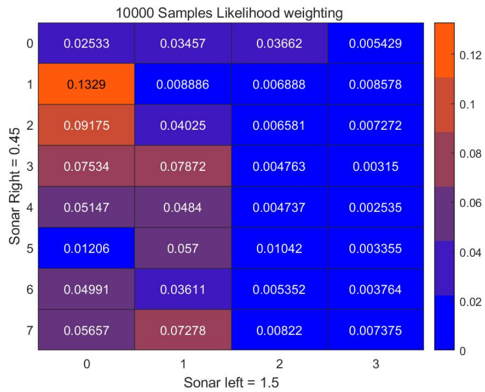

Figure 9. Sensor Uncertainty: The belief based on a sensors measurement

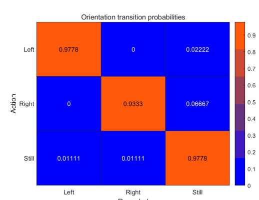

Figure 10. Nao 10° Rotation Non-determinism: The heatmap of beliefs concerning position after a 10 degrees rotation. The ordinate shows the command and abscisae shows probability of the obtained state

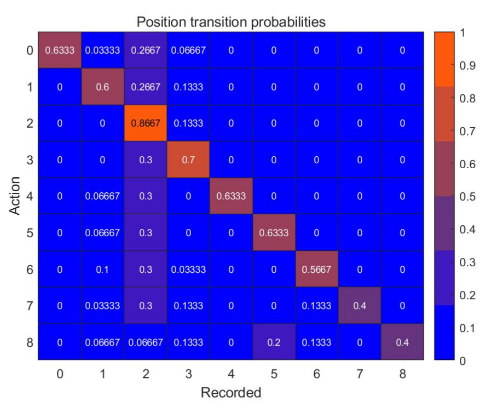

Figure 11. Nao Stepping Non-determinism: The heatmap of beliefs concerning position after a step

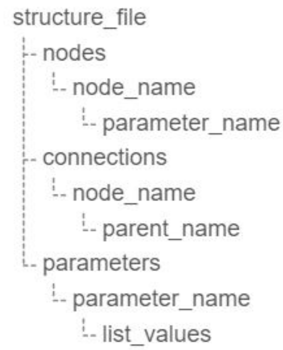

Figure 12. Layout of JSON file containing Bayesian Network

5.3 Model Guide

The goal of this work was to provide baseline models upon which the locational awareness of the Nao robot can be improved. This section serves as a reference on how the obtained models, which are made publicly available in the Nao database, can be used.

```
BN estimate_CPTs(BN_structure, training_data, BN)
```

The Python function `estimate_CPTs` calculates the conditional probability structures, given the Bayesian network structure and the corresponding training_data measurements, which are stored in a file. It accepts three file names, with the BN_structure and output BN files having JSON format, and the measurements having csv format. The format of the BN_structure is shown in Figure 12.

The requirement for the measurements file is that the first row contains the column names, which correspond to the nodes of the network. The output file only stores the matrices, and no structure information regarding the network, since that can be extracted from the structure file.

```
distrib belief(l_sonar, r_sonar, landmarks, theta)
```

The function `belief` is used to compute a probability distribution for all the possible current locations (x, y, α) within a cell given the sonar and landmark readings, as well as the head-torso angle `theta`.

```
distrib likelihood_weighting(DBN, query_vars,
                 evidence_vars, samples_nb)
```

The likelihood weighting function calculates the probability distribution over the query variables given the evidence variables. It accepts the DBN as a filename for conditional probability tables and for its structure, the query variables as a dictionary with keys being the needed node names, the evidence as a dictionary with keys being the node names and values being the measured node state, and a number of samples to be run. The more samples are run, the higher the accuracy of the likelihood weighting algorithm.

API of the Bayesian Network.
The Bayesian Network was developed using Python to parse the data results from the CSV files and generate the expected node results. The generated conditional probability tables (CPTs) are written to CSV files.

API of Likelihood Weighting Module.
The Likelihood Weighing Module was developed using Python. The code parsed the Bayesian Network conditional probability tables (CPTs), the user input specifying the positional goals of the robot the expected current position of the robot, and the robot's sonar and landmark detection data based on it's current position. The code produced the transitional probabilities for the next timestep at that location, and the estimated final robotic location with head orientation.

6. Conclusions

The problem of lack of support for easy development of intelligent and autonomous Nao robot applications with walking mobility is addressed by extending on the preliminary prior work in our organization, that introduced a "sensors and transition measurements database" for probabilistic reasoning in restricted environments.

This work extends prior contributions not only by adding new data measurements with new sensor and landmark sensor measurements, but also by contributing a general sensor model, and a Dynamic Decision Network framework with libraries for training conditional probability tables and for likelihood weighting inference steps compatible with the data formats in the Nao sensors and transition database.

References

[1] A. Hasanain, T. Weekes, M. Person, K. Paul, Y. Chang, A. Rothman, Z. Rui, R. Rahil, M. Syed, C. Wodd, and M. Silaghi, "Models and framework for supporting humanoid robot planning & exploration," in *FCRAR*, 2018.

[2] K. Alghamdi, J. Torres, J. Burden, V. Maddineni, R. Alharbi, T. Neelam, P. Tarsoly, A. AlHaif, J. Nke, S. Jadhav, S. Dodle, and M. Silaghi, "Towards sensor and motion measurements databases for training models for the nao robot," in *FCRAR*, 2019.

[3] 2020.

[4] Honda, "Asimo technical information," 2007.

[5] S. J. Russell and P. Norvig, *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,, 2016.

[6] K. P. Murphy, *Dynamic Bayesian Networks: representation, inference and learning*. PhD thesis, University of California, Berkeley, 2002.

[7] J. Luo, S. Zhang, X. Dong, and H. Yang, "Designing states, actions, and rewards for using pomdp in session search," in *Advances in Information Retrieval* (A. Hanbury, G. Kazai, A. Rauber, and N. Fuhr, eds.), pp. 526–537, Springer International Publishing, 2015.

[8] N. Roy, G. Gordon, and S. Thrun, "Finding approximate pomdp solutions through belief compression," *Journal of Artificial Intelligence Research*, vol. 23, p. 1–40, Jan 2005.

[9] Y. Zhang, *Real-time SLAM for Humanoid Robot Navigation Using Augmented Reality*. PhD thesis, Applied Sciences: School of Mechatronic Systems Engineering, 2014.

[10] M. Ocana, L. M. Bergasa, M. A. Sotelo, and R. Flores, "Indoor robot navigation using a pomdp based on wifi and ultrasound observations," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2592–2597, Aug 2005.

[11] S. Iijima and I. Kobayashi, "A pomdp-based multimodal interaction system using a humanoid robot," in *Proceedings of the 30th Pacific Asia Conference on Language, Information and Computation: Posters*, pp. 519–525, 2016.

[12] R. Kaplow, A. Atrash, and J. Pineau, "Variable resolution decomposition for robotic navigation under a pomdp framework," in *2010 IEEE International Conference on Robotics and Automation*, pp. 369–376, 2010.

[13] D. Shepard, "A two-dimensional interpolation function for irregularly-spaced data," in *Proceedings of the 1968 23rd ACM national conference*, pp. 517–524, 1968.