Models and Framework for Supporting Humanoid Robot Planning & Exploration

Ahmad Hasanain, Troy Weekes, Michael Person, Kepinski Paul, Yonghui Chang, Aaron Rothman, Zhu Rui, Rajaa Rahil, Muntaser Syed, Chris Wodd, Marius Silaghi
Florida Institute of Technology

ABSTRACT

While the Nao humanoid is among the most advanced intelligent robots accessible to the global public, it lacks high level support for advanced intelligent walking activities. The robot has remarkable low level intelligence embedded into itself, such as artificial life with breath motions, and reasonable situation awareness with human face recognition and tracing. It also has motion control support in terms of unreliable distance or velocity-based commands for walking. However, building high level intelligent applications on such robots still requires significant research and effort. In this work we propose a framework of Artificial Intelligence (AI) models that create a platform for easy high level tasks specification and implementation. The models are trained using a Nao humanoid undergoing experimentation and the results are evaluated based on a set of high-level tasks implemented with standard AI algorithms. The models are described, as well as the test-beds and benchmarks used for their evaluation. The work proves the large potential that humanoids hold in the area of applications offering human support in daily activities.
Keywords: Humanoid, framework, model, AI, Nao.

1. Introduction

In this work we propose a probabilistic model for addressing the Nao humanoid robot non-deterministic actions and sensors. These models are designed for integration with high level artificial intelligence techniques, usable for advancing towards a long term goal of providing a software architecture for supporting Nao's motion and localization activities for general tasks.

A plethora of robotic hardware and test-beds are available for educational, research, and application opportunities. However, while the Nao humanoid robots are among the most advanced widely available robots for household applications, there is a lack of widely available software for supporting it in high level intelligent activities. This may be explained in part by the uncertainty introduced due to the significant non-determinism in walking actions presented by the Nao robot. Each single movement command and corresponding localization with respect to the base introduce errors of dozen of degrees in rotation, and dozens of centimeters in translation. Such errors can be compensated with inputs from sensors such as sonars and vision, but sonars have high unreliability themselves, and the vision needs reliable landmarks as well as previous knowledge about these landmarks. While the robot has two cameras, they are not configured from factory for usage as a stereo rig, being oriented in different directions and lacking synchronization of capture.

The Nao robot is a humanoid robot that is designed to serve as a human companion and can sustain well household environments.

Figure 1. The benchmark setting: mazes with landmarks.

While its relatively small dimensions limit its usability for tasks requiring strength or height and dexterity, it is well adapted to home, hospital, and office environments to provide, for example, entertainment and emergency support for children and seniors. While its communication and emotional intelligence is meeting many expectations, its movement and ability to localize with standard software is reduced, and insufficient for most other tasks. A number of research efforts have led to the description of successful applications of *simultaneous localization and mapping* (SLAM) to certain tasks such as wall following in mazes and localization in spaces filled with landmarks [1]. However there is a lack of general software packages applicable to new scenarios. In this work we take a step towards designing such a general mapping and localization support software architecture.

After introducing a benchmark problem based on mazes with landmarks (Figure 1), and reviewing some of the related work, the rest of the paper details the definition and training of probabilistic models for the Nao sonars and visual landmark localization. These probabilistic models are being tested by integration into particle-filtering based SLAM reasoning or partially observable Markov decision process (POMDP) formulation of planning problems. These are significant building blocks for the software architectures that we plan de address. Preliminary experimental results are presented before the conclusions.

2. Problem Formalization

The problem is described by a grid of $m \times n$ cells where some of the separating walls are missing, creating a maze. This can be formalized as two Boolean matrices $H_{m+1,n}$, and $V_{m,n+1}$, where in our example above, $m = 4$ and $n = 4$. The matrix $H_{m+1,n}$ with $m+1$ rows and n columns specifies the presence of separating walls between east-west neighboring cells. The matrix $V_{m,n+1}$ specifies the presence of separating walls between north-south neighboring cells. The functions $h : [1..m+1] \times [1..n] \times \{S,N\} \rightarrow \mathbb{N}$ and

```
+ ~ + ~ + ~ + ~ +
|       |       |
+ ~ +   +   + ~ +
|               |
+ ~ +   + ~ +   +
|       |   |   |
+   +   +   +   +
| X |   |       |
+ ~ + ~ + ~ + ~ +
```

Figure 2. The shape of a maze used in experiments, in ASCII representation.

$v : [1..m] \times [1..n+1] \times \{E, W\} \to \mathbb{N}$ assign a distinguishing label to each wall, corresponding to the elements in $H_{m+1,n}$ and $V_{m,n+1}$, respectively. The image of the functions h and v is not defined on inputs where $H_{m+1,n}$ and $V_{m,n+1}$ are false, respectively. The sets $\{$S,N$\}$ and $\{$E,W$\}$ specify the sides of the wall whose label is considered: *south, north, east*, and *west*, respectively. The image of the function can be a special pattern recognizable by a classifier, or something as simple as a NAOmark label. In an extreme case, where there is no distinguishing information between walls, the image of the two functions would consist in a single value.

A robot is placed in this maze in a potentially unknown original position and has to learn the elements H, V, h, and v. The robot can move with non-deterministic actions inducing transition probabilities that can be modeled as Gaussian. It also has noisy sensors that can detect H and V (sonars) or v and h (vision).

The position π of the robot at a moment in time is jointly defined by a maze cell (i,j) with $i, j \in [1..m] \times [1..n]$, as well as a position and orientation in the cell defined with respect to its south-west corner (x, y, θ).

The past trajectory of the robot, acts as a weight gauging the confidence of the robot in particular components of the belief. The trajectory can be maintained with various levels of detail. The a temporal trajectory of the robot is defined as a Boolean matrix $T_{m,n}$, where elements are set to true for maze cells that have been visited in the past.

A belief concerning the structure of the maze and position of the robot is a probability distribution b over the complete set of possible data structures T, H, V, h, v, and π.

For mapping or SLAM problems in this setting, the above data structures, namely H, V, h, and v, as well as the belief concerning the final position of the robot, are requested outputs while the inputs may consist in:

- values for m and n,
- specification of landmarks for distinguishing walls (e.g., NAO marks at 1/2 size)
- dimensions for cells (e.g., $21in \times 21in$), as well as
- a belief regarding the current position of the robot in the described space.

The task is to have a mobile robot explore and localize itself within a maze. The map of the maze is assumed initially unknown to the robot other than that it consists of cells, which are analogous to rooms or parts of rooms in an office environment, and the cells' shapes are in a predefined finite set of possibilities. For simplicity, the cells shapes are assumed empty squares with 53.34cm sides, each being possibly a wall/object as depicted in Figure 2.

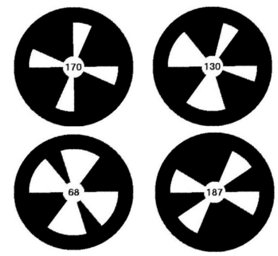

Figure 3. Landmark examples, each consists of a unique integer conveyed in the NAOmarks pattern.

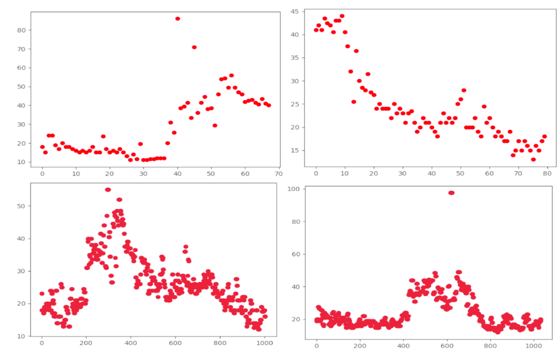

Figure 4. Probabilistic Sonar Model.

3. Nao5 Robot Specifications

Our experiments, and corresponding probabilistic models obtained for sensors and actions, are constructed with a Nao robot at version 5. This robot boasts 2 sonars on the chest, bumpers on the feet, and two cameras placed vertically on the face. Nao has multiple other sensors in its actuators. The sonars and landmark recognition modules have significant noise, as highlighted by the described experiments. The landmarks utilized, which require the software to only be informed about their size, are recognized by the built-in ALLandMarkDetection software package, and are shown in Figure 3. The package provides an estimate for the abscissa, ordinate, and projection angle of the smallest line of sight relative to the robot's camera and the NAOmarks. Furthermore, adjusting the yaw of the head and joints of the limbs can be performed with high accuracy, but walking and turning have significant error.

4. Probabilistic Sensor Model

Measurements of the Nao sensors response as a function of the distance between the Nao torso and walls, are shown in Figure 4. Sonar readings are taken while the robot is rotating its body 360°. In each image in the Figure 4, the robot was placed in the same cell but with different position and angle. The sonar reading are in cm in the y-axis, while the x-axis is in a unit of time, which is proportional to the robot's body rotational angle captured uniformly while the robot motion was active.

Although the robot was supposed to take 360° (degrees) to turn around, it usually needs approximately 390°. Consequently, the direct scattering of these distance readings in an x-y plane is apparently not straightforward. The x- and y- axes in the Figure 6 are in cm. The motion started when the robot was exactly facing the positive x-axis and ended around the axis. Notice that the initial distance to the wall was 10cm while it landed at 15cm from the same wall.

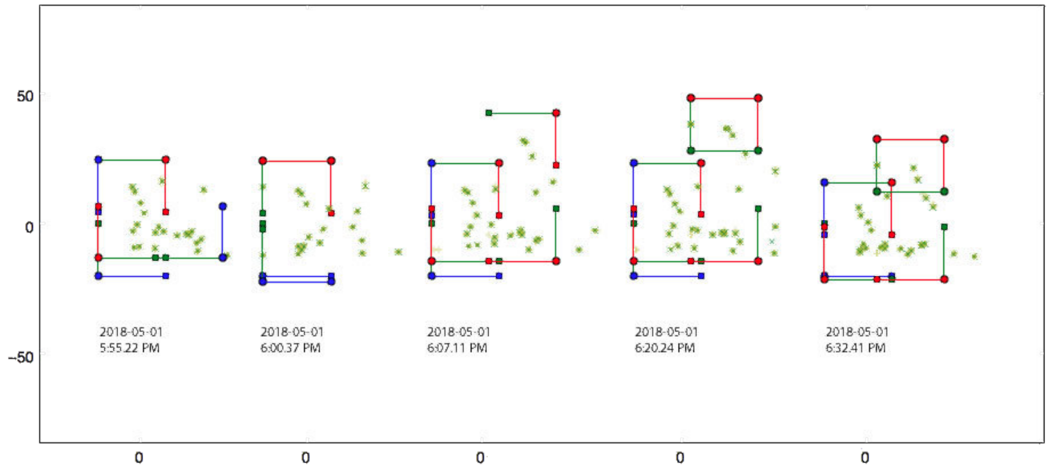

Figure 5. The localization results of five tests.

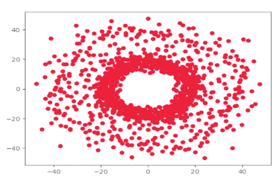

Figure 6. Shape of sonar signal during robot rotation.

Based on such measurements, statistics are used for training a probability distribution of the sensor readings with respect to wall distance.

5. Landmark Detection Models

A visually based localization algorithm is proposed which utilizes a probabilistic sensor fusion approach. On-board the humanoid Nao robot are a stereo camera and two sonars. The Nao is preprogrammed to perform an image space search for NAOmarks, seen in Figure 3, and provide the vector separating the Nao robot from the marker. However, it is unknown what the underlying algorithm for this search is and it also provides noisy measurements, most likely due to its required generalizability to all scenarios. A naive approach to reduce the uncertainty due to the noise of the Nao's built-in localization would be to form an estimator of the true value by forming a sample and computing the expected value. The sample variance could also be computed to act as an error bound for the estimator. In any case, it is inconvenient to form a large enough sample required to place small enough bounds on the error.

An improvement is proposed, as follows. The robot is surrounded by at most three Nao marks if a wall is present. The Nao captures images using both camera's of what is directly in front of it. The images are then rectified using the known camera intrinsics and then the epipolar lines are searched to compute the stereo pair's disparity map. The disparity map is then mapped to a depth map using the camera extrinsics. This depth map provides two purposes; the first is to augment each pixel location with a depth value and also to create a Cartesian 3D point cloud using the camera's Projection matrix.

Next the reference camera's image is passed through the Hough Circle Transform to retrieve any possible circular like regions of interest (ROI) in the image. In order to separate out the NAOmarks from this noisy set of ROI's, a Histogram of Oriented Gradients (HOG) feature vector is extracted from each region. Each ROI's feature vector is passed through a binary Support Vector Machine (SVM) to classify whether the ROI is a Nao mark or an extraneous detection. After the true Nao mark pixel locations have been identified, the mean depth for each mark is computed from the depth map and the mean offset, in both axis, is computed using the point cloud.

This provides a set of vector's to any potential Nao markers in the Nao's current field of view. During this processing time, samples from the Nao's application programming interface (API) have been collected and stored to be used later. If any Nao marks have been found, the sonar values are also aggregated. Although the sonar measurements will not directly pick up the Nao mark, they will pick up the distance to the wall it is placed upon. Finally we have a set of three different measurement types; our own visually based set of vectors, a set of Nao API based vectors, and a set of sonar radial distance measurements. The expected value of these three sets are then computed however each set has a different probability associated with it. We place a probability of 50% to our own measurement, a probability of 30% to the Nao API's sample mean, and finally a probability of 20% to the sonar depth measurement. This expected value performs a fusion on the three different sets of data to provide a more accurate localization than a single stream of data alone.

The process is then repeated with the Nao's head turned left and right. This eliminates the data stream from the sonar's however this is the more optimal solution to having Nao's entire body turn which introduces motion noise in addition to sensor noise.

6. Motion Planning

The Nao robot has a significant amount of noise in it's movements. The noise is primarily attributed to the asynchronicity of it's different operational threads; thus, making the Nao an ideal robot for developing probabilistic models. The first step in developing any autonomous capabilities of the Nao is creating an accurate motion model. In order to simplify the transition model, a priori knowledge of the maze and the Nao is used to reduce the set of possible transitions. For example, the Nao will be assumed to not be able to move in the direction opposite to the command. Since the location the Nao will occupy in a single cube is continuous, the position is discretized into a four by four grid of possible locations the Nao could occupy. This discretization reduces the complexity of the transition model. The transition between the Nao going from one cube to another can be seen in Figure 7. The Nao is marked with a red circle, the current discretized cube is shown in green and the set of possible states it can transition to are shown in blue. The blue region extends past a single cube because its non-deterministic motion model allows for a small probability to pass beyond its given command.

In order to help decrease uncertainty that the Nao has moved to the expected location of the transition model, a NAOmark was located in the cell that the transition is set to move into. When the Nao is ready to move, it will lock with the NAOmark and gather readings continuously while transitioning into the next cell. The readings consist of x, y, z positions from the marker as well as rotation along each axis which can be converted to angles of view of the NAOmark. These angles of view allow the Nao to align perpendicularly to the NAOmark it has a lock onto. From there, the next set of readings can be taken to localize in that cell. The results of five tests are illustrated in Figure 5. It is assumed that there will be 2 or 3 NAOmarks in each cell, having distinct numbers within the cell; however, other cells in the maze can contain NAOmarks with the same number. It is also assumed that each NAOmark is placed in the center of a wall; with this known and the length of a single wall known, using just the sensor readings, the Nao can position itself very close to the center of the cell during transition. This can be used as input motion model to the Sequential Monte Carlo Method described in the next section.

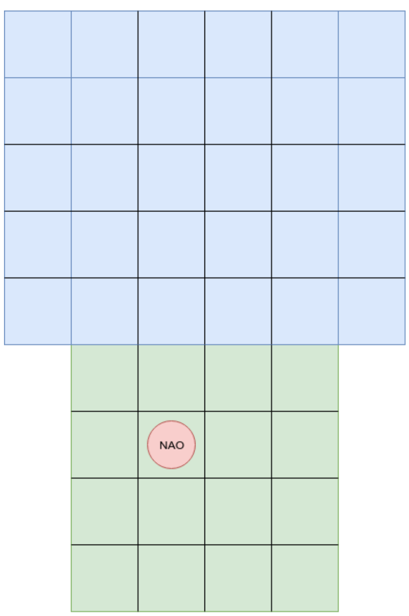

Figure 7. Transition of the Nao from one cell to another

7. Sequential Monte Carlo Method

The Kalman Filters are a powerful class of algorithms commonly used for a variety of probabilistic tasks [2]. However, they have strong requirements for both Gaussian distributions and linear models. In order to address these stringent limitations for real world applications, Extended Kalman filters and Unscented Kalman filters were introduced, allowing for the use of non-linear models [3, 4]. However, many applications cannot be assumed to be Gaussian, making the Kalman Filter family of algorithms unsuitable. For example, skewed normal distributions also appear in several applications. On the other hand, the Sequential Monte Carlo Method, or Particle Filter, is a robust probabilistic method that is able to deal with any distribution and non-linear transitions. Although there is no general proof of convergence for particle filters, they empirically provide accurate results with much less time complexity. For our applications, particle filters will be used for both localization and for Simultaneous Localization and Mapping (SLAM).

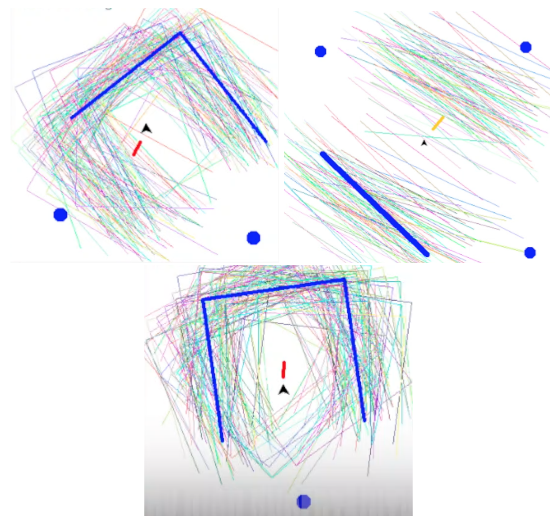

Figure 8. Particle filtering for Monte-Carlo Localization in a maze cell. The thick line represents a generated hypothesis whose attributes are proportional to the arithmetic mean of the population of the hypotheses.

For Monte Carlo localization, particles described by specific hypotheses (H, V, h, v, π), are generated and weighted according to the likelihood of the evidence (i.e. history of sensor readings) in the corresponding world. The robot first considers all possible world cases, in which the robot may be placed. The hypotheses space is weighted after each sensor reading taken after a robot motion. While the hypotheses are weighted, new particles are generated (resampled) according to the new weight distribution. In our settings, a particle directly conveys the state of the world as oppose to the state of the robot in the world, and thus the shape of the cell, in which the robot is, is one of the discrete parameters that a particle can describe. The other attributes, such as the x- and y- axes as well as the rotation of the hypotheses, have continuous dimensions. Since the Nao's sensors behave differently when the sensed objects are closer than 20cm, a Boolean distance measure sets the weights of the particles that match to a corresponding value (e.g., 0.5). Additionally, the particles whose features are beyond the hypothesis space are regenerated. Finally, a constant weight (e.g., 10 percent) is given to particles that have not been observed by the sensor model, to preserve the unseen hypotheses unless particles with very high certainty emerge in the hypothesis population.

Figure 9 depicts three iterations during a Monte Carlo Localization search in a situation when the robot is rotating in a single cell. Each assembly of same-color lines resembles a particle. The number of particles in each conducted experiment is 3000, and a momentum was added to the weights to compensate for the first degree Markov transition assumption. The later the iteration number, the more aligned the particles are, as can be seen in Figure 9. The thick blue lines, in the figure, resemble a particle whose placement was at the arithmetic means of the continuous attributes of

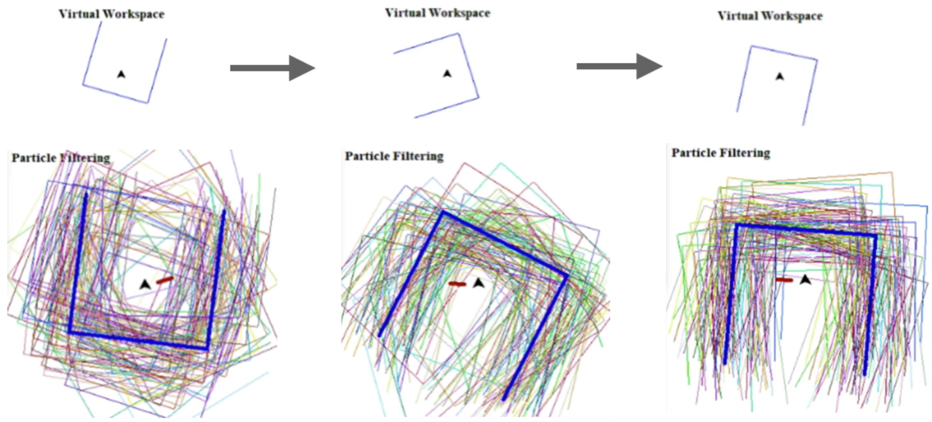

Figure 9. Particle filtering for Monte-Carlo Localization in a maze cell. The first row shows three iteration states. Time is illustrated from left to right, and the second row presents the particles in the three states.

the 3000 particles. The only discrete attribute is the shape of the cell; therefore, the shape of the resulting hypothesis is the shape of the majority of the particles. The module gave consistent results as the shape of the cell differed. Figure 10 portrays two other examples, from left to right, when the number of walls modeled by the particles was one and three respectively. Our configurations allow for more complex settings than the assumed simple squared cells. The stopping criterion of our algorithm was when 99% of the population had the same shape; hence, the convergence of the particles. The configuration with two parallel walls was also tested. The large blue and green discs picture the centers of the adjacent cells to which the robot could move. The module was backed by the breadth-first shortest path search and a database that records the cells and passages which were visited and found.

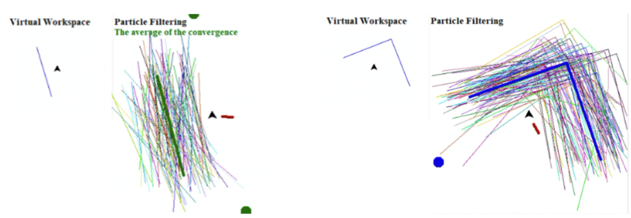

Figure 10. Two examples of the particle-filtering-based localization when the number of cell sides vary.

8. Related Work

The problem of simultaneous localization and mapping (SLAM) for a Nao robot placed in a room with multiple Nao marks placed at random locations, at the height of the cameras of Nao, was addressed in [1]. The effort led to new SLAM approaches for real-time incorporation of new landmarks in exploration.

Probabilistic reasoning for localization has been used with mobile robots to address various types of problems. If the map of the explored world is known, then Monte Carlo Localization can employ particle filtering [5].

Motion planning in partially observable non-deterministic environments of this type can be modeled with Partially Observable Markov Decision Processes (POMDPs), for which various techniques have been proposed to tame the complexity challenges. All these techniques can employ a dynamic belief network representation of the problem [6].

9. Conclusion

In this work we build a probabilistic model of the humanoid robot Nao sensors and actions, enabling the application of high level intelligent algorithms for tasks such as localization, mapping, and planning. Preliminary results are described and several compact methods of representation that enable efficient reasoning were developed.

The Nao robot is found to have significant non-determinism in walking and rotation actions. Additionally, its sonar sensors have significant noise and built-in visual NAOmark landmark detection shows noise that increases when the head's yaw rotational angles are larger than 80 degrees. The proposed models can be used in POMDPs and SLAM solvers.

References

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

[2] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[3] N. J. Muga and A. N. Pinto, "Extended kalman filter vs. geometrical approach for stokes space based polarization demultiplexing," *IEEE/OSA Journal of Lightwave Technology*, vol. 33, pp. 4826–4833, December 2015.

[4] E. A. Wan and R. V. D. Merwe, "The unscented kalman filter for nonlinear estimation," pp. 153–158, 2000.

[5] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, "Robust monte carlo localization for mobile robots," *Artificial intelligence*, vol. 128, no. 1-2, pp. 99–141, 2001.

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

[7] N. d. F. . N. G. Arnaud Doucet, ed., *Sequential Monte Carlo Methods in Practice*. Springer, 2001.