Sensor fusion for interactive real-scale modeling and simulation systems

M. Ali Mirzaei, Jean-Rémy Chardonnet, Christian Père, Frédéric Mérienne
Arts et Métiers ParisTech, CNRS, Le2i, Institut Image
Chalon-sur-Saône, France
{ali.mirzaei, jean-remy.chardonnet, christian.pere, frederic.merienne}@ensam.eu

***Abstract*—This paper proposes an accurate sensor fusion scheme for navigation inside a real-scale 3D model by combining audio and video signals. Audio signal of a microphone-array is merged by Minimum Variance Distortion-less Response (MVDR) algorithm and processed instantaneously via Hidden Markov Model (HMM) to generate translation commands by word-to-action module of speech processing system. Then, the output of an optical head tracker (four IR cameras) is analyzed by a non-linear/non-Gaussian Bayesian algorithm to provide information about the orientation of the user's head. The orientation is used to redirect the user toward a new direction by applying quaternion rotation. The output of these two sensors (video and audio) is combined under the sensor fusion scheme to perform continuous travelling inside the model. The maximum precision for the traveling task is achieved under sensor fusion scheme. Practical experiment shows promising results and new insights for computer games.**

Keywords—sensor fusion; navigation; speech processing; optical head tracking

I. INTRODUCTION

Whenever something is too dangerous, expensive, and time consuming, there have been attempts to simulate it. One of the nowadays known real-scale simulation technology is virtual reality (VR). Unlike TV and movies which convey passive real or imagined experiences, new VR technologies bring a measure of the conveyed experience with interactivity.

Simulation has been a backbone for designers, car manufacturers, and airplane pilots during the past years, ever since the first VR system came to be. The newest generations of simulators have incorporated more VR techniques and concepts to bring even greater level of realism into practice. For example, architects use more of VR simulator to study the effect of different design aspects on the habitants [1]. The benefits and savings in time and cost are enormous. This kind of simulation is mostly referred in the literature as Serious Games (SG) that are similar to computer games in terms of interaction and navigation. If a navigation device is designed for a VR system, it can be used in computer games as well.

Immersive virtual environment (VE) [2] is a real-scale 3D computer-generated "Environment" in which a person can move around and interact as if he/she actually was in this imaginary place. This is accomplished by totally immersing the person's senses in the "virtual world" using a set of interaction devices such as fly-stick, gamepad, data-gloves, and so on.

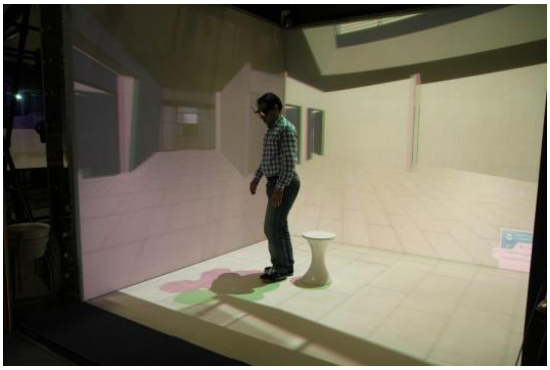

Fig. 1. Example of navigation task in a real-scale simulator system

Recently, different methods based on natural language, gesture, gait using speech processing and body skeleton analysis have been proposed [3].

The head-tracker is an input device (a configuration of multiple IR-D cameras and laser beamers) which calculates the head position and orientation in a multi-screen VE (for instance a CAVE™ system). One can "move" through the VE in a direction that the user is heading to. The model can be anything which can be experienced three dimensionally.

Going from one point to another along a specific direction usually needs a navigation device such as a gamepad, however it can be performed by analyzing the data coming from the head-tracker. Sometimes the navigation device has orientation, position and command buttons altogether. Speech processing and turning the detected words into action can create a potential alternative for navigation in VE.

Our contribution in this paper is to use a voice command for translation and head orientation for rotational movement and to combine these two sensors data (audio, video) to perform a complete navigation task continuously (Fig. 1).

The paper is organized as follow: in section II, the signal associated to two types of sensor, voice and image, will be processed to extract useful features. Then, the features will be combined under a novel sensor fusion scheme to generate commands for rotational and translation movement in section III. The details of the software development platform will be discussed in section IV. Hardware apparatus and devices will be explained in section V. Finally we will have a short discussion over the accuracy of the results and the conclusion in the last section.

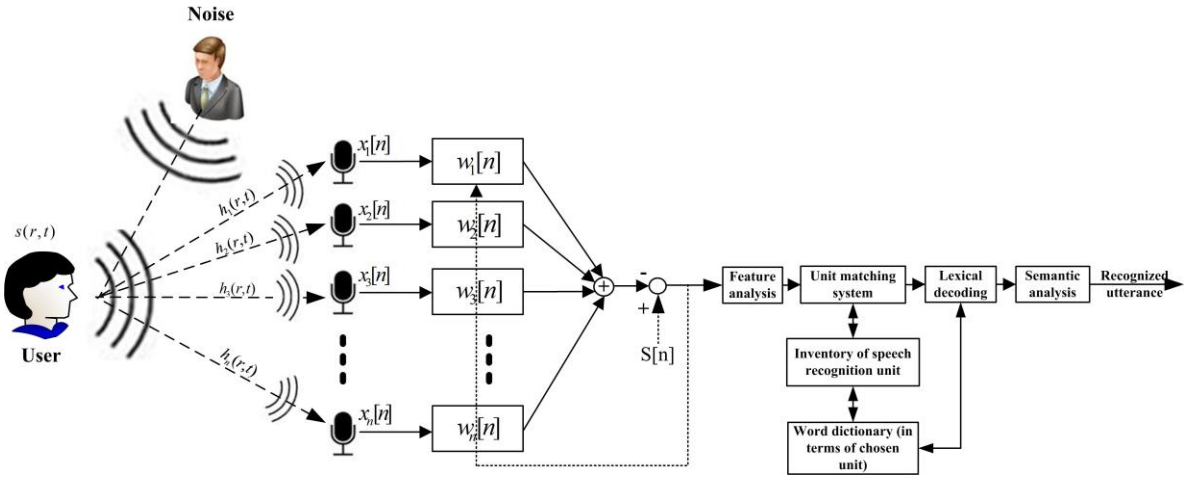

Fig. 2. Scheme for microphone-array sound capturing and speech processing in a noisy acoustical environment

II. SIGNAL PROCESSING APPROACH FOR TWO TYPES OF SENSORS

Two sensor configurations will be introduced in this study: 1) microphones-array (voice) and 2) infrared depth (IR-D) sensor (video). Each sensor employs its own unique configuration and processing approaches to capture and analyze the associated signal.

A. Audio sensor (microphone-array)

1) Audio system configuration

A robust real-time speech processing approach with the presence of the noise while creating less delay will be addressed here. We consider a sound capture situation, noise, echoes and reverberation as sketched in Fig. 2. The channel impulse responses $h_i(r,t)$ describe sound propagation from the source to the individual microphones. The discrete-time beamformer is modeled by an FFT overlap-add filter bank [4]. The MVDR [5] beamformer algorithm in the frequency domain is used to analyze this multi-channel system. An MVDR beamformer optimizes the power of the output signal under constraint that signals from the desired direction are maintained [6]. Optimization constraints (1) can be solved using Lagrange's method (2):

$$w_0 = \arg\min w^H S_{xx} w, \text{ with } w^H h = 1 \quad (1)$$

$$\nabla_w \left[w^H S_{xx} w + \lambda \left(w^H h - 1 \right) \right] = S_{vv} w + \lambda h = 0 \quad (2)$$

where, S_{xx}, w, ∇_w are the spatio-spectral correlation matrix, beamformer weights and the gradient with respect to the weight vector, respectively. Superscript H and h denote conjugate transpose and the channel transfer function vector. Combining the constraint equation from (1) with (2) leads to the well-known solution for the optimum weight vector

$$w_0 = \frac{S_{vv}^{-1} h}{h^H S_{vv}^{-1} h} \quad (3)$$

If the noise is assumed as homogeneous diffuse noise and if we estimate S_{vv} for each signal frame with index m by

$$S_{vv}\left(e^{j\theta}, m\right) = \alpha S_{vv}\left(e^{j\theta}, m-1\right) + (1-\alpha) v\left(e^{j\theta}, m\right) v^H\left(e^{j\theta}, m\right) \quad (4)$$

where $\theta = 2\pi \frac{f}{f_s}$ is the frequency variable and $\alpha \approx 0.8$ (v is the vector of the noise spectra), the optimum weight vector can be found iteratively with a steepest descent algorithm expressed by (5):

$$w_{k+1} = w_k - \mu \nabla_w \left[w_k^H S_{xx} w_k + \lambda \left(w_k^H h - 1 \right) \right] = w_k - \mu \left(S_{vv} w_k + \lambda h \right) \quad (5)$$

Lagrange multiplier, λ, is calculated by substituting second constraint of (1) in (2). By eliminating λ from (5) we finally get the update equation (6):

$$w_{k+1} = w_k - \mu \left(1 - \frac{h h^H}{\|h\|^2} \right) S_{vv} w_k = w_k - \mu g_k \quad (6)$$

In (6), the weight vector updates by using S_{vv} estimated from (4) and iterate in each frame. Furthermore, convergence speed is improved by computing an optimum step size factor μ. We choose a step size so that it minimizes the noise power at the beamformer output for each iteration:

$$\frac{\partial \left(w_{k+1}^H S_{vv} w_{k+1} \right)}{\partial \mu} = 0 \quad (7)$$

Combining (6) and (7) results in

$$\mu_k = \frac{g_k^H S_{vv} w_k}{g_k^H S_{vv} g_k} \quad (8)$$

Then we can summarize the whole process in the flowchart shown in Fig. 3.

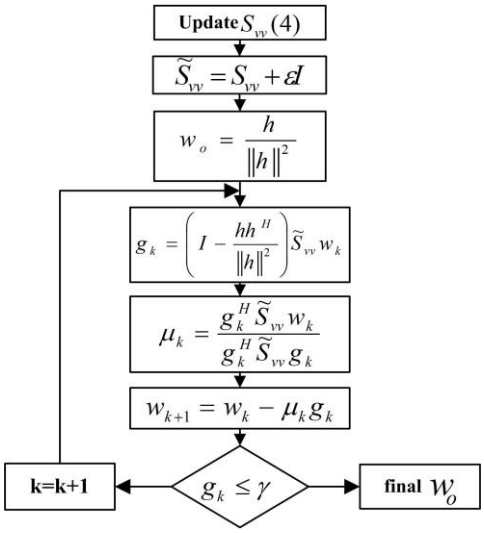

Fig. 3. Flow chart for weight vector calculation

TABLE I. THE VOCABULARY LIST IN THE DICTIONARY OF SPEECH PROCESSING SYSTEM (LEXICON)

	Vocabulary					
	start	***stop***	***forward***	***backward***	***Turn right***	***Turn left***
Code	00001	00010	000100	001000	010000	100000

2) Speech processing

Fig. 2 shows our speech recognition approach and system components. The key signal processing components include Feature analysis, Unit matching system, Lexical coding, and Syntactic analysis.

The speech signal processing is performed in the frequency and time domain to extract observation vectors which can be used to train the HMM [7]. The HMM is a strong algorithm to characterize various speech sounds.

First, a choice of speech recognition must be made by unit matching system. Possibilities include linguistically based sub-word units as well as derivative units. For a specialized application (current application), it is both reasonable and practical to consider the word as a basic speech unit. We will consider such systems exclusively in this literature. For that, we use only words included in Table I.

Lexical coding process places constraints on the unit matching system so that the paths investigated are those corresponding to sequences of speech units which are in a word dictionary (lexicon). This procedure implies that the speech recognition word must be specified in terms of basic units chosen for recognition. Syntactic analysis adds further constraints to the set of recognition search paths. One way in which semantic constraints are utilized is via a dynamic model of the state of the recognizer.

When the vocabulary is recognized by the speech processing system, associated code (second row in Table I) will be selected and sent to the navigation application and the appropriate command is generated for the simulator.

B. Vision sensor(IR-D with laser beamer)

1) IR-D and laser beamer configuration

IR-D cameras and a laser beamer were arranged in a precise configuration to localize and calculate the position of the user's head. The setup is a real-time optical tracking system including four IR-D cameras. The system uses the TOF (Time of Flight) theory, image processing based on morphology and triangulation to estimate the position of the head. Five balls (indicators in Fig. 4) were attached to an E-shape marker and mounted on 3D stereoscopic glasses.

2) IR image processing for head tracking

The optical tracking system uses non-linear/non-Gaussian Bayesian algorithm [8] to track the markers. From a Bayesian perspective, the tracking problem is to recursively calculate some degree of belief in the state of x_k at time k, taking different values, given the data $z_{1:k}$ up to time k. Thus it is required to construct the Probability Distribution Function (PDF) $p(x_k|z_{1:k})$. PDF $p(x_k|z_{1:k})$ may be obtained, recursively, in two stages: prediction and update. Suppose PDF $p(x_{k-1}|z_{1:k-1})$ at time k-1 is available. The prediction stage involves using (9) (process model) to obtain the prior PDF (prediction stage) of the state at time k via (10).

$$x_k = f_k(x_{k-1}, v_{k-1}),\ z_k = h_k(x_{k-1}, n_k) \tag{9}$$

$$p(x_k|z_{1:k-1}) = \int p(x_k|x_{k-1}) p(x_{k-1}|z_{1:k-1}) dx_{k-1} \tag{10}$$

At time step k, a measurement z_k becomes available, and this may be used to update the prior stage via Bayes' rule (11):

$$p(x_k|z_{1:k}) = \frac{p(z_k|x_k) p(x_k|z_{1:k-1})}{p(z_k|z_{1:k-1})} \tag{11}$$

Where, the normalizing constant is calculated by (12)

$$p(z_k|z_{1:k-1}) = \int p(z_k|x_k) p(x_{k-1}|z_{1:k-1}) dx_k \tag{12}$$

Now by having the position and the orientation of the head we can implement the navigation task (rotation by head orientation). Any rotation in a 3D space can be represented as a combination of an axis vector and an angle of rotation. Quaternions give a simple way to encode this axis-angle representation and apply the corresponding rotation to a position vector representing a point relative to the origin in R^3.

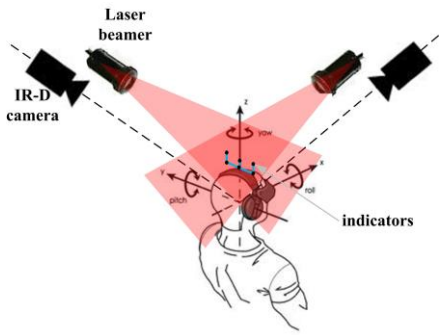

Fig. 4. Vision sensors and data acquisition setup

A unit vector such as (a_x, a_y, a_z) in Cartesian coordinate system can be rewritten as $a_x i + a_y j + a_z k$, making it a pure imaginary quaternion. A rotation in a 3D space with an angle of θ around the axis defined by a unit vector $\bar{u}$ is represented by a quaternion using an extension of Euler's formula (13).

It can be shown that this rotation can be applied to an ordinary vector p in 3D space. Vector p can be considered as a quaternion with a real coordinate equal to zero and the rotation operation can be expressed by (14) (well known as Hamilton product), where, (p'_x, p'_y, p'_z) in (14) is the new vector after the rotation. If p and q are quaternions representing two continuous rotations, then (15) is the result of the final rotation which is the same as rotating by q and then by p.

$$q = e^{\frac{1}{2}\theta(u_x i + u_y j + u_z k)} = \cos\left(\tfrac{1}{2}\theta\right) + (u_x i + u_y j + u_z k)\sin\left(\tfrac{1}{2}\theta\right) \quad (13)$$

$$p' = qpq^* \quad (14)$$

$$pq\vec{v}(pq)^* = pq\vec{v}q^* p^* = p(q\vec{v}q^*)p^* \quad (15)$$

III. Fusion of Sound and Vision Data

A sensor fusion scheme for combining video and audio data is shown in Fig. 5. A complete navigation task in the real-scale 3D system or simulator consists in translational and rotational movements. However a combination of these two tasks is applied to user navigation inside the 3D model. Audio and video data are analyzed and processed to provide command for translational and rotational movements respectively.

When the word "start" is said, the translational movement begins and by saying "stop", the movement is terminated. Velocity and acceleration can be automatically/manually adjusted by the navigation function. Rotation angle around the "row" axis (Fig. 4) is used for the orientation of the movement.

IV. Architecture of Development Platform

Fig. 6 shows the architecture of the development platform. All the C++ functions are wrapped under Java code. A set of functions have been developed for navigation inside a 3D model. The velocity and the acceleration of the translational and the rotational movement can be controlled via a pair of buttons on the navigation devices. The resolution of the changes is the same in each step, 0.05 units. Devices are connected via Virtual-Reality Peripheral Network (VRPN) to the platform while generated output is projected in the display system by MPI.

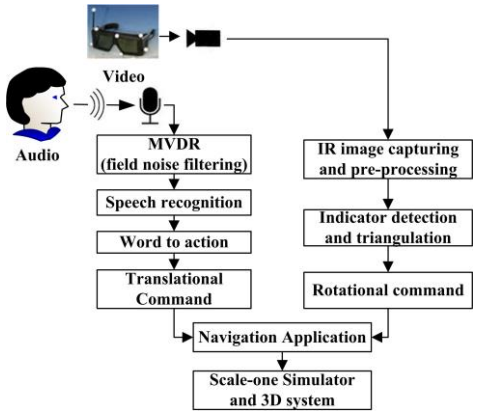

Fig. 5. Sensor fusion scheme for combining audio and vision data

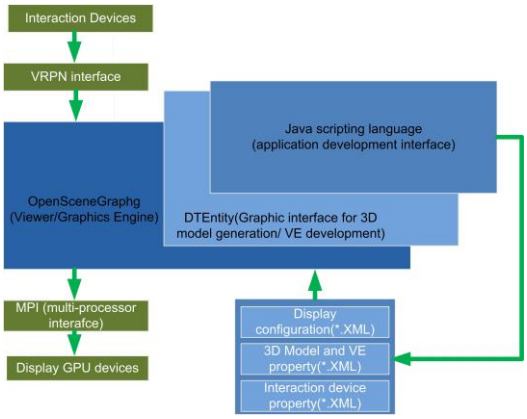

Fig. 6. Development platform

VRPN also provides an abstraction layer that makes all devices of the same base class look the same, however, it does not mean that all trackers produce the same report. Each of these abstracts is a set of semantics for a certain type of device.

V. Hardware Apparatus and Devices

A 4-side CAVE™ system, with two projectors per side, is used to implement and test the navigation system. An infrared based head-mounted tracking system (Fig. 4) is used to find the user location. A microphone array with 4 microphones was used to acquire user's voice. A custom made platform called Petriiv (see section IV) was developed to manage the connection between display projectors, infrared camera, sound source and the networking. The platform uses OpenSceneGraph on the top of OpenGL to render the 3D model. Then the model is projected into the display system by MPI and four NVidia Quadroplex GPUs (Fig. 6).

VI. Experiment

Navigation by three methods as shown in Fig. 7 and Fig. 8 was carried out along the same path and trajectory.

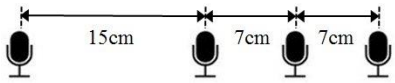

Fig. 7. Experiment 1: with microphone-array

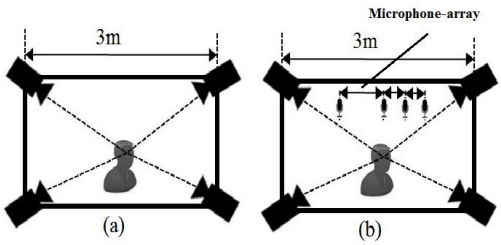

Fig. 8. a) Experiment 2: head tracking system b) Experiment 3: sensor fusion

Experiment 1: only microphone-array was used to command and travel inside a 3D model while the user was standing 1.5 meters away from the sensors (Fig. 7) and commanding the system. The vocabulary list in Table I was applied to this experiment.

Experiment 2: head-tracking with a configuration shown in Fig. 8.a were imposed to perform the navigation task.

Two sensors in previous experiments were combined to make experiment 3 (Fig. 8.b). The translational movement in experiments 2 and 3 can be initiated and terminated by both a fly-stick equipped with tracker and the user head movement (the person imitates walking in place and an algorithm is designed to capture and analyze the movement).

VII. Results and Discussion

Velocity (speed) is one of the important parameters for navigation system. Three different devices (voice, vision, sensor fusion) were employed to perform traveling task inside a 3D model. Precision of each navigation device was measured during the traveling through a target path with different speeds. Precision was calculated by accumulating rotational and translational special error (%) and subtracting from 100%. As seen, after sensor fusion, maximum precision is achieved (from 70% to 85%). However, maximum precision happens between 3 and 6 ($m.s^{-1}$) as shown in Fig. 9 (at 5 $m.s^{-1}$).

Besides, commanding the system with voice is more precise for lower speed while head tracking works more precisely in higher speed. It is because speech processing has a delay to process the speech and this delay is not so perceivable in the lower speed.

The precision of each method is differentiated for translational and rotational task for the three devices. Voice command is more precise for translational while video command is more precise for rotational movement. Sensor fusion benefited from positive aspect of each navigation method to provide more precision as shown in Table II.

Another parameter is the precision of voice command with/ without pre-processing (MVDR) step. The speech processing system with MVDR is 2 and 1.5 times more accurate for rotation and translation command respectively as seen in Table III.

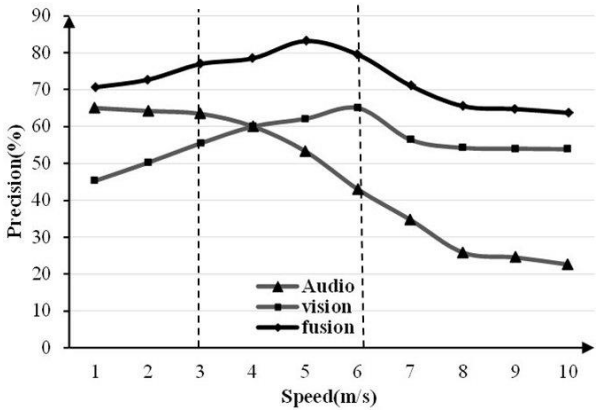

Fig. 9. Comparison between after and before fusion

TABLE II. Precision of Rotation and Translation Sub-Task Before and After Sensor Fusion

Method	Precision (%)		
	Rotation	***Translation***	***Trajectory (rotation and translation)***
Audio (speech processing)	65.31	85.58	53.64
Video (IR-D image processing)	95.23	70.87	62.90
Sensor fusion (audio, video)	96.47	87.72	92.56

TABLE III. Effect of MVDR on the Accuracy of Speech Processing

Method	Speech processing without MVDR		Speech processing with MVDR	
	Rotation	***Translation***	***Rotation***	***Translation***
Precision (%)	35.28	63.45	58.51	84.21

VIII. Conclusion

Audio and video signals along with their associated processing approaches for generating translational and rotational commands were discussed. These two sensor data were combined under a sensor fusion scheme to perform complete travelling tasks inside either a 3D model or a simulator. The result shows 20% improvement in the total performance of the navigation system after the sensor fusion. Moreover, the maximum precision (85%) of the navigation system appears to be between the speed of 3 and 6 $m.s^{-1}$. Besides, speech processing system with MVDR preprocessing will yield two times precision (58.51%) than without MVDR.

Acknowledgment

This paper is supported under national project FUI Callisto.

References

[1] A. Kanamgotov, A. Christopoulos, M. Conrad, and S. Prakoonwit, "Immersion in virtual worlds-but not second life!," in International Conference on Cyberworlds (CW), pp. 107–113, 2012.

[2] G. Bruder, V. Interrante, L. Phillips, and F. Steinicke, "Redirecting walking and driving for natural navigation in immersive virtual environments," IEEE Transactions on Visualization and Computer Graphics, vol. 18(4), pp. 538–545, 2012.

[3] O. M. Vultur, S. G. Pentiuc, and A. Ciupu, "Navigation system in a virtual environment by gestures," in 9th International Conference on Communications (COMM), pp. 111-114, 2012

[4] H. Pulakka and P. Alku, "Bandwidth extension of telephone speech using a neural network and a filter bank implementation for highband mel spectrum," IEEE Transations on Audio, Speech, and Language Processing, vol. 19(7), pp. 2170–2183, 2011

[5] W. Shao and Z. Qian, "A new partially adaptive minimum variance distortionless response beamformer with constrained stability least mean squares algorithm," Advanced Science Letters, vol. 19(4), pp. 1071–1074, 2013.

[6] M. Rubsamen and A. B. Gershman, "Robust adaptive beamforming using multidimensional covariance fitting," IEEE Transactions on Signal Processing, vol. 60(2), pp. 740–753, 2012.

[7] Rune Lyngsø, "Hidden markov models," Narotama, 1:1–24, 2012

[8] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," IEEE Transations on Signal Processing, vol. 50(2), pp. 174–188, 2002.