Navigation of a mobile robot on the temporal development of the optic flow

Anuj Dev, Ben Kröse, Frans Groen

RWCP Novel Functions SNN*Laboratory
Department of Computer Science, University of Amsterdam

Abstract

The robot navigation task presented in this paper is to drive through the center of a corridor, based on a sequence of images from an on-board camera. Our measurements of the system state, the distance to the wall and orientation of the wall, are derived from the optic flow. Whereas the structure of the environment is usually computed from the spatial derivatives of the optic flow, we use the structure contained in the temporal derivatives of the optic flow to compute the environment structure and hence the system state. The algorithm is used to control a 'remote brain' robot and results on the accuracy of the state estimates are presented.

1 Introduction

Autonomous mobile robots are being seen more often in various real-world applications, such as transportation, cleaning or surveillance tasks. The navigation task contains two problems: the localization task and the obstacle avoidance task. For this last task often a sensor-based, reactive controller is designed.

In this paper we present research on using vision for reactive behavior: based on the optic flow field induced by its motion, our mobile robot has to drive through the middle of the corridor. In order to perform this task, the sensing system has to provide information about the ego-motion of the system and the structure of the environment. The optic flow field provides such information: the ego-motion and relative depths can be derived directly from the flow vectors [3, 7]. To compute the spatial structure of the environment, such as surface normals, usually the spatial derivatives of the optic flow field are used. In this paper we show that the *temporal* derivative of the flow field can be used to provide a robust estimate of the environment structure. Our approach differs from other presented approaches on visual 'wall following' [5] in the sense that we are able to extract estimates of the orientation of the wall, and not only scaled distances, which makes it possible to make a more robust controller.

We will start with a general framework of optic flow and review how ego motion, relative depth and orientation can be derived from the optic flow.

*Dutch Foundation for Neural Networks

2 Optic Flow

First we describe the general framework of optic flow. The Coriolis equation of a moving (camera) coordinate system with translational velocity $\vec{T}$ and rotational velocity $\vec{\omega}$ results in motion of a point $\vec{P}$ in the environment with respect to this coordinate system:

$$\dot{\vec{P}} = -\vec{T} - \vec{\Omega} \times \vec{P},$$

with

$$\vec{P} = (P_x, P_y, P_z),$$
$$\vec{T} = (T_x, T_y, T_z),$$
$$\vec{\omega} = (\omega_x, \omega_y, \omega_z).$$

We use the pinhole model of the camera: the projection of a point $\vec{P}$ in the environment onto, here in homogeneous coordinates, the point $\vec{r}$ on the image plane $Z = 1$.

$$\vec{r} = \frac{\vec{P}}{P_z} \quad , \quad \vec{r} = (r_x, r_y, 1)$$

Projecting the motion of $\vec{P}$ onto the image plane results in the motion $\dot{\vec{r}}$ of the point $\vec{r}$ on the image plane. This is known as the image motion or confusingly as the optic flow:

$$\dot{r}_x = \frac{T_x - r_x T_z}{P_z} + \omega_x(r_x r_y) - \omega_y(1 + r_x^2) + r_y \omega_z$$

$$\dot{r}_y = \frac{T_y - r_y T_z}{P_z} + \omega_x(1 + r_y^2) - \omega_y(r_x r_y) - r_x \omega_z$$

The camera on the mobile robot is mounted such that it faces in the tangent of its path, i.e. facing forwards. This means that:

$$T_x = T_y = \omega_x = \omega_z = 0$$

Proc. IROS 97 0-7803-4119-8/97/$10©1997 IEEE

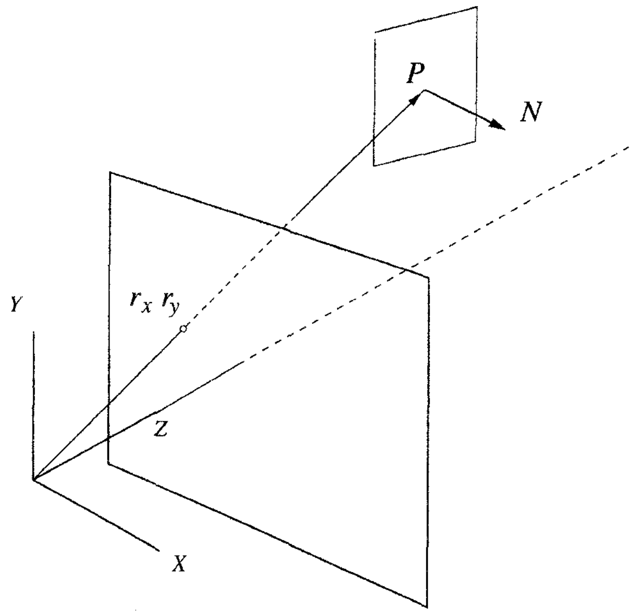

Figure 1: The camera coordinate system

such that the image motion is given by

$$\dot{r}_x = -\frac{r_x T_z}{P_z} + \Omega_x \quad , \quad \Omega_x = \omega_y(1 + r_x^2) \qquad (1)$$

$$\dot{r}_y = -\frac{r_y T_z}{P_z} + \Omega_y \quad , \quad \Omega_y = \omega_y(r_x r_y) \qquad (2)$$

Note that we cannot measure P_z because this parameter is always divided by the translational velocity, the velocity scaling property of the optic flow. We thus can only measure the relative depth P_z/T_z.

3 Ego-motion, depth and orientation from the flow

Rotation

To estimate the rotation we eliminate the depth T_z/P_z from the estimated image motion equations (1) and (2) and obtain [1]:

$$r_y \dot{r}_x - r_x \dot{r}_y - r_y \omega_y = 0$$

However, we must keep in mind that the optic flow vectors are estimated from an image sequence (see [2]), and that we only have noisy measurements $\underline{\dot{r}}_x$ and $\underline{\dot{r}}_y$. In a region R of the image where there is sufficient confidence in the flow-vectors ([2]) the flow is used to make a robust estimate of the camera rotation. A least mean square fit is used, giving:

$$\underline{\omega}_y = \frac{\sum_R \left(r_y^2 \underline{\dot{r}}_x - r_x r_y \underline{\dot{r}}_y\right)}{\sum_R \left(r_y^2\right)} \qquad (3)$$

Relative depth

We can now easily estimate the relative depth from the image motion vectors by subtracting the rotational components ($\underline{\Omega}_x$ and $\underline{\Omega}_y$) from the image motion. For each point (r_x, r_y) in the image plane (onto which the point P projects), equations 1 and 2 give estimates of the relative depth P_z/T_z (time to contact τ) of that point. For example, from the x-component of the flow:

$$\tau = \underline{\left(\frac{P_z}{T_z}\right)_x} = -\frac{r_x}{\underline{\dot{r}}_x - \underline{\Omega}_x}, \qquad (4)$$

It is clear that τ can also be computed from the y-component. In section 6 we discuss when to use the x and when to use the y component.

Orientation

Koenderink has analyzed the spatial variation $\nabla_r \dot{\vec{r}}$ of the image motion, concluding that this matrix can be decomposed in terms of differential invariants of the vector field [3]. Here we study the *temporal* variation of the image motion at a fixed location in the image:

$$\mathcal{H} \equiv \frac{\partial}{\partial t} \dot{\vec{r}}|_{\vec{r}=const}$$

From 1 and 2 we find:

$$\mathcal{H} = \begin{pmatrix} \dot{r}_x - \Omega_x \\ \dot{r}_y - \Omega_y \end{pmatrix} \frac{\dot{P}_z}{P_z} \qquad (5)$$

The relative change of depth $\frac{\dot{P}_z}{P_z}$ is a function of the orientation of the surface under consideration. Suppose that the surface is characterized by a surface normal $\vec{N}(t)$ and a perpendicular distance to the center of the camera coordinate system $d(t)$. Both are a function of time since the robot is moving. Since

$$P_z = \frac{d(t)}{\vec{r} \cdot \vec{N}(t)}$$

we can write

$$\frac{\dot{P}_z}{P_z} = \frac{\dot{d}}{P_z \vec{r} \cdot \vec{N}} - \frac{\vec{r} \cdot \dot{\vec{N}}}{\vec{r} \cdot \vec{N}}.$$

The change in N and d can be expressed in the motion parameters: $\dot{d} = \vec{N} \cdot \vec{T}$ and $\dot{\vec{N}} = -\vec{\omega} \times \vec{N}$, resulting in

$$\frac{\dot{P}_z}{P_z} = \frac{\vec{N} \cdot \vec{T}}{P_z \vec{r} \cdot \vec{N}} + \frac{\vec{r} \cdot \vec{\omega} \times \vec{N}}{\vec{r} \cdot \vec{N}}. \qquad (6)$$

The relation between the temporal derivative and the orientation can now be written as:

$$\mathcal{H} = \begin{pmatrix} \dot{r}_x - \Omega_x \\ \dot{r}_y - \Omega_y \end{pmatrix} \frac{(\vec{T}/P_z + \vec{r} \times \vec{\omega}) \cdot \vec{N}}{\vec{r} \cdot \vec{N}}. \qquad (7)$$

Since $\mathcal{H}, \dot{r}_x, \Omega_x, -\vec{T}/P_z$ are measurable, we can derive the surface normal. However, counting the unknowns reveals that the orientation of the viewed surface can only be determined in the direction of the tangent of the observer. We will show that for the restrictions in our application the orientation of the robot can be recovered.

4 Wall following

The camera is mounted rigidly on the vehicle, with the camera axis in the driving direction. A summary of the system variables, with reference to figure 2, is given below:

d : distance of the wall to center of camera
ϕ : orientation of camera w.r.t. the wall normal
ν : translational velocity of the camera
$\dot{\phi}$: rotational velocity of the camera

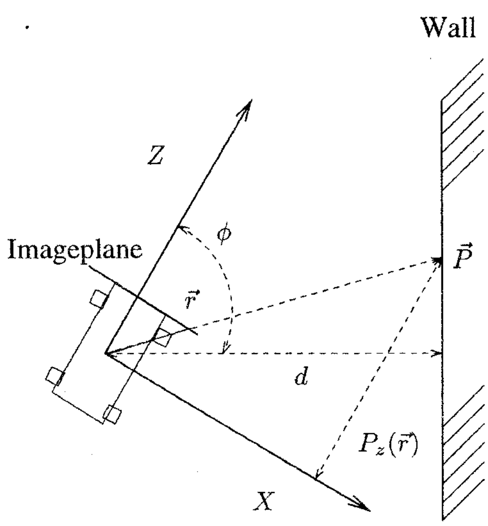

Figure 2: Top view of the sensor coordinate system

The goal of the controller is to align the camera parallel to the wall and at a certain distance from the wall. However, as is well known when using optic flow, distances can only be measured scaled with the translational velocity of the camera. This means that we have the following set-points (we denote set-points of the controller with a superscript ^{s}):

$$\left(\frac{d}{\nu}\right)^s = c = \text{constant} \quad , \quad \phi^s = \frac{\pi}{2}$$

In this paper we only control the steering angle of the vehicle. The input to the system is $\dot{\phi}$. We use a simple proportional controller:

$$\dot{\phi} = K_\phi \left[\phi - \phi^s\right] + K_d \left[\frac{d}{\nu} - \left(\frac{d}{\nu}\right)^s\right]$$

In the case of a corridor follower, the constant (c) need not be specified beforehand but follows from the condition that the distance from the camera to the left wall and right wall should be equal. Let the distance to the left wall be d_l and to the right wall be d_r. We then use the following controller:

$$\dot{\phi} = K_\phi \left[\phi - \phi^s\right] + K_d \left[\frac{d_l}{\nu} - \frac{d_r}{\nu}\right]$$

In order to control the vehicle, we need to estimate ϕ and d/ν from the optic flow. In the following section we will show how that can be done and how ϕ and d/ν relate to measurable quantities from the optic flow.

5 Relating the state to the observable features from the flow

In section 3 we described how to estimate the rotation ω_y of the camera, the relative depth τ along a viewing direction and derived an expression relating the surface normal and temporal derivative of the optic flow. In this section we will describe how these observables from the flow relate to the state d/ν and ϕ of the mobile robot. Refer to figure 2

Because the camera is fixed to the car, we know that the translational velocity ν is identical to T_z. The walls are vertical, yielding $N_x = -\sin(\phi), N_y = 0, N_z = -\cos(\phi)$ In this notation it can be show that

$$\vec{r} \cdot \vec{N} = -r_x \sin(\phi) - \cos(\phi)$$

$$\mathcal{H} = \begin{pmatrix} \dot{r}_x - \Omega_x \\ \dot{r}_y - \Omega_y \end{pmatrix} \left(\frac{-\frac{(\dot{r}_x - \Omega_x)}{r_x}\cos(\phi) + \dot{\phi}(r_x\cos(\phi) - \sin(\phi))}{r_x \sin(\phi) + \cos(\phi)}\right)$$

From this we can solve ϕ:

$$\phi = -\arctan\left(\frac{\mathcal{H}_x r_x + (\dot{r}_x - \Omega_x)^2 - \dot{\phi}(\dot{r}_x - \Omega_x) r_x^2}{r_x(\mathcal{H}_x r_x + \dot{\phi}(\dot{r}_x - \Omega_x)}\right) \tag{8}$$

Once we have the orientation of the vehicle, we can use this the relative distance to the wall:

$$\left(\frac{d}{\nu}\right) = \tau r_x \sin(\phi) + \cos(\phi) \tag{9}$$

As we see, the state variables of the system, d/ν and ϕ, are completely observable from the image motion. However, the estimation of the image motion from a sequence of images is difficult, in the sense that the resulting time to contact τ is very noisy. A more robust state estimation is necessary.

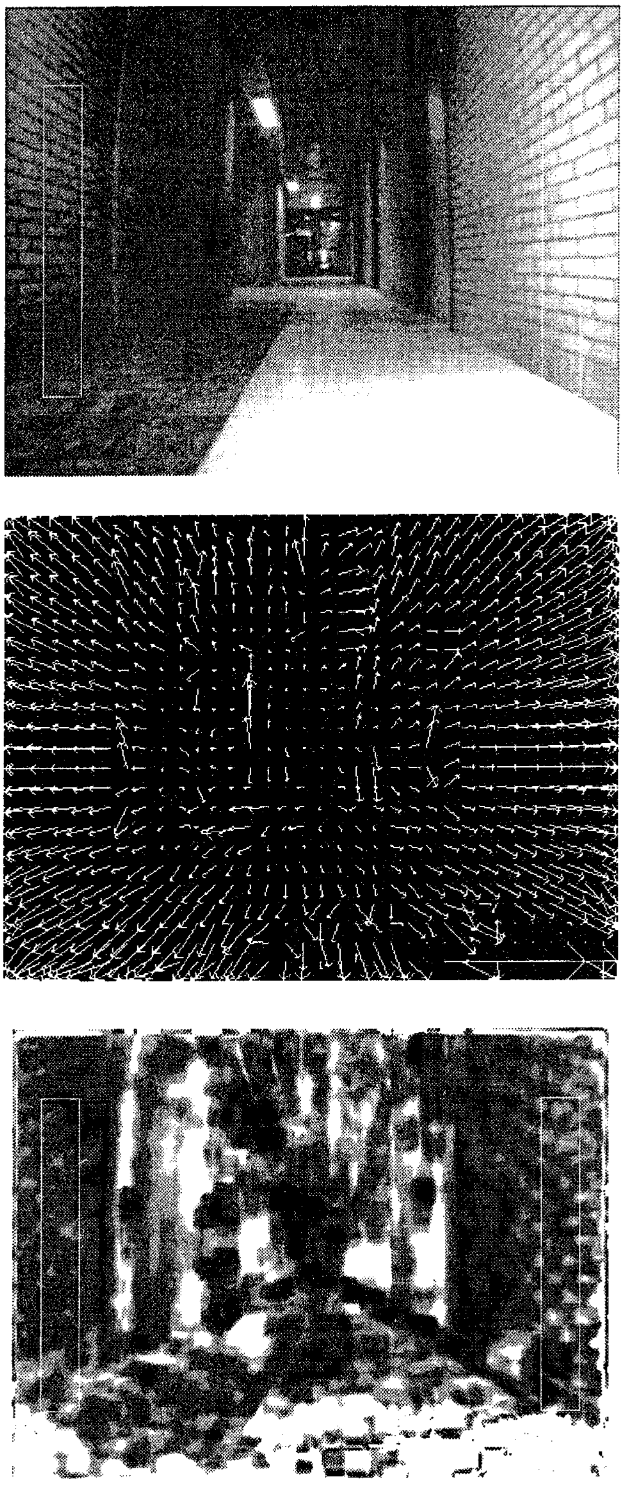

Figure 3: Upper figure: the camera image and the regions in which the optic flow is used. Middle: the optic flow field. Note that in some area's many erroneous estimates are made. Lower: the confidence measure. Darker regions correspond to lower variance in the image motion estimates. In the calculation of $\underline{\tau}$ the τ's in the regions L and R are weighed with this confidence value.

6 Estimating the state

From a parameter estimation point of view we wish to suppress the noise in the image motion as much as possible, such that the estimates of the state are less sensitive to this noise. So at what location in the image should or can we measure the image motion such that the states are more reliable?

For example, a sensitivity analysis of the error in τ towards noise in the image motion shows that:

$$\tau = -\frac{r_x}{\dot{r_x} - \Omega_x} \rightarrow \frac{\delta\tau}{\delta\dot{r_x}} = \frac{\tau^2}{r_x} \rightarrow \sigma_\tau^2 = \sigma_{\dot{r}_x}^2 \frac{\tau^4}{r_x^2}$$

It should be directly clear the r_x should be large to damp the noise if we measure the τ from $\dot{r}_x$ (and similar for r_y). The relation of the sensitivity of the noise in the τ and the noise in the image motion is given by a parabola function of the location in the image plane. This off course means that corners in the image are the best locations to measure the τ, second best are the borders of the image plane.

In order to obtain a robust estimate of the states we take the average at some locations in the image. As pointed out above, these regions should be in the corners or at the borders of the image. Now we use some prior knowledge in the sense that the walls in the corridor are vertical: we take two thin vertical patches, left and right, as shown in figure 3.

We now determine $\underline{\phi}$ and $\underline{\left(\frac{d}{v}\right)}$ from the equations 8 and 9, using the the average $\dot{r}_x$ and $\mathcal{H}_x$ in the windows.

These measurements are the input to the P controller.

7 Experiments

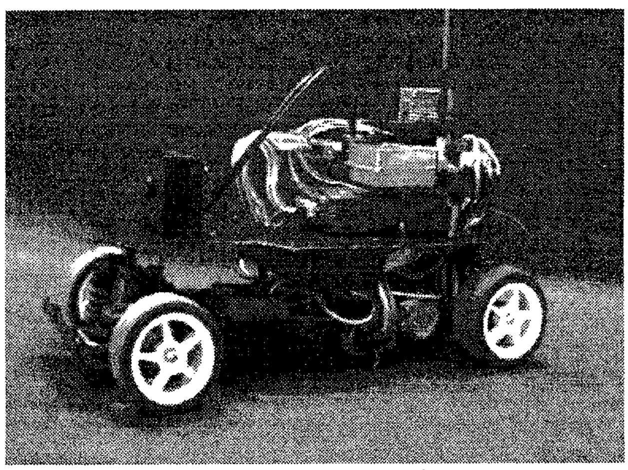

Figure 4: The robot vehicle

The robot vehicle is a customized small radiographic controlled cart (Figure 4), of which we removed the receiver. A simple 8 bit μprocessor on the cart takes care

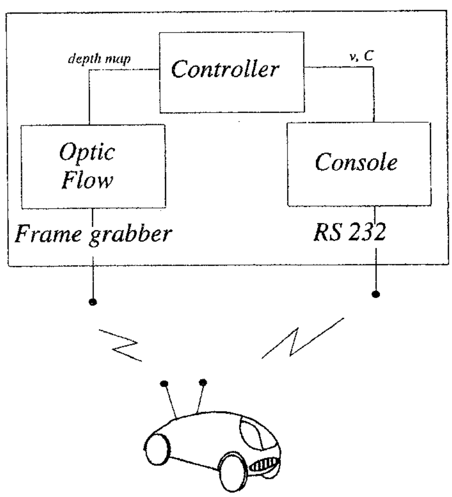

Figure 5: The set-up of the system

of the servo motors for propulsion and steering, and takes care of the emergency stops and watchdog. The processor communicates via a wireless RS232 connection with a host computer, which is a Pentium-based PC. A process running on the PC sends the desired speed and curvature to the vehicle.

A small CCD camera is mounted on top of the vehicle. The video-signal is sent via a UHF-link to a receiver, which is connected to the frame-grabber mounted in the PC system. A block diagram is given in Figure 5. Images can be grabbed and stored in memory continuously. From a sequence of images the motion field is computed with the method reported in [2].

A description of the dynamical behavior of the system, the controller and some experiments in simulation are described in [4].

Here we present experimental results on the accuracy of the estimation of the state of the system. In order to have a calibrated image sequence, in which the path of the camera was known, we used a robot arm with a camera mounted in the end effector. This sequence is made up of a circular trajectory towards a wall. The camera is mounted such that only one joint is used to generate the trajectory. At each frame of the sequence we know the position and velocity of the camera and wall from the sensor which measures the state of the robot, and hence are able to compute the variables $d(t)/\nu$, $\phi(t)$ and $\dot{\phi}$.

Using the described method we will now give some preliminary results. In figure 6 we have plotted the real values of $\tau(t)$, $d/\nu(t)$ and $\phi(t)$ as computed from our calibrated motion of the robot. The results of the measurements $\underline{\tau}(t)$, $\underline{d}(t)/\nu$ and $\underline{\phi}(t)$ are shown in figure 7. [1] It can directly be seen that there is a systematic error in the measurements. This is due to the fact that the camera is poorly calibrated. The camera parameters may be as far of as 20% from the real values. The noise in the estimation of the $\underline{\tau}$ propagates to the estimate of the $\dot{\underline{\tau}}$. This directly influences the state variables, but we expect that the low pass structure of the mobile robot system will damp this noise.

8 Conclusions

We presented a control system for a vision guided mobile vehicle, which has as task to drive through the middle of a corridor with constant speed and parallel to the walls. Our approach is different from other approaches in visual robot guidance in that it does not use the spatial derivatives of the optic flow field to estimate surface slants, but uses the *temporal* derivative of the optic flow field. We implemented the algorithms on a standard PC without any special hardware, and achieved real-time performance. Robot navigation based on optic flow is now possible using standard components. Experiments with a calibrated camera motion showed the accuracy of the methods.

References

[1] A.R. Bruss and B.K. Horn. Passive navigation. Computer Graphics and Image Processing, 21, pp3-20, 1983.

[2] A. Dev, B.J.A. Kröse and F.C.A. Groen. Confidence measures for image motion estimation. RWCP symposium, Tokyo, 1997.

[3] J.J.Koenderink. Optic Flow. Vision Research, 26(1), 161-180, 1986.

[4] B.J.A. Kröse, A. Dev, X. Benavent and F.C.A. Groen. Vehicle navigation on optic flow. RWCP symposium, Tokyo, 1997.

[5] J. Santos-Victor, G. Sandini, F. Curotto and S. Garibaldi. Divergent Stereo in Autonomous Navigation: From Bees to Robots. Int. Journal of Computer Vision, 14, 159-177 (1995).

[6] H.C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. Nature, 293 (10), pp 133-135, 1981.

[7] S. Negahdaripour, S. Lee. Motion Recovery from Image Sequences Using Only First Order Optic Flow Information. International Journal of Computer Vision, 9(3), 163-184, 1992.

[1] The "glitches" in the estimate of $\underline{\tau}(t)$, and hence in the state variables, are due to the erroneous reversal of odd and even frames in our frame grabber.

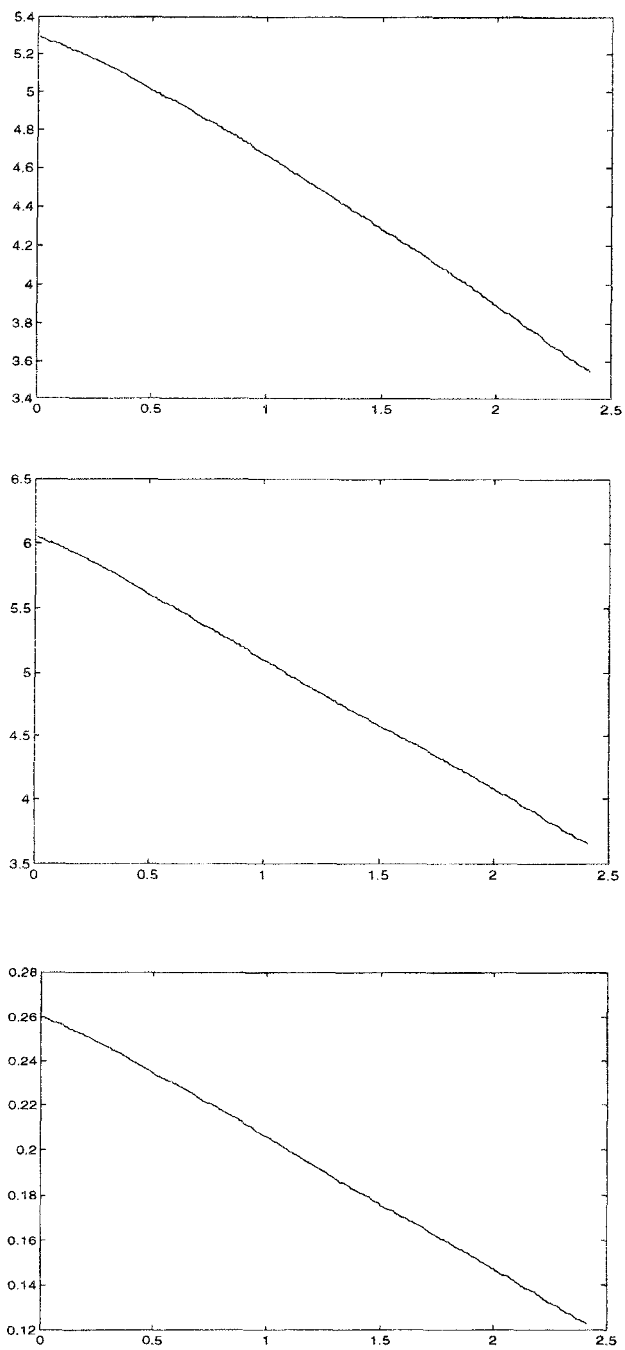

Figure 6: The 'real' values, against the time in seconds on the horizontal axis, of (from top to bottom): $\tau(t)$ in seconds, $d(t)/\nu$ in seconds and $\phi(t)$ in rad.

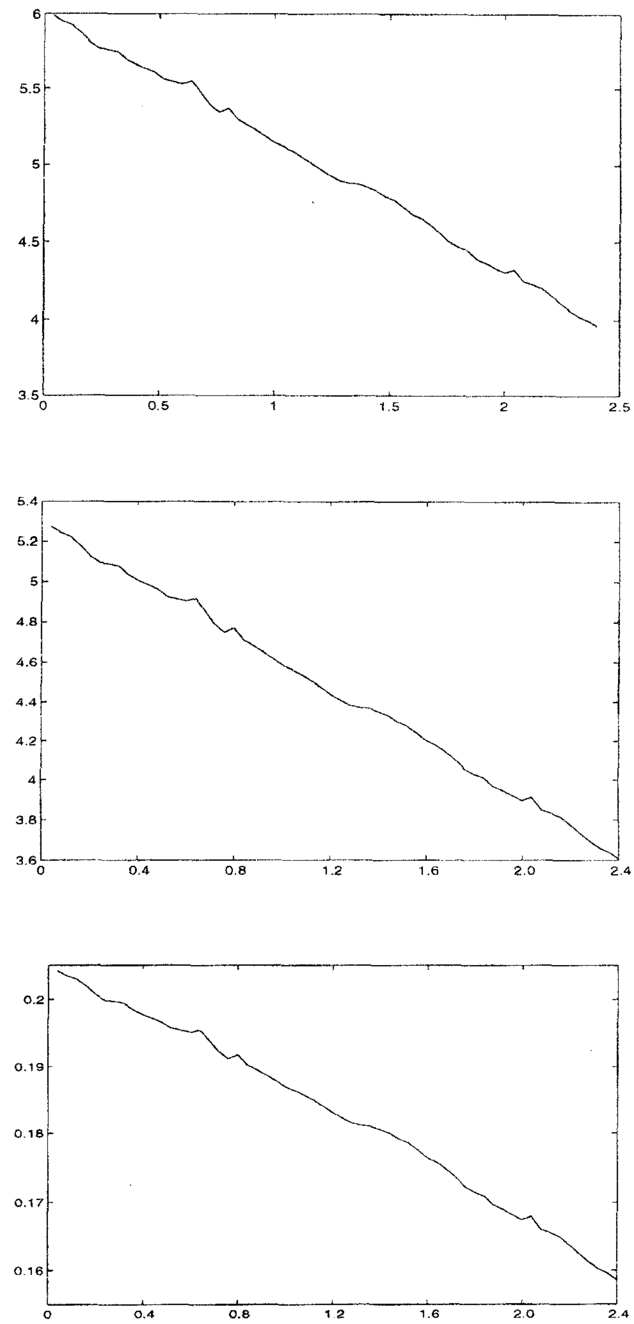

Figure 7: The estimated values, against the time in seconds on the horizontal axis, of (from top to bottom): $\underline{\tau(t)}$ in seconds, $\underline{d}(t)/\nu$ in seconds and $\underline{\phi}(t)$ in rad.