Compensating Intermittent Delayed Visual Feedback in Robot Navigation

K. L. Koay and G. Bugmann
School of Computing, Communications and Electronics
University of Plymouth, Plymouth PL4 8AA, UK
K.L.Koay@herts.ac.uk, gbugmann@plymouth.ac.uk

Abstract

When a mobile robot uses vision for self-localization and obstacle avoidance, the results of image processing become available after a variable computational delay. If the robot keeps moving during image processing, results become available after the robot has left the position where the image was taken. This is the problem of control in presence of intermittent delayed feedback. This paper proposes a solution in the form of a modification of the Smith Predictor. Several previously proposed methods, in particular those using retro-active updating, are reframed here in the context of the Smith Predictor. Preliminary navigation results are shown.

1. Introduction

Delayed visual measurements used to cause robots to exhibit a stop-and-go motion (Moravec, 1983; Kosaka and Kak, 1992). When a mobile robot relied on vision for its navigation process, it had to wait for the results of image processing and planning to become available before the navigation process could resume. Such a stop-and-go motion is not a practical solution in many applications, such as autonomous wheelchairs or cars.

A method to achieve continuous motion was proposed at CMU (Goto et al., 1988) in the form of a "driving pipeline". In this approach, the space ahead of the robot is divided into a series of "driving units". The robot calculates a path in a driving unit a few steps ahead, while executing the path defined for the current unit. In this work, the time delay problem is minimized by reducing the speed of the vehicle. It is generally assumed that the increased speed of modern computers will render the time delay problem negligible.

However, more powerful computers induce the implementation of more intelligent and complex algorithms (Bak et al., 1998) which take more time. Further, not all sources of delays are of computational nature, such as inter-process communication delays or the sampling time in the camera. Thus, the time delay problem is likely to stay and needs to be addressed in a principled way.

A series of papers have proposed a solution in the form of "retroactive updating" (Kosaka, Meng and Kak, 1993; Maeyama, Ohya and Yuta, 1995). In this approach, delayed visual measurements are used to update the belief of the robot about its position at the time the picture was taken. The estimate of the current position is then obtained by integrating odometric measurements from that time.

A standard solution to the problem of control with delayed feedback is the Smith Predictor (Smith, 1959). However, it was not designed for handling intermittent feedback with variable delays. In this paper, we will show that, with a few modifications, the use of the Smith Predictor can be extended to these cases. We will also show that the concept of retroactive updating was already present in the design of the Smith Predictor.

2. The Problem

In order to control a robot's motion along a path, its motion control system needs to be provided with the robot's current position $P(t)$ and its current goal position $P_G(t)$. The robot's current position can be estimated from the robot's shaft encoders feedback while its current goal position can be obtained through path planning.

Due to the unreliability of the shaft encoder's feedback, visual feedback is often used to improved the accuracy of the estimated robot's position $\hat{P}(t)$.

The problem with visual feedback is the computational time delay, as visual feedback usually involves several time-consuming processes. Therefore, the robot's position $P(t)$ obtained through visual feedback only becomes available at time $t+n\tau$, where τ is the controller cycle time. If the robot moves while processing visual information for self-localization, $P(t)$ determined from vision is no longer the robot's current position but its past.

The aim here is to obtain a better estimate $\hat{P}'(t)$ of the position at time t based on the delayed visual feedback $P(t-n\tau)$ and the position obtained from shaft encoders measurements $\hat{P}(t)$, which we will consider to be an estimate due to the unreliability of shaft encoders. We consider visual measurements to be errorless.

3. Smith Predictor

The Smith Predictor is well known as an effective Dead-time Compensator (DTC) for a stable process with a large dead-time (Smith, 1959). The classical configuration of a Smith Predictor is shows in Figure 1. The presence of a large dead-time (e.g. $n\tau$) in the process $P(s)$ causes the feedback of $y(t)$ to be delayed and forces conventional controllers to operate with a low gain. The Smith Predictor improves the closed-loop performance by introducing a minor feedback loop around the primary controller to produce $v(t)$, which is an estimation of the variation of $y(t)$ during the last n units of time. This variation $v(t)$ added to the delayed measurement constitutes an estimate of the current value of y, which will become available as the later measurement $y(t+n\tau)$. This is subtracted form the requested value r to produce the error e' that is fed into the controller. This eliminates the sluggish responses or over-correction associate with conventional controllers (Levine, 1996).

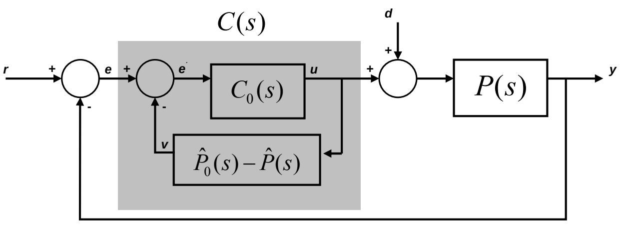


Figure 1. A Classical diagram of a control system incorporating a Smith Predictor. r in the figure representing the reference signal, $P(s)$ is the transfer function of the process with large dead-time, $\hat{P}(s)$ and $\hat{P}_0(s)$ are the process models with and without dead time respectively. The shaded area $C(s)$ is the Smith Predictor or Dead-time Compensators (DTC).

The Smith Predictor was developed for dealing with dead-time problems common to industrial process where feedback from the processes is continuous, i.e. in each control cycle a new delayed feedback is available. However, with visual measurements, the feedback is intermittent. This is either due to the long processing time of each captured frame, or simply due to the fact that images are captured at a given frame rate. Further, the delay is variable, depending on the complexity of a given image. Therefore, the original Smith Predictor is not suited for applications with visual feedback. In addition, it can not handle corrections of the estimated orientation that are required in navigation applications (this will become more clear in the discussion). The modifications proposed in the next section show how these limitations can be removed.

4. Modified Smith Predictor

It is useful to re-examine how the Smith Predictor operates. The error e' that drives the controller can be rewritten as follows:

$$e' = r - y(t) - (\hat{P}_0(s) - \hat{P}(s)) \quad (1)$$

$$e' = r - (y(t) - \hat{P}(s)) - \hat{P}_0(s) \quad (2)$$

Where r is the desired output of the plant, $y(t)$ is the delayed measurement of the output, $\hat{P}_0(s)$ is an estimate of the current output of the system, and $\hat{P}(s)$ is the value of the estimate made n time steps earlier. $\hat{P}_0(s)$ is usually calculated using a model of the system's dynamics and the history of past control actions. Equation 2 shows that, in the conventional Smith Predictor, a delayed copy $\hat{P}(s)$ of the model's output $\hat{P}_0(s)$ is in effect compared in each control cycle (time step) with the actual delayed measurement $y(t)$. The difference between represents the error made by the model up to time $(t-n\tau)$. Thus the classical Smith Predictor performs a form of retroactive updating where the error made up to n time steps in the past is corrected at each time step.

In the case of an intermittent feedback, the delayed measurement is available only once very n time steps and model errors can only be corrected every n time steps. One solution is to store the correction factor and use it unchanged until the next measurement becomes available. This is the basis for the modifications of the Smith Predictor (figure 2). Retroactive updating is here of an intermittent nature, very similar to that proposed by other authors (see discussion).

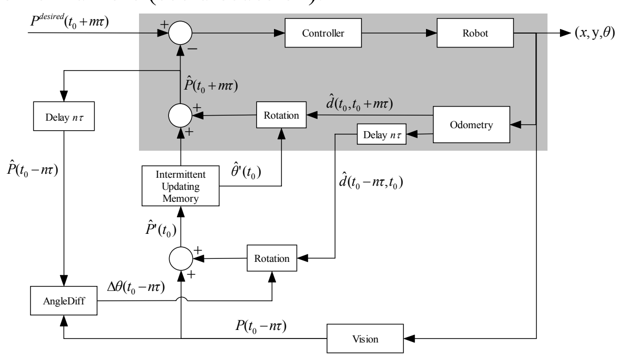


Figure 2. The modified Smith Predictor for navigation application with intermittent feedback. The area shaded in grey is the "fast loop" that is updated at every control cycle. The components outside the shaded area are updated with a lower frequency, every time that vision provides new self-localization data. Time t_0 is the time at which visual feedback becomes available, with information pertinent to time $t_0 - n\tau$.

In our mobile robot application, the estimation of the current position (at time $t_0 + m\tau$, $m \leq n$) is provided by odometry (rather than by an internal model), with the "last best estimate" $\hat{P}'(t_0)$ as a starting point.

$$\hat{P}(t_0 + m\tau) = R(\hat{\theta}'(t_0))\hat{d}(t_0, t_0 + m\tau) + \hat{P}'(t_0) \quad (3)$$

The odometric displacement $\hat{d}(t_0, t_0 + m\tau)$ is calculated in a reference frame with the same orientation as the last best estimate $\hat{P}'(t_0)$. $\hat{P}'(t_0)$ is calculated from the actual position at time $t_0 - n\tau$, which is known at time t_0, updated by odometric measurements between $t_0 - n\tau$ and t_0.

$$\hat{P}'(t_0) = R(\Delta\theta(t_0 - n\tau))\hat{d}(t_0 - n\tau, t_0) + P(t_0 - n\tau) \qquad (4)$$

Note that the value of $\hat{P}'(t_0)$ is obtained by rotating the displacement vector $\hat{d}$ before adding it to the measured position $P(t_0 - n\tau)$ (see figure 3). *R*() is the rotation matrix and the angle of rotation $\Delta\theta(t_0 - n\tau)$ is the angular difference between the estimated and actual orientation of the robot at time $t_0 - n\tau$.

Equations 4 and 3 show that the estimated position is affected by odometric errors occurring over a time span of at least $n\tau$ and at most 2 $n\tau$. Without intermittency, this is always $n\tau$ exactly. Note that *n* does not need to be a constant, it can vary from image to image.

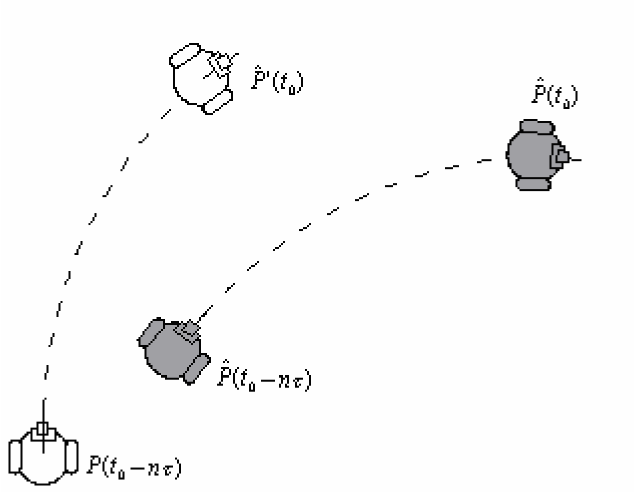


Figure 3. The shaded robots indicate their estimated positions and the white robots indicate actual positions. At time t0, the visual feedback informs the robot of its correct position at time t0-nt. By replacing the erroneous belief $\hat{P}(t - n\tau)$ with the measured value $P(t - n\tau)$, one obtains a better estimate $\hat{P}'(t_0)$ of the position at time t_0.

5. Robot Experiment

5.1. The Vision-Based Navigation System

The modified Smith Predictor was used with a vision-based navigation system consisting of a robot and a remote computer system which functions as the robot's remote brain. The task of the robot is to perform vision-based navigation toward the goal while avoiding any detected obstacles.

The remote brain processed images sent wirelessly for self-localizing the robot based on landmarks (to determine the pose $P(t) = (x, y, \theta)$ and for detecting obstacles. The remote brain also performed path planning to determine a collision free path to the goal. This path was communicated wirelessly to the robot in the form of a series of waypoints implementing a receding horizon control strategy. Note that the path starts at the position where the robot was when the image was captured. When the robot receives the path specifications, it has already travelled parts of it, and heads towards the next waypoint along the path (figure 4). More details on planning and the use of waypoints can be found in (Koay, 2003).

We have initially attempted to estimate the robot's displacement $\hat{d}$ using a dynamic model of the robot, but actual displacement depended strongly on the discharging battery voltage, so we used odometry instead.

The modified Smith Predictor is implemented on-board the robot, where odometric information is continuously available. Retroactive updating takes place each time the remote brain sends the result of self-localization. For this to work, the robot needs to know when the image was taken. In that case, it is well approximated by the time of arrival of the previous data, as the remote brain captures a new image as soon as data have been sent.

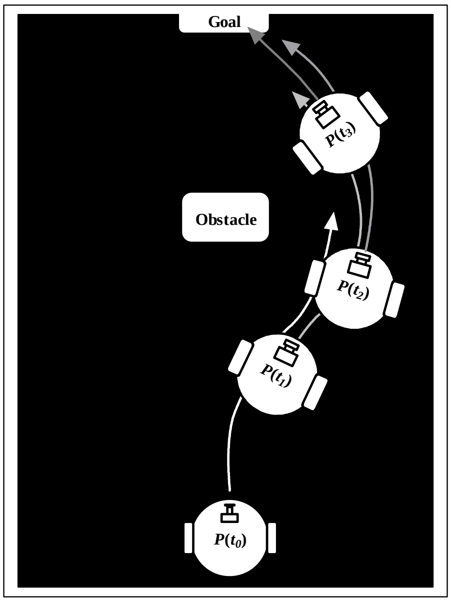


Figure 4. The concept of receding horizon control strategy. Using the Receding Horizon Strategy, the robot's remote brain is regularly searching for new paths toward the goal based on the robot latest coordinate (obtained through vision). Each line in figure here represents an obstacle free path $P(t_x)$ based on the image captured at time t_x.

5.2. Experimental Setup

The aim of the experiment was to investigate and demonstrate a control technique that addresses concurrent image processing and planning while the robot is in motion. To achieve this aim, the following simplifications were used:

The goal is at a predefined location that can be varied by the experiment, but does not requires competences in visual object localization.

Obstacles are simple white rectangular blocks, each with the height of 1.5 centimetre, small enough for 2-D approximation, that can be detected visually by the robot's onboard camera as 2-D forbidden areas standing out from the dark floor of the environment. These obstacles can be placed anywhere.

The environment is a small-scale box of size 125×89 centimetres developed to emulate a room in the real world. The walls and the floor of the small-scale robot's environment were painted with white colour and black colour respectively. This setup simplifies image processing and frees time for exploring other issues related to the overall aim of this research.

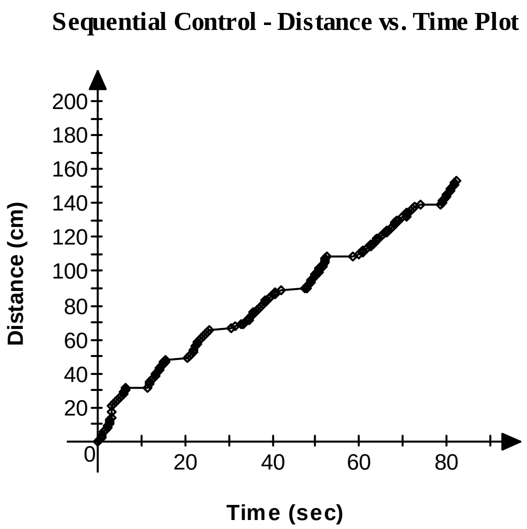
A robot with a circular cross-section robot and two drive wheels (which enable the robot to spin around a centre point) is used as a prototype of a domestic robot. This cylindrical robot is free from both the geometric constraints and the piano-mover's problem (Schwartz and Sharir, 1983), therefore the 3-D planning problem is simplified to a 2-D planning problem.

An overhead camera is mounted above the working area for recording the robot's trajectories.

5.3. Result

The results are collected from one travel from the start to the goal, similar to the one suggested in figure 4, to demonstrate that the proposed method is effective. Figure 5 shows two examples of distances-versus-time plots for the system that uses sequential control (stop-and-go) and the system that uses concurrent control (modified Smith Predictor).

Figure 5(a) shows the of stop-and-go motion effect in the system that uses the sequential control method. This system takes about 20 seconds longer to complete the navigation task compared to the system that uses concurrent control shown in figure 5(b). Figure 5(b) shows that the system exhibits continuous motion.



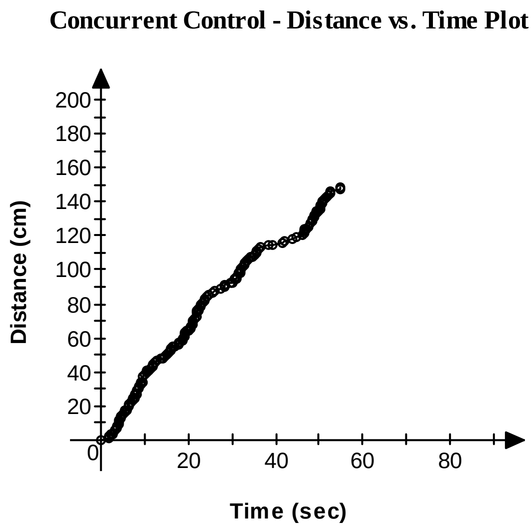


Figure 5. The Distance vs. Time plot of two systems that use different control methods. (a) This figure shows the stop-and-go motion of the system that uses sequential control, while (b) shows that the system that uses concurrent control does not exhibits continuous motion and completes the task 20 seconds quicker than (a). Variations in speed in (b) are dues to a programmed slower motion in curves and faster in straight lines.

6. Discussion

Solving the problem of delayed measurement by retroactive updating was initially proposed by Kosaka, Meng and Kak (1993) although one could argue, as done later in this section that the concept was already present in the design of the Smith Predictor (Smith, 1959).

Kosaka, Meng and Kak (1993) wanted to solve the stop-and-go motion problem by integrating visual information that was $n\tau$ time steps old into the tracking system. For that purpose, they stored a history of all commands (or shaft encoder readings) from the image capture time $t_0 - n\tau$ to the time t_0 when the delayed measurement becomes available. They also stored the measured position $\hat{P}(t_0 - n\tau)$ at time t_0–$n\tau$. When the delayed measurement $P(t_0 - n\tau)$ becomes available, the new estimation of the current position $\hat{P}'(t_0)$ is produced by recalculating the total displacement vector $\hat{d}(t_0 - n\tau, t_0)$ from past commands, then rotates the displacement vector by the error $\Delta\theta$ between the measured heading $P(t_0 - n\tau)$ and $\hat{P}(t_0 - n\tau)$, and add it to the new measurement for time $P(t_0 - n\tau)$:

$$\hat{P}'(t_0) = R(\Delta\theta)\hat{d}(t_0 - n\tau, t_0) + P(t_0 - n\tau) \qquad (5)$$

where R is the rotation matrix.

The requirement to store the history of commands in Kosaka, Meng and Kak (1993) was due to the incremental method used to calculate the position uncertainty. As noted in Maeyama, Ohya and Yuta

(1995), for re-estimating the position only, the total displacement is sufficient.

In Maeyama, Ohya and Yuta (1995) a new method is proposed to re-estimate the uncertainty without using the history of commands. This problem is not present in this work, as images are acquired at the maximum possible rate, thus there is no advantage in having access to uncertainty information to decide when to recalibrate, as done in Kosaka, Meng and Kak (1993) and Maeyama, Ohya and Yuta (1995).

The method for recalibration of the position used in Maeyama, Ohya and Yuta (1995) differs from that in Kosaka, Meng and Kak (1993) in that in the former a more complex data fusion process is used for generating the new position $\hat{P}'(t_0)$. This consists of a maximum likelihood estimation including all measurement available at time t_0-$n\tau$. Otherwise the principle is the same as in Kosaka, Meng and Kak (1993) where the total displacement since t_0 is estimated from odometric measurements.

In a more recent work, Larsen, Andersen and Ravn (1998) are concerned with how to set Kalman Filter parameters given that parts of the measurements are delayed. The proposed solution is to extrapolate the delayed measurement $P(t_0 - n\tau)$ to the current time by adding to it the displacement $\Delta\hat{P}(t_0 - n\tau, t_0)$ as determined from all other sensors

$$\hat{P}^{extrapolate}(t_0) = P(t_0 - n\tau) + \Delta\hat{P}(t_0 - n\tau, t_0) \qquad (6)$$

This extrapolated data is then fused with other measurements available at time t_0 to produce the best estimation of the position at time t_0.

The essential difference with the method proposed by Maeyama, Ohya and Yuta (1995) is that data fusion takes place here at time t_0 rather than at time t_0-$n\tau$.

Very similar principles are used in the design of the original Smith Predictor. There, the delayed measurement $P(t_0 - n\tau)$ is available at each time step, hence the recalibration takes place at every time step.

$$\hat{P}'(t_0) = R(\Delta\theta(t_0 - n\tau))\hat{d}(t_0 - n\tau, t_0) + P(t_0 - n\tau) \qquad (7)$$

where

$$\hat{d}(t_0 - n\tau, t_0) = \hat{P}(t_0) - \hat{P}(t_0 - n\tau) \qquad (8)$$

Note that this method requires updating at each time step a list of past position vectors $\hat{P}(i) = (\hat{x}(i), \hat{y}(i), \hat{\theta}(i))$, where i = t_0-$n\tau$,…,t_0. This is required to calculate the orientation error $\Delta\theta$ used to rotate the displacement vector (alternatively, one could keep in memory the list of displacements in every time step for the time span from t_0-$n\tau$ to t_0).

$$\Delta\theta(t_0 - n\tau) = AngleDiff(\hat{\theta}(t_0 - n\tau), \theta(t_0 - n\tau)) \qquad (9)$$

Note that the need to compare $\hat{P}(t_0 - n\tau)$ and $P(t_0 - n\tau)$ within the fast loop is usually not mentioned in standard descriptions of the Smith Predictor which are not concerned with navigation applications. Therefore the modified diagram of the Smith Predictor (figure 2) shows in more details the pieces of information needed for its operation in the case of a navigation application.

When the feedback is intermittent, a further modification is needed. This is because $P(t_0 - n\tau)$ is only available every $n\tau$ steps and consequently also $\Delta\theta$.

In the proposed additional modification of the Smith Predictor (figure 2) only one value of the displacement between t_0 - $n\tau$ and t_0 needs to be updated and only the previous estimate $\hat{P}(t_0 - n\tau)$ needs to be stored. The best estimate $\hat{P}'(t_0)$ is then kept in memory for use at every time step, and is changed only when new measurements become available (Figure 2). This model therefore copes with variable delays and maximizes localization accuracy for a given computing power, by processing images as fast as possible.

7. Conclusion

This paper proposed a modification of the classical Smith Predictor to enable the control of systems with intermittent delayed feedback. It showed that the Smith Predictor performs retroactive updating, and that intermittent retroactive updating in the Smith Predictor framework is conceptually equivalent to other methods proposed in different frameworks (Kosaka, Meng and Kak, 1993; Maeyama, Ohya and Yuta, 1995; and Larsen, Andersen and Ravn, 1998). Basically, all methods produce an estimate of the current position by adding the estimated displacement to the delayed measurement.

References

Bak, M., Larsen, T. D., Norgaard, M., Andersen, N. A., Poulsen, N. K. and Ravn, O. (1998). Location Estimation using Delayed Measurements. 5th International Workshop on Advanced Motion Control 98, Coimbra, Portugal, Pages 180-185. (Available at http://www.iau.dtu.dk/~tdl/amc98paper.ps.gz)

Goto Y., Schafer S.A., and Stentz A. (1988) The Driving Pipeline: A Driving Control Scheme for Mobile Robots. Tech. report CMU-RI-TR-88-08, Robotics Institute, Carnegie Mellon University, June, 1988.

Kosaka, A. and Kak, A. C. (1992). Fast Vision-Guided Mobile Robot Navigation Using Model-Based Reasoning and Prediction of Uncertainties. *Computer Vision, Graphics, and Image Processing – Image Understanding*, 56(3):271-329.

Koay K.L (2003) Intelligent vision-based navigation system. Thesis. University of Plymouth, United Kingdom.

Kosaka, A., Meng, M. and Kak, A. C. (1993). Vision-Guided Mobile Robot Navigation Using Retroactive Updating of Position Uncertainty. *Proceeding of IEEE International Conference on Robotics and Automation*. 2:1-7

Larsen, T. D., Andersen, N. A. and Ravn, O. (1998). Incorporation of Time Delayed Measurements in a Discrete-time Kalman Filter. *In To appear in proceedings for CDC '98*, Tampa, Florida.

Levine, W. S. (1996). *The Control Handbook*. CRC Press, ISBN: 0849385709.

Maeyama, S., Ohya, A. and Yuta, S. (1995). Non-stop outdoor navigation of a mobile robot---Retroactive positioning data fusion with a time consuming sensor system---. *In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Pages 130-135.

Moravec, H. P. (1983). The Stanford Card and the CMU Rover. *Proceedings of the IEEE*, 71(7):872-884.

Ohya, A., Kosaka, A. and Kak, A. (1998). Vision-Based Navigation by a Mobile Robot with Obstacle Avoidance using Single-Camera Vision and Ultrasonic Sensing. *IEEE Transactions on Robotics and Automation*, 14(6):969-978.

Schwartz, J. T. and Sharir, M. (1983). On the Piano Movers' Problem III: Coordinating the Motion of Several Independent Bodies: The Special Case of Circular Bodies Moving Amidst Polygonal Barriers. *Robotics Research*, 2(3): 46-75.

Smith, O. J. (1959). A Controller to Overcome Dead Time. *ISA J.*, 6(2):28-33.